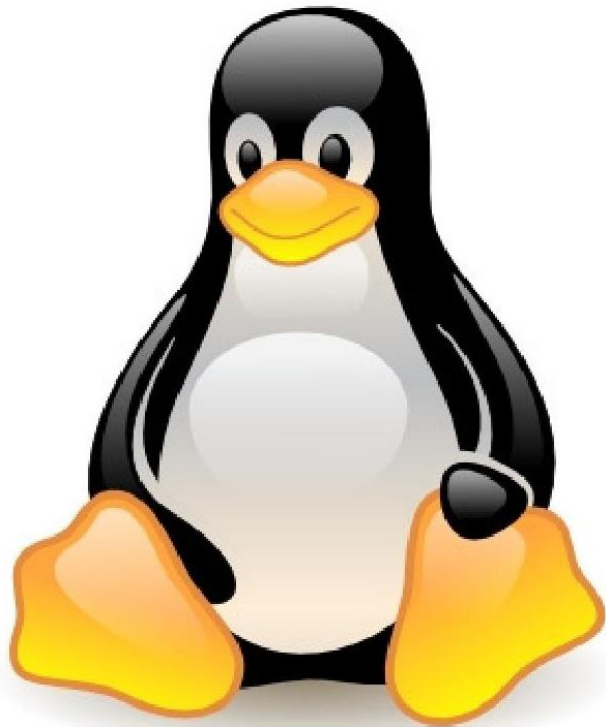


Advanced Embedded Linux & Device Driver Development



Training Highlights:

- Learn through Practical's
- Work on Latest ARM Core like Cortex A5/A8/A9
- Real World Examples and Projects
- Assured Post Training Support
- Unlimited Access to the Hardware Boards for Practical's
- Training Tutorials & data available online

Module-1 : Linux Basics & Internals

Duration: 4 Days

Timings: 9:30 AM to 5:30 PM

S.no	Topic	Activities
1	Linux Introduction & Installation	<ul style="list-style-type: none">➤ What is Linux, How it has been evolved, GNU License & Kernel➤ How Linux was designed,➤ Sub systems of Linux [Scheduler, Process, Memory Management, File System, Device Management]➤ Ways to Install Linux [1. Dual Boot, 2. Within Windows, 3. Using Virtual Machine]➤ How to update Linux and install required package
2	Linux Shell Commands	<ul style="list-style-type: none">➤ Basic Commands➤ Dir & File Command
3	Linux Shell Commands	<ul style="list-style-type: none">➤ System Commands➤ Misc Command
4	Shell Scripting	<ul style="list-style-type: none">➤ Writing Basic Linux Shell scripting➤ Variables & Operators in Shell scripting➤ Command Line Arguments➤ Logical Structures in Shell Scripting
5	C Programming in Linux	<ul style="list-style-type: none">➤ Writing C program on Linux➤ Compiling and executing Linux➤ Linux Executable format info & tools➤ Debugging C application on Linux using GDB
6	Makefiles	<ul style="list-style-type: none">➤ Understanding Makefiles➤ Writing Makefiles➤ Compiling Multiple source directory's using Makefile➤ Advanced methods used in writing Makefiles

Module-1 : Linux Basics & Internals

S.no	Topic	Activities
7	Process Management	<ul style="list-style-type: none">➤ Understanding Linux Process➤ How to create child process using [system, exec, fork & clone]➤ Managing Linux process
8	File Operation in Linux	<ul style="list-style-type: none">➤ How to write application to access files in Linux➤ System Calls used in Linux to control special files like device nodes➤ How to write a serial port access program in Linux
9	Linux Multi-Threading Programming	<ul style="list-style-type: none">➤ Basics of Multithreading in Linux➤ How to create multi-threading applications in Linux➤ Managing & communication between Multiple threads
10	Inter Process Communication	<ul style="list-style-type: none">➤ Data sharing between multiple processes using IPC Mech.➤ Writing apps using PIPEs, FIFOs, Msg Queues, Shared Memory
11	Network Programming in Linux	<ul style="list-style-type: none">➤ How to develop client server based network application in Linux➤ When and how to use TCP and UDP Protocols
12	Programming & Debugging Tools	<ul style="list-style-type: none">➤ strace : Tracing System calls➤ ltrace : Tracing Library calls➤ Tools used to detect memory access error and Memory➤ leakage in Linux : mtrace, valgrind➤ Using gdb and ddd utilities➤ Core Dump Analysis etc.

Module-2 : Embedded Linux Porting

Duration: 2 Days

Timings: 9:30 AM to 5:30 PM

S.no	Topic	Activities
1	Introduction, Setup & Hardware	<ul style="list-style-type: none">➤ Introduction to Embedded Linux➤ ARM Processor Basics & Families➤ ARM Board Details and Schematic Overview➤ Boot Process➤ Host PC Setup for eLinux Development
2	Toolchain & Hardware Practical's	<ul style="list-style-type: none">➤ Board Boot Options➤ Flashing Bootloader & Linux Kernel on Board➤ Setting up TFT and Running Application on Board➤ Toolchain & its components➤ How to build toolchain
3	Bootloader U-Boot	<ul style="list-style-type: none">➤ Introduction to Bootloader➤ Primary Bootloader (TI X-Loader)➤ Bootloader Commands and their usage
4	U-Boot Porting	<ul style="list-style-type: none">➤ Bootloader Source Code Structure➤ Compiling Bootloader➤ How to port Bootloader on ARM Based Hardware➤ Patching Bootloader
5	Customizing Bootloader	<ul style="list-style-type: none">➤ 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

Module-2 : Embedded Linux Porting

S.no	Topic	Activities
6	Linux Kernel	<ul style="list-style-type: none">➤ Introduction to Linux Kernel Arch➤ Kernel Dir Structure➤ Kernel Layers H/W dependent and independent (BSP)➤ Kernel Build System (KConfig)
7	Kernel Porting & Compilation	<ul style="list-style-type: none">➤ 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
8	Kernel Modification	<ul style="list-style-type: none">➤ How to modify the Kernel code➤ How to integrate new driver / module in kernel image➤ Building static and dynamic kernel modules
9	Root File System	<ul style="list-style-type: none">➤ Components of Roofs➤ Types of Roofs➤ Different types of Flash Device (NOR / NAND)➤ Building Roofs from scratch and using Build System (Buildroot)
10	Embedded Application Development	<ul style="list-style-type: none">➤ How to develop embedded applications➤ Debugging application on target using GDB➤ Running sample Web-Server Application➤ Using Eclipse for embedded application development

Module-3 : Device Driver Development

Duration: 2Days

Timings: 9:30 AM to 5:30 PM

S.no	Topic	Activities
1	Introduction and Arch of Linux Device Drivers	<ul style="list-style-type: none">➤ Introduction to Kernel Space and User Space➤ Memory Management in Kernel➤ How to develop Kernel Device Driver➤ Layers of LDD➤ Processor Memory Layout➤ Device Register Access from Code
2	Kernel Module Programming	<ul style="list-style-type: none">➤ Kernel Module Programming➤ Module Parameters➤ Exporting Symbols between modules
3	Character Device Drivers	<ul style="list-style-type: none">➤ Linux Kernel Device Driver Framework➤ Virtual File System as bridge between Driver and Application➤ Implementing basic character driver
4	Character Device Drivers	<ul style="list-style-type: none">➤ Writing Makefile to compile Device driver➤ Compiling and running on X86➤ Cross Compiling and running on ARM Hardware
5	Advance options in Character Device Drivers	<ul style="list-style-type: none">➤ Implementing advance api like ioctl in character device driver➤ Standards to follow while implementing ioctl➤ Writing and testing LED driver with IOCTL on ARM Hardware
6	Interrupts in Device Driver	<ul style="list-style-type: none">➤ Interrupts in ARM Processor➤ Interrupts Mechanism in Linux Kernel➤ How to implement Interrupts in device driver
7	Interrupt Handling & Bottom Half	<ul style="list-style-type: none">➤ Writing and testing Interrupt for Button press on ARM Target➤ Writing and testing multiple Interrupts in single driver➤ How to implement Shared Interrupts➤ How to handle lengthy ISR using Bottom Half (Soft IRQ, Tasklet & Workqueues)

Module-3 : Device Driver Development

S.no	Topic	Activities
8	Special File Systems ProcFS & SysFS	<ul style="list-style-type: none">➤ Ram based files systems in Linux➤ Using procs for special purpose and accessing kernel data structure➤ How to implement procs➤ Sysfs implementation in device drivers for easy application access.
9	Introduction to Block Device and Network Device Drivers	<ul style="list-style-type: none">➤ Introduction to block and network device drivers➤ Case study of Network Device Drivers
10	Advance Device Drivers and debugging	<ul style="list-style-type: none">➤ 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 debugs / target debugging

Optional:

- If you are having any technical queries related to syllabus covered (LAB & Theory), we provide an additional weekend session for the same.

Embedded Linux Training With Free Hardware Kit

Embedded Linux
Porting

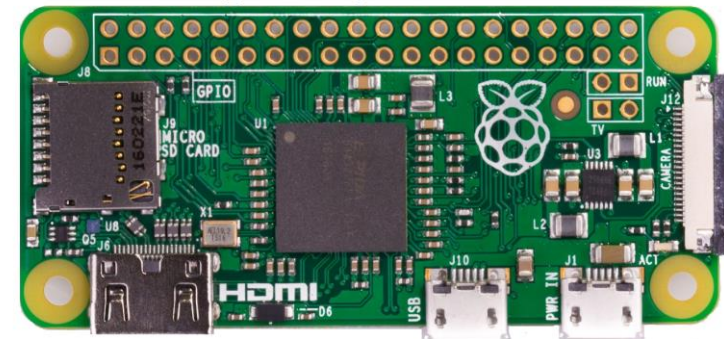
Linux Basics &
Internals

Device Driver
Development

+

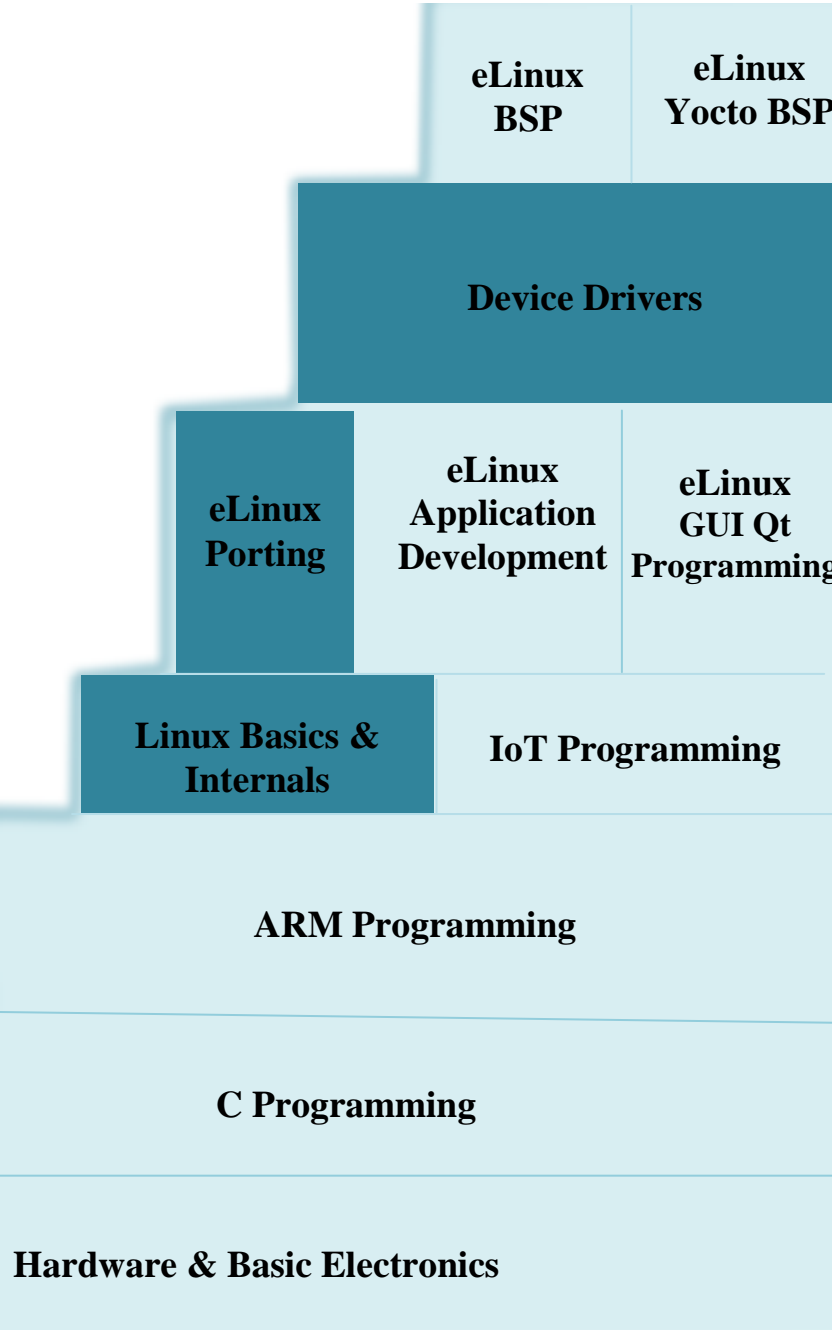
Free Hardware

Raspberry PI Zero



Embedded Engineering Steps

Locations / Fees / Duration



Embedded Engineering

Locations

- Bangalore

Duration:

- 8 Days

Fees:

- 18000.00 INR
- [Register Now](#)

After development workshop:

- When you return to work, you are entitled to schedule a technical discussion with the course instructor for help and guidance as you apply your new skills to your projects.

Address:

- AESLAB | #9/1 1st Floor, 3rd Main, 8th Block, Opp. Police Station, Kormangala, Bengaluru, Karnataka 560095
- Email: info@easyarm.com, Phone: +91-80-41307589 || +91-9972039671, Web: www.aeslab.com