Focus Training Services

Your Training Partner

Course Content

Embedded Systems

Key Features:-

1- Limited Students in Batch(Max 15)
2- Experience and Certified Trainers
3- Placement Support for 1 year
4- Project Based Training
5- Training Material
6- Global Certification
7- Full functional Lab & Lab Support
8- Weekly Assessment
9- Resume Preparation and Soft Skills Training
10- Internship opportunities

Ameerpet-

#301, Nagasuri Plaza
Ameerpet, Hyderabad-81
Landmark- Above Bank of India (BOI)
PH- 040 6610099, 8801105544
USA- +1 908 333 5455
info@focustech.in, www.focustech.in

Madhapur-

#301, KSR Tower
Madhapur, Hyderabad-81
Landmark- Beside Indian Oil Petrol Pump
PH- 040 48505572, 9030945188
USA- +1 908 333 5455
info@focustech.in, www.focustech.in

Focus Training Services
#301, 3rd floor, Nagasuri Plazza,
Satyam theatre Road Ameerpet
Above bank of India (BOI)

Embedded Systems Content

Focus Training Services
#301, 3rd floor, KSR Towers,
Madhapur,
Beside Indian Oil Petrol pump

Embedded System

As its name suggests, Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. It offers many benefits such as sophisticated control, precision timing, low unit cost, low development cost, high flexibility, small size, and low weight. These basic characteristics can be used to improve the overall system or device in various ways:

- Improved performance
- More functions and features
- Reduced cost
- Increased dependability

Because of these benefits, billions of microcontrollers are sold each year to create embedded systems for a wide range of products. In other words : An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task.

Prerequisite

Any Technical Graduate with interest of working in core electronics

C and Data structures

Introduction to 'C' language

- Features of C
- History
- Structure of C Program
- Keyword, Identifiers & Constants

Data types

- Primitive Data Types
- Aggregated Data Types

Operators

- Binary Operators
- Unary Operators
- Ternary Operators
- Special Operators
- Order of Evaluation

Selections

- Simple if
- if..else
- Nested if
- if..else ladder
- Goto Statement
- Break and Continue Statement
- Switch..Case statement

Iteration

- While
- For
- Do..While
- Nested loop Statements

Arrays

- Introduction to arrays
- Need for Arrays
- Types of arrays
- One Dimensional Arrays
- Two Dimensional Arrays
- Multi Dimensional Arrays

String manipulation

- Declaring String
- Initializing String
- String FunctionsString Formatted Specifiers
- Multiple Strings

Functions

- Interdiction to Functions
- Need for Functions
- Classification of Functions
- Function Prototype
- Defining Function
- Calling Function
- Function with Arrays
- Function with Strings
- Recursive Functions

Storage class specifiers

- Automatic
- Extern
- Static
- Register

Structures, unions, enum

- Introduction to structures
- Declaring a Structure
- Introduction to Structures
- Structures with Arrays
- Structures with Function
- Nested Structures
- Introduction to Union
- Declaring Union
- Difference between Structures and Unions
- Enumerations
- Typedef

Focus Training Services
#301, 3rd floor, Nagasuri Plazza,
Satyam theatre Road Ameerpet
Above bank of India (BOI)

Embedded Systems Content

Focus Training Services
#301, 3rd floor, KSR Towers,
Madhapur,
Beside Indian Oil Petrol pump

Pointers

- Introduction to Memory
- Introduction to Pointers
- Operations on Pointers
- Pointer to Pointer
- Pointer to Array
- Array to Pointers
- Void pointers
- Call by Value and Call by Reference
- Passing Pointers to Functions
- Functions returning Pointers
- Pointer to Functions
- Dynamic Functions Call with Function Pointer
- Pointers with Structures

Dynamic memory allocation

- Allocation (Malloc, Calloc & Realloc)
- De – Allocation (Free)

Files

- Introduction
- File Input, Output Operations
- Sequential Files
- Random Access Files
- Command Line Arguments
- Handling Errors
- Database vs File System

Data Structures (DS)

- Introduction
- Stacks using Arrays
- Stacks using Linked List
- Queues using Arrays
- Queues using linked List
- Circular Queues
- De – Queues
- Singly Linked List
- Circular Linked List
- Doubly Linked List
- Infix, Prefix and Postfix Expressions
- Trees
- Binary Trees
- Directed Graphs, Undirected Graphs
- Searching Methods – BFS, DFS
- Searching Techniques – Linear, Binary
- Sorting Techniques – Bubble, Insertion,
- Selection, Quick, Heap and Merge

8051 and Embedded C

- Introduction to Embedded Systems
- History & Need of Embedded System
- Basic components of Embedded System
- Hardware Classification of Embedded System
- Programming Language Classification of Embedded System
- Advantage & Disadvantage of Low level & High level
- programming language of Embedded System

Mircoprocessor and Microcontroller Classification

- Difference between Microprocessor & Microcontroller
- Classification based on architecture
- Classification based on Instruction Set
- Type of Microcontroller
- Memory Classification

Introduction to 8051 microcontroller

- Introduction of ATMEL 8051 family
- Block diagram description of AT89C51
- Special feature of AT89C51
- Pin description of AT89C51

Registers & Memory of AT89C51

- Description of RAM
- Description of CPU Registers
- Function of SFR

AT89C51 Development Board

- Board Features
- Schematic reading

Introduction to Software Development Tools for 8051

- Introduction to IDE
- Components of IDE
- Keil µVision IDE
- Introduction to Proteus
- Using Proteus Simulator

Embedded Systems Content

Assembly language programming of AT89C51

- Addressing modes of AT89C51
- Directives of Assembly Language
- Data Transfer Instruction
- Jump Instruction
- Arithmetic Instruction
- Logical Instruction
- Branching Instruction

Interfacing of LED

- Introduction of LED's
- Interfacing Circuit Description of LED's
- Programming of LED's Interfacing

Interfacing of 7-Segment Display

- Introduction to 7 Segment Display
- Types of 7 Segment Display
- Interfacing Circuit Description of 7 Segment Display
- Programming of 7 Segment Display Interfacing

Interfacing of LCD

- Introduction to 16 x 2 LCD
- Commands of 16 x 2 LCD
- Interfacing Circuit Description of 16 x 2 LCD
- Programming of 16 x 2 LCD

Interfacing of switches and keyboard matrix

- Introduction to Switches & Keyboard Matrix
- Interfacing Circuit of Switches & Keyboard Matrix
- Programming of Keyboard Matrix & Switches
- Controlling of LED's by using Switches
- Key board Matrix & LCD Interfacing Program

Interfacing of Motors

- Introduction to Motors
- Types of Motors used in Embedded System
- Programming & Controlling of Motors in Embedded System

Timers and counter programming

- Introduction to Timer & Counter
- Difference between Timer and Counter
- Description of SFR associated with Timer & Counter
- Programming of Timer & Counter

Serial communication programming

- Introduction to Serial Communication
- Types of Serial Communication
- Description of SFR associated with Serial Communication
- Introduction & Interfacing of UART
- Programming of UART

Interrupts Programming

- Introduction to Interrupts
- Types of Interrupts
- Programming of Software & Hardware Interrupts

Interfacing of ADC

- Introduction to ADC
- Interfacing circuit of ADC
- Working & Interfacing of Temperature Sensor (DS1621 & Lm35)

Interfacing of external memory

- Introduction to External Memory Interfacing
- Introduction to I2C Protocol
- Using I2C library to read/write External Memory

Introduction of Embedded C

- Introduction to Embedded C
- Different between C & Embedded C
- Data Type of Embedded C
- Operators of Embedded C
- Statements & Loops of Embedded C

Interworking of Assembly & Embedded C

- Inline Function
- Inline Assembly Routines

Programming and interfacing using Embedded C

- Programming of Timer & Counter
- Programming of Serial Port
- Programming of Interrupt
- LCD Interfacing
- Motor Interfacing
- Key board Matrix Interfacing

PIC Microcontroller

Introduction to PIC

- History
- PIC Families
- How to choose/select a PIC microcontroller

PIC16F877 Introduction

- Architecture and Memory Organization
- Register Memory Organization
- EEPROM/Flash Memory Organization
- Input/Output Ports and TRIS Registers
- Timer Modules
- CCP Modules
- USART Modules
-
- Advanced Features in Peripheral Interface Controller
- Instruction Sets
- Data Transfer Group Instructions
- Arithmetic and Logic Operations Group
- Bit Operation Instructions
- Program Flow Control Group

PIC Programming

- PIC16F877 board features
- How to burn bootloader code using PICkit2 programmer
- MicroC PRO IDE for PIC
- PIC On-Chip peripheral Programming : Ports: Input/Output,
- Timers & Counters, USART, I2C,SPI, A/D converter, Interrupts
- Device Interfacing : LEDS, Switches, Seven Segment Display,
- LCD, Keypad Matrix

ARM Processor & 32-bit MCU

ARM Introduction

- What is ARM?
- Family, Architecture, Core
- History of ARM
- Pipeline
- Processor Modes
- Registers
- Exception Handling
- ARM/Thumb instruction set
- Jave byte code engine (Jazelle)

LPC2148 Features & Architecture

- LPC2148 features
- LPC2148 architectural overview
- ARM7TDMI-S processor
- On-chip Flash
- In-System Programming (ISP)
- Bootloader
- In-Application Programming (IAP)
- On-chip SRAM
- LPC2148 Block diagram
- Memory Addressing
- Memory Acceleration Module
- Pin Configuration
- Pin Connect Block

LPC2148 Development Board

- Board features
- Schematic reading

Getting Started with Keil µVision

- ARM Compiler toolchain
- Creating a new µVision Project
-

- µVision Workspace and Building the Project
- Simulation with Debug Mode

LPC2148 On-chip peripheral Programming

- Assembly and Embedded C programming
- Startup code
- Phase locked loop
- VPB Divider
- GPIO/FGPIO
- Vectored Interrupt Controller
- External Interrupts
- UART
- Analog to Digital Converter
- Digital to Analog Converter
- Timer
- Pulse Width Modulation
- Real Time Clock
- Watchdog Timer
- Serial Peripheral Interface (SPI)
- Synchronous Serial Port (SSP)
- Inter-Integrated Circuit (I2C)
- System Control

Device Interfacing

- LED
- Buzzer
- Switches
- Potentiometer
- LM35 Temperature Sensor
- LDR Sensor
- IR Sensor
- Alphanumeric LCD
- DC Motor
- Servomotor
- EEPROM
- Headphone
- Speaker
- PS/2 Keyboard
- SD Card with FAT filesystem
- VGA

Cortex-M Series MCUs

- Cortex M0/M3/M4 overview
- CMSIS Standard
- On-chip peripheral Programming : GPIO
- SysTick
- Timers
- ADC
- I2C
- SPI
- SSP
- UART
- WDT
- SYSTEM CONTROL
- NVIC
- I2S
- Quadrature Encoder
- Direct Memory Access
- USB
- CAN
- Ethernet

RS232/RS485/RS422 Standard

Microcontroller Debugging Techniques

- Sensors
- 7-Segment Display
- Multiplexed 7-Segment Display
- 14-Segment Display
- Keypad
- Stepper Motor
- DS1307 RTC
- MMA7455 Accelerometer
- GSM
- GPRS
- GPS
- RF Module
- RFID
- Infrared
- Zigbee
- Bluetooth
- WiFi/WLAN
- Graphical LCD
- Touchscreen
- Biometric Fingerprint
- Data encryption

REAL TIME OPERATING SYSTEM (RTOS)

Introduction

- Operating system
- Monolithic vs Microkernel Architecture
- Why use an RTOS?

- What should be considered when choosing an RTOS?
- Types of RTOS
- Features and Advantage of RTOS
- Introduction to various RTOS available in market

RTOS Concepts

- What is task?
- Multitasking
- Task Control Block
- Task Scheduling and algorithm
- Timer Tick
- Scheduler
- Context Switch
- Task states
- Foreground and background systems
- Task Priorities
- Non-preemptive kernel and pre-emptive kernel
- Data sharing/reentrancy
- Resources and shared resources
- Critical section
- Deadlock
- Starvation
- Priority Inversion
- Priority inheritance
- Task Synchronization
- Binary Semaphore
- Counting Semaphore
- Mutual Exclusion (Mutex) Semaphore
- Intertask Communication
- Message Queues
- Interrupts, interrupt latency and Jitter
- Timer Management
- Memory Management
- I/O Management

Introduction to FreeRTOS

- Free RTOS overview
- Why choose FreeRTOS?
- Understanding the FreeRTOS Architecture and features
- Coding Standard and Style

Cygwin and GNU toolchain Installation

FreeRTOS Programming on LPC2148 Development Kit

- Creating and deleting a task
- Start scheduler
- Using the task parameter
- Task control
- Priorities
- Priority-equally tasks
- Forcing a context switch
- Starvation
- Creating a queue
- Reading in a queue
- Writing to a queue
- Creation of a binary semaphore
- Taking a binary semaphore
- Giving a binary semaphore
- Gatekeeper task and mutex
- Priority inheritance
- Creation of a counting semaphore
- Taking a counting semaphore
- Giving a counting semaphore
- Use of Banker's al gorithm
- Multiple reader/writer problem
-
- Managing interrupts using a binary semaphore
- Suspend interrupts
- Stop the scheduler
- Suspend scheduler
- Resume scheduler
- Memory management
- Prototypes
- Memory allocated once for all
- Constant sized and numbered memory
- Free memory allocation and deallocation
- Implementing a command using CLI

Linux Internals

Introduction to Linux

- Overview of Operating System
- Types of kernel
- The GNU Project and the Linux kernel
- Advantages of Linux

Embedded Systems Content

- Linux Architecture
- Linux Flavors

Shell commands

Vim Editor

Cscope and ctags with Vim

Shell scripting

- What is shell scripting?
- Importance of shell scripting
- Different types of shells
- Creating Shell script
- Making shell script executable
- Variables
- Operators
- Decision making
- Looping

GNU make & Makefile

Soft and Hard Link

Static and Shared Libraries

Linux Boot Process

Filesystem Hierarchy Standard(FHS)

System Calls

- Communicating with the kernel
- APIs, POSIX and the C Library
- Syscalls
- System Call Handler
- System Call Implementation

The Virtual Filesystem

- Filesystem Abstraction Layer
- Linux Filesystems
- The super_block structure
- Superblock operations
- The inode structure
- Inode operations
- The dentry structure
- Dentry operations
- The file structure
- File operations
- Datastructures associated with filesystems
- Datastructures associated with a Process

Process Management

- What is Process?
- Types of processes
- Process related commands
- Process Descriptor
- Process State
- The task_struct structure
- Process creation
- Process scheduling
- context switching
- Process priority
- Process termination
- Symmetric multiprocessing(SMP)
- Threads
- Memory management
- Logical and physical address space
- Virtual Memory
- MMU and TLB
- Contiguous memory allocation
- Paging
- Segmentation
- Segmentation with paging
- Page fault
- Demand Paging
- Swapping
- Fragmentation
- vmalloc()
- kmalloc()

POSIX Threads Programming

- What are pThreads?
- The pThreads API
- Creating and terminating Threads
- Passing arguments to Threads
- Joining and Detaching Threads
- Stack Management
- Mutex Variables
- Condition Variables

Signals

- Signal types and default actions
- signal() system call
- Signal Handlers
- sigaction() system call

Inter Process Communication

- Pipes
- Fifo
- Semaphores
- Message queue
- Shared memory

Block I/O Layer

- Anatomy of a block device
- Buffers and Buffer heads

- The bio structure
- Request Queues
- I/O Schedulers

Sockets

- Client-server model
- TCP/IP connection
- Sockets as means for IPC
- System calls related to TCP and UDP sockets
- TCP client-server
- UDP client-server
- Netlink socket interface

Debugging

- strace
- ltrace
- mtrace
- gdb
- ddd
- core dump analysis

Board Bringup & Embedded Linux Porting

Embedded Linux and ARM Processors

- Introduction to Embedded Linux
- Embedded Linux vs Desktop Linux
- Embedded hardware overview
- Embedded Linux components
- Embedded Linux distributions
- What is ARM?
- Family, Architecture, Core
- History of ARM
- Introduction to ARM9, ARM11, Cortex-A8 and Cortex-A9
- Overview of FriendlyARM mini2440 board features
- Raspberry Pi board features
- Beaglebone Black board features
- Radxa rock lite board features

Toolchain

- What is toolchain?
- Toolchain components
- Types of toolchain
- Prebuilt toolchain
- Toolchain building systems
- Environmental variables
- GNU Make & Makefile
- Lab : Configuring and Compiling the ARM Compiler toolchain manually
- Cross-compiling HelloWorld.c example for ARM Processor

U-Boot

- Bootloader
- Role of a bootloader
- Bootloader on x86
- Bootloader on embedded architectures
- U-Boot directory structure
- Porting U-Boot to a new board
- Configuring, Compiling and Installing U-Boot
- U-Boot shell
- U-Boot commands
- Environmental variables
- Creating boot-scripts
- Introduction to Barebox
- Lab : Adding our own command

Kernel

- Linux Kernel directory structure
- Board file
- Device Tree structure
- Device Tree overlay
- Linux Kernel porting to a new board
- Kernel build system
- Lab : Configuring, Compiling and Installing Linux Kernel

Root file system

- FHS: File System Hierarchy Standard
- File System Layout
- Minimal root file system
- Manually building your root file system
- Busybox
- Cross-compiling applications and libraries
- The init Process
- Initrd vs Initramfs

- Lab : Manually building our own root file system
- Booting from TFTP-NFS
- Booting from SD Card

Buildroot

- Embedded Linux build system
- Introduction to Buildroot
- Buildroot directory structure
- Lab : Adding a new package
- Configuring Buildroot and Building (external toolchain)
- Booting from TFTP-NFS
- Booting from SD Card
-
-

- Configuring Buildroot and Building (buildroot toolchain)
- Booting from TFTP-NFS
- Booting from SD Card

Yocto

- Introduction to Yocto
- Lab : Configuring Yocto and Building
- Booting from TFTP-NFS
- Booting from SD Card

Flashing Prebuilt images

- Lab : Flashing a prebuilt image onto eMMC and Booting
- Flashing Debian distribution onto the SD Card and Booting
- Flashing Angstrom distribution onto the SD Card and Booting
- Flashing Android 4.4 KitKat onto the SD Card and Booting
- Flashing Android 4.4 Kitkat onto the SD Card using fastboot and Booting

Eclipse

- Eclipse and the GNU Cross Development Toolchain
- Work on a remote project with Eclipse via ssh/scp
- Debugging with Eclipse
- Lab : Test demo examples

Qt with Embedded Linux

Linux Device Drivers

Introduction to Linux device drivers

- Role of the device driver
- Splitting the kernel
- Classes of devices and modules

Kernel Module

- Introduction to kernel module
- Kernel module vs Application
- Writing a simple module
- Writing makefile
- Module related commands
- Version dependency
- Module parameters
- Namespace pollution
- Exporting symbols from module
- The kernel module and the GPL

Character Device Drivers

- Major and minor number
- Char dev registration
- The cdev structure
- The file_operations structure
- The file structure
- Creating device files with mknod
- GPIO driver

Procfs

- Virtual files and Directores under /proc
- Managing procfs entries
- Read and Write operation on proc entries
- Sysctl

Kobjects and Sysfs

- Virtual Directores under /sys
- Managing sysfs entries
- Read and Write operation on sys entries
- Controlling LED
- Controlling PWM
- Controlling ADC

Hotplugging with udev

- uevent
- udev rules

Embedded Systems Content

Communicating with Hardware
- I/O Ports
- I/O Memory

Interrupts and Interrupt Handlers
- Interrupts
- Interrupt Handlers
- Implementation of interrupt handling
- Top and bottom halves
- Softirq
- Tasklet
- Work queue

Kernel Synchronization
- Atomic Operations
- Spinlocks
- Semaphores
- Mutex
- Completion Variables

Block Device Drivers
- Anatomy of a block device
- Buffers and Buffer Heads
- The bio structure
- Request Queues
- I/O Schedulers
- Block device registration
- The gendisk structure
- Block device operations
- Ramdisk driver

Network Drivers
- Network Architecture
- The sk_buff structure
- The net_device structure
- Network device registration
- Network device operations
- Network card driver

Bus Driver
- I2C driver

Miscellaneous Driver
- PWM driver

Kernel Debugging

Internet Of Things (IoT)

Introduction to IoT
- Why IoT? How IoT is changing the world
- CC2650 wireless MCU Features
- Simplelink Bluetooth Smart SensorTag CC2650
- Android smartphone : Official SensorTag app
- Connect sensors to the cloud
- Node and Gateway
- IPv6 advantages for IoT
- BLE, ZigBee and 6LoWPAN
- I2C bus protocol
- MQTT protocol
- Python and Linux
- TI-RTOS
- Contiki
- RIOT

PROJECT

Build your own Home Automation System

Control Appliances through a simple mobile app

Software Development Life Cycle

What is SDLC?
- Different SDLC Models
- Tasks carried on and documents prepared in different phases of SDLC
- namely Feasibility Study, Analysis, Design, Implementation, Testing, and
- Maintenance

FINAL PROJECT