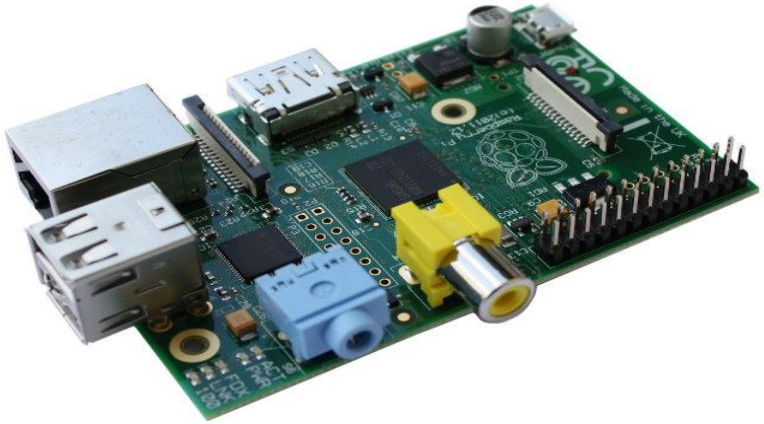
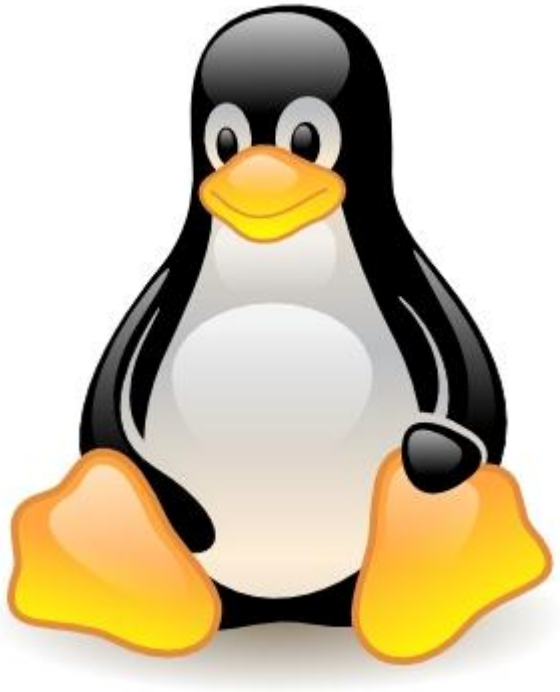
Practical World of Raspberry Pi

AESLAB
ACADEMY OF EMBEDDED SYSTEMS





Training Highlights:

- Learn through Practical's
- Real World Examples and Projects
- Assured Post Training Support
- Unlimited Access to the Hardware Boards for Practical's
- Training Tutorials & data available online

www.aeslab.com || info@aeslab.com || +91-80-41307589 || +91-9972039671

Training Modules on Raspberry-Pi

S. No.	Module Name	Duration	Price(INR)
1.	Raspberry Pi for Beginners	1 Day	500(free on purchase of RPi)
2.	Embedded Linux Porting on Raspberry Pi	2 Days	1200.00
3.	QT Embedded App Development on Raspberry Pi	2 Days	2000.00
4.	In-depth Python programming on Raspberry Pi	2 Days	2000.00
5.	The Internet of Things with Raspberry Pi	2 Days	3000.00
6.	Device driver development on Raspberry Pi	2 Days	3000.00
7.	Embedded Linux BSP BootCamp on Raspberry Pi	2 Days	5000.00

Module-1 : Raspberry Pi for Beginners

Duration: 2Days	Timings: 9:30 AM to 5:30 PM

SNO	Topic	Activities
Day-1 [Embedded Linux Intro & Board Bring-up]		
1	Introduction, Setup & Hardware	- Introduction to Raspberry-Pi & Embedded Linux - ARM Processor Basics & Families - Raspberry-Pi Board Details and Schematic Overview - Boot Process and different stages of booting - Host PC Setup for Embedded Development (Windows / Ubuntu Linux)
	Tea Break	
2	Application Programming to access Hardware peripherals	- Flashing Images on Pi - How to transfer files to/from Board & Host PC. - Programming LEDs, Relays, Switches connected to Pi
	Lunch Break	
3	Application Programming to access Hardware peripherals	- Uart Programming to communication with serial devices like GSM / GPS / GPRS / BT) - SPI Device Programming such as SPI based ADC / RTC etc. - I2C Device Programming like EEPROM / Temp Sensor etc.
4	Embedded Projects Demos	- Video Streaming using Raspberry-Pi (Digital Signage Example) - Remote device accessing (Ex. for Home Automation)
	Tea Break	
5	Embedded Projects Demos	- How to Build IoT using Raspberry-Pi - Project discussion from participants and how to implement them.

Hardware Requirements:

- Laptop with 30GB Free Space
- Install Ubuntu 12.04 LTS and additional packages using elinux_pkg.sh
- Raspberry-Pi with all accessories (Lan cable, USB to Serial Cable, Power Supply, SD-Card, Card Reader)

Who Should Attend:

- Good C Programmers seeking career in the world of Linux.
- Working Professionals from Microcontroller background, middleware C/C++.
- Programmers from Windows OS platform interested to add/migrate to Linux and Embedded Systems
- Students seeking career in Embedded Systems & Linux domain.

Module-2 : Embedded Linux Porting

Duration: 2Days	Timings: 9:30 AM to 5:30 PM

SNO	Topic	Activities
Day-1 [Embedded Linux Intro & Board Bring-up]		
1	Introduction, Setup & Hardware	- Introduction to Embedded Linux - ARM Processor Basics & Families - ARM Board Details and Schematic Overview - Boot Process - Host PC Setup for eLinux Development
	Tea Break	
2	Toolchain & Hardware Practical's	- Board Boot Options - Flashing Bootloader & Linux Kernel on Board - Setting up TFT and Running Application on Board - Toolchain & its components - How to build toolchain
	Lunch Break	
3	Bootloader U-Boot	- Introduction to Bootloader - Primary Bootloader (TI X-Loader) - Bootloader Commands and their usage
4	U-Boot Porting	- Bootloader Source Code Structure - Compiling Bootloader - How to port Bootloader on ARM Based Hardware - Patching Bootloader
	Tea Break	
5	Customizing Bootloader	- Modifying Bootloader for new feature - Modifying Bootloader to support new device - Command Line Arguments & ATAG - Booting with SD Card - Setting up NFS Server - Booting with NFS Server - Linux Kernel Compilation

Day-1 [Embedded Linux Intro & Board Bring-up]		
1	Linux Kernel	- Introduction to Linux Kernel Arch - Kernel Dir Structure - Kernel Layers H/W dependent and independent (BSP) - Kernel Build System (KConfig)
2	Tea Break	
3	Kernel Porting & Compilation	- How to configure and compile for ARM Hardware - Type of kernel images (vmlinux, zImage, uImage) - Kernel initialization process - How to port Kernel on New ARM Hardware
4	Lunch Break	
5	Kernel Modification	- How to modify the Kernel code - How to integrate new driver / module in kernel image - Building static and dynamic kernel modules
6	Root File System	- Components of RootFS -Types of RootFS -Different types of Flash Device (NOR / NAND) - Building RootFS from scratch and using Build System (Buildroot)
7	Tea Break	
8	Embedded Application Development	- How to develop embedded applications - Debugging application on target using GDB - Running sample Web-Server Application - Using Eclipse for embedded application development

Hardware Requirements:

- Laptop with 30GB Free Space
- Install Ubuntu 12.04 LTS and additional packages using elinux_pkg.sh
- Raspberry-Pi with all accessories (Lan cable, USB to Serial Cable, Power Supply, SD-Card, Card Reader)

Who Should Attend:

- Good C Programmers seeking career in the world of Linux.
- Working Professionals from Microcontroller background, middleware C/C++.
- Programmers from Windows OS platform interested to add/migrate to Linux and Embedded Systems
- Students seeking career in Embedded Systems & Linux domain.

Module-3 : QT Embedded App Development on RaspberryPi

Duration: 2 Days	Fee: INR 2000

–: Write us for more details (info@aeslab.com) :–

Module-4 : In-depth Python programming on Raspberry-Pi

Duration: 2 Days	INR 2000

–: Write us for more details (info@aeslab.com) :–

Module-5 : The Internet Of Things with Raspberry Pi

Duration: 2 Days	INR 3000

–: Write us for more details (info@aeslab.com) :–

Module-6 : Device Driver Programming on Pi

Duration: 2Days	Timings: 9:30 AM to 5:30 PM

SNO	Time	Topic	Activities
Day-1			
1	09:30 - 11:00	Introduction and Arch of Linux Device Drivers	- Introduction to Kernel Space and User Space - Memory mgmt in Kernel - How to develop Kernel Device Driver - Layers of LDD - Processor Memory Layout - Device Register Access from Code
2	11:00 -11:15	Tea Break	
3	11:15-12:45	Kernel Module Programming	- Kernel Module Programming - Module Parameters - Exporting Symbols between modules
4	12:45-13:30	Lunch Break	
5	13:30-14:30	Character Device Drivers	- Linux Kernel Device Driver Framework - Virtual File System as bridge between Driver and Application - Implementing basic character driver
6	14:30-15:30	Character Device Drivers	- Writing Makefile to compile Device driver - Compiling and running on X86 - Cross Compiling and running on ARM Hardware
7	15:30-15:45	Tea Break	
8	15:45-17:30	Advance options in Character Device Drivers	- Implementing advance api like ioct in character device driver - Standards to follow while implementing ioctl - Writing and testing LED driver with IOCTL on ARM Hardware
Day-2			
1	09:30 - 11:00	Interrupts in Device Driver	- Interrupts in ARM Processor - Interrupts Mechanism in Linux Kernel

			- How to implement Interrupts in device driver
2	11:00 -11:15	Tea Break	
3	11:15-12:45	Interrupt Handling & Bottom Half	- Writing and testing Interrupt for Button press on ARM Target - Writing and testing multiple Intterupts in single driver - How to implement Shared Interrupts - How to handle lengthy ISR using Bottom Half (Soft IRQ, Tasklet & Workquees)
4	12:45-13:30	Lunch Break	
5	13:30-14:30	Special File Systems ProcFS & SysFS	- Ram based files systems in Linux - Using procfs for special purpose and accessing kernel data structure - How to implement procfs - Sysfs implementation in device drivers for easy application access.
6	14:30-15:30	Introduction to Block Device and Network Device Drivers	- Introduction to block and network device drivers - Case study of Network Device Drivers
7	15:30-15:45	Tea Break	
8	15:45-17:30	Advance Device Drivers and debugging	- MTD Subsystem for Flash Memory Devices - Nand and Nor Device Drivers - USB Subsystem Introduction - How usb gadget drivers are used in Embedded Applications -Debugging Techniques like debugfs / target debugging

Hardware Requirements:

- Laptop with 30GB Free Space
- Install Ubuntu 12.04 LTS and additional packages using elinux_pkg.sh
- Raspberry-Pi with all accessories (Lan cable, USB to Serial Cable, Power Supply, SD-Card, Card Reader)

Who Should Attend:

- Working Professionals from Microcontroller background, middleware C/C++ to start with Linux Device driver programming.
- Programmers from Windows OS platform interested to add/migrate to Linux and Embedded Systems

Module-7 : Embedded Linux BSP on Raspberry Pi

Day-1	Day-2

1st Half: **[Get Comfort with ARM Target Board]**
- PhyBoard-WEGA System Arch
- Boot Process from PowerOn
  o Primary Boot Loader
  o Main Boot Loader (U-Boot / Barebox)
  o Kernel & Root File System
- Flashing Images using TFTP
  o (BootLoader, Kernel, RFS)
- Cross compile C app and Execute on Target
 	
2nd Half: **[Bootloader & Kernel Deep Dive HANDS-ON]**
- Bootloader Code Walk through
- Basic Hardware Configuration in Bootloader
  o RAM / FLASH (NAND)
  o Communication Ports (Serial, Ethernet)
- Lab Modify/Add GPIO Driver to power LED status in Bootloader code and test it on Target
 	
- Kernel Code Walk through
- Kernel Modifications for basic Board Booting
  o Board File Structure & Components
  o Understanding Pin Muxing
  o Adding devices inits in Board File
  ▪ Understanding Platform data
  ▪ Adding PMIC configuration for power domains
  ▪ Modifying Voltage Regulator for different devices
  ▪ Adding NAND support
  ▪ Adding Serial Port
  o Make file modifications & Board Config file creation
	1st Half: **[Kernel Deep Dive Cont… HANDS-ON]**
	- Walkthrough **MMC** domain in AM335x & its implementation
	- Lab Add SD-CARD support to Board file and enable root file system to be mounted from SD-Card partition.
	
	- Walkthrough **GPIO's** and its driver implementation
	- Lab Modify Board file to configure any pin of WEGA Board and test it using Linux user application.
	
	- Understanding **UARTs** in AM335x and its driver components
	- Lab Modify Board file to configure UART-2 & UART-3 on WEGA Board and test it using Linux user application.
	
	2nd Half: **[Kernel Deep Dive Cont… HANDS-ON]**
	- Intput Subsytem in Linux
	- Lab: Modify Board file to Configure Switches on WEGA board to generate input events & test it from user app.
	
	- I2C Subsytem in Linux
	- Lab: Modify Board file to add support of i2c based EEPROM or RTC and test it using user app.
	
	- SPI Subsytem in Linux
	- Lab: Modify Board file to add SPI based External ADC device to WEGA Board and test it from user app.
	
	- Display Sub-System in Linux
	- Lab: Configure the 7" LCD Display and test it using fbtest utils in linux.

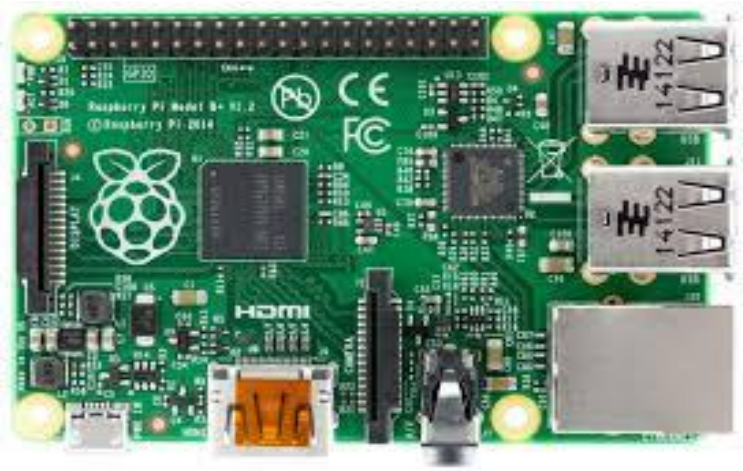
Training Objective

- Learn Linux BSP Development from scratch and get ready for the next project engaged in Board Bring-up and BSP Development.
- Deep dive in to the Boot-loader & Kernel code to become familiar for BSP modifications.
- Make your hands dirty with Board file writing, modifying pin Muxing & device driver code.

Pre-requisites:

- Should have hands-on in board bring-up and knowledge of writing Linux Device drivers.

RaspberryPi Board:



INR 2880* /- (* Tax Extra)

Features:

- **Broadcom BCM2835 SoC full HD multimedia applications processor in Chip**
- **700 MHz Low Power ARM1176JZ-F Applications Processor in CPU**
- **Dual Core VideoCore IV® Multimedia Co-Processor in GPU**
- **512 MB SDRAM @ 400 MHz in RAM**
- **MicroSD**
- **4x USB Ports**
- **1x 10/100mb Ethernet RJ45 Jack**

Address:
#9/1 1st Floor, 3rd Main, 8th Block, Opp. Police Station, Kormangala, Bangalore-560095, India. Ph: +91-80-41307589, Mob: +91-9972039671, www.aeslab.com, info@aeslab.com