Manipal School of Information Sciences

Manipal Academy of Higher Education, Manipal

Outcome Based Education (OBE) Framework

Two Year full time Postgraduate Program

Master of Engineering - ME (Embedded Systems & Instrumentation)

TABLE OF CONTENTS

Contents

NATURE AND EXTENT OF THE PROGRAM

An engineering graduate skillset requirement is changing with invent of the new technologies. In particular the impact of Embedded Systems & Instrumentation provide a high employability in the industry.

Master of Engineering - ME (Embedded Systems & Instrumentation) Program is a comprehensive two-year postgraduate program, which aims to provide hands-on experience to prepare industry-ready ESI professionals. The program Master of Engineering - ME (Embedded Systems & Instrumentation) helps engineering graduates to specialize in the field of electronics, instrumentation and enables them to learn how embedded devices can be programmed, regulating, networked for the data communication and its analysis. Students will also understand the security issues, validating, debugging the circuit boards. This two year master program will cover various domain like communication, sensors and actuators, cloud, data analytics.

Master of Engineering - ME (Embedded Systems & Instrumentation) postgraduate degree would welcome graduates from any discipline with 50% mark in qualifying exam. Students after successfully completing the program will get career opportunities such as automotive, aerospace and defence, industrial electronics, robotics, chip and circuit board design validation.

PROGRAM EDUCATION OBJECTICE (PEO)

The overall objectives of the Learning Outcomes-based Curriculum Framework (LOCF) for **Master of Engineering - ME (Embedded Systems & Instrumentation) program are as follows.**

PEO No	Education Objective
PEO 1	Successfully engage in challenging careers with professional approach in the areas of embedded systems, instrumentation and related domains of engineering.
PEO 2	Demonstrate competence in identifying and analysing technical problems, suggest feasible and innovative solutions using their core competence in embedded systems, instrumentation design and thereby support the technological growth of the nation.
PEO 3	Impart quality technical education, engage in research and contribute to knowledge creation and sharing.
PEO 4	Possess analytical, communicative, leadership skills, and demonstrate the ability to work in multidisciplinary and multi-cultural environments.
PEO 5	Be Self-motivated and remain continuously employable by engaging in lifelong learning.

GRADUATE ATTRIBUTES

S No.	Attribute	Description
1	**Scholarship of Knowledge**	Acquire in-depth knowledge of specific discipline or professional area, including wider and global perspective, with an ability to discriminate, evaluate, analyse and synthesise existing and new knowledge, and integration of the same for enhancement of knowledge.
2	**Critical Thinking**	Analyse complex engineering problems critically, apply independent judgement for synthesising information to make intellectual and/or creative advances for conducting research in a wider theoretical, practical and policy context.
3	**Problem Solving**	Think laterally and originally, conceptualise and solve engineering problems, evaluate a wide range of potential solutions for those problems and arrive at feasible, optimal solutions after considering public health and safety, cultural, societal and environmental factors in the core areas of expertise.
4	**Research Skill**	Extract information pertinent to unfamiliar problems through literature survey and experiments, apply appropriate research methodologies, techniques and tools, design, conduct experiments, analyse and interpret data, demonstrate higher order skill and view things in a broader perspective, contribute individually/in group(s) to the development of scientific/technological knowledge in one or more domains of engineering.
5	**Usage of modern tools**	Create, select, learn and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities with an understanding of the limitations.
6	**Collaborative and Multidisciplinary work**	Possess knowledge and understanding of group dynamics, recognise opportunities and contribute positively to collaborative-multidisciplinary scientific research, demonstrate a capacity for self-management and teamwork, decision-making based on open-mindedness, objectivity and rational analysis in order to achieve

		common goals and further the learning of themselves as well as others.
7	**Project Management and Finance**	Demonstrate knowledge and understanding of engineering and management principles and apply the same to one's own work, as a member and leader in a team, manage projects efficiently in respective disciplines and multidisciplinary environments after consideration of economical and financial factors.
8	**Communication**	Communicate with the engineering community, and with society at large, regarding complex engineering activities confidently and effectively, such as, being able to comprehend and write effective reports and design documentation by adhering to appropriate standards, make effective presentations, and give and receive clear instructions.
9	**Life-long Learning**	Recognise the need for, and have the preparation and ability to engage in life-long learning independently, with a high level of enthusiasm and commitment to improve knowledge and competence continuously.
10	**Ethical Practices and Social Responsibility**	Acquire professional and intellectual integrity, professional code of conduct, ethics of research and scholarship, consideration of the impact of research outcomes on professional practices and an understanding of responsibility to contribute to the community for sustainable development of society.
11	**Independent and Reflective Learning**	Observe and examine critically the outcomes of one's actions and make corrective measures subsequently, and learn from mistakes without depending on external feedback.

QUALIFICATIONS DESCRIPTORS

1. Demonstrate

   (i) A systematic, extensive, coherent knowledge and understanding of an academic field of study as a whole, its applications, and links to related disciplinary areas/subjects of study; including a critical understanding of the established theories, concepts, and of a number of advanced and emerging issues in the field of Embedded Systems & Instrumentation;

   (ii) Procedural knowledge that creates different types of professionals related to the Embedded Systems & Instrumentation, including research and development, teaching, government and public service;

   (iii) Professional skills in the domain of Embedded Systems, Instrumentation, system control, Communication protocolsdata structures, web-services, Security protocols and architectures, sensors, data analytics, actuators including a critical understanding of the latest developments, and an ability to use established techniques in the domain of Embedded Systems and Instrumentation.

2. Demonstrate comprehensive knowledge about Instrumentation, embedded systems, microcontrollers, Internet of Things, embedded programming including current research, scholarly, and/or professional literature, relating to essential and advanced learning areas pertaining to the Embedded Systems and Instrumentation techniques and skills required for identifying problems and issues related.

3. Demonstrate skills in identifying information needs, collection of relevant quantitative and/or qualitative data drawing on a wide range of sources, analysis and interpretation of data.

4. Methodologies as appropriate to the subject(s) for formulating evidence based solutions and arguments.

5. Use knowledge, understanding and skills for critical assessment of a wide range of ideas, complex problems and issues relating to the chosen field of study.
6. Communicate the results of studies undertaken in an academic field accurately in a range of different contexts using the main concepts, constructs and techniques of the Embedded Systems and Instrumentation studies.
7. Address one's own learning needs relating to current and emerging areas of study, making use of research, development and professional materials as appropriate, including those related to new frontiers of knowledge.
8. Apply one's disciplinary knowledge and transferable skills to new/unfamiliar contexts, to identify, analyse problems, issues, and seek solutions to real-life problems.

PROGRAM OUTCOMES

After successful completion of Master of Engineering - ME (Embedded Systems & Instrumentation) Program, Students will be able to:

PO No	Attribute	Competency
PO 1	**Scholarship of Knowledge**	Acquire in-depth knowledge of ESI domain, with an ability to discriminate, evaluate, analyze, synthesize the existing and new knowledge, and integration of the same for enhancement of knowledge.
PO 2	**Critical Thinking**	Analyze complex ESI Eco System critically, apply independent judgement for synthesizing information to make intellectual and/or creative advances for conducting research in a wider theoretical, practical and policy context.
PO 3	**Problem Solving**	Think laterally and originally, conceptualize and solve ESI problems, evaluate a wide range of potential solutions for those problems and arrive at feasible, optimal solutions after considering public health and safety, cultural, societal and environmental factors in the core areas of expertise.
PO 4	**Research Skill**	Extract information pertinent to unfamiliar problems through literature survey and experiments, apply appropriate research methodologies, techniques and tools, design, conduct experiments, analyze and interpret data, demonstrate higher order skill and view things in a broader perspective, contribute individually/in group(s) to the development of scientific/technological knowledge in one or more domains of engineering.
PO 5	**Usage of modern tools**	Create, select, learn and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities with an understanding of the limitations.

PO 6	**Collaborative and Multidisciplinary work**	Possess knowledge and understanding of group dynamics, recognize opportunities and contribute positively to collaborative-multidisciplinary scientific research, demonstrate a capacity for self-management and teamwork, decision-making based on open-mindedness, objectivity and rational analysis in order to achieve common goals and further the learning of themselves as well as others.
PO 7	**Project Management and Finance**	Demonstrate knowledge and understanding of engineering and management principles and apply the same to one's own work, as a member and leader in a team, manage projects efficiently in respective disciplines and multidisciplinary environments after consideration of economical and financial factors
PO 8	**Communication**	Communicate with the engineering community, and with society at large, regarding complex engineering activities confidently and effectively, such as, being able to comprehend and write effective reports and design documentation by adhering to appropriate standards, make effective presentations, and give and receive clear instructions.
PO 9	**Life-long Learning**	Recognize the need for and have the preparation and ability to engage in life-long learning independently, with a high level of enthusiasm and commitment to improve knowledge and competence continuously.
PO 10	**Ethical Practices and Social Responsibility**	Acquire professional and intellectual integrity, professional code of conduct, ethics of research and scholarship, consideration of the impact of research outcomes on professional practices and an understanding of responsibility to contribute to the community for sustainable development of society.

PO 11	**Independent and Reflective Learning**	Observe and examine critically the outcomes of one’s actions and make corrective measures subsequently and learn from mistakes without depending on external feedback.

COURSE STRUCTURE, COURSEWISE LEARNING OBJECTIVE, AND COURSE OUTCOMES (COS)

FIRST YEAR:

Semester: 1 (Manipal, India) **Semester: 2 (ESIGELEC, France)**

Subject Code	Subject Title	L	T	P	C	Subject Code	Subject Title	L	T	P	C
CSE 601	Data Structures and Algorithms	3	-	-	3	ESI 621	Virtual Instrumentation	2	-	3	3
ESD 602	Microcontrollers and its Applications	3	-	-	3	ESI 622	Specific Instrumentation	2	-	3	3
ESD 605	Embedded Systems	3	-	-	3	ESI 623	Embedded C Programming	2	-	3	3
BDA 601	Fundamentals of Machine Learning	3	-	-	3	ESI 624	Artificial Intelligence for Smart Systems	2	-	3	3
	Elective - 1	3	-	-	3	ESI 625	Smart Sensors	2	-	3	3
CSE 601L	Data Structures and Algorithms Lab	-	-	3	1	ESI 626	Project Management	-	2	-	2
ESD 602L	Microcontrollers and its Applications Lab	-	-	3	1		Elective - 2	2	-	3	3
ESD 605L	Embedded Systems Lab	-	-	3	1	ESI 628	Oral Communication	1	-	-	1
BDA 601L	Fundamentals of Machine Learning Lab	-	-	3	1	ESI 629	R&D Project	-	4	3	5
	Elective – 1 Lab	-	-	3	1	ESI 630	French Language – 2	4	-	-	4
ESI 695	Mini Project - 1	-	-	4	-						
ESI 697	Seminar - 1	-	-	1	-						
ESI 609	French Language-1 *	5	-	-	-						
Total		**20**	**-**	**15**	**25**	**Total**		**17**	**6**	**21**	**30**

*** Audited and not considered for CGPA calculation**

SECOND YEAR (FINAL YEAR):

III and IV Semester		
ESI 799	Project Work	25
Total Number of Credits to Award Degree		75

List of Electives (Theory)

Manipal, India		ESIGELEC, France	
Elective - 1		Elective - 2	
Code	**Subject**	**Code**	**Subject**
CSE-618	Dot Net Technologies	ESI-631	Embedded Java
CSE-620	Linux and Scripting Languages	ESI-632	Real Time Operating Systems
CSE-622	Advanced Programming Techniques	ESI-633	Embedded Linux
IOT-607	Internet of Things	ESI-634	Mobile Robotics and Perception
		ESI-635	EMC Automotive Systems

List of Electives (Lab)

Manipal, India	
Elective – 1	
Code	**Subject**
CSE-618L	Dot Net Technologies Lab
CSE-620 L	Linux and Scripting Languages Lab
CSE-622 L	Advanced Programming Techniques Lab
IOT-607 L	Internet of Things Lab

Note:

For the students who are studying second semester at ESIGELEC, France only credit will be transferred and it will not be used for calculation of GPA/ CGPA. For the students who are opting out of Study Abroad – Credit Transfer program and continuing II Semester in SOIS:

1. The exit policy is ME (Embedded Systems)
2. For the subject French Language-I in I Semester
   - This credit will not be considered for the calculation of GPA/ CGPA.
3. Third & Fourth Semesters - Internship:
   - Number of credits for project work / Internship is 25
   - Minimum duration of internship is 10 months

Name of the Institution / Department: Manipal School of Information Sciences

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Data Structures and Algorithms
Course Code: CSE 601	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: **3**	**Prerequisites:** Basic Programming – preferably C
Synopsis:	This Course provides insight on 1. This course introduces students to elementary data structures and design of algorithms. 2. Students learn how to design optimal algorithms with respect to time and space 3. Students learn how to implement link list, stack, queues, searching and sorting techniques, sets, trees and graphs. 4. Students learn the design of divide and conquer technique, dynamic programming, greedy technique and back tracking.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Specify and analyse algorithms.
CO 2:	Learn and design programs for implementation of linear and non linear data structure.
CO 3:	Learn and design programs for sorting and searching.
CO 4:	Illustrate application of divide and conquer technique, dynamic programming, greedy technique and back tracking.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*			*							
CO 2	*	*				*					
CO 3	*					*					
CO 4	*	*				*					

Course content and outcomes:

Content	***Competencies***
Unit 1: Introduction	
Algorithm Specification, Performance Analysis	At the end of the topic student should be able to: 1. Define algorithms (C1) 2. Analyse algorithms. (C6)
Unit 2: Algorithm Analysis Techniques	
Analysis of Recursive Programs, Solving Recurrence Equations, General Solution for a large class of Recurrences.	1. Define recursive programs (C2) 2. Design simple recursive programs (C6) 3. Solve recurrence relations (C6)
Unit 3: Elementary data structures	
Implementation of Lists, Stacks, Queues	1. Design singly linked list (C6) 2. Design doubly linked list(C6) 3. Explain the concepts of array-based stacks (C2) 4. Explain the concepts of pointer-based stacks (C2) 5. Design and implement Queues. (C6)
Unit 4: Sorting & Searching Techniques	
Quick sort, Heap sort, Merge sort, Binary search, linear search, Fibonacci search	1. Develop algorithm for insertion sort, bubble sort and selection sort. (C6) 2. Develop and analyse algorithm for quick sort (C6) 3. Develop and analyse algorithm for heap sort (C6)

	4. Develop and analyse algorithm for merge sort (C6) 5. Design and analyse algorithms for binary, linear and Fibonacci search (C6)
Unit 5: Operations on Sets	
Introduction to Sets, A Linked- List implementation of Set, The Dictionary, The Hash Table Data Structure	1. Develop data structures for sets (C6) 2. Design a linked list-based implementation of sets (C6) 3. Design a Dictionary (C6) 4. Design Data structure for hash table (C6)
Unit 6: Trees	
Basic Terminology, Implementation of Trees, Binary Trees, Binary Search Trees	1. Examine the concepts of trees. (C3) 2. Design and implement general trees (C6) 3. Design and implement binary trees (C6) 4. Design and implement binary search trees (C6)
Unit 7: Graphs	
Basic definitions, Representation of Graphs, Minimum Cost Spanning Tree, Single Source Shortest Paths, All-Pairs Shortest Path	1. Define graphs (c6) 2. Design data structure for graphs (c6) 3. Formulate an algorithm to solve minimum cost spanning tree(c6) 4. Formulate an algorithm to solve Single source shortest path (c6)

	5. Formulate an algorithm to solve All- pair shortest path(c6)
Unit 8: Algorithm Design Techniques	
Divide-and-Conquer Algorithms, Dynamic Programming, Greedy Algorithms, Backtracking	1. Design of divide and conquer algorithms (C6) 2. Solve max min, Strassen's matrix multiplication, multiplication of long integers problem. (C6) 3. Design of dynamic programming techniques (C6) 4. Solve matrix chain order problem (C6) 5. Design of greedy algorithms(C6) 6. Solve Knap-sack, job scheduling with deadlines and optimal storage on tapes problems. (C6) 7. Design of Back tracking algorithms (C6)

Learning strategies, contact hours and student learning time		
Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	30	60
Seminar	-	-
Quiz	02	04
Small Group Discussion (SGD)	02	02
Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Clinic	-	-
Revision	02	-

Assessment	06	-
TOTAL	**44**	**74**

Assessment Methods:	
Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3	CO 4
Sessional Examination 1	*	*		
Sessional Examination 2		*	*	*
Assignment/Presentation	*	*	*	*
End Semester Examination	*	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. “Introduction to Algorithms” Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. 2. “Data Structures& Algorithms” Aho, Hopcroft and Ulmann 3. “Data structures and algorithm analysis in C” Mark Allen Weiss 4. “Computer Algorithms” : Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Microcontrollers and its Applications
Course Code: ESD 602	**Course Instructor:**
Academic Year: 2020-2021	**Semester:** First Year, Semester 1
No of Credits: **3**	**Prerequisites:** Microprocessor architecture, Assembly language and Number systems

Synopsis:	This Course provides insight on 1. This course provides the knowledge of Intel 8051 and ARM Microcontrollers. 2. This course provides the knowledge of Microcontroller architecture, Registers and Instruction sets to write Assembly and Embedded C Programming. 3. This course provides the concept of Interfacing and Programming Sensors and Peripherals to Microcontrollers. 4. This course provides the concept of Designing Embedded Systems using Microcontrollers.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Employ the knowledge of Microcontrollers to build Embedded systems.
CO 2:	Explain the concept of Programming Microcontrollers using Assembly and Embedded C.
CO 3:	Design Embedded Systems by interfacing Sensors and Actuators.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*										
CO 2	*	*			*						
CO 3	*		*		*						

Course content and outcomes:

Content	***Competencies***
Unit 1: Introduction to Microprocessor & Microcontrollers	

Comparison – Variants – Types – General – ASIC – PLD – Introduction to Motherboard(Desktop) - Introduction to Embedded Board – Compare and Contrast - Application Types – Single Tasking – Multitasking – Multi-Application	1. Explain about the differences of Microprocessor and Microcontrollers(C2) 2. Describe Microcontroller Architecture (C2) 3. Explain the Register sets, Programming model and Memory map of Microcontroller(C2) 4. Describe about Microcontroller Instruction set. (C2) 5. Write the Applications using Microcontrollers. (C3)
Unit 2: Introduction to ARM Microcontrollers	
Programming Model – Processor Modes – ARM vs Thumb Introduction to LPCxxxx Microcontrollers – Features – Detailing of Pins - Memory Map Concepts – RAM & ROM - Interrupts Concepts (Internal & External)	1. Describe ARM Microcontroller architecture. (C2) 2. Describe the architecture of ARM Microcontrollers. (C2) 3. Apply knowledge of ARM Microcontroller architecture to rig up Embedded system circuits(C3) 4. Develop a Prototype of Embedded systems using ARM Microcontroller(C5, P3)
Unit 3: Reset Circuitry	
Crystals - Introduction to GPIO – Registers – Input /Output Configuration – Pull Up and Pull Down Resistor Concept – Interfacing with LED – Interfacing Push Buttons – LCD – Stepper Motor – DC Motor	1. Describe Crystal oscillator. (C2) 2. Describe Pull Up and Pull Down Resistor Concept.(C2) 3. Illustrate Interfacing LED, Push Buttons, LCD, Stepper Motor – DC Motor with microcontroller. (C2)
Unit 4: Relays	
Types of Relays – Interfacing	1. Describe Relay and its with interfacing external peripherals to Microcontrollers. (C4)
Unit 5: Timer, Counter Introduction	

Configuration – Programming	1. Describe about timers, counters and its usage with Microcontrollers(C4)
Unit 6: Serial vs Parallel Bus	
Serial vs Parallel Bus - Compare and Contrast – Terminology: Baud Rate – Bit Rate – RS232 – DB9 handshaking concepts - Configuring Registers – Programming for UART modules.	1. Describe about Serial and Parallel communication protocols(C2)
Unit 7: Introduction to SPI and I2C Protocol	
Detailed Discussion – Bit Banging – Interfacing with SPI and I2C Devices – RTC / ADC /DAC.	1. Describe SPI, I2C standards and its Interfacing with SPI and I2C Devices – RTC / ADC /DAC.(C3) 2. Explain about how to establish multi controller communications using communication protocols (C3)
Unit 8: Introduction to ADC and DAC	
Types – Chips - Register Configuration – Interfacing	1. Summarize types of ADC, DAC and its usage with Microcontroller. (C2)
Learning strategies, contact hours and student learning time	

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	30	60
Quiz	02	04
Small Group Discussion (SGD)	02	02
Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Revision	02	-
Assessment	06	-
TOTAL	**44**	**74**
Assessment Methods:		

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos			
Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	
Sessional Examination 2		*	*
Assignment/Presentation	*		*
End Semester Examination	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. William Hohl, Christopher Hinds,"ARM Assembly Language: Fundamentals and Techniques",2nd Edition, ISBN-13: 978-1482229851, ISBN-10: 1482229854 2. Andrew Sloss, Dominic Symes, Chris Wright,"ARM System Developer's Guide: Designing and Optimizing System Software",1st Edition,The Morgan Kaufmann Series in Computer Architecture and Design, ISBN-13: 978-1558608740, ISBN-10: 1558608745 3. David Seal, "ARM Architecture Reference Manual", 2nd Edition, Addison-Wesley Professional. 4. Steve Furber,"ARM System-on-Chip Architecture",2nd Edition,Addison-Wesley Professional, ISBN-13: 078-5342675191,ISBN-10: 0201675196 5. Douglas V. Hall,"Microprocessors and Interfacing",Mcgraw Hill Educatin ,ISBN-10 1259006158,ISBN-13 9781259006159,2012. 6. Websites & Transaction Papers

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Embedded Systems
Course Code: ESD 605	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 2
No of Credits: 3	**Prerequisites:** Microprocessor architecture, Microcontroller Architecture, Assembly language and Number systems.

Synopsis:	This Course provides insight on 1. This course provides the knowledge of ARM Cortex M3 Processor architecture 2. This course provides the knowledge of Microcontroller based on ARM Processor architecture and its Registers and Instruction sets to write Assembly and Embedded C Programming. 3. This course provides the concept of Interfacing and Programming Sensors and Peripherals to Microcontrollers. 4. This course provides the concept of Communication Protocols required for multi-processor communication. 5. This course provides the concept of Real time operating systems on Microcontrollers. 6. This course provides the concept of Designing Real Time Embedded Systems using ARM Microcontroller.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Employ the knowledge of Microcontrollers to build Embedded systems.
CO 2:	Explain the concept of Programming ARM Microcontrollers using Assembly and Embedded C.
CO 3:	Design a Real time Embedded Systems by interfacing Sensors, Actuators and porting Real time operating systems.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*

CO 1	*	*									
CO 2	*	*	*		*						
CO 3	*	*	*		*						

Course content and outcomes:

Content	***Competencies***
Unit 1: Introduction to Embedded Systems	
Design Challenges, Processors Technology, Design Technology	At the end of the topic student should be able to: 1. Describe the Design issues in designing the Embedded Systems.(C1) 2. Discuss the design technology associated with Embedded Systems.(C2)
Unit 2: Introduction to ARM Cortex processor	
Variants of Cortex and ARM versions, Comparison of M-series processor, Architecture, Programmers Model, APSR register, Memory Model, Exception, Interrupts, Reset	1. Explain about ARM Processor architecture (C2) 2. Describe ARM Cortex m3 processor data path, Register set, Programming models and memory map (C2) 3. Describe about ARM Cortex M3 Processor Instruction set. (C2) 4. Describe about ARM Processor system bus and Interrupt controller (C2) 5. Describe about interrupt and Exception handling (C2) 6. Describe ARM Microcontroller architecture. (C2)
Unit 3: Instruction Set Architecture	
More on Memory System, Exceptions and Interrupts, NVIC, Memory Protection Unit, Assembly Programming, Embedded C programming, CMSIS, Startup Code	1. Describe ARM Cortex memory system. 2. Describe interrupt and Exception handling (C2) 3. Describe NVIC, Memory Protection Unit. (C2) 4. Discuss CMSIS implementation in ARM Cortex.(C2)

Unit 4: Introduction to LPC13/17xx Microcontroller	
Memory Mapping, Registers involved and programming with GPIO, PWM	1. Discuss Memory Mapping, Registers involved and programming with GPIO, PWM. (C3) 2. Apply knowledge of ARM Microcontroller architecture to rig up Embedded system circuits(C3)
Unit 5: Data Acquisition System	
ADC, Types of ADC, Choosing the ADC, DAC	1. Identifying various types of ADC. (C1) 2. Review ADC and DAC selection criteria. (C2)
Unit 6: Serial Communication	
UART, I2C, SPI, Interfacing	1. Discussing various types of Serial Communication mechanism. (C2)
Unit 7: USB BUS	
Speed Identification on the bus, States, Packets, Data flow types, Enumeration, Descriptors, USB Interface –C Programs	1. Identify USB types, Firewire devices, ports, cables. 2. Describing Enumeration, Descriptors mechanism in USB.(C2)
Unit 8: CAN BUS	
Introduction, Frames, Bit stuffing, Types of errors, Nominal Bit Timing, A simple application with CAN	1. Describe the nature of CAN and the basic CAN protocol, and the basic structure of a CAN network. (C2) **2.** Prepare a simple application with CAN. (C3)
Unit 9: Introduction to Multitasking in Microcontrollers	
Variants of RTOS, FreeRTOS, UCOS, uCLinux, FreeRTOS on Cortex based Microcontrollers, TASK CREATION, QUEQUES, SEMAPHORE, MUTEX, Application development	1. Describe about Real time operating systems role in building real time systems (C3) 2. Describe about Designing Real Time Embedded systems by interfacing peripherals and actuators (C2)

	3. Design a Real time Embedded system by writing applications on top of Real time operating systems (C5)
Unit 10: Designing a Digital Camera	
Introduction, Requirement, Specifications, Implementation, Testing	**1.** Summarize the stages involved in designing a digital camera. (C2)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	30	60
Quiz	02	04
Small Group Discussion (SGD)	02	02
Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Revision	02	-
Assessment	06	-
TOTAL	**44**	**74**

Assessment Methods:

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	
Sessional Examination 2		*	*
Assignment/Presentation	*	*	

End Semester Examination	*	*	*
Feedback Process	• End-Semester Feedback		
Reference Material	1. Joseph Yiu, “The definitive guide to the ARM Cortex-M3”, Elsevier, 2nd Edition, 2010. 2. Frank Vahid, Tony Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, Wiley India, ISBN:81-265-0837-X, 2007. 3. Richard Barry, “NXP Semiconductors, LPC13xx/17xx User Manual”, 2012. 4. NXP Semiconductors, “LPCzone Examples”, 2012. 5. “FreeRTOS Reference Manual”, Real Time Engineers Ltd., 2016.		

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Fundamentals of Machine Learning
Course Code: BDA 601	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: **3**	**Prerequisites:** Basic Programming – preferably Python

Synopsis:	This Course provides insight on 1. This course provide the concept of machine learning, applications, techniques, design issues and approaches to machine learning. 2. This course provide the fundamental knowledge about concept learning, hypothesis and bias. 3. To implement machine learning algorithms such as Decision Tree learning, Probably Approximately Correct (PAC) learning, Bayesian learning, Instance-based learning, Principal Component Analysis (PCA) and Ensemble methods in real time data set for various analysis.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Identify the goals, applications, types and design issues of machine learning techniques.
CO 2:	Relate concept learning and hypothesis space.
CO 3:	Apply PCA learning approach to reduce the dimension.
CO 4:	Analyse different machine learning algorithms.
CO 5:	Design ensemble methods.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*										
CO 2		*									
CO 3			*								
CO 4				*							
CO 5				*							

Course content and outcomes:

Content	*Competencies*
Unit 1: Introduction	
Definition of Machine Learning, Goals and applications of machine learning, Basic design issues and approaches to machine learning, Types of machine learning techniques	1. Define Machine Learning (C1) 2. Describe about any three applications for which machine learning approaches seem appropriate. (C2) 3. Illustrate different types of machine learning techniques (C3)
Unit 2: Inductive Classification	
The concept learning task, Concept learning as search through a hypothesis space, General-to-specific ordering of hypotheses, Finding maximally specific hypotheses, Version spaces and the candidate elimination algorithm, Inductive bias.	1. Relate concept learning and hypothesis space (C4). 2. Apply different algorithms to obtain most general and most specific hypotheses from the training examples. (C3)
Unit 3: Decision Tree learning	
Representing concepts as decision trees, Recursive induction of decision trees, Picking the best splitting attribute, Entropy and information gain, Searching for simple trees and computational complexity.	1. Apply decision tree algorithm to find the hypothesis space (C3) 2. Construct decision tree machine learning algorithm (C5) 3. Explain the method of choosing training examples and target function in the design of a machine learning system (C2) 4. Explain different validation technique to find the accuracy in training and testing of data set (C5)
Unit 4: Computational learning theory	
Models of learnability: learning in the limit, Probably Approximately Correct (PAC) learning, Sample Complexity:	1. Define various terms related to computational learning approach (C1).

quantifying the number of examples needed to PAC learn, Computational complexity of training. Sample complexity for finite hypothesis spaces, Noise Learning Multiple Classes, Bias-variance trade-off, under-fitting and over-fitting concepts	2. Describe different models learning in the limit (C2) 3. Calculate the number of training examples required in different types of learning approaches (C4).
Unit 5: Bayesian learning	
Probability theory and Bayes rule, Naive Bayes learning algorithm - Parameter smoothing, Generative vs. discriminative training, Logistic regression, Bayes nets and Markov nets for representing dependencies	1. Write the applications of Bayes theorem (C3) 2. Describe the use of Logistic Regression in Machine Learning (C2) 3. Predict the target value for the new instance using Naïve Bayes classifier. (C3)
Unit 6: Instance-based learning	
Constructing explicit generalizations versus comparing to past specific examples, K-Nearest Neighbour learning algorithm, Case-based reasoning (CBR) learning	1. Construct explicit generalizations (C5) 2. Discriminate Instances Based and Case-based learning (C4) 3. Explain K-nearest neighbour learning (C5)
Unit 7: Continuous Latent Variables	
Principal Component Analysis (PCA), Applications of PCA	1. Describe use of Principal Component Analysis for the complex data set (C2). 2. Apply PCA to choose principal components for the given data set (C3)
Unit 8: Ensemble methods (bagging and boosting)	
Using committees of multiple hypotheses, Bagging, Boosting, DECORATE, Active learning with ensembles	1. Choose a suitable method of ensemble learning approach (C3). 2. Explain various ensemble techniques (C5)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	30	60
Quiz	02	04
Small Group Discussion (SGD)	02	02
Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Revision	02	-
Assessment	06	-
TOTAL	**44**	**74**

Assessment Methods:

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3	CO 4	CO 5
Sessional Examination 1	*	*			
Sessional Examination 2			*	*	
Assignment/Presentation	*	*	*	*	
End Semester Examination	*	*	*	*	*

	•
Reference Material	1. T. Mitchell, “Machine Learning”, McGraw-Hill, 1997. 2. E. Alpaydin, “Machine Learning”, MIT Press, 2010. 3. C. Bishop,” Pattern Recognition and Machine Learning”, Springer, 2006.

	4. E. Hart, R. Duda and D. Stork, “Pattern Classification”, Wiley-Interscience, 2000. 5. T. Hastie, R. Tibshirani and J. Friedman, “The Elements of Statistical Learning: Data Mining, Inference and Prediction”, Springer, 2nd Edition, 2009. 6. Jason Bell, “Machine Learning for Big Data”, Wiley Big Data Series, 2016. 7. Rama Murthy G,” Multidimensional Neural Networks Unified Theory”, New Age International, 2008.

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Dot Net Technologies
Course Code: CSE-618	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: **3**	**Prerequisites:** Basic knowledge of programming concepts and OOP's concepts.
Synopsis:	This Course provides insight on 1. This course provides students with elementary knowledge of Dot Net framework and C sharp programming language. 2. Students learn to design and develop C sharp Dot net applications. 3. Students learn to explore various features of ADO .Net and build C sharp Dot net applications with database support.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Explain Dot Net architecture and framework
CO 2:	Analyse C sharp language fundamentals
CO 3:	Discover the use of visual studio IDE
CO 4:	Design and develop OOP based Dot net application us with database connectivity using ADO
CO 5:	Develop basic Web Applications using ASP .net
Mapping of COs to POs	

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*										
CO 2		*									
CO 3	*	*									
CO 4		*	*	*							
CO 5			*	*							

Course content and outcomes:

Content	*Competencies*
Unit 1: Introducing C# and the .NET Platform:	
The philosophy of .NET, Building C# Applications	1. Explain Dot Net architecture (C2) 2. Define CLR, CTS, CLS and base class libraries. With sketch explain their relationships (C2) 3. Identify different features in Visual Studio (C1) 4. Explain the workflow that takes place between the source code, a given .NET compiler, and the .NET execution engine (C2)
Unit 2: The C# Programming Language:	
C# Language Fundamentals, Object-Oriented Programming with C# 2.0, Understanding Object Lifetime, Understanding Structured Exception Handling, Interfaces and Collections, Callback Interfaces, Delegates, and Events.	1. Explain C# Language Fundamentals. (C2) 2. Discuss the concepts of OOP. 3. Define Object Lifetime. 4. Discuss Structured Exception Handling. (C2) 5. Explain and Discuss Callback Interfaces, Delegates, and Events. (C3)
Unit 3: Programming with the .NET Libraries:	
The System.IO Namespace, Understanding Object Serialization, Building Better Window with System Window Forms, Rendering Graphical Data with GDI, Programming with Window Forms Controls, Database Access with ADO.NET	1. Discuss the System.IO Namespace. (C2) 2. Discuss building Better Window with System Window Forms. (C2) 3. Discuss Rendering Graphical Data with GDI. (C3) 4. Explain programming with Window Forms Controls. (C2) 5. Explain database Access with ADO.NET. (C3)

Unit 4: Web Applications and XML Web Services:	
ASP.NET 2.0 Web Pages and Web Controls, ASP.NET 2.0 Web Applications	1. Discuss creating a Simple Web Application and Creating a Web Form. (C2) 2. Explain adding and Configuring Server Controls in a Web Form and Asp.Net Standard Controls. (C2)
Learning strategies, contact hours and student learning time	

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	30	60
Quiz	02	04
Small Group Discussion (SGD)	02	02
Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Revision	02	-
Assessment	06	-
TOTAL	**44**	**74**

Assessment Methods:	
Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva
Mapping of assessment with Cos	

Nature of assessment	CO 1	CO 2	CO 3	CO 4	CO 5
Sessional Examination 1	*	*	*		
Sessional Examination 2			*	*	*
Assignment/Presentation	*	*	*	*	*
End Semester Examination	*	*	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. Pro C# with .NET 3.0 by Andrew Troelsen, Apress.

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Linux & Scripting languages
Course Code: CSE-620	**Course Instructor:**
Academic Year: 2020-2021	**Semester:** First Year, Semester 1
No of Credits: 3	**Prerequisites:**

Synopsis:	The goal of the course is to 1. Study of scripting languages such as Bash and Perl in Linux environment. 2. The study of usage of scripting languages in VLSI field. 3. To provide the basic knowledge about different tools available to automate the task
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1	Discover shell script programmatically using different features and debugging the code
CO 2	Apply SED & AWK commands to do more complex task in easy way
CO 3	Apply PERL scripts that create and change scalar, array and hash variables

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*	*	*								
CO 2	*	*		*							
CO 3		*	*	*							

Course content and outcomes:

Content	***Competencies***
Unit 1: Essentials	
Structure of a Linux Based Operating System, Hardware, Kernel, files & file system; Processes; networking; version control.	1. Summarize the Structure of a Linux Based Operating System 2. Discuss Hardware, Kernel (C2) 3. Explain files & file system, Processes; networking; version control(C2)

Unit 2: Introduction to Scripting: Shell, Tcl/tk, perl, python	
Getting started with Shell Programming: Writing shell scripts, Variables in shell, User defined variables (UDV), Rules for Naming variable name (Both UDV and System Variable), Printing or accessing values of UDV (User defined variables), echo Command, Shell Arithmetic, More about Quotes, Exit Status, The read Statement, Wild cards (Filename Shorthand or meta Characters), More commands on one command line, Command Line Processing, Requirements for Command Line arguments, Redirection of Standard output/input i.e. Input - Output redirection, Pipes, Filter, What is Processes, Why Process required, Linux Command(s) Related with Process Shells (bash) structured Language Constructs: Decision making in shell script, test command or [expr], if...else...fi, Nested ifs, Multilevel ifthen-else, Loops in Shell Scripts, for loop, Nested for loop, while loop, The case Statement, Debugging the shell script. Advanced Shell Scripting Commands: /dev/null - to send unwanted output of program, Local and Global Shell variable (export	1. Explain Variables, User defined variables (UDV) (C2) 2. Examine the Rules for Naming variable name (C3) 3. Write basic shell script using echo Command, Shell Arithmetic, Quotes, Exit Status, Wild cards, Command Line arguments; Redirection, Pipes, constructs. (C3)

command) Conditional execution i.e. && and ||, I/O Redirection and file descriptors, Functions, User Interface and dialog utility-Part I, User Interface and dialog utility-Part II, Message Box (msgbox) using dialog utility, Confirmation Box (yesno box) using dialog utility, Input (inputbox) using dialog utility, User Interface using dialog Utility - Putting it all together, trap command, The shift Command, getopts command.	
Unit 3: Awk utility	
Getting Starting with awk, Predefined variables of awk, Doing arithmetic with awk, User Defined, variables in awk, Use of printf statement, Use of Format Specification Code, if condition in awk, Loops in awk, Real life examples in awk, awk miscellaneous, sed - Quick Introduction, Redirecting the output of sed command, Writing sed scripts.	1. Illustrate Data manipulation using awk utility(C3) 2. Experiment Regular expression using awk utility (C4) 3. Experiment script using conditional statement using awk (C4)
Unit 4:	
Scalar Variables: What is Scalar? , Defining Scalar Variables, Literal Representation, Scalar Operators. Arrays: What is a List or Array? , Defining Array variables, Literal Representation, Array Operators. Hash Arrays: What is a Hash Array?, Hash Key and its value, Defining Array variables,	1. Illustrate scalar variables, arrays, and hash arrays (C3)

Literal Representation, Accessing Hash Array values, Hash Array Operators, How a Scalar Operator determines, Strings, Numbers.	
Unit 5: Perl	
Introduction to perl: What is PERL?, The structure of a Perl CGI script, Informing the Server software where Perl, CGI scripts are stored, Concept of granting permissions for everyone, to be able to use the Perl scripts. The perl programming environment: Creating a Perl CGI script, Invoking a Perl CGI script, Executing a Perl CGI script, Placing comments in a Perl script. Perl functions and procedures: Scalar Functions, Scalar Procedures, Array Functions, Array Procedures, Hash Array Functions, Hash Array Procedures.	1. Experiment Perl program using Perl constructs (C4)
Unit 6: Pattern Matching Subroutines	
Stdin/Stdout: Input from STDIN (Server Default Port 80), Output to STDOUT (Server Default Port 80), Makefile – create a makefiles, shortcuts	1. Illustrate stdin/stdout and makefile (C3)
Learning strategies, contact hours and student learning time	

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	30	60
Quiz	02	04
Small Group Discussion (SGD)	02	02

Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Revision	02	-
Assessment	06	-
TOTAL	**44**	**74**

Assessment Methods:	
Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos			
Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	
Sessional Examination 2		*	*
Assignment/Presentation			*
End Semester Examination	*	*	*

Feedback Process	● End-Semester Feedback
Reference Material	1. “Introduction to Linux – A Beginner’s Guide”, Machtelt Garrels 2. “Unix shell programming”, Stephen G. Kochan, Patrick H. Wood 3. “Sed & awk “,Dale Dougherty, Arnold Robbins 4. “Programming Perl”, Larry Wall, Tom Christiansen, Jon Orwant

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Advanced Programming Techniques
Course Code: CSE-622	**Course Instructor:**
Academic Year: 2020-2021	**Semester:** First Year, Semester 1
No of Credits: 3	**Prerequisites:** Basic programming knowledge
Synopsis:	1. This course would provide fundamental knowledge of various object oriented programming concepts. 2. The course will also provide skill sets to design and develop window based java applications. 3. The course will provide essential knowledge about multi thread programming, collection framework and utility library.

Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Explain major principles of object oriented programming concepts
CO 2:	Illustrate the use of OOP's concepts in java applications
CO 3:	Design UI applications using swing components
CO 4:	Apply multi thread programming , collection framework and utility library in java applications

Mapping of COs to POs

Cos	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*	*PO 12*
CO 1	*											
CO 2			*		*							
CO 3		*			*							
CO 4		*		*								

Course content and outcomes:

Content	*Competencies*
Part - 1: Introduction to Object Oriented Programming Concept	
Structural vs. procedural languages - Introduction to OOPS concept – Data Abstraction - Encapsulation – Polymorphism –	1. Discuss the differentiation between structural and procedural language (C2) 2. Discuss Core Concepts of OOPs concept (C2) 3. Apply the Data Abstraction, Encapsulation, Polymorphism (C3)

Inheritance. Introduction to Coding patterns	4. Discuss Coding patterns (C2)
Part - 2: Introduction to JAVA	
Introduction to JAVA – Data types – Variables - Array – Operators – Control Statements – Classes – Methods – Inheritance – Package – Interface – Exception Handling – Multithreading – I/O – Applets – Applet Lifecycle	1. Discuss Data types, Operators, Control statements available in Java language (C2) 2. Differentiate between checked Vs unchecked exceptions (C2) 3. Apply the techniques of OOPs into the Java applications (C3) 4. Write java program to read data from different types of files (C3) 5. Discuss life cycle of java applets (C2) 6. Explain thread synchronization (C2) 7. Illustrate inter thread communication using java application (C3)
Part 3: Introduction to GUI Programming with Swing	
AWT – Events – Layouts – Menus – JAVA Beans – Swings	1. Distinguish between AWT components and swing components (C2) 2. Define features of swing components (C1) 3. Apply different swing components in java applications (C3) 4. Discuss event delegation model (C2)
Part 4: Introduction to Java Library	
The Java Library – The Collection Framework, Utility classes, More on Applets and Swings	1. Discuss various interfaces and classes available in collection framework (C2) 2. Explain the use of utility classes in java applications (C2) 3. Apply collection framework classes in java applications (C3)
Learning strategies, contact hours and student learning time	

Learning strategy	*Contact hours*	*Student learning time (Hrs)*

Lecture	30	60
Quiz	02	04
Small Group Discussion (SGD)	02	02
Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Revision	02	-
Assessment	06	-
TOTAL	**44**	**74**

Assessment Methods:	
Formative:	**Summative:**
Theory Assignment	Sessional Examination
Lab Assignment	University End Semester Examination
Lab Test	Viva
Viva	

Mapping of assessment with Cos				
Nature of assessment	CO 1	CO 2	CO 3	CO 4
Sessional Examination 1	*	*		
Sessional Examination 2			*	*
Assignment/Presentation	*	*	*	*
End Semester Examination	*	*	*	*

Feedback Process	1. End-Semester Feedback
Reference Material	1. Herbert Schildt – "The Complete Reference - JAVA" Seventh Edition, TATA McGRAW-HILL EDITION

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Internet of Things
Course Code: IOT-607	**Course Instructor:**
Academic Year: 2020-2021	**Semester:** First Year, Semester 1
No of Credits: **3**	**Prerequisites:** Computer Networks, Programming aspects.

Synopsis:	This Course provides insight on 1. Various elements involved in the development of application for IoT. 2. Understanding of protocols across IoT stack. 3. Scripting languages like shell and python. 4. Client Server architecture and Python APIs of Socket programming. 5. Database and Python Database connectivity, Python Web Programming, IoT Framework.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Describe the developmental aspects of the application in IoT.
CO 2:	Demonstrate the usage of networking protocols across IoT stack.
CO 3:	Demonstrate the fundamental concepts in Client Server architecture and database implementation and usage with Python API's.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*										
CO 2	*	*			*						
CO 3	*		*		*						

Course content and outcomes:

Content	***Competencies***
Unit 1: Internet of Things	
IoT Protocols – Logical Design - Enabling Technologies - Levels – IoT vs M2M – Design Methodology – Domain Specific Applications	1. Outline the integration of various elements of IoT ecosystem. (C2)

Unit 2: Introduction to Python	
Datatypes - Constructs – Packages	1. Employ Datatypes, Constructs, Packages in python programming. (C2)
Unit 3: Wireless Sensor Networks	
Protocol Standards – Issues – Routing – Applications	1. Describe Protocol Standards, Routing, Issues in Wireless Sensor Networks. (C2)
Unit 4: Bluetooth	
Introduction – Protocol Stack - RF Classes – Radio Technologies – Service Discovery – Device Discovery – Profiles – Security (Discovering Bluetooth) - Hardware	1. Explain the aspects of Bluetooth technology. (C2)
Unit 5: Zigbee	
- Frequency - Channels – Topology - Zigbee Protocol Stack - PHY - MAC Layer - Working – Frame Structure – Beacon – Non-Beacon Communication - Zigbee PDU – Zigbee Hardware – API Mode and AT mode communication.	1. Describe Protocol Standards, Routing, Issues in Zigbee. (C2)
Unit 6: Internet Protocol	
Introduction to IPv4 and IPv6 – IPv4 Headers – Ipv6 Headers	1. Demonstrate the implementation of IPv4 and IPv6 protocol in TCP/IP protocol stack . (C3)
Unit 7: 6LoWPAN - 6LoWPAN architecture	
simple, extended and ad-hoc networks. Issues in determining IPv6 links in LLNs and illustration of the undetermined link addressing model. IPv6 addressing in 6LoWPAN.	1. Indicate the 6LoWPAN architecture for resource constrained devices. (C2)
Unit 8: Sockets	

Introduction to Sockets – Client Server Architecture –Unix Sockets – PORTS – Python APIs of Sockets – TCP socket programming using Python – UDP – RAW packets python programming.	1. Outline Client Server Architecture. (C1)
Unit 9: Databases & Web Programming	
Introduction to Databases – File System vs RDBMS – ER Diagram – Python Database connectivity (CRUD) - Web Server Concepts - Python Web Programming – IoT Framework.	1. Illustrate the socket communication using python API's for RWA, stream and datagram-oriented use cases. (C3)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	30	60
Quiz	02	04
Small Group Discussion (SGD)	02	02
Self-directed learning (SDL)	-	04
Problem Based Learning (PBL)	02	04
Case Based Learning (CBL)	-	-
Revision	02	-
Assessment	06	-
TOTAL	**44**	**74**

Assessment Methods:

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	
Sessional Examination 2		*	*
Assignment/Presentation		*	*
End Semester Examination	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. Arshdeep Bhaga, Vijay Madishetti, “Internet of things:A hands on Approach”, Universities Press, ISBN:978172719547 2. Robert Faludi,”Building Wireless Sensor Networks”,Orielly, 2012 3. Jean-Philippe Vasseur,Adam Dunkels,"Interconnecting Smart Objects with IP: The Next Internet",Morgan Kaufmann Publishers,2010,ISBN:0123751659 9780123751652 4. Marco Schwartz,"Internet of Things with the Arduino Yun",Packt Publishing,2014 5. Charalampos Doukas,"Building Internet of Things With the Arduino: Volume 1",CreateSpace Independent Publishing Platform,2012 6. Todor Cooklev ,“Wireless communication standards”, IEEE Press 7. Houda Labiod, Hossam Afifi, Costantino De Santis, “Wi-Fi, Bluetooth, Zigbee and WiMAX”, Springer Publications 8. Madhushree Ganguli , “Getting started with Bluetooth”, Premier Press, 2002, ISBN 1931841837, 9781931841832.

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Data Structures and Algorithms Lab
Course Code: CSE 601L	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: **1**	**Prerequisites:** C Programming

Synopsis:	This Course provides insight on 1. This course introduces students to elementary data structures and design of algorithms. 2. Students learn how to design optimal algorithms with respect to time and space 3. Students learn how to implement link list, stack, queues, searching and sorting techniques, sets, trees and graphs. 4. Students learn the design of divide and conquer technique, dynamic programming, greedy technique and back tracking
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Specify and analyse algorithms
CO 2:	Learn and design programs for implementation of linear and non linear data structure.
CO 3:	Learn and design programs for sorting and searching.
CO 4:	Illustrate application of divide and conquer technique, dynamic programming, greedy technique and back tracking.
CO 5:	Learn to organise the code for scalability and maintainability.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1		*									
CO 2		*	*		*			*			
CO 3		*	*		*			*			
CO 4		*	*		*			*			
CO 5		*	*		*			*			

Course content and outcomes:	
Content	***Competencies***
Unit 1: Elementary data structures	
Implementation of Lists, Stacks, Queues	1. Design and Implement singly linked list 2. Design and Implement doubly linked list 3. Design and Implement array-based stack 4. Design and Implement pointer-based stack 5. Design and Implement array-based queues. 6. Design and Implement pointer-based queues.
Unit 2: Sorting & Searching Techniques	
Quick sort, Heap sort, Merge sort, Binary search, linear search, Fibonacci search	1. Design and implement programs for insertion sort, bubble sort and selection sort. 2. Design and implement programs for quick sort 3. Design and implement programs for heap sort 4. Design and implement programs for merge sort 5. Design and implement programs for binary, linear and Fibonacci search
Unit 3: Trees	
Basic Terminology, Implementation of Trees, Binary Trees, Binary Search Trees	1. Write a program to implement binary trees 2. Write a program to implement binary search trees 3. Tree traversal technique

Unit 4: Graphs	
Basic definitions, Representation of Graphs, Minimum Cost Spanning Tree, Single Source Shortest Paths, All-Pairs Shortest Path	1. Write programs to represent a graph using adjacency matrix and adjacency list techniques 2. Write a program to implement minimum cost spanning tree 3. Write a program to solve Single source shortest path problem 4. Write a program to solve All- pair shortest path problem
Unit 5: Algorithm Design Techniques	
Divide-and-Conquer Algorithms, Dynamic Programming, Greedy Algorithms, Backtracking	Write a program to solve max min problem 2.Write a program to solve Strassen's matrix multiplication problem 3. Write a program to solve matrix chain order problem 4.Write programs to solve knap-sack, job scheduling with dead line and optima storage on taps problems. 5. Write programs to solve n queens and graph colouring problems

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-
Clinic	-	-

Practical	24	-
Revision	03	-
Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3	CO 4	CO 5
Sessional Examination 1	*	*			
Sessional Examination 2		*	*	*	
Assignment/Presentation	*	*	*	*	*
Laboratory Examination	*	*	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. “Introduction to Algorithms” Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. 2. “Data Structures& Algorithms” Aho, Hopcroft and Ulmann 3. “Data structures and algorithm analysis in C” Mark Allen Weiss 4. “Computer Algorithms” : Ellis Horowitz, SartajSahni, SanguthevarRajasekaran

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Microcontroller and its Applications Lab
Course Code: ESD 602L	**Course Instructor:**
Academic Year: 2020-2021	**Semester:** First Year, Semester 1
No of Credits: 1	**Prerequisites:** Microprocessor architecture, Assembly language and Number systems

Synopsis:	This Course provides insight on
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Employ the knowledge of Microcontrollers to build Embedded systems.
CO 2:	Explain the concept of Programming Microcontrollers using Assembly and Embedded C.
CO 3:	Design Embedded Systems by interfacing Sensors and Actuators.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*		*		*						
CO 2	*	*			*						
CO 3	*	*	*		*						

Course content and outcomes:

Content	***Competencies***
Unit 1: Introduction to Microprocessor & Microcontrollers	
Comparison – Variants – Types – General – ASIC – PLD – Introduction to Motherboard(Desktop) - Introduction to Embedded Board – Compare and Contrast - Application Types –	1. List different IDE's to program Microcontrollers (C1) 2. Design a Environment with tools required to build Embedded systems using Microcontrollers (C3)

Single Tasking – Multitasking – Multi-Application	
Unit 2: Introduction to ARM Microcontrollers	
Programming Model – Processor Modes – ARM vs Thumb Introduction to LPCxxxx Microcontrollers – Features – Detailing of Pins - Memory Map Concepts – RAM & ROM - Interrupts Concepts (Internal & External)	1. Demonstrate ARM Processor architecture specification using LPC 2148 Microcontroller Board (C3) 2. Demonstrate a peripherals of ARM Microcontroller using LPC 2148 Microcontroller Board (C3)
Unit 3: Reset Circuitry	
Crystals - Introduction to GPIO – Registers – Input /Output Configuration – Pull Up and Pull Down Resistor Concept – Interfacing with LED – Interfacing Push Buttons – LCD – Stepper Motor – DC Motor	1. Design an Digital notice board using LPC 2148 Microcontroller board to understand Peripherals on board (C3) 2. Design an Automated Fan / AC / Temperature control system using on chip sensors and peripherals of LPC 2148 Microcontroller board (C3)
Unit 4: Relays	
Types of Relays – Interfacing	1. Demonstrate working of Relay by controlling High voltage devices like DC Motor interfacing to ARM Microcontroller (C4)
Unit 5: Timer, Counter Introduction	
Configuration – Programming	1. Design a Digital clock using ARM Microcontroller using on chip Timer and Counter (C3)

Unit 6: Serial vs Parallel Bus	
Serial vs Parallel Bus - Compare and Contrast – Terminology: Baud Rate – Bit Rate – RS232 – DB9 handshaking concepts - Configuring Registers – Programming for UART modules.	1. Design a Master and Slave architecture using Microcontrollers and establish communication using on chip serial UART (c4)
Unit 7: **Introduction to SPI and I2C Protocol**	
Detailed Discussion – Bit Banging – Interfacing with SPI and I2C Devices – RTC / ADC /DAC.	1. Design a Serial wired communication among multiple Microcontrollers and sensors using I2C (c4) 2. Design a Serial wired communication among Microcontroller and multiple sensors in Master and Slave Architecture using SPI (c4)
Unit 8: Introduction to ADC and DAC	
Types – Chips - Register Configuration – Interfacing	1. Design a Data Acquisition system ARM Microcontroller (C4)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-

Clinic	-	-
Practical	24	-
Revision	03	-
Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	
Sessional Examination 2		*	*
Assignment/Presentation	*	*	
Laboratory Examination	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. William Hohl,Christopher Hinds,"ARM Assembly Language: Fundamentals and Techniques",2nd Edition, ISBN-13: 978-1482229851, ISBN-10: 1482229854 2. Andrew Sloss, Dominic Symes, Chris Wright,"ARM System Developer's Guide: Designing and Optimizing System Software",1st Edition,The Morgan Kaufmann Series in Computer Architecture and Design, ISBN-13: 978-1558608740, ISBN-10: 1558608745 3. David Seal, "ARM Architecture Reference Manual", 2nd Edition, Addison-Wesley Professional.

	4. Steve Furber,"ARM System-on-Chip Architecture",2nd Edition,Addison-Wesley Professional, ISBN-13: 078-5342675191,ISBN-10: 0201675196 5. Douglas V. Hall,"Microprocessors and Interfacing",Mcgraw Hill Educatin ,ISBN-10 1259006158,ISBN-13 9781259006159,2012.

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Embedded Systems Lab
Course Code: ESD 605L	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: 1	**Prerequisites:** Microprocessor architecture , Microcontroller Architecture , Assembly language and Number systems

Synopsis:	This Course provides insight on 1. This course provides the knowledge of ARM Cortex M3 Processor architecture. 2. This course provides the knowledge of Microcontroller based on ARM Processor architecture and its Registers and Instruction sets to write Assembly and Embedded C Programming. 3. This course provides the concept of Interfacing and Programming Sensors and Peripherals to Microcontrollers. 4. This course provides the concept of Real time operating systems on Microcontrollers.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Illustrate the features of embedded systems, architecture of ARM7, Instruction set and development tools of ARM.
CO 2:	Experiment the architectural features of LPC13/17XX microcontrollers, interfacing peripheral devices to LPC2148.
CO 3:	Design a Real time Embedded Systems by interfacing Sensors and Actuators and porting Real time operating systems.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*	*	*		*						
CO 2	*	*			*						
CO 3	*	*	*		*						

Course content and outcomes:	
Content	***Competencies***
Unit 1: Introduction to LPC13/17xx Microcontroller	
Introduction to LPC13/17xx Microcontroller - Hardware, SW.	At the end of the topic student should be able to: 1. Summarise LPC13/17xx Microcontroller architecture and developmen t tools of ARM. (C2)
Unit 2: Interfacing LPC13/17xx Microcontroller	
Interfacing With LED, LCD Seven Segment Display, UART, HEX Keypad.	Experiment interfacing LPC13/17xx Microcontroller with I/O devices. (C2)
Unit 3:	
Introduction to Free RTOS, FreeRTOS API Calls, Task Creation, Queques, semaphore, mutex, RTOS application development.	1. Summarise FreeRTOS architecture. (C2) 2. Practise different API call in FreeRTOS. (C2) 3. Design a Real time Embedded system by writing applications on top of Real time operating systems (C5)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-
Clinic	-	-

Practical	24	-
Revision	03	-
Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	*
Assignment		*	*
Laboratory Examination	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. Joseph Yiu, “The definitive guide to the ARM Cortex-M3”, Elsevier, 2nd Edition, 2010. 2. Frank Vahid, Tony Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, Wiley India, ISBN:81-265-0837-X, 2007. 3. Richard Barry, “NXP Semiconductors, LPC13xx/17xx User Manual”, 2012. 4. NXP Semiconductors, “LPCzone Examples”, 2012. 5. “FreeRTOS Reference Manual”, Real Time Engineers Ltd., 2016.

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Fundamentals of Machine Learning Lab
Course Code: BDA 601L	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: **1**	**Prerequisites:** Basics of Programming

Synopsis:	This Course provides insight on 1. This course provide the concept of machine learning, applications, techniques, design issues and approaches to machine learning. 2. This course provide the fundamental knowledge about concept learning, hypothesis and bias. 3. To implement machine learning algorithms such as Decision Tree learning, Probably Approximately Correct (PAC) learning, Bayesian learning, Instance-based learning, Principal Component Analysis (PCA) and Ensemble methods in real time data set for various analysis.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Identify the software and tools for designing machine-learning applications.
CO 2:	Apply concept learning and hypothesis space.
CO 3:	Apply machine-learning approach to reduce the dimension.
CO 4:	Analyse different machine learning algorithms.
CO 5:	Design ensemble methods.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*										
CO 2		*									
CO 3			*								
CO 4				*							
CO 5				*							

Course content and outcomes:

Content	***Competencies***
Unit 1: Introduction	
Definition of Machine Learning Goals and applications of machine learning Basic design issues and approaches to machine learning Types of machine learning techniques	1. Identify programming environments available for the machine learning (C1) 2. Classify the pros and cons of various environments for ML coding (C2)
Unit 2: Inductive Classification	
The concept learning task. Concept learning as search through a hypothesis space. General-to-specific ordering of hypotheses. Finding maximally specific hypotheses. Version spaces and the candidate elimination algorithm. Inductive bias.	1. Design a machine learning model to get a Maximally Specific Hypothesis for the given training examples (C5). 2. Construct a machine learning model to obtain most general and most specific hypotheses for the given training examples (C5)
Unit 3: Decision Tree learning	
Representing concepts as decision trees. Recursive induction of decision trees. Picking the best splitting attribute Entropy and information gain. Searching for simple trees and computational complexity.	1. Develop a machine learning classifier using decision tree and random forest (C5) 2. Examine different applications of decision tree and random forest (C4)

Unit 4: Computational learning theory	
Models of learnability: learning in the limit. Probably Approximately Correct (PAC) learning. Sample complexity: quantifying the number of examples needed to PAC learn. Computational complexity of training. Sample complexity for finite hypothesis spaces. Noise. Learning Multiple Classes. Bias-variance trade-off, under-fitting and over-fitting concepts.	1. Design a learning method to determine the sample complexity of training examples (C5) 2. Analyse bias-variance trade-off, under-fitting and over-fitting concepts (C4)
Unit 5: Bayesian learning	
Probability theory and Bayes rule. Naive Bayes learning algorithm - Parameter smoothing. Generative vs. discriminative training Logistic regression. Bayes nets and Markov nets for representing dependencies	1. Design a machine learning model using Bayes learning (C5). 2. Develop a machine learning classifier models using different approach (C5) 3. Design Bayes nets and Markov nets for representing dependencies (C5)
Unit 6: Instance-based learning	
Constructing explicit generalizations versus comparing to past specific examples. K-Nearest Neighbour learning algorithm.	1. Design machine learning models to classify the instances using K-NN and CBR approaches (C5).

Case-based reasoning (CBR) learning.	
Unit 7: Continuous Latent Variables	
Principal Component Analysis (PCA), Applications of PCA	1. Apply PCA for different complex applications (C3)
Unit 8: Ensemble methods (bagging and boosting)	
Using committees of multiple hypotheses. Bagging Boosting DECORATE Active learning with ensembles.	1. Design a Bayesian Networks (C5) 2. Develop machine learning models using Ensemble models. (C5)

Learning strategies, contact hours and student learning time		
Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-
Clinic	-	-
Practical	24	-
Revision	03	-
Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:	
Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3	CO 4	CO 5
Sessional Examination 1	*	*			
Sessional Examination 2			*	*	
Assignment/Presentation	*	*	*	*	*
Laboratory Examination	*	*	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. Machine Learning, T. Mitchell, McGraw-Hill, 1997 2. Machine Learning, E. Alpaydin, MIT Press, 2010 3. Pattern Recognition and Machine Learning, C. Bishop, Springer, 2006 4. Pattern Classification, R. Duda, E. Hart, and D. Stork, Wiley-Interscience, 2000 5. T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, 2nd Edition, 2009 6. Machine Learning for Big Data, Jason Bell, Wiley Big Data Series 7. Multidimensional Neural Networks Unified Theory, Rama Murthy G 8. Current literature

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Dot Net Technologies Lab
Course Code: CSE 618L	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: **1**	**Prerequisites:** Basic Programming Skills, Innovative Thinking, Enthusiasm to learn OOP concepts.

Synopsis:	To design and develop C sharp Dot net windows-based applications with database support while implementing OOP concepts.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Explore and use Visual Studio IDE to build c sharp applications.
CO 2:	Provide database support to the windows-based applications while discover and apply various features of ADO .Net
CO 3:	Design and develop ASP .Net web applications

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*	*									
CO 2	*	*	*								
CO 3			*	*							

Course content and outcomes:

Content	***Competencies***
Unit 1: Introducing C# and the .NET Platform:	
Overview of C#	1. Able to create simple programs using C sharp language.
Unit 2: The C# Programming Language:	
Variables, data types	1. Usage of Variables, data types in the programs.

Unit 3: Programming with the .NET Libraries:	
Window based applications	1. Able to build a full-fledged windows based application with database support and use various features of ADO .net.
Unit 4: Web Applications and XML Web Services:	
Using reference type variables	1. Creating and designing a Simple Web Application with Web Forms, Asp.Net Standard Controls and Configuring Server Controls in a Web Form,

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-
Clinic	-	-
Practical	24	-
Revision	03	-
Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:

Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination

Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	*
Sessional Examination 2		*	*
Assignment/Presentation	*	*	
Laboratory examination	*	*	*

Feedback Process	End-Semester Feedback
Reference Material	1. Pro C# with .NET 3.0 by Andrew Troelsen ,Apress.

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Linux & Scripting languages Lab
Course Code: AES 615.1	**Course Instructor:**
Academic Year: 2020-2021	**Semester:**
No of Credits: **1**	**Prerequisites:** Problem solving, basic programming
Synopsis:	This Course provides insight on 1. Study of scripting languages such as Bash and Perl in Linux environment. 2. The study of usage of scripting languages.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Experiment shell script programmatically using different features and debugging the code
CO 2:	Operate SED & AWK commands to do more complex task in easy way
CO 3:	Experiment PERL scripts that create and change scalar, array and hash variables

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*										
CO 2			*								
CO 3	*				*						

Course content and outcomes:

Content	***Competencies***
Unit 1:	
Essentials	1. Understand the basic concepts of shell, kernel, operating system (C2).

	2. Able to create user account (c3)
Unit 2:	
Introduction to Scripting: Shell, Tcl/tk, perl, python	1. Able to write shell script and debug the script (C3) 2. Understand the importance of shell script in real wold. (C2)
Unit 3:	
Awk utility	1. Generate report using awk script (C3)
Unit 4:	
Sed & Make	1. Perform file handling function using sed script (C4) 2. Appraise the importance of MAKE file (C3)
Unit 5:	
Perl	1. Create pattern matching , report generation and perform file handling function using Perl Script (C3)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-
Clinic	-	-

Practical	36	-
Revision	-	-
Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:	
Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	
Sessional Examination 2		*	*
Assignment/Presentation	*	*	
Laboratory examination	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. “Introduction to Linux – A Beginner’s Guide”, Machtelt Garrels 2. “Unix shell programming”, Stephen G. Kochan, Patrick H. Wood 3. “Sed & awk “,Dale Dougherty, Arnold Robbins “Programming Perl”, Larry Wall, Tom Christiansen, Jon Orwant

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Advanced Programming Techniques Lab
Course Code: CSE-622L	**Course Instructor:**
Academic Year: 2020-2021	**Semester:** First Year, Semester 1
No of Credits: 1	**Prerequisites:** Basic programming knowledge
Synopsis:	1. This course would provide fundamental knowledge of various object oriented programming concepts. 2. The course will also provide skill sets to design and develop window based java applications. 3. The course will provide essential knowledge about multi thread programming, collection framework and utility library.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Apply OOP's concepts in java applications
CO 2:	Practice UI java applications using swing components
CO 3:	Write a java application for multi thread programming
CO 4:	Apply collection framework and utility library in java applications

Mapping of COs to POs

Cos	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*	*PO 12*
CO 1		*	*		*							
CO 2		*	*		*							
CO 3		*	*		*							
CO 4		*	*		*							

Course content and outcomes:

Content	*Competencies*
Part - 1: Installation of JDK tool and supporting editor	
Installation of Java SDK, supporting editor, environment setting, Project creation, building a project, running a sample project	1. Demonstrate to install JDK and other supporting software. (C3)

Part - 2: Implementing OOP's concepts using Java	
Apply OOP's concepts such as abstraction, inheritance and polymorphism in java applications	1. Practice various OOP's concepts by writing java program (C3)
Part 3: Implementation of GUI applications	
Understand basic steps involved in building UI based java applications	1. Practice developing UI applications using various AWT components and Swing components (C3)
Part 4: Usage of Event Delegation Model	
Understand usage of Event Delegation Model in UI application	1. Experiment of interactive UI applications (C4)
Part 5: Multi thread Programming	
Creation and testing multi tread programming	1. Practice thread synchronization & inter thread communication (C3)
Part 6: Java Library	
Understand usage of Collection frame work and utility classes	1. Experiment of java applications using collection framework (C4)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-
Clinic	-	-
Practical	24	-
Revision	03	-

Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:	
Formative:	**Summative:**
Theory Assignment	Sessional Examination
Lab Assignment	University End Semester Examination
Lab Test	Viva
Viva	

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3	CO 4
Sessional Examination 1	*	*		
Sessional Examination 2		*	*	*
Assignment/Presentation	*	*	*	*
Laboratory examination	*	*	*	*

Feedback Process	1. End-Semester Feedback
Reference Material	1. Herbert Schildt – "The Complete Reference - JAVA” Seventh Edition, TATA McGRAW-HILL EDITION

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Internet of Things Lab
Course Code: IOT-607L	**Course Instructor:**
Academic Year: 2020-2021	**Semester:** First Year, Semester 1
No of Credits: **1**	**Prerequisites:** Computer Networks, Programming aspects.

Synopsis:	This Course provides insight on 1. Various elements involved in the development of application for IoT. 2. Understanding of protocols across IoT stack. 3. Scripting languages like shell and python. 4. Client Server architecture and Python APIs of Socket programming. 5. Database and Python Database connectivity, Python Web Programming, IoT Framework.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Explain basic principles of Python programming language.
CO 2:	Demonstrate the usage of networking protocols across IoT stack using Raspberry Pi and Cloud.
CO 3:	Demonstrate the fundamental concepts in Client Server architecture, database implementation and web programming with Python API's.

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*										
CO 2	*	*		*	*						
CO 3	*		*		*						

Course content and outcomes:

Content	***Competencies***
Unit 1: Python	
Introduction to Python datatypes, constructors, functions, Python Class, Modules, exception Handling, Python Packages	Employ Datatypes, Constructs, Packages in python programming. (C2)

Unit 2: Raspberry PI IoT Board	
Introduction to RPI, Raspberry Pi - Installation, first boot configuration, Raspberry Pi - Sensor Interfacing, Sending data to Cloud.	Demonstrate the usage of RPI in IoT Application Scenario. (C3)
Unit 3: Things Board Cloud	
Installation of things board Platform, Device, assets & dashboard Creation, population of data.	Illustrate the usage of things board Platform. (C4)
Unit 4: Socket Programming	
Unix Socket Programming - Client Server Architecture, Python Socket Programming - Client Server Architecture, RAW packets python programming	Illustrate the socket communication using python API's for RWA, stream and datagram-oriented use cases. (C3)
Unit 5: Databases	
Python Database connectivity (CRUD) - Web Server Concepts - Python Web Programming – IoT Framework.	Demonstrate the usage of databases, web programming using Python API . (C3)

Learning strategies, contact hours and student learning time

Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	12	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	-	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	03	-
Clinic	-	-
Practical	24	-

Revision	03	-
Assessment	06	-
TOTAL	**48**	**-**

Assessment Methods:	
Formative:	**Summative:**
Internal practical Test	Sessional examination
Theory Assignments	End semester examination
Lab Assignment & Viva	Viva

Mapping of assessment with Cos

Nature of assessment	CO 1	CO 2	CO 3
Sessional Examination 1	*	*	*
Assignment/Presentation		*	*
Lab Semester Examination	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	1. Arshdeep Bhaga, Vijay Madishetti, “Internet of things:A hands on Approach”, Universities Press, ISBN:978172719547 2. Robert Faludi,”Building Wireless Sensor Networks”,Orielly, 2012 3. Jean-Philippe Vasseur,Adam Dunkels,"Interconnecting Smart Objects with IP: The Next Internet",Morgan Kaufmann Publishers,2010,ISBN:0123751659 9780123751652 4. Marco Schwartz,"Internet of Things with the Arduino Yun",Packt Publishing,2014 5. Charalampos Doukas,"Building Internet of Things With the Arduino: Volume 1",CreateSpace Independent Publishing Platform,2012 6. Todor Cooklev ,“Wireless communication standards”, IEEE Press 7. Houda Labiod, Hossam Afifi, Costantino De Santis, “Wi-Fi, Bluetooth, Zigbee and WiMAX”, Springer Publications

	8. Madhushree Ganguli , “Getting started with Bluetooth”, Premier Press, 2002, ISBN 1931841837, 9781931841832.

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Mini Project - 1
Course Code: ESD 695	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: 4	**Prerequisites:** Any programming language and circuit basics

Synopsis:	Students are expected to select a problem in the area of their interest and the area of their specialization that would require an implementation in hardware / software or both in a semester
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Apply the objectives of the project work and provide an adequate background with a detailed literature survey
CO 2:	Breakdown the project into sub blocks with sufficient details to allow the work to be reproduced by an independent researcher
CO 3:	Compose hardware/software design, algorithms, flowchart, methodology, and block diagram
CO 4:	Evaluate the results
CO 5:	Summarize the work carried out

Mapping of COs to POs

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1				*							
CO 2					*			*			
CO 3							*			*	
CO 4						*					*
CO5:							*				

Course content and outcomes:

Content	***Competencies***
Phase 1	

Problem identification, synopsis submission, status submission, mid evaluation.	At the end of the topic student should be able to: 1. Identify the problem/specification (C1) 2. Discuss the project (C2) 3. Prepare the outline (C3) 4. Describe the status of the project (C2) 5. Prepare a mid-term project presentation report (C3) 6. Prepare and present mid-term project presentation slides (C3, C5) 7. Develop project implementation in hardware/software or both in chosen platform (C5)
Phase 2	
Status submission, final evaluation.	1. Prepare the progress report (C3) 2. Prepare the final project presentation report (C3) 3. Prepare and present final project presentation slides (C3, C5) 4. Modify and Develop implementation in hardware/software or both in chosen platform (C3, C5) 5. Justify the methods used and obtained results (C6)

Learning strategies, contact hours and student learning time		
Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	-	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	48	-
Self-directed learning (SDL)	-	-

Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	-	-
Clinic	-	-
Practical	-	-
Revision	-	-
Assessment	03	-
TOTAL	**51**	**09**

Assessment Methods:	
Formative:	**Summative:**
Project Problem Selection	Mid-Term Presentation
Synopsys review	Second status review
First status review	Demo & Final Presentation

Mapping of assessment with Cos					
Nature of assessment	CO 1	CO 2	CO 3	CO 4	CO 5
Mid Presentation	*	*			
Presentation	*	*	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	Particular to the chosen project

Name of the Program:	Master of Engineering - ME (Embedded Systems & Instrumentation)
Course Title:	Seminar - 1
Course Code: ESD 697	**Course Instructor:**
Academic Year: 2020 - 2021	**Semester:** First Year, Semester 1
No of Credits: 1	**Prerequisites:** Communication Skill
Synopsis:	1. To select, search and learn technical literature. 2. To identify a current and relevant research topic. 3. To prepare a topic and deliver a presentation. 4. To develop the skill to write a technical report. 5. Develop ability to work in groups to review and modify technical content.
Course Outcomes (COs):	On successful completion of this course, students will be able to
CO 1:	Show competence in identifying relevant information, defining and explaining topics under discussion.
CO 2:	Show competence in working with a methodology, structuring their oral work, and synthesizing information.
CO 3:	Use appropriate registers and vocabulary, and will demonstrate command of voice modulation, voice projection, and pacing.
CO 4:	Demonstrate that they have paid close attention to what others say and can respond constructively.
CO 5:	Develop persuasive speech, present information in a compelling, well-structured, and logical sequence, respond respectfully to opposing ideas, show depth of knowledge of complex subjects, and develop their ability to synthesize, evaluate and reflect on information.
Mapping of COs to POs	

COs	*PO 1*	*PO 2*	*PO 3*	*PO 4*	*PO 5*	*PO 6*	*PO 7*	*PO 8*	*PO 9*	*PO 10*	*PO 11*
CO 1	*							*	*		*
CO 2	*							*	*		*
CO 3	*							*	*		*
CO 4	*							*	*		*

CO5:	*							*	*		*

Learning strategies, contact hours and student learning time		
Learning strategy	*Contact hours*	*Student learning time (Hrs)*
Lecture	-	-
Seminar	-	-
Quiz	-	-
Small Group Discussion (SGD)	14	-
Self-directed learning (SDL)	-	-
Problem Based Learning (PBL)	-	-
Case Based Learning (CBL)	-	-
Clinic	-	-
Practical	-	-
Revision	-	-
Assessment	-	-
TOTAL	**14**	**-**

Assessment Methods:	
Formative:	**Summative:**
Seminar Topic Selection	
Synopsys review	
PPT Review	

Mapping of assessment with Cos					
Nature of assessment	CO 1	CO 2	CO 3	CO 4	CO 5
Presentation	*	*	*	*	*

Feedback Process	• End-Semester Feedback
Reference Material	Particular to the chosen Seminar

PROGRAM OUTCOMES (POS) AND COURSE OUTCMES (COS) MAPPING

Sl.No.	Course Code	Course Name	Credits	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
1	CSE 601	Data Structures and Algorithms	3	*	*		*		*					
2	ESD 602	Microcontrollers and its Applications	3	*	*	*		*						
3	ESD 605	Embedded Systems	3	*	*	*		*						
4	BDA 601	Fundamentals of Machine Learning	3	*	*	*	*							
5	CSE-618	Dot Net Technologies	3	*	*	*	*							
6	CSE-620	Linux and Scripting Languages	3	*	*	*	*							
7	CSE-622	Advanced Programming Techniques	3	*	*	*	*	*						
8	IoT 607	IoT	3	*	*	*		*						
9	CSE 601L	Data Structures and Algorithms Lab	1		*	*		*			*			
10	ESD 602L	Microcontrollers and its Applications Lab	1	*	*	*		*						
11	ESD 605L	Embedded Systems Lab	1	*	*	*		*						
12	BDA 601L	Fundamentals of Machine Learning Lab	1	*	*	*	*							
13	CSE-618L	Dot Net Technologies Lab	1	*	*	*	*							
14	CSE-620 L	Linux and Scripting Languages Lab	1	*		*		*						
15	CSE-622 L	Advanced Programming Techniques Lab	1		*			*						
16	IOT-607 L	Internet of Things Lab	1	*	*	*	*	*						

17	EDA 695	Mini Project - 1	4				*	*	*	*	*		*	*
18	EDA 697	Seminar - 1	1	*							*	*		*

Semester 2: ESIGELEC, FRANCE

Virtual Instrumentation

Module code: MSCCEIxx Duration: 30h

Objectives: At the end of this module, students will be able to:

- Use LabVIEW to create applications
- Understand front panels, block diagrams, and icons and connector panes
- Use built-in LabVIEW functions
- Create and save programs in LabVIEW so students can use them as subroutines
- Create applications that use plug-in DAQ devices. The application must respect standard LabVIEW practices (taken from the Certified LabVIEW Developer (CLD) test) and use a modular and evolving architecture
- Design a program with LabVIEW for an electrocardiogram that monitors real and “noisy” data. This program must:
    - Respect design standards
    - Use standard programming

List of topics:

- Fundamental programming notions in LabVIEW
- LabVIEW programming
- Creating an interface
- Learning good LabVIEW practices for form and structure in programming

Specific Instrumentation

Module Code: MSCCEIxx Duration: 30h

Objectives: At the end of this module, students will:

- Manage the entire information sampling chain in an instrumentation-type embedded system

List of topics:

- The measurement chain:
    - From the physical signal to digital processing

Sensors:

- Types
- Technology

Signal conditioning:

- Transport
- Filtering
- Amplification

Sampling:

- Period
- Response time

Information security:

- Accuracy
- Lifetime
- Redundancy

Embedded C Programming

Module Code: MSCCEIxx Duration: 30h

Objectives:

At the end of this module, students will be:

- Familiar with the C coding practices for embedded systems
- Familiar with the elements and tools for embedded software validation
- Able to develop, write and test a C language program (as per design specifications) to be used with a microprocessor with respect of good practices like MISRA-C rules
- Able to analyse and enumerate the various phases of development for a software project: the V cycle
- Able to program a microcontroller and develop embedded applications. These applications will deal with digital inputs/ outputs, analog signals and will create delays and time events by means of hardware timer
- Able to apply techniques and rules to ensure software quality and best coding practices (A sizeable part of the course is devoted to programming the microcontroller)

List of topics:

- Specificities of C Language for embedded systems (variables, memory organization, physical address access, etc.)
- Introduction to embedded system and programming methods
- Software analysis and validation tools and principles for embedded systems
- C language for embedded systems

- Best coding practices
- Programming the MSP430 microcontroller

Artificial Intelligence for Smart Systems

Module Code: MSCCEIxx Duration: 30h

Objectives:

At the end of this module, students will be able to:

- Identify artificial intelligence problems in the smart embedded systems field.
- Describe the principle of some of the most widespread artificial intelligence method
- Develop a basic scenario as an application for a smart embedded system: for example autonomous mobile robot problem, using existing building blocks and software tools

List of topics:

- Artificial intelligence issues
- Possible applications in the field of mobile robotics: recognition of road signs, obstacles, pedestrians, faces, etc.
- Study of some of the most widespread methods
- Existing systems in the automotive field
- Implementation: C/C++ programming, Scilab, python, and use of the OpenCV library

Smart Sensors

Module Code: MSCCEIxx Duration: 30h

Objectives:

At the end of this module, students will be able to:

- Describe the typical internal architecture of such a sensor, the advantages and disadvantages associated with it and the current uses of this type of system
- Understand the complexity and the benefits of using this kind of technology

List of topics:

- “Smart” vs “dumb” sensors
- Observer (human) effect and Schrödinger’s cat dilemma in sensing
- Statistical modelling of sensing/measurements
- Signal processing for smart sensing
- Communication systems
- Case studies

Project Management

Module Code: MSCCEIxx Duration: 26h

Objectives:

At the end of this module, students will be able to:

- Appreciate the need for project management including formal methods, as a recognized discipline.
- Understand the complexities of different types of computing projects and some of the methods used to manage them.
- Appreciate the need to break up complex projects.
- Appreciate the need for effective planning, monitoring and control mechanisms.
- Understand the need for formal project management organizational structures.
- Understand the importance and management of stakeholders in an international project.
- Apply some of the skills and knowledge learned in any future project (including during other module(s) of this course, and, in particular, documentation for development project).
- Understand the complexity of a technical project and the need for formal methods

List of topics:

- What is a project? The need for PM, formal methods
- Managing large, complex, international projects
- Un peu de franglais (PM culture and language in English and in French)

- Management of projects, project life cycle, roles of the project manager and stakeholders.
- Stakeholder management, scope, creep
- Work planning, project breakdown structures and estimating.
- Resource planning, estimating, management.
- Risk identification, analysis, management.
- PERT and Gantt charts, their use and shortcomings.
- PM planning tools (including practical sessions with MS Project)
- Change control, documentation, configuration management.
- Project control, quality, documentation, delivery management.
- Project closure; maintenance projects.
- Types of computing projects and risks; computing PM methods.
- Cost-benefit analysis and project accounting may be touched upon, but are not in the scope of this course.

Embedded Java

Module Code: MSCCEIxx Duration: 30h

Objectives:

At the end of this module, students will:

- Be familiar with a computer language which can be used to develop graphic applications under Windows for personal embedded systems like Pocket PCs

List of topics:

- Java ME environment: interface and syntax
- Basics of programming in the Java ME environment

Real-time Operating Systems

Module Code: MSCCEIxx Duration: 30h

Objectives:

At the end of this module, students will be able to:

- Understand why real-time executive is used in embedded systems
- Describe the four major categories of services provided by an executive
- Describe the necessary required materials to implement an executive in real-time
- Learn the various commercial aspects of executive suppliers
- Describe the role of scheduling, how it works and the major variations
- Calculate task times for simple situations
- List attribution rules for task priority
- Describe how the principle elements for synchronization are presented in executives
- Describe the characteristics and how an email inbox works
- Design and develop a simple multitasking application with MicroC / OSI

List of topics:

- Fundamentals of multitasking and real-time
- A scheduler: its role and how it works
- Why real-time executives are used in embedded systems
- Necessary materials
- Categories of executives and their markets
- A real-time kernel: MicroC/OSII (Micro-Controller Operating Systems Version 2)
- Memory management

- Intertask communication and synchronization tools
- Using MicroC/OSII and microcontrollers

Embedded Linux
Module Code: MSCCEIxx Duration: 30h

Objectives:
At the end of this module, students will:

- Understand the possibilities and uses of the Linux kernel for an embedded IT project.
- Learn the principle software tools used in the Linux/Unix world and how to use them to develop.
- Write a device driver for specific Linux run material
- Combine tools to create advanced functions with a minimum of programming

List of topics:

- Introduction to Linux.
- How an OS fits in an embedded system.
- History of Linux and Unix systems.
- Linux compared to other embedded operating systems.
- Fundamental tools: command lines, shell scripts.
- Linux development tools.
- C programming with embedded systems.
- Linux drivers.
- Web connections and Remote Administration Tools (RATs)

Mobile Robotics & Perception

Module Code:MSCCEIxx Duration: 30h

Objectives:

At the end of this module, students will be able to:

- Name the name and function of the different elements of a mobile robot
- Describe the architecture of a mobile robot
- Design, code and test an algorithm allowing the robot to move while avoiding obstacles
- Cite the problems of mobile robotics: modelling, trajectory planning, localization, navigation

List of topics:

- Introduction to Mobile Robotics
- Sensors used in mobile robotics
- Actuators used in mobile robotics
- The different mobile platforms
- Modelling and Control Laws in Mobile Robotics
- Location
- Navigation and trajectory planning

EMC Automotive Systems

Module Code: MSCCEIxx Duration: 26h

Objectives:

At the end of this module, students will:

- Understand EMC System architecture
- Understand Integrity signal and how to calculate it
- Understand EMC of components and how to protect electronic system
- Understand near field and interactions with the environment

List of topics:

- EMC Introduction
- Integrity Signal (IS)
- EMC of components
- Near-field

Oral Communication

Module Code: MSCCEIxx Duration: 15h

Objectives:

At the end of this module, students will:

- Have a clear model of what constitutes successful and unsuccessful presentations
- Have practiced giving formal presentations in English
- Be more aware of their own shortcomings when presenting
- Practice and perfect final presentation skills
- Learn the importance of structure and how formal prepared speech differs from everyday social interactions
- Work with their presenting strengths and weaknesses via several short practice presentations and a final (individual and/or group) presentation

List of topics:

- Methods for creating a final presentation
- Practice

R&D Project

Module Code: MSCCEIxx Duration: 60h

Objectives:

At the end of this module, students will be able to:

- Design, develop and realize an embedded system in mobile robotics and embedded electronics
- Develop technical solutions-based electronic equipment or an electronic board: hardware and software
- Test the platform developed
- Develop and carry out an embedded system platform successfully
- Manage a technical project

List of topics:

Project Management:

- Benchmarking study
- Technical and Functional specifications
- Architecture Design and Risk analysis
- Test protocol

Technical Development: Image processing and computer vision systems:

- Image segmentation
- Pattern recognition
- Object detection and tracking
- Artificial Intelligence and Deep Learning Applications for mobile robotics and electronic applications

- Dataset collection
- Mobile robotics and autonomous navigation
- IoT and sensors
- Smart mobility

French Language 2

Module Code: MSCAESLANG Duration: 64h

Objectives:

At the end of this module, students will be able to:

- Understand standard French used in everyday situations at work, school, etc. (Oral comprehension)
- Understand texts written in standard French used in everyday situations such at work, school, etc. (Written comprehension)
- Participate in a regular day-to-day conversation on familiar topics (Oral expression)
- Ask and exchange information (Oral expression)
- Prepare and give a short formal presentation (Oral expression)
- Write short, clear and coherent texts on familiar/everyday situations with basic grammar and vocabulary (Written expression)

List of topics:

- Revision of grammar and vocabulary
- Preparation for the Test of French Language (TCF or TEF)