

20ES007 - Principles of Embedded Software

UNIT I:

Introduction: Compilers, Languages, and Object-Oriented Programming Embedded System Development – Debugging – More Challenges - Principles to Confront Those Challenges - Creating System Diagrams - The Block Diagram - Hierarchy of Control - Layered View - From Diagram to Architecture - Encapsulate Modules - Delegation of Tasks - Driver Interface: Open, Close, Read, Write, IOCTL - Adapter Pattern - Getting Started with Other Interfaces - Example: A Logging Interface.

UNIT II:

Hardware/SoftwareIntegration: IdealProjectFlow - BoardBring-Up - ReadingaDatasheet. - Evaluating Components Using the Datasheet - Your Processor Is a Language - Reading a Schematic - Having a Debugging Toolbox (and a Fire Extinguisher) - Keep Your Board Safe - Keep Your Board Safe – Toolbox - Digital Multimeter - Oscilloscopes and Logic Analyzers - Testing the Hardware (and Software) - Building Tests - Flash Test Example - Command and Response - Command Pattern - Dealing with Errors - Consistent Methodology - Error-Handling Library - Debugging Timing Errors.

UNIT III: Outputs, Inputs, and Timers:

Toggleing an Output - Starting with Registers - Set the Pin to Be an Output - Turn On the LED - Blinking the LED – Troubleshooting - Separating the Hardware from the Action - Board-Specific Header File - I/O-Handling Code - Main Loop - Facade Pattern - The Input in I/O - A Simple Interface to a Button - Momentary Button Press - Interrupt on a Button Press - Configuring the Interrupt - Debouncing Switches - Runtime Uncertainty - Dependency Injection - Using a Timer - Timer Pieces – Doing the Math.

UNIT IV: Managing the flow of Activity:

Scheduling and Operating System Basics – Tasks - Communication Between Tasks - Avoiding Race Conditions - Priority Inversion - State Machines - State Machine Example: Stoplight Controller - State-Centric State Machine - State-Centric State Machine with Hidden Transitions - Event-Centric State Machine - State Pattern - Table-Driven State Machine - Choosing a State Machine Implementation.

UNIT V : Interrupts andCommunicatingwithPeripherals

An IRQ Happens - Save the Context - Get the ISR from the Vector Table - Calling the ISR - Restore the Context - When to Use Interrupts - How Not to Use Interrupts – Polling - System Tick - Time-Based Events - A Very Small Scheduler - The Wide Reach of Peripherals - External Memory - Buttons and Key Matrices – Sensors – Actuators – Displays - PuttingPeripheralsandCommunicationTogether DataHandling - AddingRobustnesstotheCommunication ChangingData - Changing Algorithms.

TEXT BOOKS:

1. Making Embedded Systems (O'Reilly) - Elecia White, 2012, ISBN-13: 978-1-449-30214-6

2. Embedded Systems Fundamentals with ARM Cortex-M based Microcontrollers: A Practical Approach (ARM Education Media UK) - Alexander G. Dean, 2017, ISBN-13: 978-1911531036

Lab Experiments

To be able to learn about various C Programming tools

To be able to learn about Arduino in detail.

1. Introduction to C Programming tools
2. Using Standard I/O
3. Using Conditionals
4. Using Loops
5. Intro to Addresses, Pointers and Handles
6. Interfacing with Arduino
7. Arduino Digital Output
8. Arduino Digital Input
9. Arduino Analog Input
10. Arduino Reaction Timer
11. Arduino Reaction Timer Redux
12. Arduino Analog Output via PWM
13. Arduino Event Counter
14. Arduino Arbitrary Waveform Generator