Online Training on Linux SPI & UART Drivers
By Pradeep Tewani

Description
The Linux SPI & Uart 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 helps to apply the important concepts such as Platform Drivers and DTB. And finally the low level driver is integrated with Linux SPI framework, thereby helps in developing the complete understanding of various SPI components such as SPI Master, SPI Client device & driver. In addition, the course covers the important concept of Direct Memory Access (DMA) using the Linux DMA Engine. Next part covers the low level uart driver and correspondingly the integration with the TTY framework.

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 Direct Memory Access (DMA)
- To enable pariticipants to understand the Linux SPI & TTY Framework
- To develop the indepth understanding of complete end to end flow from the user space to the low level driver.

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 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 SPI & UAtRT drivers
- Setting up the host environment

- Patching & Building the kernel
- Overview of SPI Protocol

Exercises

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

+ *Session 2: Low Level I2C Driver*

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

Exercises

- Write a low level SPI driver with loopback

Session 3: Linux SPI framework

- SPI framework components
- Registering SPI Client & Master driver
- Understanding the SPI Client probing mechanism
- Adding device specific nodes in the device tree

Exercises

- Writing a dummy SPI Client and dummy Master
- Integrating the low level driver with SPI framework

+ *Session 4: SPI Driver with Linux DMA Engine*

- Understanding the Linux DMA Engine
- Enhancing SPI driver to use DMA

Exercises

- Enhance the driver to add the support for DMA

+ *Session 5: Platform Specific UART Driver*

- UART Protocol overview
- Understanding the UART registers for the platform
- Writing a framework independent low level UART driver

Exercises

- Writing a low level UART driver

+ ***Session 6: Linux TTY Framework***

- Understanding the Linux TTY Framework
- Understanding the data flow for TTY Framework
- Writing a dummy UART driver

Exercises

- Writing a dummy UART driver using the TTY Framework

+ *Session 7: Integrating the UART driver with TTY Framework*

- Understanding the need for interrupts
- Integrating the bottom halves

Exercises

- Enhancing the low level driver to use TTY Framework

+ *Session 8: Wrap Up*

- Conclusion
- Next Steps