Embitude

Let's have fun with Embedded

Online Training on Linux I2C Drivers
By Pradeep Tewani

Description
The Linux I2C drivers course provides a deep insight into the Embedded Linux Driver in hardware platform specific context. The course starts with basic low level device driver and helps develop the understanding of accessing hardware specific registers in Linux driver. Further to this, it provides the deeper understanding of driver design philosophy and covers the important concepts such as Device model, Platform Drivers and DTB. And finally the low level driver is integrated with Linux I2C framework providing the indepth understanding of interrupt management & bottom halves in the device drivers.

Course Objective
The Linux kernel programming course attempts to serve multiple objectives:

- To enable participants develop the low level driver from scratch by accessing the hw specific registers
- To enable participants develop the solid understanding of device driver fundaments as device tree blob (DTB), Platform Driver and Interrupt Mangement
- To enable pariticipants to understand the Linux I2C Framework
- To develop the indepth understanding of complete end to end flow from the user space to the low level

Target group:
Professionals looking to sharpen their skills in Linux Device Driver

Pre-requisite
Knowledge of C & Good Understanding of Linux Character Drivers

Learning Outcome
- Acquaintance with writing & testing the low level driver
- Indepth understanding of how the various linux driver components are linked together
- Developing the confortability with adding the device tree node and integrating the complete driver

Methology
Every theoretical topic is accompanied by corresponding hands-on/assignment to get the deep understanding of the topic.

Assessment
Assignment Based

+ Session 1: BBB Set up & Building the Kernel
- Readying BBB for I2C driver
- Setting up the host environment
- Patching & Building the kernel
- I2C Protocol overview

Exercises

- Configure & build the kernel
- BBB Setup & booting up the board
- Develop the character driver framework for low level I2C driver

+ Session 2: Low Level I2C Driver

- Understanding the AM335x specific I2C registers
- Understanding the flow for transferring the single byte

Exercises

- Develop the logic for transferring a single bytes on the I2C bus

Session 3: Enhance the Low Level I2C Driver

- Sending the multiple bytes
- Interfacing with on-board EEPROM

Exercises

- Enhance the driver to Transfer the multiple bytes
- Enhance the driver to perform read & write operations on EEPROM

+ Session 4: Linux Device Model

- Need for Linux Device Model
- Role of Platform Drivers
- Adding the device specific nodes in the device tree blob (DTB)

Exercises

- Enhance the low level driver to use the platform bus
- Enhance the driver to use the DTB

+ Session 5: Linux I2C Framework

- I2C Framework Components
- Registering I2C Client & Adapter driver
- Understanding the client driver probe mechanism

Exercises

- Writing a hardware independent Dummy Adapter & Dummy Client Driver

+ Session 6: Integrating low level driver with framework

- Integrating platform specific controller driver
- Integrating the EEPROM client driver

Exercises

- Making suitable enteries in DTB for adapter & client driver
- Tetsting the driver

+ Session 7: Enhancing the driver with Interrupts

- Understanding the need for interrupts
- Integrating the bottom halves

Exercises

- Integrate ISR & the bottom halves
- Register the tasklet as the bottom half
- Register the work queue as the bottom half

+ *Session 8: Wrap Up*

- Conclusion
- Next Steps