

RTOS with PIC Microcontroller Programing

24 Hours

Course Description

This comprehensive course is designed to teach you the fundamentals of PIC microcontroller programming from scratch, making it suitable for beginners without any prior programming knowledge. You will not only learn how to program PIC microcontrollers but also gain hands-on experience building your own hardware by utilizing microcontrollers and other circuit components. The course emphasizes register-level coding, ensuring you have a deep understanding of PIC microcontroller programming without relying on code configurators or peripheral libraries.

Additionally, this course delves into Real-Time Operating System (RTOS) concepts, using the open-source FreeRTOS kernel. You will explore multitasking, task scheduling, intertask communication, and more. The code you write throughout the course will run on the hardware you build, making you proficient in both hardware and firmware development.

Audience

- Beginners in electronics seeking to learn PIC microcontroller programming.
- Engineering students looking to enhance their knowledge of embedded systems.
- DIY enthusiasts interested in building and programming their own hardware projects.
- Professionals in the embedded systems field who want to deepen their understanding of register-level coding and RTOS integration.

Pre-requisite Knowledge/Skills

- Hardware components as listed in the course materials.
- PICKIT3 Programmer.
- Computer with MPLABX IDE.
- No prior programming experience required; the course covers everything you need to know.

Course Objectives

By the end of this course, you will:

- Have a solid understanding of PIC microcontroller programming from scratch.
- Be proficient in building your own hardware projects and circuits.
- Be able to write code for basic tasks by referring to device datasheets.
- Gain expertise in register-level coding for PIC microcontrollers.
- Understand and implement various functionalities without relying on code configurators or peripheral libraries.
- Be familiar with Real-Time Operating System (RTOS) concepts using FreeRTOS.
- Learn multitasking, task scheduling, intertask communication, and memory management.

- Develop software for hardware you have built, ensuring a complete understanding of both hardware and firmware development.

Course Outline

The course comprises 24-hours of theory and labs. It's divided into 5 different modules.

Section 1: Course Introduction

- Overview of the course
- Introduction to PIC microcontrollers
- Hardware components and software installations

Section 2: Circuit Design and Hardware Setup

- Building circuits for PIC microcontrollers
- Power supply considerations
- Basic connection requirements
- Bluetooth connections
- Using LEDs for GPIO
- LCD interface
- Flashing the PIC microcontroller
- Introduction to KiCAD PCB files

Section 3: PIC Microcontroller Programming

- Installing the XC32 Compiler for PIC32M
- General Purpose Output (GPIO)
- General Purpose Input (GPIO)
- Core Timer
- External Interrupts
- Input Change Notification
- LCD Interface
- Building a Quiz Buzzer
- Timer1 and 32-Bit Timers
- Watchdog Timer
- Input Capture
- Output Compare
- Analog-to-Digital Converter (ADC)

- UART for BLE
- MPLAB Harmony Configurator (MHC)
- Assignments and quizzes

Section 4: Introduction to FreeRTOS

- Understanding FreeRTOS
- Setting up FreeRTOS projects in MPLABX
- Task management and scheduling
- Inter-task communication
- Idle task and critical sections
- Memory management
- Software timers
- Interrupt management
- BLE tasks demonstration
- Resource management
- Mutexes

Section 5: Building RTOS Projects with FreeRTOS

- Hands-on projects integrating FreeRTOS into PIC microcontroller applications