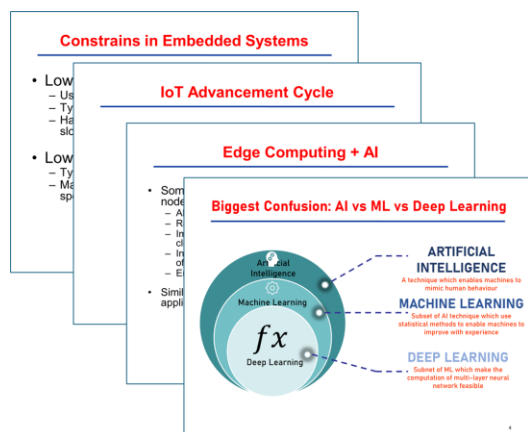


ME2110

Embedded Systems and AI Courseware

Teaching slides

- Editable Microsoft® PowerPoint® slides
- Covers 45 hours of teaching



Training kit

- Multicores MCUs with AI Accelerators
- Hardware Building Blocks
- Lab sheets & model answers
- Problem-based assignments
- Covers 18 hours of labs



Target university subject	Target year of study	Prerequisite(s)
Embedded System Design, Embedded Systems & AI	3 rd -year undergraduate	Microcontroller and Microprocessor Systems, Python and C++ programming

ME2110 Embedded Systems and Artificial Intelligence introduces students to the design, implementation, and validation of real-time embedded systems, culminating in the deployment of embedded machine learning models on microcontroller platforms with hardware acceleration.

The course emphasizes system-level integration, combining sensing, signal processing, machine learning inference, and actuation under realistic memory, performance, and power constraints.

Learning Outcomes:

1. Design and implement embedded systems using microcontrollers, timers, communication interfaces, sensors, and actuators.
2. Analyse and validate real-time system behaviour using appropriate software tools and measurement instruments.
3. Train, optimize, and deploy lightweight neural networks on embedded platforms with hardware acceleration.
4. Evaluate trade-offs between accuracy, memory usage, and performance in embedded AI applications.
5. Integrate sensing, inference, and actuation into a continuous real-time execution workflow.

Benefits of the ME2110 courseware

- **Strong embedded systems competence**, including microcontroller configuration, peripheral interfacing, real-time execution, and hardware validation.
- **Practical experience deploying embedded AI**, covering the full lifecycle from dataset preparation and CNN training to model quantization and deployment on resource-constrained microcontrollers with hardware acceleration.
- **System-level engineering skills**, enabling students to integrate sensing, inference, and actuation into continuous real-time systems and reason about execution behavior, timing, and performance.
- **Ability to evaluate engineering trade-offs**, including accuracy versus memory footprint, performance versus power consumption, and software versus hardware-accelerated execution.
- **Industry-relevant, hands-on experience**, aligned with modern embedded and edge-AI applications such as robotics, intelligent sensing, and automation.
- **Improved job readiness**, providing transferable skills applicable to embedded firmware development, embedded AI/TinyML, robotics, and edge computing roles.



Teaching Slides

More than 400 editable Microsoft PowerPoint teaching slides, covering 45 hours of teaching for one full semester, are provided. The slides cover the following topics:

- Overview of embedded system design, artificial intelligence, and edge AI.
- Analog and digital sensor interfacing techniques, covering signal conditioning circuits, analog-digital (ADC), and digital-analog (DAC) conversions.
- Introduction to timers, interrupts, and commonly used communication protocols in embedded systems design (I2C, SPI, UART, etc.)
- Actuator: controlling external devices (relay, motor, etc.) and managing display.
- Machine learning hardware and software frameworks for deploying on embedded devices (TinyML).
- Building neural networks and Deep learning (DL) applications.
- Case studies



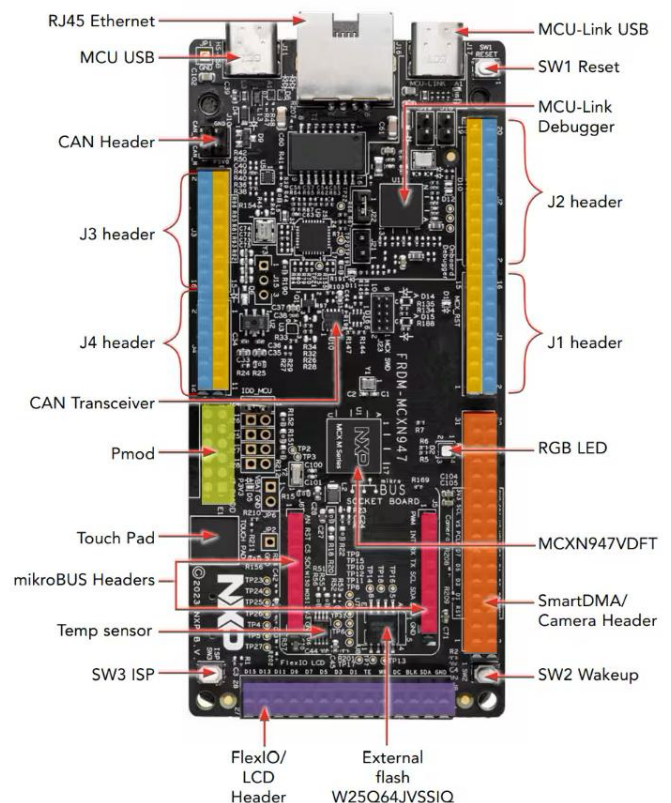
Training Kit

MCXN947 MCUs development board

FRDM-MCXN947 is a compact and scalable development board for the rapid prototyping of MCX N94 and N54 MCUs. They offer industry-standard headers for easy access to the MCU's I/Os, integrated open-standard serial interfaces, external flash memory, and an on-board MCU-Link debugger.

Features

- Microcontroller
 - MCX-N947 Dual Arm™ Cortex™ -M33 cores @150MHz each with optimized performance efficiency, Up to 2MB dual-bank flash with optional full ECC RAM, External flash
 - Accelerators: Neural Processing Unit, PowerQuad, Smart DMA, etc.
- Memory Expansion: DNP Micro SD card socket
- Connectivity
 - Ethernet Phy and connector
 - HS USB Type-C connectors
 - SPI/I2C/UART connector (PMOD/mikroBUS, DNP)
 - WIFI connector (PMOD/mikroBUS, DNP)
 - CAN-FD transceiver
- Debug: On-board MCU-Link debugger with CMSIS-DAP, JTAG/SWD connector
- Sensor: P3T1755 I3C/I2C Temp Sensor, Touch Pad
- Expansion Options
 - Arduino® Header (with FRDM expansion rows)
 - FRDM Header
 - FlexIO/LCD Header
 - SmartDMA/Camera Header
 - ™ *DNP
 - mikroBUS™
- User Interface: RGB user LED, plus Reset, ISP, Wakeup buttons



Software Tools and Frameworks

The MCUXpresso IDE will be used for developing embedded system software (C programming) in this courseware. This IDE provides developers with a user-friendly Eclipse-based development environment tailored for NXP® MCUs. Additionally, the MCUXpresso IDE's debug connections are compatible with all general-purpose Arm® Cortex-M® based EVKs and custom development boards. These connections are supported by optimized open-source and commercial debug probes from NXP, P&E Micro®, and SEGGER®.

TensorFlow is an open-source software library for numerical computation that makes machine learning faster and easier. It is designed to facilitate research in deep neural networks, as well as to allow for the easy deployment of machine learning-based applications in production environments. TensorFlow uses a system of multi-layered nodes that allows you to quickly set up, train, and deploy artificial neural networks with large datasets.

Lab sheets

Labs progress from MCU configuration and peripheral interfacing to system integration and embedded AI deployment with hardware acceleration. The labs introduce machine learning workflows, including dataset preparation, CNN training, quantization, and deployment of neural networks on embedded targets with hardware acceleration.

Each lab is estimated to require 3 hours to complete. Model answers are provided with all lab sheets.

Lab Sheet	Hardware Kit					Accelerometer
	FRDM-MC94 Board	DC Motor Driver + Motors	Camera and LCD	Analog and Digital Sensors	Oscilloscope (note 1)	
1 Introduction to MCU Programming & GPIO Introduces microcontroller project setup and GPIO control, focusing on correct hardware initialization, timing control, and verification of real-time execution through observable outputs.	√				√	
2 Analog Data Acquisition Using ADC Develops skills in analog signal acquisition using ADC peripherals, including configuration, data sampling, conversion to physical values, and validation of sensor measurements during runtime execution.	√			√	√	
3 Sensor Interfacing and Data Display Covers digital sensor interfacing using communication protocols (e.g. I ² C) and real-time data presentation on displays, emphasizing multi-peripheral integration and stable execution behavior.	√		√		√	√
4 PWM and Motor Control Introduces pulse-width modulation for actuator control, enabling students to control motor speed and direction while analyzing timing behavior, duty-cycle effects, and validating physical system response.	√	√			√	
5 An attendance record system using a face detection CNN - Convert a pre-trained face detection CNN into the appropriate format (e.g., ONNX). - Convert the CNN into a hardware-friendly format using the NXP eIQ Toolkit. Execute the CNN inference and record the performance. - Compare the difference in performance between software implementation and hardware acceleration.	√		√			
6 Train a small CNN to perform a specific task and deploy it on a microcontroller - Develop and train a small CNN for detecting square objects on a host machine. - Convert the CNN into a hardware-friendly format using the NXP eIQ Toolkit. Execute the CNN inference and record the performance. - Deploy the network and compare the inference speed: with and w/o the NPU acceleration.	√		√			

Note 1: not provided, minimum 50MHz, 2 analog channels.

Hardware Building Blocks and Accessories

The following accessories are provided with the training kit.

1. ADXL345 3-axis Accelerometer Breakout
2. Infrared sensor module
3. VGA Camera Module
4. 4-Wheel Robot Base with Simplifying H-Bridge Motor Driver
5. Linear temperature Sensor
6. 3.5" 480x320 IPS TFT LCD Module.
7. USB-C-USBA cable

Problem-based assignments

Students are required to design systems that operate in a continuous real-time execution model, involving hardware initialization, sensor or camera data acquisition, embedded inference, output generation, and actuation.

The problem-based assignments below allow students to enhance their problem-solving skills.

1. AI-Based Robotic Control with Hand Detection

The final system shall detect a hand from the live camera image, display the detection result on the LCD, and control the robot according to the detected hand position.

2. Fruits classification

This assignment mimics a production line that classifies certain types of fruits and controls the motor movement accordingly. Train a CNN model to perform fruit classification with acceptable accuracy. Deploy the trained CNN model on MCXN947. Apply appropriate techniques to utilize the hardware accelerator available in a microcontroller.

3. Investigating Embedded AI Platform Constraints - hand gesture recognition

This assignment uses the deployment of an Embedded AI application on the MCXN947 platform as an experimental vehicle to identify and analyse the practical constraints of running AI on resource-constrained embedded systems

Ordering Information

Description	Package	Product Number
Teaching Slides	1 user license	ME2110-100
Training Kit	1 unit	ME2110-200
Teaching Slides + Training Kit	1 user license + 1 unit	ME2110-300

Training courses related to subject matter are available on request. Visit dreamcatcher.asia for details.

For more information or enquiries: Website: dreamcatcher.asia/cw E-mail: cw.sales@dreamcatcher.asia Acehub Vista Sdn Bhd (785702-P) 70-03-79, D'Piazza Mall, Jalan Mahsuri 11900 Bayan Lepas, Penang Malaysia	© 2010-2011 Acehub Vista Sdn Bhd We reserve the right to change or alter the information in this material without prior notice. The information provided in this material is accurate as of the print date. Microsoft, Windows, and Office Programs are trademarks of Microsoft Corporation in the United States and/or other countries. All other copyrights and trademarks belong to their respective owners. updated on 7 July 2026
--	--