SREE EMBEDDED TECHNOLOGIES
#F-208, RajiniGandha Complex, ChaithanyaPuri, Hyderabad - 500060.
e-mail : info@sreeembedded.com Call : 040 4856 9842 , 70758 00077.
www.sreeembedded.com

About Us :

Sree Embedded Engineers is an organization focused on delivering complete embedded system training. We provide training on complete practical oriented methods on Embedded domain, wireless technologies and mobile platforms.

The institute was founded by experienced engineers and Technical managers and Architects worked in various organizations in telecom and Embedded systems domains. Our training program is aimed at practical oriented and matching to the industry standards.

The teaching staff is highly experienced and good teaching skills and worked around 15 years in organizations like Samsung, Qualcomm,Texas Instruments on various technologies like Wireless,Audio device drivers,IOT technologies, Android Framework,HAL and Device drivers etc.

Teaching Staff:

The faculty obtained Masters degree in Embedded Systems and around 15 years of organization experience.

Course Contents for Embedded Training:

Module I. Introduction to Embedded Systems

1. Embedded systems
2. Embedded h/w units and devices
3. Embedded s/w in a systems
4. Classification of ES
5. Memory types and interfaces
6. CPU types and characteristics
7. I/O controllers - polled I/O or programmed I/O
8. Development environment - host-target environment,
   cross compilers, downloading methods, emulators, target based debugging, debug monitors and host based source level debugging
9. ARM arvhitecture
   - Inroduction to ARM Architecture
   - Processor Modes
   - ARM Registers
   - ARM Endian Support
   - ARM Instruction Set
   - Addressing Modes
10. IOT basic concepts

Module II. C Programming and data structures

'C' Programming in Embedded Systems:

Why C in Embedded
ANSI Standard
Fundamentals of C

Datatypes and Constants

Simple & Formatted I/O
Memory Usage
Operators & Expressions
Flow Control
Loops

Functions

Role of Functions
Pass by value / reference
Returning values from Functions
Recursive Functions
Call Back Functions
Implications on Stack
Library Vs User defined function
Passing variable number of arguments

Arrays

Defining, initializing and using arrays
Multi Dimensional Arrays
Arrays of Characters and Strings
Arrays and Pointers
Passing arrays to functions
String handling with and without library functions

Storage Classes

Scope and Life
Automatic, Static, External, Register

Memory(CPU / RAM)

Structures & Unions

What structures are for
Declaration, initialization
Accessing like objects
Nested Structures
Array of Structures
Passing structures through functions
Allocation of memory and holes
Structure Comparison
Structure bit operation
Typedef for portability
Unions
Overlapping members

Enumerated data types

Enum, Indexing, enum Vs #define

Bit Operations

AND (&), OR (|), XOR (^)
Compliment (~)
Left-Shift (<<), Right Shift (>>)
Masking, Setting, Clearing and Testing of Bit / Bits

Pointers

The purpose of pointers
Defining pointers
The & and * operators
Pointer Assignment
Pointer Arithmetic
Multiple in-directions
Advanced pointer types
Generic and Null Pointer
Function Pointers
Pointers to Arrays and Strings
Array of Pointers
Pointers to Structure and Union

Pointers to Dynamic memory
Far, Near and Huge Pointers
Pointer Type Casting

Dynamic Memory Allocation

Malloc(), Calloc(), Realloc(), Free()
Farmalloc(), Farcalloc()

File Handling Concepts

Concept of a FILE data type
Inode, FILE structure
File pointer
Character handling routines
Formatted Data Routines
Raw data Routines
Random Access to FILE

Command line Arguments

Argc, argv
Variable Inputs to the main

Compiler in Practical

Pre-processor Directives
Compiler, Assembler, Linker
Conditional Compilation
Multiple File Compilation
Code Optimization techniques
Volatile, #pragma

Data Structures

Linear & non-linear
Homogeneous & non-homogeneous
Static & Dynamic
Single, Double & Circular Linked Lists
Stacks & Queues
Binary Trees

Sorting and Searching Techniques

Insertion, Selection, Bubble, Merge, Quick, Heap

Concepts and Real-time Exposure

Development Tools and Environment
Make Utility
MISRA Coding Standards
Object / Executable File Format
Debugger

Mini Project 1: using data structures and pointers

Module III. gnu toolchain

GCC
GNU Makefile
GNU binutils
GDB (GNU Debugger)

Module IV. Firmware or Embedded C programming using IOT architecture:

1. 8051,ARM7 and ARM9 architecture
2. IOT architecture
3. Ayla IOT architecture (ST and Murata board)
4. Firmware development
5. Cloud architecture
6. Interfacing concepts of uart,i2c,leds,adc and spi

Module V. Linux basics and Utilities

1. Unix /Linux/Windows history
2. Free s/w foundation
3. GNU Project
4. Booting process of OS
5. Linux architecture
6. Linux root file system and virtual file system
   (file management/process management/synchronization/IPC/memory management)
7. Linux Kernel introduction
8. Shell programming

9. System calls, file handling, environment variables, inodes and file permissions
10. Error handling in Linux

Module VI. Fundamentals of Linux kernel and Device Drivers

1. Kernel Classification (Micro/Monolithic)
2. Kernel programming and source tree overview
3. Proc and Sys file systems
4. Device drivers - types and roles
5. Getting the sources and configuring the kernel
6. Diff and patching utilities
7. Compiling ,installing and booting the kernel

Module VII. Modular programming

1. Introduction to modular programming
2. Features of Modular programming
3. Kernel module vs Applications
4. Compiling and loading
5. The kernel symbol table
6. Module parameters

Module VIII. Character device drivers

1. The design and architecture
2. Major and Minor numbers
3. Data structures
4. Char device registration, open and release
5. Memory usage
6. read and write
7. Implementation of Char driver

Module IX. Race Conditions

1. concurrency and management
2. Semaphore Vs Mutexes
3. completions
4. Spinlocks
5.Locking traps
6.Alternatives to Locking

Module X. Block device drivers

1. ioctl
2. Blocking I/O
3. poll and select
4. Asynchronous notifications

Module XI. Hardware communication

1. I/O ports and I/O memory
2. I/O port example
3. Using I/O memory

Module XII. Interrupt handling

Preparing the parallel port
Installing an Interrupt Handler
Implementing a Handler
Top and Bottom Halves
Interrupt Sharing
Interrupt-Driven I/O

Module XIII. Memory mapping and DMA

Memory Management in Linux
The mmap Device Operation
Performing Direct I/O
Direct Memory Access

Module XIV. Block drivers

Registration
The Block Device Operations
Request Processing

Module XV.Network drivers

Connecting to the Kernel
The net_device Structure in Detail
Opening and Closing
Packet Transmission
Packet Reception
The Interrupt Handler
Receive Interrupt Mitigation
Changes in Link State
The Socket Buffers
MAC Address Resolution
Custom ioctl Commands
Multicast

Module XVI. Board bring up

1. Embedded Linux: Benefits of Linux and open source tools components of linux booting, embedded linux Focus system arch
2. cross compiler tool chain: need for cross compile using pre-built cross toolchain and building our own cross tool chain
3. Boot loader: advantages, Uboot source code overview, building u-boot for target boot time optimization
4. Linux kernel: supported h/w arch,croos compiling the kernel for target,understanding kernel boot arguments, system call implementation and boot-time optimizations
5. File system: understanding NAND/NOR flash, file system hierarchy,Busy box/Build root/LTIB/Yocto cross compiling apps and libs, creating file system images
6. Flashing : boot loader/kernel/file system and images
7. Android and linux portingon TI/samsung/freescale/broadcom SOC's

Module XVII. Android (need to change)

Introduction to Android
Android source code and compiling
Boot loaders for Android
Android changes to the Linux kernel
Supporting new hardware
Development and debugging with ADB
Android's build system
The Android file system
Android native layer and calling a C program from Android
Android framework and applications
System customization Advise and resources

Module XVIII. C++ Programming

1. Introduction to cpp
2. Data types, storage classes,constants, modifier types, operators,loops, functions, arrays, strings, pointers, references, date and time,
3. classes and objects
4. Inheritance
5. Overloading
6. polymorphism
7. Abstraction
8. Encapsulation
9. Interfaces
10.constructors and destructors
11. Static members and functions
12. friend classes and functions
13. Data conversions
14. Exception handling
15. Templates
16. Standard template library - STL

Module XIX. Debugging Techniques

Debugging Support in the Kernel(user space,kernel space and Android tools)
Debugging by Printing
Debugging by Querying
Debugging by Watching
Debugging System Faults
Debuggers and Related Tools

Module XX. Repository Tools

1. GIT Overview
2. Git commands
3. SVN Overview

Module XXI. Python

1. Basics
2. Variable types
3. Operators,loops, numbers, strings,lists,tuple,dictionary,Date and time
4. Functions
5. Modules
6. Files and IO
7. Exceptions

8. Classes and objects
9. Database access
10. Networking
11. Email
12. multithreading
13. XML
14. Frameworks

REAL TIME PROJECTS

Project -1 : Understanding RS232 driver
Project-2 : Understanding parallel port device driver
Project-3 : Understanding Keyboard device driver
Project-4 : Understanding keyboard device driver
Project-5 : Understanding Mouse driver
Project-6 : Understanding USB pen device driver
Project-7 : Audio device driver porting and feature development
Project-8 : WiFi device driver porting and feature development
Project-9 : WPA Supplicant porting and feature development
Project-10: Radio interface layer(RIL)-HAL understanding and development
Project-11: IOT understanding and development of AYLA networks board
Project-12: IOT understanding and development of Alljoyn framework
Project-13: IOT understanding and development using Googles Android things
Project-14: I2C and SPI drivers on Dragon board

The hardware kits available:

1. Panda board
2. Beagle bone black
3. Rasperry-Pi
4. Ayla board
5. IMX7 board
6. Aurdino board
7. 8051 controller boards
8. Dragon board