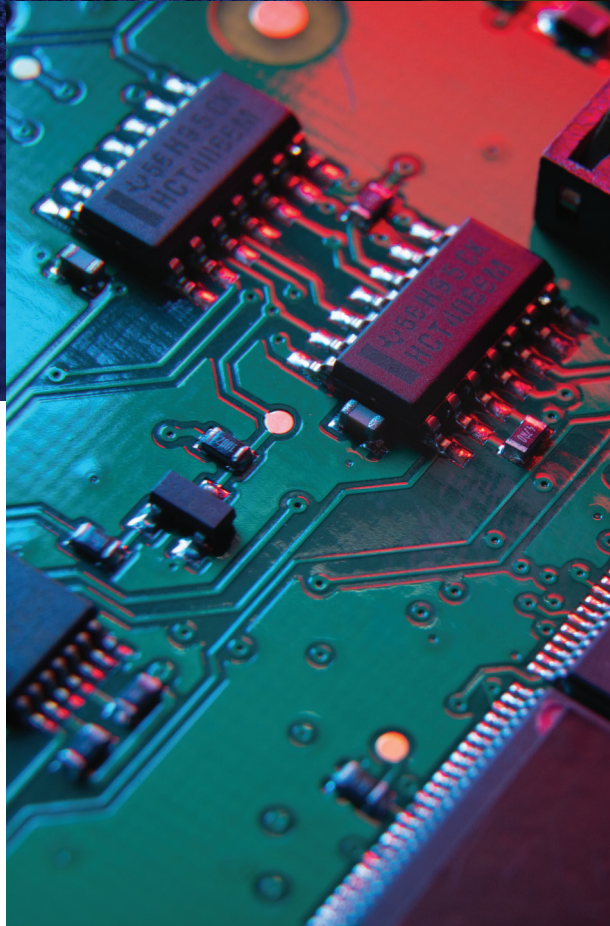


ESSENTIALS OF MICROCONTROLLER PROGRAMMING



Length: 3 Days

Hardware: STM32 Microcontroller from STMicroelectronics

Software: STM32CubeIDE, C/C++ Compiler

Training Board: STM32MP157F-DK2

ENROLL NOW

OVERVIEW

This course provides a fundamental understanding of embedded C programming for STM32 microcontrollers. Participants will explore the essentials of microcontroller architecture, GPIO programming, interrupts, and peripheral interfacing. Through hands-on labs using STM32CubeIDE, attendees will gain practical experience in writing efficient and reliable embedded C code.

COURSE OBJECTIVE

After completing this course, you will:

- Understand the basics of embedded systems and the STM32 microcontroller architecture.
- Gain proficiency in programming GPIOs, handling interrupts, and utilizing peripherals.
- Be able to develop and debug embedded C applications using STM32CubeIDE.
- Acquire practical experience through hands-on labs with STM32 Development Board.

• TARGET AUDIENCE

Embedded system developers, electronics engineers, and enthusiasts entering the field of microcontroller programming with a focus on STM32.

• PREREQUISITES

Basic knowledge of C programming.

TEACHING METHODOLOGY

- **Instructional strategies** include lectures, discussions, demonstrations, and workshop-based learning which are used to convey information, facilitate learning, and promote understanding.
- **Technology integration** hands-on labs, and interactive discussions.
- **Learning activities** Practical exercises with STM32 Development Board to reinforce theoretical concepts.

COURSE FRAMEWORK

DAY 1 : Introduction to STM32 Microcontrollers and GPIO Programming

- ◇ Overview of Embedded Systems and STM32 Architecture
- ◇ Introduction to STM32CubeIDE
- ◇ GPIO Configuration and Programming
- ◇ **Lab exercise:** Blinking LED using GPIO (Lab1 Manual)

DAY 2 : Interrupts and Peripheral Interfacing

- ◇ Understanding Interrupts and NVIC
- ◇ Configuring and Handling Interrupts
- ◇ Introduction to STM32 Peripherals (e.g., UART, Timer) and Peripheral Interfacing
- ◇ **Lab exercise:** Signal-Driven Processes (Lab2 Manual)

DAY 3 : Peripheral Programming and Project Development

- ◇ Advanced Peripheral Configuration (e.g., SPI, I2C)
- ◇ Memory-mapped Registers and Bit Manipulation
- ◇ Real-world Project Development
- ◇ **Lab exercise:** Implementing a Practical Project (Lab3 Manual)