



myTectra
Unlock Learning Potentials!

ISO 9001:2008
Certified Company

Course Details:

Course Code – MYT1162 – Embedded System

Course duration: Regular - 3 Months / Week End- 12 Week End / Fast Track – 45 Days

Training mode: Instructor led Class training | Live Online Training

Server access : Provided at no extra cost

Study material : Provided in the form of PDFs, PPTs & Word docs / On-Line materials

Weekend, weekday, Super Fast track, fast track & normal training can be provided

Contact: +91 90191 91856

Email: info@mytectra.com

Web: www.mytectra.com

About myTectra:

“myTectra a global learning solutions company helps transform people and organization to gain real, lasting benefits “

myTectra provides training across the technologies and functions using time tested delivery mode such as class room training , virtual training , send me a trainer , trainer connect .We are now one of the largest online training company and expanding our class room training across multiple cities.

Join myTectra! Explore world of possibilities in learning and acquiring professional Skills across domains and technologies irrespective of what kind of functional or technical background your are from..

myTectra Learning Solutions Private Limited
Bangalore – BTM Layout | Indiranagar | Basaveshwarnagar
+91 90191 91856 | info@myTectra.com | www.myTectra.com

WEEK: 1

CHAPTER 1: GETTING STARTED

- ✓ What is C?
- ✓ Data Types
- ✓ Variables
- ✓ Naming Conventions for C Variables
- ✓ Printing and Initializing Variables

CHAPTER 2: CONTROL FLOW CONSTRUCTS

- ✓ if
- ✓ if else
- ✓ while
- ✓ for
- ✓ Endless Loops
- ✓ do while
- ✓ break and continue
- ✓ switch
- ✓ else if

CHAPTER 3: THE C PREPROCESSOR

- ✓ #define
- ✓ Macros
- ✓ #include
- ✓ Conditional Compilation
- ✓ #ifdef
- ✓ #ifndef

CHAPTER 4: MORE ON FUNCTIONS

- ✓ Function Declarations
- ✓ Function Prototypes
- ✓ Returning a Value or Not

- ✓ Arguments and Parameters
- ✓ Organization of C Source Files
- ✓ Extended Example

WEEK: 2

CHAPTER 1: BIT MANIPULATION

- ✓ Defining the Problem Space
- ✓ A Programming Example
- ✓ Bit Wise Operators
- ✓ Bit Manipulation Functions
- ✓ Circular Shifts

CHAPTER 2: STRINGS & ARRAY

- ✓ Fundamental Concepts
- ✓ Aggregate Operations
- ✓ String Functions
- ✓ Array Dimensions
- ✓ An Array as an Argument to a Function
- ✓ String Arrays
- ✓ Example Programs

CHAPTER 3: POINTERS (PART 1)

- ✓ Fundamental Concepts
- ✓ Pointer Operators and Operations
- ✓ Changing an Argument with a Function Call
- ✓ Pointer Arithmetic
- ✓ Array Traversal
- ✓ String Functions with Pointers
- ✓ Pointer Difference
- ✓ Prototypes for String Parameters
- ✓ Relationship Between an Array and a Pointer
- ✓ The Pointer Notation *p++

WEEK: 3

CHAPTER 1: POINTERS (PART 2)

- ✓ Dynamic Storage Allocation - malloc
- ✓ Functions Returning a Pointer
- ✓ Initialization of Pointers
- ✓ gets - a Function Returning a Pointer
- ✓ An Array of Character Pointers
- ✓ Two Dimensional Arrays vs. Array of Pointers
- ✓ Command Line Arguments
- ✓ Pointers to Pointers
- ✓ Practice with Pointers
- ✓ Function Pointers

CHAPTER 2: STRUCTURES

- ✓ Fundamental Concepts
- ✓ Describing a Structure
- ✓ Creating Structures
- ✓ Operations on Structures
- ✓ Functions Returning Structures
- ✓ Passing Structures to Functions
- ✓ Pointers to Structures
- ✓ Array of Structures
- ✓ Functions Returning a Pointer to a Structure
- ✓ Structure Padding

CHAPTER 3: STRUCTURE RELATED ITEMS (UNION)

- ✓ typedef - New Name for an Existing Type
- ✓ Bit Fields
- ✓ unions
- ✓ Non-Homogeneous Arrays
- ✓ Enumerations

WEEK: 4

CHAPTER 1: FILE I/O

- ✓ System Calls vs. Library Calls
- ✓ Opening Disk Files
- ✓ fopen
- ✓ I/O Library Functions
- ✓ Copying a File
- ✓ Character Input vs. Line Input
- ✓ scanf
- ✓ printf
- ✓ fclose
- ✓ Servicing Errors - errno.h
- ✓ Feof

CHAPTER 2: SCOPE OF VARIABLES

- ✓ Block Scope
- ✓ Function Scope
- ✓ File Scope
- ✓ Program Scope
- ✓ The auto Specifier
- ✓ The static Specifier
- ✓ The register Specifier
- ✓ The extern Specifier
- ✓ The Const Modifier
- ✓ The Volatile Modifier

CHAPTER 3: INTRODUCTION TO EMBEDDED

- ✓ What is Embedded Systems?
- ✓ Difference b/w Micro processor & Micro Controller
- ✓ CISC Vs RISC
- ✓ Architecture of 8,16,32-bit Processor
- ✓ Software Used, Compilation, Debugging
- ✓ Example Programs (LCD, RELAY, STEPPER MOTOR)

Embedded Software Life Cycle Testing

SPI, ADC, Serial Communication, Protocols (I2C , CAN, Ethernet)

WEEK 5

Linux Internals

Chapter 1: Linux OS Architecture

- ✓ Linux Features
- ✓ Linux Kernel Source Directory Structure
- ✓ Linux Kernel Components
- ✓ User Mode Vs Kernel Mode
- ✓ System Initialization – Booting Process

Chapter 2: Introduction to Linux tools, compilers and utilities

- ✓ Introduction To Makefile
- ✓ How to write Makefile to compile programs on Linux
- ✓ Building static and dynamic libraries
- ✓ LABs

Chapter 3: Kernel compilation

- ✓
 - Importance of Makefiles
- ✓ Procedure to recompile the kernel
- ✓ LAB

WEEK 6

Detail study of Linux OS components

Chapter 1: Process Management

- ✓ What is Process?
- ✓ Process Control Block (PCB)

- ✓ Types Of Processes
- ✓ States Of Process
- ✓ How to Create Process?
- ✓ Process Scheduling
- ✓ LABs

Chapter 2: Thread Management

- ✓ What is Thread?
- ✓ Thread Control Block (TCB)
- ✓ User level Vs Kernel level Threads
- ✓ How to create and cancel threads?
- ✓ Thread Scheduling
- ✓ Process Vs Threads
- ✓ LABs

WEEK 7

Chapter 1: Interrupt Management

- ✓ What are interrupts?
- ✓ Types Of Interrupt
- ✓ Interrupt Handling
- ✓ Interrupt Service Routine (ISR)
- ✓ Interrupt Latency

Chapter 2: Signal handling

- ✓ What are signals in Linux OS?
- ✓ Signal Implementation
- ✓ Signal Handling
- ✓ LABs

Chapter 3: Inter-Process communication (IPCs)

- ✓ Introduction To Inter-processcommunication mechanism

- ✓ Pipes, Message Queue and Shared Memory
- ✓ Semaphores and Mutex
- ✓ LABs

Chapter 4: Socket programming on Linux

- ✓ TCP/IP and UDP socket programming
- ✓ LABs

Linux Kernel and Device driver Programming

Chapter 5: .Linux File system and System call interface

- ✓ Introduction To System Call Mechanism
- ✓ Significance Of System Calls
- ✓ LABs

Chapter 6: Memory Management Unit

- ✓ Segmentation and Paging
- ✓ Swapping and demand paging
- ✓ malloc(), kmalloc() and free()

WEEK 8

Linux Kernel and Device driver Programming

Chapter 1: Module Basics

- ✓ Introduction to Modules
- ✓ Writing Your first kernel module
- ✓ Module Related Commands
- ✓ Statically linked vs Dynamically linked
- ✓ Exporting symbols from modules
- ✓ The kernel symbol table
- ✓ Concurrency in the kernel

- ✓ Module Parameters
- ✓ Version dependency
- ✓ LABs

Chapter 2: An introduction to device drivers

- ✓ Role of the Device Drivers
- ✓ Role of Virtual file system
- ✓ Classes of devices

Chapter 3: Character Device Drivers

- ✓ Registering a character device driver
- ✓ File operations and ioctls
- ✓ Reading and writing into char devices
- ✓ LABs

WEEK 9

Chapter 1: Block Device Drivers

- ✓ Registering block driver
- ✓ File operations and ioctls
- ✓ Handling requests
- ✓ Write RAM type of disk driver
- ✓ LABs

Chapter 2: Network Device Drivers

- ✓ The net_device structure in detail
- ✓ Packet transmission
- ✓ Packet reception
- ✓ Simulating a network device
- ✓ LABs

WEEK 10

Embedded Linux On ARM9

Chapter 1: Intro to the target board (Samsung mini2440 – ARM 9)

- ✓ Introduction to ARM9 architecture

Chapter 2: Installation of cross compilation tool chain

- ✓ Installing Sources, Patching
- ✓ Installing the GCC toolchain
- ✓ Set Cross Compiling Environment
- ✓ LABS

Chapter 3: Porting Linux on ARM9

- ✓ Recompilation and flash Kernel on ARM9 board
- ✓ LABS

WEEK 11

Chapter 1: The Root Filesystem

- ✓ Creating a new root filesystem
- ✓ Busybox
- ✓ A Small Application example
- ✓ Flashing the new root filesystem
- ✓ Importance of the Makefile
- ✓ Compilation procedure
- ✓ LABS

Chapter 2: Write device driver for ARM9 board

Chapter 3: Services on Board

- ✓ Compiling and setting up services
- ✓ An example service – LABS

WEEK 12

Chapter 1: Structure and implementation of open source RTOS

- ✓ RTOS Source Organization
- ✓ File System in RTOS
- ✓ Configuration Of RTOS
- ✓ Implementation Of RTOS

Chapter 2: Port RTOS on ARM Board

- ✓ Steps for porting RTOS On ARM7
- ✓ LAB(Demostration)- Port Open Source RTOS On ARM Board

Chapter 3: Real Time Operating Fundamentals

- ✓ Task Management
- ✓ Multitasking
- ✓ Context Switching
- ✓ Inter Process/Task Communication (IPC)
- ✓ LABs (Demostration)

References and Guideline for Linux base embedded system

Skills developed after completion of course:

- ✓ Learn about Key principles of Linux OS
- ✓ Expertise on device driver for target board
- ✓ Porting Linux on advanced cross platform i.e. ARM 9
- ✓ Get good expertise on Linux base embedded system

Thanks!