Keonics Certified Advanced Embedded Systems

Duration: 3 Months

1. Introduction to Embedded Systems

1.1 Brief Introduction
1.2 Classifications of Embedded Systems
1.3 Characteristics of an Embedded System
1.4 Applications of an Embedded Systems

2 Introduction to Embedded C Programming

2.1 Assembly Vs. C
2.2 What is Embedded C
2.3 C Vs. Embedded C
2.4 C Programming
2.4.1 Data types, operators and Variables
2.4.2 Storage Classes
2.4.3 Stack Frame Analysis
2.4.4 Functions
2.4.5 Arrays
2.4.6 Pointers
2.4.7 Structures
2.4.8 Advance C Programming Level 1
2.4.9 Pointers and Arrays
2.4.10 Pointers and Functions
2.4.11 Variable Argument Function
2.4.12 Memory Management
2.4.13 Structures
2.4.14 Structures and Array
2.4.15 Structures and Functions
2.4.16 Unions
2.4.17 Enumerations
2.4.18 Macros
2.4.19 GDB – Debug tool
2.5 Advance C programming level 2
2.5.1 Introduction to Data Structures
2.5.2 Types of Arrays
2.5.3 Linked Lists
2.5.4 Singly Linked List
2.5.5 Doubly Linked Lists
2.5.6 Circular lists
2.5.7 Sorting algorithms
2.5.8 Applications of Sorting Algorithm
2.5.9 Searching Algorithm
2.5.10 Stacks

2.5.11 Applications of Stack
2.5.12 Queues
2.5.13 Applications of Queues
2.5.14 Deque
2.5.15 Circular Queues
2.5.16 Priority Queues
2.5.17 Trees

3 Basics of Micro controllers

3.1 What are Micro Processors?
3.2 What is Micro Controller?
3.3 Difference between Micro controller and Microprocessor
3.4 Difference between Von Newman and Harvard Architectures
3.5 Difference between CISC & RISC Architectures
3.6 Overview of MCS51 Family

4 Intel 8051 Architecture

4.1 Features of 8051 micro controller
4.2 Internal ROM memory Architecture
4.3 Internal RAM Memory Architecture
4.4 Pin out detail of 8051 micro controller
4.5 Instruction Set
4.5.1 Data Transfer Instruction Set
4.5.2 Arithmetic Instruction Set
4.5.3 Bitwise and Byte wise Logical Instruction Set
4.5.4 Rotate Instruction Set
4.5.5 Call and Jump Instruction Set
4.5.6 Stack Operation and its architecture

5 Programming 8051 in Assembly

5.1 Introduction to Assembly level Programming
5.2 Assembly Programming
5.3 IO Port Programming
5.4 Timer and Counter Programming
5.5 Serial Port Programming
5.6 Interrupt Handling

6 Keil C Programming and Interfacing 8051 with peripherals and devices

6.1 Need of Cross Compiler
6.2 Popular Cross Compilers
6.3 What and Why Compiler Directives
6.4 Overview of µ vision Development. Tools
6.5 Environmental settings required to run a Keil C program
6.6 Introduction to Keil C
6.7 Intermixing C and Assembly
6.8 Interfacing Techniques

6.8.1 IO Port Interface
6.8.2 Interfacing LED
6.8.3 Interfacing Liquid Crystal Display (LCD)
6.8.4 Interfacing a Keypad
6.8.5 Interfacing EEPROM
6.8.6 Interfacing ADC
6.8.7 Interfacing Sensors
6.8.8 Interfacing DAC
6.8.9 Interfacing Stepper Motor
6.8.10 Interfacing DC Motor
6.9 Advanced Programming Issues
6.10 What is and Why Optimization.
6.11 What are 8051 Specific Optimization supported by Cx51
6.12 Difference between Keil C and ANSI C

7 Embedded Protocols

7.1 RS232
7.1.1 Difference between Serial and Parallel Communication
7.1.2 Introduction RS232 Protocol
7.1.3 Introduction to MAX232
7.1.4 UART Programming
7.1.5 Advantages and Disadvantages of RS 232
7.1.6 Applications of RS232
7.2 I2C
7.2.1 Introduction and Features to I2C
7.2.2 I2C Frame Format
7.2.3 Interfacing I2C based peripherals with Micro controller
7.2.4 Advantages and Disadvantages of I2C
7.2.5 Applications of I2C
7.3 RTC
7.3.1 Introduction and Features of RTC
7.3.2 Programming RTC
7.3.3 Advantages and Disadvantages of RTC
7.3.4 Applications of RTC
7.4 CAN
7.4.1 Introduction and Features of CAN
7.4.2 Characteristics of CAN network
7.4.3 Bus States
7.4.4 CAN Layers
7.4.5 CAN Frame Formats
7.4.6 Inter Frame Space, Bus Idle
7.4.7 Message Validation
7.4.8 Type of Errors

7.4.9 CAN Device States
7.4.10 Interfacing CAN Controllers to a micro controller
7.4.11 Using micro controllers with inbuilt CAN Controllers
7.4.12 Advantages and Disadvantages of CAN
7.4.13 Applications of CAN

8 ARM Processors

8.1 Introduction to ARM 7 Architecture
8.2 ARM instructions set
8.3 Thumb instructions set
8.4 Programmer’s model
8.5 Memory and system architecture
8.6 ARM 7 mini project

9 Overview of operating system

9.1 Process Management System
9.2 Memory Management
9.3 Device Management
9.4 File Management
9.5 Network Management
9.6 Introduction to Linux device drivers