

Hands-on, Trainer-led, Offline Course

Embedded Systems with AI

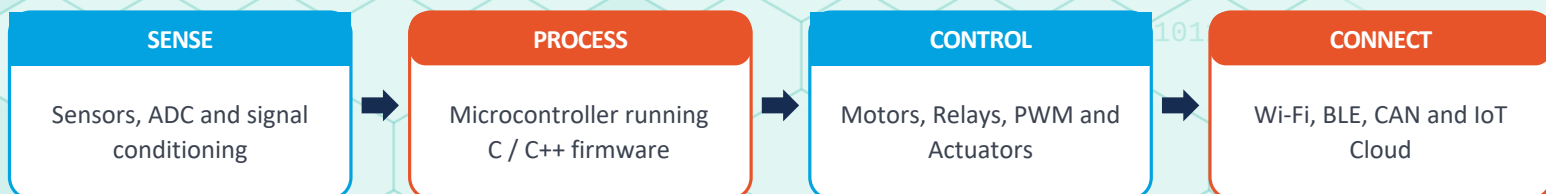
Build smart devices that sense, think and act - from microcontrollers to Edge AI.

Different domains. One technology. A product approach.

From the phone in your pocket to satellites in orbit, embedded systems quietly run the world.



How every embedded system works?



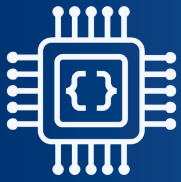
You build and program every one of these layers in this program.

100%
Hands-on

Kit
Included

Capstone
Projects

On/Offline
Options



Embedded Systems

Engineering Program · Arduino to Embedded AI

Duration: 26–32 Weeks (6–8 Months) | 2 Sessions × 1.5 Hours Daily | 17 Minor + 17 Major Projects

Mode: Offline & Online | Hands-on | Project-based Learning

Embedded systems are the hidden computers inside cars, medical devices, home appliances, drones and factory machines. This program takes you from basic electronics and C programming to building complete embedded products writing C and C++ that talks directly to hardware, then moving from Arduino and ESP32 to professional STM32, Raspberry Pi boards, Real-time operating systems, Linux networking and Embedded AI. Every concept is applied the same day in our hardware lab.

Electronics + C/C++ Programming + Hardware + Embedded AI

Course Details:

ELECTRONICS & PROGRAMMING	MICROCONTROLLERS, SYSTEMS & AI
- Module 1 Basic Electronics - Module 2 Analog & Digital Circuits - Module 3 Sensing & Measurement - Module 4 Systems Fundamentals - Module 5 C Programming & Data Structures - Module 6 Embedded C - Module 7 GPIO Programming - Module 8 C++ for Embedded - Module 9 Debugging	- Hardware Lab Arduino & ESP32 - Module 10 ARM & STM32 - Module 11 RTOS - Module 12 Basic Linux - Module 13 Shell Scripting - Module 14 TCP/IP Networking - Module 15 Python - Module 16 Artificial Intelligence - Module 17 Embedded AI

17 Minor Projects + 17 Major Projects + 34 Assessments

Each module features one minor project, one major project, and assessments to reinforce practical skills.

Pursue your dream of becoming an Embedded Systems Engineer with hands-on skills in electronics, C/C++ firmware, microcontrollers, RTOS, Linux, and Embedded AI. Our hardware-first program combines daily theory sessions and practical labs, building your verified portfolio through 34 projects (17 minor and 17 major) and 34 assessments, culminating in a final integrated demo.



Module 1 - Basic Electronics

- Electronic Components and Circuits
- Voltage, Current and Ohm's Law
- Power Supplies and Circuit Protection
- Resistors, Capacitors and Diodes
- Multimeter and Oscilloscope Basics

Module 2 - Analog & Digital Circuits

- Analog Electronic Components
- Diodes, Transistors and Op-Amps
- Logic Gates and Number Systems
- Combinational and Sequential Circuits
- Signal Conditioning and Circuit Design

Module 3 - Sensing & Measurement

- Sensors and Transducers
- Analog and Digital Signal Measurement
- Sensor Interfacing Techniques
- Voltage, Current and Signal Monitoring
- Data Acquisition and Calibration Basics

Module 4 - Systems Fundamentals

- Embedded System Architecture
- Signals, Timing and Interfaces
- State Machines and System Behaviour
- Hardware and Software Interaction
- System Design Fundamentals

Module 5 - C Programming & Data Structures

- C Programming Fundamentals
- Arrays, Strings and Pointers
- Functions, Structures and Unions
- Linked Lists, Stacks and Queues
- Searching, Sorting and Memory Management

Module 6 - Embedded C

- Embedded C Programming Fundamentals
- Bitwise Operations and Registers
- Memory Maps and Peripheral Control
- Interrupts and Resource Management
- Firmware Development and Debugging

Module 7 - GPIO Programming

- GPIO Input and Output Configuration
- Digital Pin Control and Register Access
- LED, Switch and Button Interfacing
- Pull-Up, Pull-Down and Interrupt Handling
- GPIO-Based Hardware Control Projects

Module 8 - C++ for Embedded

- Classes, Objects and Constructors
- Object-Oriented Programming Concepts
- Inheritance and Polymorphism
- References and Templates
- Hardware Abstraction and Modular Design

Module 9 - Debugging

- Hardware and Software Debugging
- Multimeter and Oscilloscope Usage
- Breakpoints and Serial Log Analysis
- Fault Identification and Root-Cause Analysis
- Firmware Testing and Error Resolution

Hardware Lab - Arduino & ESP32

- Arduino Programming and Digital I/O
- Sensor and Actuator Interfacing
- PWM and Motor Control
- ESP32 Wi-Fi and Bluetooth Communication
- Hardware Integration and Prototyping

Module 10 - ARM & STM32

- ARM Cortex-M Architecture
- STM32 GPIO and Peripheral Programming
- Timers, PWM, ADC and DMA
- UART, SPI, I²C and CAN Protocols
- STM32 Firmware Development and Debugging

Module 11 - RTOS

- RTOS Fundamentals and Architecture
- Task Creation and Scheduling
- Queues, Semaphores and Mutexes
- FreeRTOS Timers and Interrupt Handling
- Real-Time Multitasking Projects

Module 12 - Basic Linux

- Linux Commands and File Management
- Files, Permissions and Processes
- Linux System Utilities
- GCC, Make and GDB Fundamentals
- Embedded Linux Development Basics

Module 13 - Shell Scripting

- Shell Commands and Variables
- Conditional Statements and Loops
- Functions and Script Automation
- File and Log Management
- Automated Testing and System Tasks

Module 14: TCP/IP Networking

- Networking Fundamentals and Protocol Layers
- IP Addressing, Ports and TCP/UDP
- Socket Programming
- Client-Server Communication
- Device Connectivity and Remote Diagnostics

Module 15: Python

- Python Syntax and Data Types
- Lists, Dictionaries and Functions
- File Handling and Data Processing
- Serial Communication and Device Control
- Automated Testing and Report Generation

Module 16: Artificial Intelligence

- AI and Machine Learning Fundamentals
- Data Collection and Preprocessing
- Features, Training and Validation
- Classification and Anomaly Detection
- Model Evaluation and Performance Metrics

Module 17: Embedded AI

- Sensor Data Collection and Labelling
- AI Model Deployment on Embedded Devices
- Model Quantization and Optimization
- RAM, Flash and Latency Management
- Real-Time AI Inference on Microcontrollers

Real-time Projects

1 Minor + 1 Major Project in Every Module

- **Embedded C:** bare-metal sensor/actuator platform
- **C++:** modular hardware service framework
- **Analog:** multi-sensor analog front end
- **Digital:** digital access-control unit
- **Debugging:** diagnostic firmware & crash report
- **ARM & STM32:** industrial data acquisition
- **RTOS:** real-time monitoring controller
- **TCP/IP:** networked device-monitoring gateway
- **Linux:** edge-gateway service
- **Python:** automated hardware test & report tool
- **AI:** sensor anomaly-detection model
- **Embedded AI:** optimised intelligent edge node

SIGN OFF

Final Integrated Demo

- Working hardware, end to end
- Arduino → ESP32 → STM32 → RTOS/Linux → AI
- Presented and explained to faculty

***Learner to sign NDA for the intellectual property*

Engineering Portfolio

- Code repositories & Project reports
- Measurements, Screenshots & Demos
- Verified LMS Record of all assessments

Program Outcomes

- Build circuits - Basic, Analog & Digital electronics
- Programming - C, Data Structures, C++ & Emb. C
- Control hardware - From Arduino to ESP32 & STM32, Raspberry Pi
- Debug systems - Instruments, logs & debuggers
- Connect devices - RTOS, Linux, TCP/IP & scripting
- Add Intelligence - Python, AI & Embedded AI

How You Learn

- Session 1 (90 min): Concept + Guided demo
- Session 2 (90 min): Concept + Hands-on Lab Review
- Primary and backup faculty every day
- Your own LMS login and progress dashboard

How You Progress

- 2 assessment in every subject (34 in total)
- Assignment + minor project + major project
- Faculty review before the next stage
- Explain the circuit, code and measured result

Hardware ⇄ Software Communication

Work on Hardware - Real-time projects encourage you to write code that talks directly to hardware setting register bis to drive GPIO pins, building UART, I2C and SPI drivers in Embedded C, handling interrupts and DMA, and wrapping peripherals in reusable C++ classes. Every program is checked on a real board with a Multimeter /oscilloscope /logic analyser or ST-LINK debugger.