

# MODEL CURRICULUM



**Qualification Name:**

**Jr. Embedded Developer**

**Qualification Code: MSME/AET/60**

**Version: 2.0**

**NSQF Level: 4.5**

**Model Curriculum Version: 2.0**

**Submitted By:**

**MSME TECHNOLOGY CENTRE**

**O/o DC MSME, Ministry of Micro, Small and Medium Enterprises**

**Govt. of India**

**A-Wing, 7<sup>th</sup> Floor, Nirman Bhawan, Maulana Azad Road**

**New Delhi-110108**

**Contact No. +91-674-2654700**

**Email- msmetcab@gmail.com**

**NOS /Module: Embedded Program Development on different microcontroller**

**NOS /Module Code: MSME/AET/01**

**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++
- Understand architecture of PIC microcontroller
- Ability to interface with various peripherals
- Working with Motors & Sensors
- Ability to interface motors & sensors with microcontroller
- Understand the interfacing with various communication protocol implementation
- Ability to generate application based on industry requirements

**Practical Hours: 120**

**th Hour: 30**

**Viva Marks: - NA**

**Practical Marks: 100**

Unit No.	Unit Name	Unit level outcomes	Contents (chapters/topics)	TH hours	PR hours
UNIT-I	Introduction to C	After completion of unit Student should be able to <ul style="list-style-type: none"> <li>• The candidate will be able to prepare and maintain acknowledge-base of the known problems</li> <li>• The candidate able to develop system by interfacing as well as developing the hardware.</li> <li>• Basics of Programming</li> </ul>	<ul style="list-style-type: none"> <li>• Introduction to Programming Language</li> <li>• Concept of C/C++</li> <li>• Use of Decision Making Statement</li> <li>• Use of structure and pointer</li> <li>• Use of Arrays and Functions</li> </ul>	30	120
UNIT-II	Introduction to MPLABX compiler	<ul style="list-style-type: none"> <li>• Understanding Embedded C</li> <li>• Memory Efficiency &amp; Troubleshooting &amp; Maintenance</li> </ul>	<ul style="list-style-type: none"> <li>• Introduction to MPLABX Compiler</li> <li>• Methods of Compiling and Debugging</li> <li>• Execution of program file on hardware</li> </ul>		
UNIT-III	Embedded C -Porting and delay		<ul style="list-style-type: none"> <li>• Introduction to Embedded C/C++</li> <li>• Understanding the concept of port programming and providing delay method</li> </ul>		

			<ul style="list-style-type: none"> <li>Understanding the concept of peripherals</li> </ul>		
UNIT-IV	Introduction to PIC18F series MCU microcontroller		<ul style="list-style-type: none"> <li>Introduction to PIC Microcontroller</li> <li>Application of PIC Microcontroller</li> <li>Introduction to PIC18f4550 Family</li> <li>Understanding the instruction set used for programming. Understanding method for writing program and debugging method</li> </ul>		
UNIT-V	Interfacing with Peripherals	<p>After completion of unit student should be able to</p> <ul style="list-style-type: none"> <li>The candidate will be able to understand embedded hardware, design for telecom devices and equipment, core programming of telecom devices and equipment</li> <li>The candidate will be able to use basic communication protocols, understanding of circuits and architectures Architecture Of Microcontrollers</li> </ul>	<ul style="list-style-type: none"> <li>Understanding the concept of peripherals</li> <li>Types of Peripherals</li> <li>Interfacing with internal peripherals such as Timer/Counter, Serial Communication, Interrupt, CCP, EEPROM memory, ADC etc.</li> <li>Interfacing with external peripherals such as LED, LCD(Liquid Crystal Display), SSD(Seven Segment Display), DAC, RTC etc</li> </ul>		
UNIT-VI	Robotics: Interfacing with Motors	<ul style="list-style-type: none"> <li>Basic Peripherals</li> <li>Input &amp; Output devices</li> <li>Working with sensors</li> <li>Concept of robotics</li> <li>Types of motors &amp; sensors</li> <li>Understand interfacing of motors &amp; sensors with microcontroller</li> <li>The candidate will be able to understand embedded hardware, design for telecom devices and equipment, core programming of telecom devices and equipment</li> <li>The candidate will be able to use basic communication protocols, understanding of circuits and architectures used in telecom systems and</li> </ul>	<ul style="list-style-type: none"> <li>DC Motor in Robotics</li> <li>Gear fundamentals and types of gears</li> <li>Characteristics, Operation, Electrical model and Types of DC Motor</li> <li>Hardware interface of DC Motors</li> <li>Speed control of DC Motor</li> <li>Introduction to Stepper Motor</li> <li>Types of Stepper Motor</li> <li>Operation of Stepper Motor</li> <li>Hardware interface for driving Stepper motor</li> <li>Concept of servo motor</li> <li>Operation of Servo Motor</li> <li>Hardware interface for</li> </ul>		

		Devices	driving servomotor		
UNIT-VII	Introduction to communication protocols		<ul style="list-style-type: none"> <li>● I2C - Interfacing with micro controller using bit-banking method, I2C devices – RTC, Memory, ADC-DAC, Port-Expander,</li> <li>● SPI (Serial Peripheral Interface), Bluetooth, WI-Fi and RFID.</li> <li>● Understanding Serial Communication</li> <li>● Bluetooth Communication</li> <li>● SPI Interface</li> <li>● Wi-Fi</li> <li>● I2C</li> <li>● Infrared</li> <li>● RFID</li> <li>● GSM</li> </ul> PDH/SDH/Ethernet		
UNIT-VIII	ARM Process	At the end of this Unit the student should be able to: <ul style="list-style-type: none"> <li>● The candidate will be able to prepare and maintain acknowledge-base of the known problems</li> <li>● The candidate able to develop system by interfacing as well as developing the hardware.</li> <li>● The candidate will be able to use peripherals of ARM processor for different applications</li> <li>● The candidate will be able to understand the concept of instruction and execution</li> <li>● The candidate will be able to</li> </ul>	<ul style="list-style-type: none"> <li>● Introduction to ARM Processor</li> <li>● Architecture of ARM7TDMI Processor</li> <li>● Advantages of 32-bits over 8-bits controller</li> <li>● Architecture of ARM7TDMI Processor</li> <li>● Application of using ARM7TDMI Processor</li> </ul> Interfacing with various peripherals		

		<p>Produce devices based on ARM Processor and can use it in various fields as per the requirement.</p> <p>The candidate will be able to use different series of ARM Process or according to requirement</p>	<ul style="list-style-type: none"><li>● LPC2148ARMwithin-builtperipherals</li><li>● ADC</li><li>● DAC</li><li>● RTC</li><li>● UART</li><li>● Timer/Counter</li><li>● Interrupts</li><li>● PWM</li><li>● SPI based programming</li></ul> <p>LCD interfacing, Introduction to LPC1768 (Cortex-M3series).</p>		
			<ul style="list-style-type: none"><li>● Introduction to LPC1768 (Cortex-M3series).</li><li>● Architecture of LPC1768 (Cortex-M3series) core.</li><li>● Interfacing with peripherals of LPC1768(Cortex-M3series)</li></ul> <p>Application of LPC1768 (Cortex-M3series).</p>		

**NOS /Module: IoT application development on R-Pi and Node-MCU**

**NOS /Module Code: MSME/AET/02**

**Outcomes:**

After completion of course Student should be able to:

- Introduction to Python Programming
- Understand Decision Making Statements and Loop Statements
- Understand Functions, Structures and Union in python
- OOPs Concept in Python
- 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.
- Understand the architecture of Arduino devices and NodeMCU

- Understand the concept of internet of Things
- Understanding interfacing of different sensors and actuators using IoT with Arduino and NodeMCU

**Practical Hours: 60****Theory Marks: NA****Practical Marks: 100**

Unit No.	Unit Name	Unit level outcomes	Contents (chapters/topics)	TH Hours	TH Marks
UNIT-I	Introduction Python Programming	<p>At the end of this Unit the student should be able to</p> <ul style="list-style-type: none"> <li>• understand the work flow of the company's design process</li> <li>• interact with the lead engineer in order to understand the work schedules, shifts and delivery dates</li> <li>• plan work activities based the work flow and deliverables</li> <li>• Understand broad level activities involved in the stages of design</li> <li>• List the various department to interact with for completing the work</li> <li>• Minimize absenteeism and report to work on time</li> </ul>	<ul style="list-style-type: none"> <li>• Introduction to Python programming</li> <li>• Basic concept of Programming</li> <li>• Method of decision making, looping,</li> <li>• Understand the environment and features of Python Programming Language.</li> <li>• Know the history, versions and applications of Python Programming Language.</li> <li>• Learn to create and execute program in Python.</li> <li>• Learn to install and use Spyder IDE.</li> <li>• Understand the concepts of variables, data types, keywords and operators.</li> </ul>	60	NA
UNIT-II	Programming concepts of Python		<ul style="list-style-type: none"> <li>• Learn the structure of conditional statements in Python.</li> <li>• Learn using conditional statements to met certain criteria during execution of program.</li> <li>• Learn to writing looping structures.</li> <li>• Understand the concepts of higher order data structures.</li> <li>• Understand using lists, tuples, Set and dictionaries.</li> </ul>		

			<ul style="list-style-type: none"> <li>● Learn to write functions and passing arguments.</li> <li>● Understand the concept of return statement in a function.</li> </ul> <p>Understand the concept and use of Lambda function in Python.</p>		
UNIT-III	File Handling and Object Oriented Programming in Python		<ul style="list-style-type: none"> <li>● Understand the concept of file handling in Python.</li> <li>● Learn basic operations and appending the file.</li> <li>● Understand the various access modes to open a file.</li> <li>● Learn the various methods to create, remove, and change directories.</li> <li>● Understand the basic concepts of OOPs and its use in Python</li> <li>● Learn to design classes and objects in Python.</li> <li>● Learn the concepts of constructors and destructors.</li> <li>● Learn how to use class inheritance in Python for reusability. 04 16 20</li> </ul> <p>Understand the concept of overloading and overriding. Understand the concepts of data hiding</p>		
UNIT-IV	Introduction to Raspberry Pi,	<ul style="list-style-type: none"> <li>● The candidate will be able to understand the concept of instruction and execution</li> <li>● The candidate will be able to produce devices based on Raspberry Pi</li> <li>● Understand Putty Login Method to connect device</li> </ul>	<ul style="list-style-type: none"> <li>● Introduction to Raspberry Pi</li> <li>● Comparison with Microcontroller and Raspberry Pi,</li> <li>● Architecture of ARM11 processor</li> <li>● Pipelining method to execute the instruction</li> </ul>		

UNIT-V	Peripheral Interfacing using Raspberry Pi	<p>with laptop</p> <ul style="list-style-type: none"> <li>• Understand the programming concept using Raspberry Pi</li> </ul>	<ul style="list-style-type: none"> <li>• Raspberry Pi with different peripherals</li> <li>• Port Programming</li> <li>• Led Interfacing</li> <li>• LCD interfacing</li> <li>• UART</li> <li>• Bluetooth</li> <li>• Interfacing of Sensors: Ultrasonic Sensor, IR Sensor etc.</li> <li>• Interfacing with Motors: DC Motor, Stepper Motor, Servo Motor etc.</li> </ul>		
UNIT-VI	Remote Login Method : PuTTY, Hyperterminal, Ethernet		<ul style="list-style-type: none"> <li>• Remote login method to interlink Raspberry Pi with laptop</li> <li>• Debugging and executing programs using Raspberry Pi</li> </ul>		
UNIT-VII	Introduction to Arduino	<p>After completion of unit Student should be able to</p> <ul style="list-style-type: none"> <li>• The candidate will be able to understand embedded hardware, design for Arduino devices and equipment, core programming of Arduino devices and equipment.</li> </ul> <p>The candidate will be able to use programming to implement various sensors and actuators</p>	<ul style="list-style-type: none"> <li>• Introduction to Arduino Uno Board</li> <li>• Understanding programming Language used in Arduino</li> <li>• Operating the Arduino IDE, loading a simple program.</li> <li>• Writing a program to blink the onboard LED.</li> </ul>		
UNIT-VIII	Introduction to NodeMCU	<p>with Arduino devices, understanding of circuits and architectures used in NodeMCU</p> <ul style="list-style-type: none"> <li>• Implementation of various sensors and actuator on Arduino board as well as on NodeMCU using Internet of Things concept</li> </ul>	<ul style="list-style-type: none"> <li>• Introduction to NodeMCU</li> <li>• Interfacing of various peripherals with NodeMCU</li> <li>• Connecting NodeMCU with Wi-Fi network or Study of ESP8266 Wi-Fi module with Arduino Board.</li> <li>• Basics of Wireless Networking Various</li> <li>• Wi-Fi library Web server- introduction, installation, configuration</li> </ul>		

UNIT-IX	Interfacing with various Sensors and Actuators		<ul style="list-style-type: none"> <li>● Overview of Sensors working</li> <li>● Analog and Digital Sensors</li> <li>● Arduino Interfacing of Temperature, Humidity, Motion, Ultrasonic, Infrared, Current, Sound and Gas Sensor.</li> <li>● Interfacing of Actuators with Arduino.</li> <li>● Arduino - Interfacing of Relay Switch and other actuators</li> </ul>		
UNIT-X	Introduction to Internet of Things		<ul style="list-style-type: none"> <li>● Understanding IoT fundamentals</li> <li>● Various Platforms for IoT</li> <li>● Real time Examples of IoT</li> <li>● Overview of IoT components and IoT Communication</li> <li>● Technologies Challenges in IOT</li> <li>● Study of IOT Cloud platforms</li> <li>● Thing Speak API and Blynk, Adafruit I/O</li> <li>● Interfacing Clouds and Arduino with Web services.</li> </ul>		

Unit No.	Unit Name	Unit level outcomes	Contents (chapters/topics)	PR hours	PR Marks
UNIT-XI	Comparison between traditional O.S and RTOS	At the end of this Unit the student should be able to <ul style="list-style-type: none"> <li>● design flow involved in design stages , design, develop, test, debug software components</li> <li>● software module library, system testing, product</li> </ul>	<ul style="list-style-type: none"> <li>● Introduction to real time operating system</li> <li>● Understanding the concept of RTOS</li> <li>● Application of Real time Operating System</li> <li>● Comparison between OS and RTOS</li> </ul>		NA

UNIT-XII	Understanding Kernel	verification and validation, operating system such as windows, Linux ,system level integration, software fundamentals such data structures, algorithm design end-product application, i.e., industry for which embedded system is designed, Schematics and datasheets • understand scheduling process and policies required to execute task • understand the application of using real time operating system	<ul style="list-style-type: none"> <li>• What is Kernel</li> <li>• Architecture of Kernel</li> <li>• Execution of task in kernel</li> </ul>		
UNIT-XIII	Scheduling policies		<ul style="list-style-type: none"> <li>• Execution method for task</li> <li>• Type of scheduling</li> <li>• Policies for executing task</li> </ul>		
UNIT-XIV	Programming in RTOS		<ul style="list-style-type: none"> <li>• Building block policies used in RTOS</li> <li>• Programming Concept used to execute task</li> <li>• Different task scheduling while programming for RTOS</li> </ul>		

**NOS /Module: Digital Logic design and Programming on FPGA and CPLD**

**NOS /Module Code: MSME/AET/04**

**Outcomes:**

After completion of 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:-30 PRACTICAL HOURS: 60 THEORY MARKS:- NA PRACTICAL MARKS:100**

Unit No.	Unit Name	Unit level outcomes	Contents (chapters/topics)	PR hours	PR Marks
Unit-I	CPLD / FPGA architecture	At the end of this Unit the student should be able to: <ul style="list-style-type: none"> <li>● Understand the concept of VLSI design</li> <li>● Types of designing for e.g. Digital design or analog design and the irrespective design flow</li> <li>● Understanding the process of partitioning the design into different blocks</li> <li>● Selection of design types such as applications specific integrated circuit (ASIC), Field-programmable gate arrays (FPGA) and complex programmable logic device (CPLD)</li> </ul>	<ul style="list-style-type: none"> <li>● CPLD / FPGA architecture, Programming and Simulation with Xilinx ISE, VHDL programming-Full Adder</li> <li>● Introduction to complete system architecture such as memory, microcontroller, microprocessor, memory blocks, timers and oscillators, interfaces and power management</li> <li>● Introduction to design flow for the specific system</li> <li>● Introduction to VHDL programming language</li> </ul>	60	100
Unit-II	Programming and Simulation with Xilinx ISE	Use of design languages such as hardware design language (HDL), e.g. Verilog, VHDL, High level languages such as C <ul style="list-style-type: none"> <li>● Understanding the code required for design</li> <li>● Creation of code, verification, testing software</li> <li>● Testing various examples on the system</li> </ul>	<ul style="list-style-type: none"> <li>● Types of languages</li> <li>● Types of designing for e.g. Digital design or analog design and the irrespective design flow</li> <li>● Use of design languages such as hardware design language (HDL), VHDL,</li> <li>● Understanding the code required for design</li> </ul>		
Unit-III	VHDL programming	<ul style="list-style-type: none"> <li>● Understanding the synthesis and simulation process of code</li> <li>● Building simulation module as per system specification for e.g. VHDL model for ASIC design</li> <li>● Understanding types of design as per sections in VLSI processor such as high level design, operative part design, control part design,</li> </ul>	<ul style="list-style-type: none"> <li>● Introduction to design flow for the specific system</li> <li>● Introduction to VHDL programming language</li> <li>● Creation of code, verification, testing software</li> <li>● Testing various examples on the system</li> <li>● Understanding the synthesis and simulation process of code</li> </ul>		

Unit-IV	Programming through JTAG on Xilinx Spartan3 Board	memory design etc. ● Functioning and specifying the tools used for design as per requirement ● Analysis of Design code by verification engineer Installation process of design code into hardware and verifying it	● Building simulation module as per system specification for e.g. VHDL mode If or ASIC design ● Understanding types of design as per sections in VLSI processor such as high level design, operative part design, control part design, memory design etc. ● Functioning and specifying the tools used for design as per requirement ● Analysis of Design code by verification engineer ● Installation process of design code into hardware and verifying it		
---------	---	---	---	--	--

**NOS /Module: Layout Design and Fabrication of PCB**

**NOS /Module Code: MSME/AET/05**

**Outcomes:**

After completion of 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:30**

**PRACTICAL HOURS:- 60**

**THEORY MARKS: NA**

**PRACTICAL MARKS: -100**

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 <ul style="list-style-type: none"> <li>Developing schematic and creating PCB layout, converting them to Gerber format using CAD and other software tools, suitable for production and assembly process.</li> <li>Discuss with the systems analyst on</li> </ul>	<ul style="list-style-type: none"> <li>Schematic design with circuit simulation, creating footprints for customized component packages, Component</li> <li>Placement as per requirement, layout design for Single layer and Multilayer PCB, File generation for film making and PCB manufacturing.</li> </ul>	30	100
Unit II	File generation for film making and PCB manufacturing.	<ul style="list-style-type: none"> <li>the customer requirement and get inputs</li> <li>analyses the requirements of customer</li> <li>suggest any further changes to the requirement generate bill of materials(BOM) simultaneously with the schematic's creation</li> <li>select the components in the circuit by analyzing the maximum operating voltages and current levels of each node of the circuit while considering tolerance criteria</li> </ul>	<ul style="list-style-type: none"> <li>Develop schematics</li> <li>Convert schematics to PCB layout</li> <li>Create Gerber file and send to manufacturer</li> <li>Give the detailed instructions and information to be sent to manufacturer</li> <li>Achieve productivity and quality standards</li> </ul>		
Unit III	Developing schematics	<ul style="list-style-type: none"> <li>reconsider based on availability, budget and size after selecting electrically satisfactory components</li> <li>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</li> </ul>	<ul style="list-style-type: none"> <li>frame the circuit specifics before drawing, by designing schematics for the circuit</li> <li>create a new project schematic on the designing software</li> <li>add components, component values, connection between components and power connection on it</li> <li>run a check to see if there are any mistakes to be fixed</li> <li>ensure the schematic to be as per the original PCB design</li> <li>lay out clearly and logically so that it is easier at the designing stage</li> <li>make sure the trace,</li> </ul>		

			<p>pads and via are the same sizes as the original design</p> <ul style="list-style-type: none"> <li>● make short notes on the schematics in conformance with design requirements</li> <li>● create the wiring diagram, illustration of flow of connections from input to output</li> </ul>		
--	--	--	---	--	--

**NOS /Module: PROJECT**

**NOS /Module Code: MSME/AET/06**

**PRACTICAL MARKS: -100**

**HOURS: - 60**

**Outcomes:**

After completion of course Student should be able to

- Development of desired circuit schematics and printed circuit boards
- Use of different microcontrollers to realize an embedded system.
- Programming of different microcontrollers as per project requirements.
- Interfacing of different peripherals along with microcontroller in an embedded System

**COURSES / MODULE TEMPLATE**

**NOS /Module: Employability Skill**

**NOS /Module Code: MSME/ES/02**

**THEORY HOURS: 60 PRACTICAL HOURS: - THEORY MARKS: 100 PRACTICAL MARKS: -**

**Refer Standard Curriculum developed by NCVET. ([https://nqr.gov.in/downloads/pdfs/30-hours\\_MC\\_Employability\\_Skills.pdf](https://nqr.gov.in/downloads/pdfs/30-hours_MC_Employability_Skills.pdf))**