

# INTUITIVE EMBEDDED TECH

Interactive Tech Education

[www.intuitiveembedded.com](http://www.intuitiveembedded.com)

Call: 9900721303 / 9892616426 | Email: [info@intuitiveembedded.com](mailto:info@intuitiveembedded.com)

## Advanced Embedded System (FreeRTOS, ARM, Cortex)

**Total Duration for Complete Course:-**

60 Hours (Only On SATURDAY and SUNDAY)

**NOTE:** Hands-On will be done on LPC2148 (ARM 7) and LPC1769 (Cortex – M3)

**Pre-Requisite:** 1) must needed Strong C Programming Skill  
2) Understanding of 8051 is needed  
3) Understanding of OS Concepts will be add-on.

### Course Outline

Duration: (60 Hours)

#### 1. Getting started

1. Micro-controller Vs Micro-processor
2. Embedded C introduction
3. in-line Assembly
3. Cross-Compilation
4. General 'C' vs Embedded 'C'
5. Start-up Code and Linker script
6. Eclipse and CCS (Code Composer Studio)
7. How Micro-Controller Works?

#### 2. ARM Introduction

1. ARM Architecture
  1. Introduction, Features, Nomenclature, What and Why THUMB, Pipelined Architecture
  2. CISC & RISC, why RISC?
  3. Load and store Architecture, programmer's Model and Registers Model.
  4. Block Diagram, Operating States, Operating Modes and Access levels.
  5. Handling Exception, Handling Interrupt, NVIC, Interrupt priority, switching between Privileged and unprivileged mode, Interrupt Vs Exception.
  6. Bus protocols and Bus interfaces, AMBA Bus. Aligned and Unaligned data Transfer.
  7. Memory Map, Boot-loader, ISP, IAP, and Understanding Various Clock Source, PLL etc.
  8. Various Addressing modes, ARM Instruction Set, Thumb Instruction Set
  9. Installing ARM-GCC Toolchain, GNU Compiler, Cygwin, Eclipse, Makefile, Start-up file, Linker Script etc.
  10. Intro to stack memory, Stack memory model, Subroutine and stack, stacking and unstacking During Exception.
  11. Understanding Executable file format.
  12. Understanding various memory section (text, data, stack, BSS, etc.)

## 2. LPC2148 (ARM 7)

1. LPC2148 Architecture and Data-Sheet Reading
  2. LPC2148 (ARM7) Block Diagram, Pin Description, Schematic Reading
  3. GPIO Programming (Led, Buzzer, Switches)
  4. Interrupt Programming (FIQ, IRQ, Vectored Interrupt, and Non-Vectored Interrupts)
  5. UART programming (Polling Mode, Interrupt Mode)
    1. Practical Command Line Interface using Serial communication.
    2. Implementation of debug function / printf function.
  6. Timers & Counter (Match Mode, Capture Mode, Timer Interrupts, Delays)
  7. PWM Programming (Motor driving program)
  8. RTC Programming (Normal, Alarm Mode)
    1. RTC Block, Registers, RTC Interrupts, RTC alarm Mode
  9. I2C Programming (EEPROM Interfacing)
  10. SPI Programming (EEPROM Interfacing)
  11. ADC & DAC Programming (Interfacing Sensor to input of ADC and Application programming)
    1. ADC Overview, Registers, Programming, DAC Overview, registers & programming
  12. WDT (Watchdog Timer)
    1. Overview, Block Diagram, Registers, Usages & Programming
- ### 3. Peripheral Interfacing
1. EEPROM Interfacing with I2C and SPI
  2. LCD Interfacing and Programming
  3. Temp Sensor Interfacing

## 3. Cortex Introduction (Basic)

1. ARM Vs Cortex
2. Cortex Architecture Overview
3. Cortex Features over ARM TDMI
  1. Bit-Banding
  2. CMSIS
  3. Interrupt Handling
  4. MPU
4. Cortex programming using LPC1769 (Cortex-M3)
  1. GPIO
  2. UART
  3. Interrupt
  4. I2c, SPI
  5. ADC, DAC, PWM
5. Various peripherals interfacing with LPC1769
  1. Same as ARM LPC2148.

## 5. FreeRTOS

1. RTOS Introduction
  1. What is RTOS? Real-Time Concepts.
  2. Conventional OS Vs RTOS? (Linux Vs RTOS)
  3. Introduction to Real-Time Concepts, Hard Real Time, Soft Real Time
  4. RTOS Internals & Real Time scheduling & scheduling policies
  5. Performance Metrics of RTOS, Latency in Linux, Priority Inheritance
  6. Operating system structure, kernel Architecture, system calls
  7. Task Specifications, Schedulability Analysis, Task Management (TCB, Task creation etc...)
  8. Application Programming on RTOS,
  9. Porting of RTOS, Configuring RTOS,
  10. Building RTOS Image for Target platform

## 2. Getting Started with FreeRTOS

### 1. Tasks

1. What is a task?
2. Crude loop periodic delay, task parameter, priorities, starvation...
3. Creation, states, priorities, idle task, deletion...
4. TaskDelay, taskDelayUntil, polling vs. event driven, idle hook...

### 2. Scheduling

1. Determinism, multitasking, endless loop, cyclic executives, issues with interrupts, Non-/pre-emptive, prioritized pre-emptive, rate monotonic, deadline, cooperative, hybrid

### 3. Queue Management

1. What is a queue?
2. Creation, send, receive...
3. Blocking on read/write...
4. With multiple tasks blocking who will run first?
5. Large sets of data and queues.
6. Indirect/Direct synchronous and asynchronous message passing.

### 4. Interrupt Management

1. What is Interrupt vs. Polling?
2. Hardware-, Software-interrupts.
3. What if two interrupts arrive at the same time?
4. What is re-entrant code?
5. What is an event?
6. What kind of events do you know?
7. What is a Semaphore and what can it be used for?
8. What kind of Semaphores do you know?
9. Signal/Wait pattern, state diagram, event states of binary semaphore...
10. Counting semaphores
11. Queues and ISRs
12. Deferred interrupt processing, interrupt handlers, XXXFromISR(), task with Interrupt synchronization, efficient queue usage even from within an Interrupt, interrupt nesting...

### 5. Resource Management

1. Concurrency, concurrent read/writes
2. Mutual exclusion, critical sections, suspending/locking the scheduler, mutexes, priority inversion, priority inheritance, deadlock, gate-keeper tasks, mutex vs. semaphore...

### 6. Memory Management

1. Memory types, fragmentation, memory exhaustion, memory allocation patterns, Dynamic memory Allocation in kernel
2. Memory allocation schemes,
3. xPortGetFreeHeapSize(),...
3. Using FreeRTOS with Cortex Micro-controller.

### 7. Other fancy Stuff with FreeRTOS.

## 6. Real-Time Project

**THANK YOU**