MICROCHIP
UNIVERSITY

Microchip University
Online Class List

MICROCHIP

MICROCHIP
UNIVERSITY

Microchip University is now live! FREE online courses for embedded control engineers available 24 hours a day, every day.

Come learn about general embedded control topics as well as Microchip, Atmel & Microsemi products. Classes are taught by the same application and design engineers who create the products, which creates an amazing engineer-to-engineer experience. Absorb training material at all skill levels that cover topics such as Using MPLAB® Code Configurator (MCC), Embedded Linux®, Using Core Independent Peripherals (CIPs), Motor Control, Power Supply Design, Security, IoT, FPGAs, Analog System Design, Communication (USB, Bluetooth® and TCP/IP), and much more! New classes will continue to be added every month. In addition, live sessions and other educational events will be offered periodically.

To register, visit www.microchip.com/mu

Intro to the MPLAB® X IDE *(Level: Beginner)*

This class covers the basics of the MPLAB X IDE. This class will guide you through the steps of creating a simple "blink an LED" program using one of our popular PIC MCU Nano development boards. You will learn how to create a project from scratch, how to navigate the IDE, how to write and debug a simple program and then how to test your code on an actual development board.

MPLAB® X Tips & Tricks *(Level: Beginner)*

This is a collection of extremely useful tips and tricks that will help you get the most out of MPLAB® X..

MPLAB® Code Configurator *(Level: Beginner)*

The MPLAB® Code Configurator (MCC) is a free plug-in of MPLAB® X IDE, which provides an easy setup and configuration experience for supported microcontrollers. In this class, you will learn to navigate and manage project settings with MCC, set up and configure peripherals including Core Independent Peripherals, and choose from a wide variety of software libraries. MCC will generate optimized driver code tailored to your requirements, which is automatically integrated into new or existing embedded projects. Learn how to leverage the power of MCC to quickly develop an embedded application and get your project off the ground in minimal time!

Creating Unique Digital and Analog Functionality by Interconnecting Core Independent Peripherals (CIPs) *(Level: Intermediate)*

This class will show how to create a FSK demodulator, high-resolution PWM, WS2812B RGB LED protocol, ultrasonic range detector, metal detector, AC Sine direct drive and switch-mode power supply by interconnecting multiple Core Independent Peripherals (CIPs). Examples will be shown using PIC® and SAM MCUs. The CIP interconnections and circuit wave-forms will be presented along with live demonstrations of several of these designs. Exposure to these real-world examples will help you to apply CIPs in your project to reduce component count and improve system performance by replacing firmware overhead with CIP hardware.

Syntax And Structure of C *(Level: Beginner)*

This class will enable you to begin writing embedded C language firmware for microcontrollers. Most major C language constructs will be covered, including variables, constants, operators, expressions and statements, decision functionality, loops, functions, arrays, multi-file projects, and data pointers. You will learn all of these C language topics from a non-hardware framework so that you can focus on learning the C language instead of the microcontroller architecture. The presentation will be accompanied by instructor-led code demonstrations that will be conducted with the powerful MPLAB® simulator. Skills learned in this class will be applicable to any ANSI C compiler. At the end of the class, you will have the opportunity to apply your knowledge to program a microcontroller to perform basic input and output functionality and control. You will also use the MPLAB® X IDE to perform actual debugging on a microcontroller and execute some basic debugging techniques. While not required, previous experience with any programming language or experience with microcontrollers would be helpful.

Visual Debugging with MPLAB® Data Visualizer *(Level: Beginner)*

The MPLAB Data Visualizer is a tool which can help you understand your embedded application's run time behavior, on your target microcontroller. This course first introduces the MPLAB Data Visualizer's capabilities, then show you how to use these features. Although example projects are used to provide a context to show off these features, in most instances you will be able to follow along with your own project.

MPLAB® Harmony v3 Fundamentals *(Level: Intermediate)*

MPLAB® Harmony 3 is an extension of the MPLAB IDE ecosystem that simplifies development of embedded firmware for Microchip 32-bit SAM and PIC32 microcontroller devices. It includes an easy to use Graphical User Interface (GUI) for selection, management, configuration, and generation of starter code, peripheral libraries, and extensive middleware. This class provides an introduction to MPLAB Harmony 3, an overview of what it is, and an explanation of how it differs from MPLAB Harmony 2. Attendees will gain an understanding of key concepts, supported development models, and what resources are available for further learning. This class is recommended as a prerequisite to the MPLAB Harmony 3 hands-on lab classes and is a good introduction to MPLAB Harmony before attending classes focused on specific middleware stacks.

Introduction to Embedded Linux *(Level: Beginner)*

In this class, you will explore the Microchip ATSAMA5D27-SOM1-EK1 evaluation platform running a Buildroot Embedded Linux Distribution. You will be introduced to the embedded Linux boot sequence and the different components that make up a board support package. Basic concepts of Flattened Image trees and Device Tree Overlays, its need and evolution will be discussed. You will explore the tools, such as sam-ba programmer, and resources required by new users to start embedded Linux development on the Microchip wiki. The concept of user and kernel space will be introduced. This class includes hands-on exercises where you will explore the underlying hardware using different Linux tools and sub-systems. Specifically, i2c, gpio, network, device tree, udev, run-levels, start-up scripts, Linux virtual file systems - procfs, sysfs and debugfs, will be covered. You will see how to access different peripherals from user space using C and MPIO, a Python-based utility.

Exploring Linux Build Systems *(Level: Intermediate)*

In this hands-on lab class, you will configure and create a bootable embedded Linux image and project development environment for the SAMA5D27-SOM1-EK1 development kit using the Buildroot build automation tool. You will explore the basics of Buildroot allowing you to set up a production-ready build environment. We will cover concepts such as build environment configuration, external tree organization, target package selection, and build process.

Building your Linux Development System *(Level: Intermediate)*

When you are starting Linux development there are a lot of things to learn before you can write code for the embedded MPU. This class will take you through the steps required to setup a Linux development computer capable of running the Microchip tools. You will be able to select the best platform for your situation and where to locate the required tools to begin development. Having built your own Linux computer you will be well placed to take part in any of the practical sessions in the Linux series of seminars.

dsPIC33CH® Dual Core Device Architecture *(Level: Advanced)*

This course covers the architecture of the dsPIC33CH dual-core family of digital signal controllers. Intended for developers who have some experience with Microchip's single core MCU architecture, this class emphasizes the specific features and benefits found in a dual core dsPIC33CH device, including basic device architecture and peripheral integration, shared resources, run time application startup for each core, and inter-core communications. We will also review all the programming and debugging modes using the latest Microchip development tools available.

New

dsPIC33CH® Dual Core Programming and Debugging *(Level: Intermediate)*

This class will discuss the dsPIC33CH family of 16-bit dual-core dsPIC® devices. Learn how a dual-core dsPIC® device with various interconnected peripherals can be leveraged to accelerate time-sensitive embedded control applications, increase reliability in safety-critical applications, and reduce overall application cost. Multiple hands-on labs will explore various dual-core programming and debugging modes used to develop Primary and Secondary core applications separately then integrate them seamlessly using the MPLAB® X IDE environment.

MPU System and PCB Design Pitfalls and Solutions *(Level: Intermediate)*

This class will guide you through the hostile terrain of implementing high-speed/high-end MPU devices within an electronic system. Following a pragmatic and chronological approach, you will explore the different considerations to be taken at each stage of the project, from architecture definition to PCBA manufacturing. This class will review practical design cases, covering the following topics: Board layout, high-speed memory layout, high-speed communication interfaces, line balancing, impedance matching, power sequencing, low-power considerations, decoupling, power integrity, signal integrity, and EMC considerations. For each topic, design options will be explored, tips given, and pitfalls highlighted. Ultimately, ways of alleviating the MPU system design burden will be shown, exploring the MPU system solutions provided by SiP (System in Package) and SOM (System On a Module) devices.

Hello FPGA *(Level: Beginner)*

This class introduces Microchip's low-power FPGA portfolio, from our ultra-low density devices, to our newest mid-range density PolarFire FPGA and PolarFire SoC families. The presentation will include the architectural details of the most recent FPGA families, target applications, available IP blocks including microprocessor cores, development tools and the design ecosystem to implement designs. Following the lecture, the instructor will demonstrate the FPGA design flow using the Microchip Libero SoC toolset targeting the Microchip low-cost Hello FPGA platform.

New

Simple Applications Using the MPLAB® Harmony v3 Peripheral Libraries *(Level: Intermediate)*

MPLAB® Harmony v3 provides graphical tools and easy to understand peripheral libraries that simplify the use of Microchip's 32-bit microcontrollers and microprocessors. In this hands-on class, you will learn to navigate and manage project settings with MPLAB® Harmony Configurator (MHC), set up and configure peripherals, and generate optimized code tailored to your requirements, which is automatically integrated into new or existing embedded projects. Learn how to leverage the MPLAB® Harmony power to quickly develop an embedded application and get your project off the ground in minimal time!

New

Creating Advanced Embedded Applications with 32-bit MCUs/MPUs using the MPLAB® Harmony v3 Software Framework *(Level: Intermediate)*

MPLAB® Harmony is a modular framework that provides inter-operable firmware libraries for 32-bit microcontroller and microprocessor application development. This class shows how the MPLAB® Harmony drivers, system services and middleware enables you to rapidly develop bare-metal and RTOS applications.

16-Bit Bootloaders Using MCC: Device Side *(Level: Advanced)*

This class will focus on generating a 16-bit bootloader for the PIC24/dsPIC33 devices using MPLAB® X IDE and the MPLAB Code Configurator (MCC) and incorporating it into your application. The class will cover bootloader basics, the 16-Bit bootloader memory map, interrupt redirecting, verification methods, switching between the bootloader and application. The class will also walk through detailed examples on how to use the MCC bootloader module to build a complete bootloader and application as well as downloading the code to the device.

New

CAN and CAN FD Protocol and Physical Layer Basics *(Level: Beginner)*

The course will teach you the CAN and CAN FD protocol and physical layer basics. You will understand the responsibilities at system, application, and hardware level while using CAN and learn the basics of CAN transceivers and board design considerations. We also will take a short look in the future of CAN and what's coming next.

Introduction to USB 2.0 *(Level: Beginner)*

This class will provide an introduction to the basic concepts and tools of USB 2.0 such as topology, enumeration, endpoints, transfer types and classes. USB protocol analyzers, used to capture and decode USB traffic, will also be introduced.

New

USB Device Applications with MPLAB® Harmony USB Stack *(Level: Advanced)*

USB is now a standard serial communication channel to connect embedded systems to PCs. The USB Stack in MPLAB® Harmony allows you to easily develop a USB device application on PIC32 and ATSAM USB microcontrollers. In this class you will learn how to configure the MPLAB Harmony USB Stack and use the provided APIs to exchange data between your Embedded application and a PC. You will also learn how to debug your MPLAB Harmony USB device applications, avoiding possible pitfalls that you might run into.

Exploring Bluetooth® Low Energy (BLE) for Simple Applications *(Level: Intermediate)*

Struggling to understand how Bluetooth Low Energy (BLE) works? Looking to start a BLE design? This class is for engineers who want to learn BLE with little to no background at all. Concepts covered include BLE Specification, connectivity and data transfer as it relates to typical applications. If you need to replace cables in your application with BLE in the hands-on portion, you will learn how to create a serial port replacement application in 15 minutes. The hands-on labs will use the PIC/AVR-BLE board and will also cover sensor acquisition to phone communication using PunchThrough's Lightblue Explorer mobile app.

Rapid Prototyping Bluetooth® Low Energy (BLE) Android Apps using MIT App Inventor *(Level: Intermediate)*

This class will give the engineer a hands-on introduction to Android App development using MIT App Inventor. The material explained in this class will cover the steps required to develop an Android App that interacts with the AVR-BLE board. Prior knowledge of the basic Bluetooth® Low Energy (BLE) protocol is required and will not be covered in this class. Covered topics include: BLE scanning, BLE connection, BLE services, BLE characteristics, UUID's and MAC addresses.

Getting Started Developing Bluetooth® Low Energy (BLE) Android Apps using Android Studio *(Level: Intermediate)*

This class will help you create simple proof-of-concept apps using the Android™ Studio development environment. Most embedded developers are not expected to create professional mobile apps but may need a simple app to test and demonstrate the functionality of their Bluetooth product. You will learn how to use the official Android Studio development environment, how Android apps are structured, touch on key features of the Java language, and go into Bluetooth® Low Energy (BLE) support in more detail. The class will use Android phones to connect to Microchip RN4870 BLE modules. The hands-on demonstrations will step through the procedures to scan, connect, discover services, and send and receive data over a BLE connection.

Cryptography Primer *(Level: Beginner)*

The target audience are those unfamiliar with how cryptography works or those very rusty and in need of a refresher. Both symmetric and asymmetric cryptography will be covered and examples will be shown on how each are used in real-world examples. It will be shown how identity is created and used for authentication, integrity, and confidentiality. Due to time limitations, math proofs of these functions are unable to be presented. Only practical use is covered.

Design Considerations For Your First IoT Project *(Level: Beginner)*

In the Internet of Things, the Things must conform to the Internet, not the other way around. If you're just dipping your toes into the IoT, this class is for you. Adding network connectivity to an embedded product is complex. This class will explore the foundations of Internet communication. Routers, switches, IP and hardware addressing, DHCP, NAT, TCP and UDP transport layers, ports, sockets, and DNS will be explained. The MQTT eco-system will be described, including the pub/sub model, brokers, topics, and the JSON data structure. Lastly, tools for analyzing these packets and data structures will be demonstrated. This class is meant for the engineer who has no problem setting up their own home network, but has not necessarily explored the detailed requirements for an IoT device to exchange data with an internet resource.

Create a Managed IoT Device Leveraging Microsoft Azure IoT Services & the Microchip SAME54 Xplained Pro Evaluation Kit *(Level: Intermediate)*

Microsoft is a leader in IoT because they're passionate about simplifying IoT so any company can benefit from it quickly and securely. Companies are harnessing billions of IoT devices to help them find valuable insights into critical parts of their business that were previously not connected. This workshop will feature hands on labs leveraging the SAME54 Xplained Pro evaluation kit with the foundational Azure components, including Azure RTOS (formerly known as ThreadX), IoT Devices SDK featuring Plug and Play, Device Provisioning Services connected to the Azure IoT Hub Service and displaying data using the Azure IoT Central dashboard. Additionally, we will introduce new Azure IoT Services planned for future release.

Analog Design Tools I *(Level: Intermediate)*

An analog simulator can help reduce design cycle times by allowing exploration of circuit functionality and its response to various intended and unintended stimuli. The results from these experiments mitigate risks of non-testable specifications and functionality due to costs and time. This hands-on class will teach you how to use the MPLAB® Mindi™ Analog Simulator platform to analyze and perform trade offs between your requirements specifications. A real-world audio application and a switching power supply will be used as case studies for exploring the simulators functionality and capability.

Analog Tips and Tricks *(Level: Intermediate)*

This class is a collection of useful Tips and Tricks from our Analog Experts. Additional material will be posted when available.

motorBench® *(Level: Advanced)*

motorBench® Development Suite is a comprehensive software development tool for Field Oriented Motor Control (FOC).
It has 4 main features:
- performing accurate measurement of critical motor parameters, called self commissioning
- automatic configuration of control gains
- user customization and fine tuning
- generating field oriented control algorithm source code to spin the measured motor

This graphical, interactive development environment helps motor control embedded engineers to save time in starting up and running new motors, especially when the motor parameters are unknown.

New

Using X2CScope to Simplify Motor Control Development and Debugging *(Level: Intermediate)*

X2Cscope is a virtual oscilloscope tool which allows run-time debugging or monitoring of your embedded application in MPLAB® X IDE. This tool allows you to watch, plot and even modify any global variable in your embedded application at run-time without halting your CPU. X2Cscope is for generic use in any embedded application and perfectly fits for signal processing related applications. After this class you will be able to add X2Cscope to your application firmware, utilize X2Cscope GUI watch and scope views and use more advance scripting features if needed.

Motor Control Rapid Prototyping *(Level: Advanced)*

In this class you will learn about model-based design using the open source toolchain Scilab/XCOS/X2C and the commercial MATLAB/Simulink toolchain. In the first part you will do hands-on demos using Scilab/XCOS/X2C. The first demo is a basic Blinky demo to understand the toolchain and the workflow, continued from a sensored BLDC demo that allows you to identify the speed transfer function of a motor to do closed loop simulation and tuning of a speed PI controller. The third demo will demonstrate the full capabilities of the tool showing a closed loop sensorless FOC demo including closed loop simulation. In the second part of the class MATLAB/Simulink will be used to demonstrate the capabilities of this commercial toolchain in combination with Microchip microcontrollers.

Battery Charging Fundamentals *(Level: Intermediate)*

This class will introduce you to standard charge profiles for several common battery chemistries including Li-Ion, LiFePO4, NiMH and Lead Acid. We will then reference a number of real-world products and discuss their charging and battery requirements. Next we will learn how to read a datasheet to determine the charge parameters and become aware of common issues with the provided data. Finally we will review several charging topologies including linear, boost and buck, and discuss the pros/cons of each.

Getting Started with CIP Hybrid Power PWM Controllers and MPLAB® X SMPS Design Tools *(Level: Advanced)*

Microchip's mixed signal MCUs contain analog control loops with digital oversight to configure, monitor, and dynamically adjust performance with an embedded microcontroller (MCU). This allows the system to report telemetry, make operating adjustments on the fly and respond to faults with customized, application-specific code. Without this added integration, flexible robust designs could require separate chips to regulate the output, adjust the regulator operation, measure the results, and communicate the information outside the system. This class will introduce you to Microchip's PIC16F17xx family of programmable PWM controllers with an overview of the tools, architecture and 3 examples that demonstrate how each solution is created followed by a discussion of the advantages and disadvantages of each implementation.

New classes continue to be added as well as Live Sessions, so check back often!

www.microchip.com/mu

MICROCHIP