EMBEDDED SYSTEMS

6-Month Professional Training Program • 70% Practical + 30% Theory

Program	Embedded Systems Professional Training
Duration	6 Months
Approach	70% Practical + 30% Theory
Languages	Embedded C / C++
Microcontrollers	8051, Arduino Uno, PIC16F877A
Tools	Keil, Proteus, Arduino IDE, Wokwi, MPLAB X, XC8, PICSimLab
Training Style	Hands-on Labs + Mini Projects + Weekly Assessment + Final Project
Eligibility	Diploma, BE/B.Tech, ME/M.Tech, BSc/MSc in Electronics

Course Learning Path

- 01 Month 1 — Embedded C & 8051 Fundamentals
- 02 Month 2 — 8051 Interfacing & Peripherals
- 03 Month 3 — Arduino Fundamentals & Hardware Interfacing
- 04 Month 4 — Communication Protocols & Arduino Projects
- 05 Month 5 — PIC Microcontroller
- 06 Month 6 — Advanced Concepts & Final Project

Program Structure

- Foundation: Embedded Systems concepts, Embedded C, registers, memory and microcontroller architecture.
- Interfacing: Digital/analog I/O, displays, keypads, timers, interrupts, UART and sensors.
- Rapid Prototyping: Arduino programming, Wokwi simulation, sensors and actuators.
- Communication: UART, I2C, SPI and CAN fundamentals with practical applications.
- Microcontroller Development: PIC16F877A firmware using PICSimLab, MPLAB X, XC8 and Proteus.
- Industry-Oriented Integration: debugging, datasheet reading, RTOS concepts, Git/GitHub and final project development.

Training Outcome

By the end of the program, students should be able to design, program, simulate, test, debug and document practical embedded-system applications using multiple microcontroller platforms.

EMBEDDED SYSTEMS

Month 1 — Embedded C & 8051 Fundamentals

Module	01 — Embedded C & 8051 Fundamentals
Goal	Build a strong foundation in Embedded C programming, microcontroller concepts, registers, memory, and 8051 architecture.
Level	Beginner to Intermediate
Main Focus	Theory fundamentals + practical implementation + mini project
Tools	Keil + Proteus + 8051 Development Board

Module 01 Learning Path

- ■ Introduction to Embedded Systems
- ■ Microcontroller vs Microprocessor
- ■ Embedded C Fundamentals
- ■ Variables, Data Types and Operators
- ■ Conditional Statements and Loops
- ■ Functions, Arrays and Strings
- ■ Pointers and Structures
- ■ Bitwise Operations
- ■ Registers and Memory Concepts
- ■ 8051 Architecture and I/O Ports

Practical Labs

- ■ LED Blinking and LED Patterns
- ■ Push Button and Buzzer Interfacing
- ■ 7-Segment Display Interfacing
- ■ Switch-Based Applications

Mini Projects

- ■ LED Pattern Controller
- ■ Switch-Controlled Buzzer System

Module Completion Checklist

Requirement	Complete
Understand the module theory concepts	■
Complete the listed practical laboratory exercises	■
Verify the circuit/application in the selected tool or hardware	■
Complete at least one mini project	■
Document the implementation and observations	■
Complete the weekly assignment/test requirements	■

Module Outcome

Students can apply the module concepts in practical embedded-system development, simulation, testing and debugging.

EMBEDDED SYSTEMS

Month 2 — 8051 Interfacing & Peripherals

Module	02 — 8051 Interfacing & Peripherals
Goal	Develop practical skills in 8051 peripherals, timing, interrupts, serial communication, ADC/PWM fundamentals, and sensor interfacing.
Level	Beginner to Intermediate
Main Focus	Theory fundamentals + practical implementation + mini project
Tools	Keil + Proteus + 8051 Hardware

Module 02 Learning Path

- ■ LCD and Keypad Interfacing
- ■ Timers and Counters
- ■ Interrupts
- ■ UART / Serial Communication
- ■ ADC and PWM Fundamentals
- ■ Sensor Interfacing

Practical Labs

- ■ LCD Display Application
- ■ 4x4 Keypad Interface
- ■ Timer/Counter Application
- ■ Interrupt-Based Input Handling
- ■ UART Serial Communication
- ■ Sensor Data Display

Mini Projects

- ■ Digital Counter and Digital Clock
- ■ Password-Based Security System
- ■ Temperature Monitoring System
- ■ Serial Communication Application

Module Completion Checklist

Requirement	Complete
Understand the module theory concepts	■
Complete the listed practical laboratory exercises	■
Verify the circuit/application in the selected tool or hardware	■
Complete at least one mini project	■
Document the implementation and observations	■
Complete the weekly assignment/test requirements	■

Module Outcome

Students can apply the module concepts in practical embedded-system development, simulation, testing and debugging.

EMBEDDED SYSTEMS

Month 3 — Arduino Fundamentals & Hardware Interfacing

Module	03 — Arduino Fundamentals & Hardware Interfacing
Goal	Learn Arduino Uno architecture, Arduino IDE programming, I/O, sensors, actuators, PWM, and rapid prototyping.
Level	Beginner to Intermediate
Main Focus	Theory fundamentals + practical implementation + mini project
Tools	Arduino Uno + Arduino IDE + Wokwi

Module 03 Learning Path

- ■ Arduino Architecture and Arduino Uno
- ■ Arduino IDE, Digital and Analog I/O
- ■ Serial Communication and PWM
- ■ Functions and Libraries
- ■ Sensors and Actuators

Practical Labs

- ■ LEDs, RGB LEDs, Push Buttons and Potentiometers
- ■ LDR and Temperature Sensor
- ■ IR Sensor
- ■ Buzzer, Servo Motor, DC Motor and Relay
- ■ Ultrasonic Sensor

Mini Projects

- ■ Automatic Street Light
- ■ Obstacle Detection System
- ■ Temperature Monitoring System

Module Completion Checklist

Requirement	Complete
Understand the module theory concepts	■
Complete the listed practical laboratory exercises	■
Verify the circuit/application in the selected tool or hardware	■
Complete at least one mini project	■
Document the implementation and observations	■
Complete the weekly assignment/test requirements	■

Module Outcome

Students can apply the module concepts in practical embedded-system development, simulation, testing and debugging.

EMBEDDED SYSTEMS

Month 4 — Communication Protocols & Arduino Projects

Module	04 — Communication Protocols & Arduino Projects
Goal	Understand common embedded communication protocols and use Arduino to build multi-device communication projects.
Level	Beginner to Intermediate
Main Focus	Theory fundamentals + practical implementation + mini project
Tools	Arduino Uno + Arduino IDE + Wokwi

Module 04 Learning Path

- ■ UART: TX/RX, Baud Rate, Data Transmission and Reception
- ■ I2C: Master-Slave Concept, SDA/SCL, Addressing
- ■ SPI: MOSI, MISO, SCLK, CS/SS
- ■ CAN Basics: Bus Architecture, Nodes and Messages
- ■ Automotive Applications of CAN
- ■ Multi-Device Communication

Practical Labs

- ■ UART Communication
- ■ I2C LCD/OLED Interfacing
- ■ I2C Sensor Interfacing
- ■ SPI Device Interfacing
- ■ Multi-Device Communication

Mini Projects

- ■ Smart Home Automation
- ■ Sensor Monitoring System
- ■ Serial-Controlled Device

Module Completion Checklist

Requirement	Complete
Understand the module theory concepts	■
Complete the listed practical laboratory exercises	■
Verify the circuit/application in the selected tool or hardware	■
Complete at least one mini project	■
Document the implementation and observations	■
Complete the weekly assignment/test requirements	■

Module Outcome

Students can apply the module concepts in practical embedded-system development, simulation, testing and debugging.

EMBEDDED SYSTEMS

Month 5 — PIC Microcontroller

Module	05 — PIC Microcontroller
Goal	Develop practical PIC16F877A programming and interfacing skills using MPLAB X, XC8, Proteus, and a PIC development board.
Level	Beginner to Intermediate
Main Focus	Theory fundamentals + practical implementation + mini project
Tools	PICSimLab + MPLAB X IDE + XC8 + Proteus + PIC Development Board

Module 05 Learning Path

- ■ PIC Architecture — PIC16F877A
- ■ GPIO and Registers
- ■ ADC and Timers
- ■ Interrupts
- ■ PWM
- ■ LCD and Keypad
- ■ UART
- ■ Sensor Interfacing

Practical Labs

- ■ LED, Switch and LCD Interfacing
- ■ ADC-Based Sensor Reading and Temperature Monitoring
- ■ Motor Control
- ■ UART Communication
- ■ PWM Applications

Mini Projects

- ■ ADC-Based Temperature Monitor
- ■ PWM Motor Controller
- ■ UART Sensor Monitor

Module Completion Checklist

Requirement	Complete
Understand the module theory concepts	■
Complete the listed practical laboratory exercises	■
Verify the circuit/application in the selected tool or hardware	■
Complete at least one mini project	■
Document the implementation and observations	■
Complete the weekly assignment/test requirements	■

Module Outcome

Students can apply the module concepts in practical embedded-system development, simulation, testing and debugging.

EMBEDDED SYSTEMS

Month 6 — Advanced Concepts & Final Project

Module	06 — Advanced Concepts & Final Project
Goal	Integrate microcontrollers, peripherals, communication protocols, debugging, documentation, and project development into a complete embedded system.
Level	Beginner to Intermediate
Main Focus	Theory fundamentals + practical implementation + mini project
Tools	All Previous Tools + Git/GitHub

Module 06 Learning Path

- ■ Inter-Microcontroller Communication — UART / I2C / SPI
- ■ CAN Fundamentals and Interrupt-Driven Programming
- ■ Timers, PWM, ADC and Sensor Integration
- ■ Debugging Techniques and Datasheet Reading
- ■ Basics of Embedded System Design
- ■ Introduction to RTOS Concepts
- ■ Git/GitHub Basics

Practical Labs

- ■ Inter-Microcontroller Communication
- ■ Interrupt-Driven Sensor Application
- ■ Multi-Sensor Integration
- ■ Protocol-Based Device Control
- ■ Debugging and Datasheet Exercise
- ■ Git/GitHub Project Workflow

Mini Projects

- ■ Smart Home Automation System
- ■ IoT-Based Temperature Monitoring System
- ■ Smart Parking System
- ■ Obstacle Detection and Alert System
- ■ Automated Street Light System
- ■ Industrial Temperature and Safety Monitoring System

Module Completion Checklist

Requirement	Complete
Understand the module theory concepts	■
Complete the listed practical laboratory exercises	■
Verify the circuit/application in the selected tool or hardware	■
Complete at least one mini project	■
Document the implementation and observations	■
Complete the weekly assignment/test requirements	■

Module Outcome

Students can apply the module concepts in practical embedded-system development, simulation, testing and debugging.

EMBEDDED SYSTEMS

Final Project & Assessment

Final Project — Complete Embedded System

Students will develop a complete embedded system integrating a microcontroller, Embedded C, sensors, actuators, display, communication protocols, and interrupts/timers.

Suggested Projects

- ■ Smart Home Automation System
- ■ IoT-Based Temperature Monitoring System
- ■ Smart Parking System
- ■ Obstacle Detection and Alert System
- ■ Automated Street Light System
- ■ Industrial Temperature and Safety Monitoring System

Recommended Final Project Flow

Start → Requirement Definition → Block Diagram → Component Selection → Circuit Design → Firmware Development → Simulation → Hardware Integration → Debugging → Testing → Documentation → GitHub → Final Demonstration → End

Assessments

- ■ Weekly Test: Conducted at the end of every week based on topics covered.
- ■ Weekly Assignment: Embedded C programs, circuit simulations, microcontroller programming, sensor interfacing, debugging exercises, protocol-based tasks and mini-projects.
- ■ Practical Evaluation: Demonstration of selected laboratory experiments and mini projects.
- ■ Final Project Evaluation: Working demonstration, firmware quality, debugging approach, documentation and project presentation.

Overall Program Exit Criteria

Requirement	Complete
Write and understand Embedded C programs	■
Explain microcontroller architecture and memory concepts	■
Interface LEDs, switches, displays, keypads, sensors and actuators	■
Use timers, counters, ADC, PWM and interrupts	■
Implement UART communication	■
Understand and implement I2C and SPI basics	■
Explain CAN fundamentals and bus architecture	■
Develop applications using 8051, Arduino and PIC16F877A	■
Use Keil, Proteus, Arduino IDE, Wokwi, PICSimLab, MPLAB X and XC8	■
Apply debugging techniques and read datasheets	■
Understand basic RTOS concepts	■
Use Git/GitHub for project version control	■
Complete and demonstrate a final embedded system project	■

Professional Outcome

Learn online. Build real projects. Develop practical embedded-system skills. Become industry-ready.

Institute	Embedded Electronics
Website	https://embedded-electronics.github.io/