**Windows 8 Power Management for Driver Developers**

Title:

Device Power Management for Windows 8

Subtitle:

This course is intended for software development and validation engineers that want to learn more about the power management technologies incorporated into the Windows 8 operating system. At course completion, learners should have an understanding of all the components that participate in the power management of devices on Windows 8.

Course Duration:

3 Days (plus one optional day)

Participants:

Driver developers who need a solid understanding of the Windows 8 device power management.

Summary:

Students gain insight into the fundamental architecture of the Windows Driver Model (WDM) Device power management and Windows 8 sub-device (component) power management. They receive a solid introduction to the Windows Driver Foundation (WDF) Kernel Mode Driver Framework (KMDF) device power management. This includes hands-on experience of device driver development using the Windows Driver Kit (WDK) plus lots of practical tips and tricks.

Prerequisites:

-Very good knowledge of the programming language C and/or C++

- Knowledge of either WDM or KMDF driver programming models

-Basic knowledge of Microsoft development environments (Visual Studio)

-Basic knowledge of Windows system programming and system administration

**Day 1**

Introduction

* Power management requirements
* Collaborative power management in computer organization
* Brief history of power management in personal computing
* Introduction to ACPI
* Alphabet soup of Power states: G-states, S-states, C-states, P-states, B-states, D-states and F-states.
* Computer organization: PC vs. SoC
* Bus power management primer: PCI and USB
* Processor power states and platform power states
* Windows operating system architecture overview
* Kernel mode components
* WDF refresher
* KMDF refresher
* Introduction into Windows 8 device power management concepts

WDM Power Management

* Roles of WDM drivers
* Power policy owner
* Device and System Power States
* Power state transitions
* WDM power management and Plug and Play
* Handling S-IRPs
* Handling D-IRPs

**Day 2**

WDM Power Management

* Device Idle power management
* Device Wake
* Port framework power management: Video, NDIS, Streaming
* Advanced topics (paging path, inrush current, Checked and Free builds)

KMDF Power Management

* Power management defaults
* Power managed queues
* WDF power management and Plug and Play
* WDF device object power events and callbacks
* Power callbacks sequencing
* Idle and Wake support
* Windows 8 KMDF Frameworks: Serial, GPIO and SPB

Windows 8 Component Power Management

* Devices and components
* System, device and component power states
* Component power management and Plug and Play
* Power Framework (PoFx)
* WDF Power Framework support for D-States and F-States (partial)
* Power Engine Plugin (PEP)
* Component power policy ownership
* Device registration
* Handling component power callbacks
* Device unregistration

**Day 3**

Windows 8 Component Power Management

* Component Idle power management
* PoFx Device Idle power management
* PoFx Device wake support
* Advanced topics (component wake, component dependencies, managing idle residency and latency, D3Cold)
* WDM restrictions PoFx-integrated devices

Debugging and testing

* Hardware testing and debugging tools
* Software testing and debugging tools
* Debugger extensions for power management issues
* Common reliability metrics and automation
* Diagnosing and debugging common pitfalls

**Day X (NDA)**

Power Engine Plugin (PEP)

* PEP responsibilities
* PEP registration
* Device registration
* PEP messages
* PEP messages sequencing
* Device unregistration
* Handling device constraints
* Scoreboarding