# Advanced Embedded Technology

# curriculum

Ministry of Micro, Small and Medium
Enterprises, New Delhi
(MSME-Technology Centre)

COURSE NAME: PIC MICROCONTROLLER

COURSE CODE: AET01

COURSE OUTCOMES:

After completion of course Student should be able to:

- Understand the basics of Microcontroller
- Understand architecture of PIC microcontroller
- Ability to interface with various peripherals
- Understand the product requirement based on
- Understand the concept of peripherals

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Introduction to 8-bit PIC microcontroller | After completion of unit student should be able to<br>• The candidate will be able to understand embedded hardware, design for telecom devices and equipment, core programming of telecom devices and equipment<br>• The candidate will be able to use basic communication protocols, understanding of circuits and architectures<br>• Architecture of | • Introduction to PIC Microcontroller<br>• Architecture of PIC MCU Series<br>• Application of PIC Microcontroller | 10 | 10 | |
| UNIT-II | Introduction to PIC 18F series MCU | | • Introduction to PIC MCU Series<br>• Introduction to PIC18f4550 Family<br>• Architecture of PIC18f4550<br>• Understanding the instruction set used for programming.<br>• Understanding method for writing program and debugging method | 15 | 15 | |
| UNIT-III | Interfacing with Peripherals | | • Understanding the concept of peripherals<br>• Types of Peripherals | 10 | 15 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | Microcontrollers<br>• Basic Peripherals<br>• Input & Output devices<br>• Working with sensors | • Introduction to internal peripherals of PIC microcontroller<br>• Interfacing with internal peripherals such as Timer/Counter, Serial Communication, Interrupt, CCP, EEPROM memory, ADC etc.<br>• Interfacing with external peripherals such as LED, LCD(Liquid Crystal Display), SSD(Seven Segment Display), DAC, RTC etc | | | |

COURSE NAME: EMBEDDED C/C++

COURSE CODE: AET02

COURSE OUTCOMES:

After completion of course Student should be able to:

- Understand the basic concept of C/C++ programming language.
- Understand the method of writing program in C/C++.
- Understand the concept of Embedded C/C++.
- Ability to create own C/C++ program.

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Introduction to C | After completion of unit Student should be able to<br>• The candidate will be able to prepare and maintain a knowledge-base of the known problems<br>• The candidate able to develop system by interfacing as well as developing the hardware.<br>• The candidate will be able to understand embedded | • Introduction to Programming Language<br>• Concept of C/C++<br>• Use of Decision Making Statement<br>• Use of structure and pointer<br>• Use of Arrays and Functions | 5 | 10 | |
| UNIT-II | Introduction to MPLABX compiler | | • Introduction to MPLABX Compiler<br>• Methods of Compiling and Debugging<br>• Execution of program file on hardware | 10 | 10 | |
| UNIT-III | Embedded C - Porting and delay with PIC | | • Introduction to Embedded C/C++<br>• Initialization of program using microcontroller registers<br>• Understanding the concept of port | 5 | 10 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | software like embedded C/C++, python etc.<br>• The Candidate able to use compiler used for programming the PIC Microcontroller with peripherals<br>• Basics of Programming<br>• Understanding Embedded C<br>• Memory Efficiency &<br>• Optimization<br>• Troubleshooting & Maintenance | programming and providing delay method | | | |
| UNIT-IV | Peripheral programming with Embedded C. | | • Understanding the concept of peripherals<br>• Types of Peripherals<br>• Introduction to internal peripherals of PIC microcontroller<br>• Programming the peripherals with Embedded C/C++ | 15 | 10 | |

SEMESTER: I

COURSE NAME:  COMMUNICATION PROTOCOL & IMPLEMENTATION

COURSE CODE: AET03

COURSE OUTCOMES:

After completion of course Student should be able to:

- Understand the interfacing with various communication protocol implementation
- Understand the implementation method for communication protocols
- Understanding interfacing of different peripheral using communication protocol

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Introduction to communication protocols | After completion of unit Student should be able to<br>• The candidate will be able to understand  embedded hardware,  design for telecom devices and equipment, core programming of telecom devices and equipment<br>• The candidate will be able to use basic communication protocols, understanding of | • Introduction to Communication Protocol<br>• I2C - Interfacing with micro controller using bit-banking method, I2C devices – RTC, Memory, ADC-DAC, Port-Expander,<br>• SPI (Serial Peripheral Interface), Bluetooth, Wi-Fi and RFID. | 15 | 20 | |
| UNIT-II | Interfacing with various communication protocols | | • Understanding Serial<br>• Communication<br>• Bluetooth Communication<br>• SPI Interface<br>• ZigBee | 20 | 20 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | circuits and architectures used in telecom systems and devices<br>• embedded system ,system design modules, concepts of circuit design, computer architecture, design and implementation of embedded software system | • Wi-Fi<br>• I2C<br>• Infrared<br>• RFID<br>• GSM<br>• GPS<br>• PDH/SDH/Ethernet | | | |

COURSE NAME: ROBOTICS-INTERFACING WITH MOTORS AND SENSORS

COURSE CODE: AET04

COURSE OUTCOMES:

After completion of course Student should be able to:

- Working with Motors & Sensors
- Understand the types of motors & sensors
- Ability to interface motors & sensors with microcontroller

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Introduction to ROBOTICS | After completion of unit Student should be able to<br>• Understand robotics mechanism<br>• Concept of robotics<br>• Application using robot<br>• Types of motors & sensors<br>• Understand interfacing of motors & sensors with microcontroller<br>• Application of motors & sensors in robotics field<br>• Usage of motors & sensors in | • Introduction to Robotics<br>• Application of using robots<br>• Usage of motors & sensors in Robotics | 10 | 10 | |
| UNIT-II | Concepts of DC Motor | | • DC Motor in Robotics<br>• Gear fundamentals and types of gears<br>• Characteristics of DC Motors<br>• Operation of DC Motor<br>• Electrical model of DC Motor<br>• Characteristics curves of DC Motor<br>• Types of DC Motors<br>• Hardware interface of DC Motors<br>• Speed control of DC Motor<br>• Software driver for DC Motor | 15 | 10 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | robotics vehicle and product | • Speed measurements<br>• Applications of dc motors | | | |
| UNIT-III | Concept of Stepper Motor | | • Introduction to Stepper Motor<br>• Types of Stepper Motor<br>• Operation of Stepper Motor<br>• Applications of Stepper motor<br>• Hardware interface for driving Stepper motor<br>• Software driver for Stepper motor | 5 | 10 | |
| UNIT-IV | Concept of Servo Motor | | • Concept of servo motor<br>• Use of servo motor<br>• Stepper vs. servo<br>• How does servo motor works?<br>• Futaba servo s3003<br>• Applications of servo motor<br>• Hardware interface for driving servo motor<br>• Software driver for servo motor | 5 | 10 | |

COURSE NAME: ARM- Advanced RISC Machine (LPC2148)

COURSE CODE: AET05

COURSE OUTCOMES: The aim of this course student should be able to:

- Understand the knowledge of processor
- Understand the interfacing with peripherals
- Ability to generate application based on industry requirements.
- Usage of material for generating product.

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Introduction to ARM processor | At the end of this Unit the student should be able to:<br>• The candidate will be able to prepare and maintain a knowledge-base of the known problems<br>• The candidate able to develop system by interfacing as well as developing the hardware.<br>• The candidate will be able to use peripherals of ARM processor for different applications<br>• The candidate will be able to understand the concept of instruction and execution<br>• The candidate will be able to | • Introduction to ARM Processor<br>• Architecture of ARM7TDMI Processor<br>• Advantages of 32-bits over 8-bits controller | 10 | 10 | |
| UNIT-II | ARM-7 TDMI Core architecture | | • Architecture of ARM7TDMI Processor<br>• Application of using ARM7TDMI Processor<br>• Interfacing with various peripherals | 15 | 10 | |
| UNIT-III | LPC 2148 ARM with in-built peripherals | | • LPC 2148 ARM with in-built peripherals<br>• ADC<br>• DAC<br>• RTC<br>• UART<br>• Timer/Counter | 5 | 10 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | produce devices based on ARM Processor and can use it in various field as per the requirement.<br>• The candidate will be able to use different series of ARM Processor according to requirement. | • Interrupts<br>• PWM<br>• SPI based programming<br>• LCD interfacing, Introduction to LPC1768 (Cortex-M3 series). | | | |
| UNIT-IV | Introduction to LPC1768 (Cortex-M3 series). | | • Introduction to LPC1768 (Cortex-M3 series).<br>• Architecture of LPC1768 (Cortex-M3 series) core.<br>• Interfacing with peripherals of LPC1768( Cortex-M3 series)<br>• Application of LPC1768 (Cortex-M3 series). | 5 | 10 | |

COURSE NAME: Concept of RTOS

COURSE CODE: AET06

COURSE OUTCOMES: The aim of this course student should be able to:

- Explain the Real time operating system concept.
- Know about architecture of kernel.
- Understand the concept of Process scheduling, task execution etc.
- Understand the use of operating system into microcontroller.

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Comparison between traditional O.S and RTOS | At the end of this Unit the student should be able to<br>• design flow involved in design stages , design, develop, test, debug software components<br>• software module library, system testing, product verification and validation, software programming languages such as C, C++, operating system such as windows, Linux , system level integration, software fundamentals such object-oriented design, data structures, algorithm design end-product application, i.e., industry for which embedded system is designed, Schematics and data sheets | • Introduction to real time operating system<br>• Understanding the concept of RTOS<br>• Application of Real time Operating System<br>• Comparison between OS and RTOS | 10 | 10 | |
| UNIT-II | Understanding Kernel | | • What is Kernel<br>• Architecture of Kernel<br>• Execution of task in kernel | 15 | 10 | |
| UNIT-III | Scheduling policies | | • Execution method for task<br>• Type of scheduling<br>• Policies for executing task | 5 | 10 | |
| UNIT-IV | Programming in RTOS | | • Building block policies used in RTOS<br>• Programming Concept used to execute task<br>• Different task scheduling while programming for RTOS | 5 | 10 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | • understand scheduling process and policies required to execute task<br>• understand the application of using real time operating system | | | | |

COURSE NAME: VLSI Design with FPGA / CPLD

COURSE CODE: AET07

COURSE OUTCOMES: The aim of this course student should be able to:

- Ability to simulate using FPGA blocks.
- Understand Programming using FPGA/CPLD concept
- Understand simulation using Xilinx ISE design
- Ability to implement program based on application

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| Unit - I | CPLD / FPGA architecture | At the end of this Unit the student should be able to:<br>• Understand the concept of VLSI design<br>• Types of designing for e.g. Digital design or analog design and their respective design flow<br>• Understanding the process of partitioning the design into different blocks<br>• Selection of design type such as application specific integrates circuit(ASIC), Field-programmable gate arrays(FPGA) and complex programmable logic device(CPLD)<br>• Use of design languages such as hardware design language(HDL), e.g. Verilog, VHDL, High level language such as C | • CPLD / FPGA architecture, Programming and Simulation with Xilinx ISE, VHDL programming- Full Adder<br>• Introduction to complete system architecture such as memory, microcontroller, microprocessor, memory blocks, timers and oscillators, interfaces and power management<br>• Introduction to design flow for the specific system<br>• Introduction to VHDL programming language | 10 | 10 | |
| UNIT-II | Programming and Simulation with Xilinx ISE | | • Types of languages<br>• Types of designing for e.g. Digital | 15 | 10 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | • Understanding the code required for design<br>• Creation of code, verification, testing software<br>• Testing various examples on the system<br>• Understanding the synthesis and simulation process of code<br>• Building simulation module as per system specification for e.g. VHDL model for ASIC design<br>• Understanding types of design as per sections in VLSI processor such as high level design, operative part design, control part design, memory design etc.<br>• Functioning and specifying the tools used for design as per requirement<br>• Analysis of Design code by verification engineer<br>• Installation process of design code into hardware and verifying it | design or analog design and their respective design flow<br>• Use of design languages such as hardware design language(HDL), e.g. Verilog, VHDL, High level language such as C<br>• Understanding the code required for design | | | |
| UNIT-III | VHDL programming | | • Introduction to design flow for the specific system<br>• Introduction to VHDL programming language<br>• Creation of code, verification, testing software<br>• Testing various examples on the system<br>• Understanding the synthesis and simulation process of code | 5 | 10 | |
| UNIT-IV | Programming through JTAG on Xilinx Spartan 3 Board | | • Building simulation module as per system specification for e.g. VHDL model for ASIC design<br>• Understanding types of design as per sections in VLSI processor such as high level design, operative part design, control part design, memory design etc.<br>• Functioning and specifying the tools used for design as per requirement<br>• Analysis of Design code by verification engineer<br>• Installation process of design code into hardware and verifying it | 5 | 10 | |

COURSE NAME: PCB Designing

COURSE CODE: AET08

COURSE OUTCOMES: The aim of this course student should be able to:

- Developing schematic and creating PCB layout, converting them to Gerber format using CAD and other software tools, suitable for production and assembly process.
- Understand frame the circuit specifics before drawing, by designing schematics for the circuit.
- Ability to create a new project schematic on the designing software
- Ability to add components, component values, connection between components and power connection on it
- Ability to Design PCB for different electronics products

THEORY HOURS:35 PRACTICAL HOURS:35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Schematic design with circuit simulation | At the end of this Unit the student should be able to<br>• Developing schematic and creating PCB layout, converting them to Gerber format using CAD and other software tools, suitable for production and assembly process.<br>• discuss with the systems analyst on<br>• the customer requirement and get inputs<br>• analyses the requirements of customer<br>• suggest any further changes to the requirement<br>• generate bill of materials (BOM) | • Schematic design with circuit simulation, creating footprints for customized component packages, Component<br>• Placement as per requirement, layout design for Single layer and Multilayer PCB, File generation for film making and PCB manufacturing. | 10 | 15 | |
| UNIT-II | File generation for film making and PCB manufacturing. | | • Develop schematics<br>• Convert schematics to PCB layout<br>• Create Gerber file and send to manufacturer<br>• Give the detailed instructions and information to be sent to manufacturer<br>• Achieve productivity and quality | 15 | 15 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | simultaneously with the schematic’s creation | standards | | | |
| UNIT-III | Developing schematics | • select the components in the circuit by analysing the maximum operating voltages and current levels of each node of the circuit while considering tolerance criteria<br>• reconsider based on availability, budget and size after selecting electrically satisfactory components<br>• keep the BOM up to date with the schematic at all times with all details such as quantity, reference designators, manufacturer part number, value of ohms, farads, etc. and PCB footprint for each component | • frame the circuit specifics before drawing, by designing schematics for the circuit<br>• create a new project schematic on the designing software<br>• add components, component values, connection between components and power connection on it<br>• run a check to see if there are any mistakes to be fixed<br>• ensure the schematic to be as per the original PCB design<br>• lay out clearly and logically so that it is easier at the designing stage<br>• make sure the trace, pads and via are the same sizes as the original design<br>• make short notes on the schematics in conformance with design requirements<br>• create the wiring diagram, illustration of flow of connections from input to output | 10 | 10 | |

COURSE NAME: Python Programming on Raspberry Pi

COURSE CODE: AET09

COURSE OUTCOMES: The aim of this course student should be able to:

- Understand the knowledge of Raspberry Pi
- Understand the interfacing with peripherals
- Ability to generate application based on industry requirements.
- Usage of material for generating product.

THEORY HOURS: 35 PRACTICAL HOURS: 35 THEORY MARKS: 40 PRACTICAL MARKS: 60

| Unit No. | Unit Name | Unit level outcomes | Contents (chapters/topics) | TH hours | Marks | |
|---|---|---|---|---|---|---|
| UNIT-I | Introduction to Raspberry Pi, | At the end of this Unit the student should be able to<br>• understand the work flow of the company's design process<br>• interact with the lead engineer in order to understand the work schedules, shifts and delivery dates<br>• plan work activities based the work flow and deliverables<br>• Understand broad level activities involved in the stages of design<br>• List the various department to interact with for completing the work<br>• Minimize absenteeism and report to work on time<br>• The candidate will be able to | • Introduction to Raspberry Pi<br>• Comparison with Microcontroller and Raspberry Pi,<br>• Architecture of ARM11 processor<br>• Pipelining method to execute the instruction | 10 | 10 | |
| UNIT-II | Python Programming | | • Introduction to Python programming<br>• Basic concept of Programming<br>• Method of decision making, looping, branching etc<br>• Declaration of functions, array, pointer etc<br>• Python programming using Raspberry Pi<br>• Format to write the code related application | 15 | 10 | |
| UNIT-III | Peripheral Interfacing using | | Raspberry Pi with different peripherals<br>• Port Programming | 5 | 10 | |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  | Raspberry Pi | understand the concept of instruction and execution<br>• The candidate will be able to produce devices based on Raspberry Pi<br>• Understand Putty Login Method to connect device with laptop<br>• Understand the programming concept using Raspberry Pi | • Led Interfacing<br>• LCD interfacing<br>• UART<br>• Bluetooth<br>• Interfacing of Sensors: Ultrasonic Sensor, IR Sensor etc.<br>• Interfacing with Motors:  DC Motor, Stepper Motor, Servo Motor etc. |  |  |  |
| UNIT-IV | Remote Login Method : PuTTY, Hyperterminal, Ethernet |  | • Remote login method to interlink Raspberry Pi with laptop<br>• Debugging and executing programs using Raspberry Pi | 5 | 10 |  |

COURSE NAME: PROJECT

COURSE CODE: AET09

COURSE OUTCOMES: The aim of this course student should be able to:

- Concepts to address specific management needs at the individual, team, division and/or organizational level
- Practical applications of project management to formulate strategies allowing organizations to achieve strategic goals
- A perspective of leadership effectiveness in organizations
- Team-building skills required to support successful performance
- Critical-thinking and analytical decision-making capabilities to investigate complex business problems to propose project-based solutions
- Skills to manage creative teams and project processes effectively and efficiently