ac6

W4 - BSPs and drivers CE 6.0

Writing BSP and drivers for Windows CE 6.0

Windows Embedded CE is a registered trade mark of Microsoft

Goals

- Master the cross development tools
- Bring the system on the target
- Accessing I / O
- Install kernel interrupt routines and applications
- Develop drivers and Board Support Packages

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
  - ▶ One Online Linux PC per trainee for the practical activities.
  - ▶ The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
  - ▶ QEMU Emulated board or physical board connected to the online PC (depending on the course).
  - ▶ Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
  - ▶ One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
  - ▶ One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
  - ▶ An installation and test manual is provided to allow preinstallation of the needed software.
  - ▶ The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Prerequisite

- Good knowledge of the C language
- Knowledge of application programming on Windows CE 6.0 (as described in <u>W3 - Windows Embedded CE 6.0</u>course) is mandatory.

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the trainee in his company, or by the trainee himself in the exceptional case of an individual trainee.

- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

Overview of WindowsCE

- Architecture and versions of Windows CE
- What's new in version 6.0
- Technologies and processors supported
- Embedded specific settings
- Shared source code and licensing system
- Comparison with Windows XP Embedded.
- Comparison with Windows Mobile
- Development tools for Windows Embedded CE

Windows CE architecture

- The Windows CE Kernel
- The Device Manager
- The Graphical Windowing and Events System (GWES)
- Communications services
- Timers, Watchdog timers.
- Interruptions,
  - IRQs (Interrupt ReQuest)
  - ISR (Interrupt Service Routine)
  - The IST (Interrupt Service Thread)
  - Interruption APIs
- Memory architecture.

Platform Builder in brief

- Steps to build a Windows Embedded CE platform
- Creating projects and source Workspaces
- Modules and components of the OS
- The source code configuration files
  - DIRS
  - SOURCES
  - Makefile
  - Module definition
- Build phases
  - Compile
  - Sysgen
  - Release copy
  - Make image
- Configuring Debug and Release configuration

Test and debug

- Kernel Debug
- Debug areas
- Just In Time Debugging (JIT)
- Kernel Profiler, Remote Kernel Tracker, Remote Call Profiler.

- Exception handling
- CETK Tests
- Using remote tools

Developing the Board Support Package (BSP)

- Bootloader development
- Develop the OEM Abstraction Layer (OAL)
- Configuration files

Exercise: Development of a serail line KITL
Exercise: Debug using the Lauterbach probe

Device driver development

- Introduction
- Driver types
- Stream drivers
    - interface
    - installation (static and dynamic)
- User mode drivers
- Access to physical memory
- Interrupt management
- Parameter marshalling
- Power management
- Test and debug

Exercise: Writing a fully functional button driver
- *driver installation*
- *physical memory access*
- *interrupt management*
- *asynchronous access to the user buffer*
- *registration of a button as an alarm source*
- *writing a CETK test*

Exercise: Writing a LED driver
- *management of power states*

Exercise: Demonstrating the use of hardware traces (captured by the Embedded Trace Macrocell) for error detection and fix, using a Lauterbach probe.

Renseignements pratiques

Inquiry : 3 days