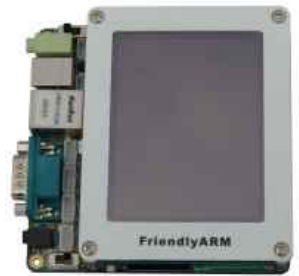


Windows CE product development using Devices Hands on Workshop (including free target board to take back home!!)

Windows Embedded CE 6.0 provides the tools and technologies to create devices that deliver immersive user interfaces and unique connected experiences. Powerful development tools like Platform Builder, a Visual Studio 2005 plug in, provide an integrated development environment (IDE) that enables you to build both application and Windows Embedded CE operating system software in a familiar environment to streamline the development process. Objective of the training is to enable the participant to learn the process of developing drivers and application for the CE product development.



Day 1

Introduction to the Platform Builder IDE

- The Win CE Development Environment
- Introduction to the IDE
- The Win CE Build Tools and Build System

Customizing the Operating System Design

- Creating and Customizing the Operating System Design
- Operating System Design Overview
- Creating an OS Design
- OS Design Customization with Catalog Components
- Build Configuration Management
- OS Design Property Pages

Configuring Windows Embedded CE Subprojects

- Windows Embedded Subprojects Overview
- Creating and Adding Subprojects to an OS Design
- Configuring a Subproject

Managing Catalog Items

- Catalog Files Overview
- Creating and Modifying Catalog Entries
- Catalog Component Dependencies

Day 2

Building and Deploying a Run-Time Image

- Building a Run-Time Image
- Build Process Overview
- Building Run-Time Images in Visual Studio
- Windows Embedded CE Run-Time Image Content



Editing Build Configuration Files

- Dirs Files
- Sources Files
- Makefile Files

Analyzing Build Results

- Understanding Build Reports
- Troubleshooting Build Issues

Deploying a Run-Time Image on a Target Platform

- Choosing a Deployment Method
- Attaching to a Device



Lab: Building and Deploying a Run-Time Image on the Emulator

- Build a Run-Time Image for an OS Design
- Configure Connectivity Options
- Change the Emulator Configuration
- Test a Run-Time Image on the Device Emulator

Day 3

Generating a Software Development Kit

- Software Development Kit Overview
- SDK Generation

Customizing a Board Support Package

- Board Support Package Overview
- Adapting and Configuring a Board Support Package
- Implementing a Boot Loader from Existing Libraries
- Adapting an OAL
- Integrating New Device Drivers
- Modifying Configuration Files

Configuring Memory Mapping of a BSP

- System Memory Mapping
- Memory Mapping and the BSP
- Enabling Resource Sharing between Drivers and the OAL

Day 4

Developing Device Drivers

- Understanding Device Driver Basics
- Native and Stream Drivers
- Monolithic vs. Layered Driver Architecture

Implementing a Stream Interface Driver

- Device Manager



- Driver Naming Conventions
- Stream Interface API
- Device Driver Context
- Building a Device Driver
- Opening and Closing a Stream Driver by Using the File API

Configuring and Loading a Driver

- Device Driver Load Procedure
- Kernel-Mode and User-Mode Drivers

Implementing an Interrupt Mechanism in a Device Driver

- Interrupt Handling Architecture
- Interrupt Identifiers (IRQ and SYSINTR)
- Communication between an ISR and an IST
- Installable ISRs

Exercise: Driver Code Examples

- Code-walk-through of:
 - Stream device driver
 - Block device driver



Day 5

Debugging and Testing the System

- Debugging and Target Device Control
- Kernel Debugger
- Debug Message Service
- Debugger support features
- CeLog Event Tracking and Processing

Configuring the Run-Time Image to Enable Debugging

- Enabling the Kernel Debugger
- KITL
- Debugging a Target Device

Testing a System by using the CETK

- Windows Embedded CE Test Kit Overview
- Using the CETK
- Creating a Custom CETK Test Solution
- Analyzing CETK Test Results
- Debug Zones

