JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

Vision and Mission of the University

Vision

The University is primarily promoting quality of education in the areas of Science, Technology, Engineering and Mathematics (STEM) as four academic pillars of education, to excel in teaching, learning, research, consultancy and placements through innovative practices with global perspective.

Mission

1. Design an Industry relevant curriculum from time to time with a Global perspective
2. Promoting quality education by embracing ICT delivery mechanism with continuous pedagogy through e-learning mechanism
3. Spread across for industry collaborations with a focus to pre-training and placements for technology transfer to society
4. Establishing centers of excellence to promote research and innovations in multidisciplinary areas to bring in patent culture and consultancy practices
5. International Collaborations for student outreach
6. Facilitating international students to study in JNTUK to infuse cross culture learning practices.

Vision and Mission of the Institute

Vision and Mission of the Department

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

Program Educational Objectives (PEOs)

PEO-1: Graduates will develop strong analytical and design skills for building efficient embedded systems using modern microcontrollers, SoCs, and real-time architectures.

PEO-2: Graduates will be proficient in the development and testing of high-performance hardware accelerators for embedded applications across domains like automotive, healthcare, and automation.

PEO-3 :Graduates will be capable of designing parallel programs using CUDA and GPU architectures to improve the performance of embedded applications.

Programme Outcomes

PO1: An ability to independently carry out research /investigation and development work to solve practical problems

PO2: An ability to write and present a substantial technical report/document

PO3: Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4 : An ability to design, test, and validate embedded hardware/software systems, including microcontroller-based and SoC-based solutions, for real-time applications.

PO5: An ability to apply advanced techniques in signal and image processing, sensor integration, and system interfacing to develop intelligent embedded systems for real-world applications.

PO6: An ability to develop and implement parallel and high-performance computing solutions using platforms such as GPUs, FPGAs, and multi-core embedded processors to accelerate embedded system functionality

Mapping of PEOs with Pos:

	PO1	PO2	PO3	PO4	PO5	PO6
PEO1	M	M	H	H	M	M
PEO2	H	M	H	H	H	H
PEO3	M	L	M	M	M	H

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

M.TECH

(EMBEDDED SYSTEMS)

Programme Course Structure & Syllabus

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

Program Structure R25 M. Tech (EMBEDDED SYSTEMS)

I - Semester

S.No	Course Title	L	T	P	C
1	Embedded Systems	3	1	0	4
2	System design with Embedded Linux	3	1	0	4
3	ARM Microcontroller based design	3	1	0	4
4	Program Elective-I	3	0	0	3
5	Program Elective-II	3	0	0	3
6	Embedded Systems Lab	0	1	2	2
7	System Design with Embedded Linux Lab	0	1	2	2
8	Seminar-1	0	0	2	1
	Total	**15**	**5**	**6**	**23**

List of Professional Elective Courses in I Semester (Electives – I & II)

S.No	Course Code	Course Title
1	PE -I	Software for Embedded Systems
2	PE -I	Hardware Software Co-design
3	PE -I	Micro electromechanical Systems (MEMS)
4	PE -I	Pervasive devices and Technology
5	PE -II	Real Time Systems
6	PE -II	Embedded System Integration
7	PE -II	Embedded Networks and Protocols
8	PE -II	RISC Processor Architecture and Programming

@ Minimum 2/3 themes per elective

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

R25-M.Tech(EMBEDDED SYSTEMS) II-Semester

S.No	Course Code	Course Title	L	T	P	C
1	PC	System design using embedded processors	3	1	0	4
2	PC	Embedded and Real Time Operating Systems	3	1	0	4
3	PC	Internet of Things	3	1	0	4
4	PE-III	Program Elective-III	3	0	0	3
5	PE-IV	Program Elective-IV	3	0	0	3
6		System Design using Embedded Processors Lab	0	1	2	2
7		Embedded and Real Time Operating Systems Lab	0	1	2	2
8		Seminar-II	0	0	2	1
		Total	**15**	**5**	**6**	**23**

List of Professional Elective Courses in II Semester (Electives – III & IV)

S.No	Course Code	Course Title
1	PE -III	AI for Embedded Systems
2	PE -III	Embedded System design using FPGA
3	PE -III	ASIC & SoC Design
4	PE -III	Robotics and Machine Vision
5	PE -IV	Network Embedded Applications
6	PE -IV	GPU Architectures and Programming
7	PE -IV	Embedded Computing System
8	PE -IV	Cryptography and Network Security

@ Minimum 2/3 themes per elective

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

R25-M.Tech(EMBEDDED SYSTEMS) III-Semester

S.No	Course Title	L	T	P	C
1	Research Methodology and UPR/Swayam 12 Week MOOC course-RM & IPR	3	1	0	4
2	Summer Internship/Industrial training(8-10)Weeks*	-	-	-	3
3	Comprehensive Viva #	-	-	-	2
4	Dissertation part -A$	-	-	20	10
	Total	**3**	**-**	**20**	**18**

*Student attended during summer /year break and assessment will be done in 3rdSem

Comprehensive viva can be conducted courses completed upto second sem

$ Dissertation-Part A internal Assessment

R25-M.Tech(EMBEDDED SYSTEMS) IV-Semester

S.No	Course Title	L	T	P	C
1	Dissertation Part – B%	-	-	32	16
	Total	-	-	32	16

% Extrenal Assessment

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	EMBEDDED SYSTEMS	L	T	P	C
		3	1	0	4

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the fundamentals of embedded systems and describe various CPU architectures along with a wide range of microcontrollers.	K2
CO2	Demonstrate knowledge of CPU processors, their operating modes, exception handling mechanisms, instruction pipelining, and basic programming techniques.	K4
CO3	Implement embedded programs using Assembly and C language specifically for ARM Cortex-M architecture.	K4
CO4	Design interfacing solutions for keys, LED/LCD displays, ADCs, and DACs.	K5
CO5	Design and develop cost-effective, real-world embedded systems by selecting appropriate hardware/software components and integrating embedded operating systems.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	H	L	L
CO2	M	L	H	H	L	L
CO3	H	M	H	H	M	M
CO4	H	M	H	H	M	L
CO5	H	M	H	H	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit No.	Syllabus	Contact Hours
Unit I	**Introduction to Embedded Systems:** Overview and characteristics of embedded systems, classification and application areas, development process, RISC vs CISC architectures, microcontroller families (8/16/32-bit), components in embedded development environments (IDE).	**12**
Unit II	**ARM Cortex M3/M4 Architecture:** Overview of ARM Cortex family, operation modes and states, registers (general and special), memory system and MPU, exception and interrupt handling, system control block, OS support features, and instruction set (arithmetic, logic, branch, etc.).	**12**

Unit III	**Programming Cortex M3/M4 in Assembly and C:** Structured programming, subroutines, 64-bit data processing, mixing C and assembly, NVIC and interrupts, fixed/floating-point arithmetic, optimized code writing, and fault handling routines.	**12**
Unit IV	**Using Embedded OS:** Introduction to embedded operating systems, tasks and threads, thread creation, inter-thread communication, signals, semaphores, message queues, and OS-based programming examples.	**12**
Unit V	**Peripheral Interfacing:** General-purpose I/O, timers, RTC, DMA, ADC, DAC, and serial communication interfaces including UART, I2C, SPI, Ethernet, CAN, etc.	**12**
	Total	**60**

TEXTBOOKS:

1. **Joseph Yiu**, A Definitive Guide to the ARM Cortex-M3 and Cortex-M4 Processors, 3rd Edition, Newnes, 2013.
2. **A. N. Sloss, D. Symes, and C. Wright**, ARM System Developer’s Guide: Designing and Optimizing System Software, Elsevier, 2004.

REFERENCE BOOKS:

3. **Yifeng Zhu**, Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C, E-Man Press LLC, 2014.
4. **Wayne Wolf**, Computers as Components: Principles of Embedded Computing System Design, 2nd Edition, Morgan Kaufmann, 2008.
5. **Prasad K. V. K. K.**, Embedded / Real-Time Systems: Concepts, Design and Programming, DreamTech Press, 2005.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	SYSTEM DESIGN WITH EMBEDDED LINUX	L	T	P	C
		3	1	0	4

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Execute Linux and File I/O Commands.	K2
CO2	Analyze Kernel Architecture and Scheduler Features.	K3
CO3	Develop Device Drivers for various peripherals.	K4
CO4	Explore Linux Root File System and concepts of Embedded Linux.	K2
CO5	Analyze RT Linux Basics and OS Safety.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	M	L	M	L
CO2	H	M	H	H	M	M
CO3	H	M	H	H	M	H
CO4	M	M	H	M	H	M
CO5	M	M	H	H	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit No.	Syllabus	Contact Hours
Unit I	**Overview of LINUX:** Introduction to UNIX/LINUX, LINUX Commands, File I/O (open, create, close, lseek, read, write), Process Control (fork, vfork, exit, wait, waitpid, exec), Embedded LINUX Vs Desktop LINUX, Embedded LINUX Distributions.	12
Unit II	**Linux Kernel:** Embedded Linux Architecture, Kernel Architecture, Hardware Abstraction Layer, Memory Manager, Scheduler, File System, I/O and Networking Subsystem, Inter Process Communication, User Space, and Start-up Sequence.	12
Unit III	**Embedded Drivers:** Board Support Package: Embedded Storage, Memory Technology Devices (MTD), Embedded Drivers: Serial, I2C, USB, Ethernet, Timer, Kernel Modules, and Embedded File System.	12

Unit IV	**Building and Debugging:** Kernel, Root File System, Case Studies: RTL LINUX, Micro C/OS-II, VxWorks, Embedded Linux, and Tiny OS.	12
Unit V	**Linux Tasks:** Porting Applications, Real-Time Linux Basics, Kernel Priority, Task Creation, Print Commands, Compilation, Safety-Critical Features, Components, Programs.	12
	Total	**60**

TEXTBOOKS:

1. Chris Simmonds, "Mastering Embedded Linux Programming" - Second Edition, PACKT Publications Limited.

2. Karim Yaghmour, "Building Embedded Linux Systems", O'Reilly & Associates

3. P Raghvan, Amol Lad, Sriram Neelakandan, "Embedded Linux System Design and Development", Auerbach Publications

REFERENCE BOOKS:

1. Christopher Hallinan, "Embedded Linux Primer: A Practical Real-World Approach", Prentice Hall, 2nd Edition, 2010.

2. Derek Molloy, "Exploring Beagle Bone: Tools and Techniques for Building with Embedded Linux", Wiley, 1st Edition, 2014

I Semester	ARM MICROCONTROLLER BASED DESIGN	L	T	P	C
		3	1	0	4

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Explore the selection criteria of ARM processors by understanding the functional level tradeoff issues.	K2
CO2	Implementations on ARM developments towards the functional capabilities	K4
CO3	Work with ASM level program using the instruction set.	K2
CO4	Programming the ARM Cortex M.	K5
CO5	Discuss about Floating Point Operations:	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	M	M	L
CO2	M	L	H	M	M	L
CO3	M	L	H	H	M	M
CO4	H	M	H	H	M	M
CO5	H	M	H	H	M	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit No.	Syllabus	Contact Hours
Unit I	**ARM Embedded Systems:** RISC design philosophy, ARM design philosophy, Embedded system hardware, Embedded system software. **ARM Processor Fundamentals:** Registers, CPSR, Pipeline, Exceptions, Interrupts and Vector Table, Core Extensions, Architecture Revisions, ARM Processor Families. **Architecture of ARM Processors:** Programmer's model, modes and states, special and floating-point registers, APSR, Memory system, MPU, Exceptions, NVIC, vector table, Fault handling, SCB, Debug, Reset sequence.	12

Unit II	**ARM Instruction Set:** Data processing, branch, load-store, software interrupt, program status register instructions, loading constants, ARMv5E extensions, Conditional execution. Thumb Instruction Set: Thumb Register Usage, ARM-Thumb Interworking, Branch, Data Processing, Load-Store, Stack, and Software Interrupt Instructions.	12
Unit III	**Technical Details of Cortex M Processors:** Overview of Cortex-M3 and M4: architecture, instruction set, block diagram, memory system, exception and interrupt support. Features: Performance, code density, low power, MPU, OS support, Cortex-M4-specific DSP features, Debug support, Scalability, Compatibility.	12
Unit IV	**Instruction Set of Cortex M:** Instruction set background, comparison across Cortex-M processors, UAL syntax, instruction suffixes, Cortex-M4-specific instructions, Barrel shifter, Special instructions and register access.	12
Unit V	**Floating Point Operations**: Floating point data and FPU overview (CPACR, FP registers, FPSCR, FPCCR, FPCAR, FPDSCR, MVFR0, MVFR1). DSP Applications: Dot product, Biquad filter, FIR, FFT and optimized DSP code writing for Cortex-M4.	12
	Total	**60**

TEXTBOOKS:

1. Andrew N.SLOSS, Dominic SYMES, Chris WRIGHT-ARM System Developer's Guide Designing and Optimizing System Software, Elsevier Publications, 2004.
2. Joseph Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors by Elsevier Publications, 3rdEd.,

REFERENCE BOOKS:

1. Steve Furber-Arm System on Chip Architectures–EdisonWesley,2000.
2. David Seal-ARM Architecture Reference Manual, Edison Wesley,2000.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	SOFTWARE FOR EMBEDDED SYSTEMS	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the fundamentals of real-time and embedded systems, including software issues, development processes, and requirement analysis using use case modeling.	K2
CO2	Design embedded software architecture using design patterns and implement it using appropriate programming languages, compilers, and embedded operating systems.	K5
CO3	Develop timing and delay routines using hardware timers (Timer 0, Timer 1) and implement accurate portable hardware delays for embedded systems.	K4
CO4	Analyze and implement timeout mechanisms using loop-based and hardware-supported techniques to ensure system reliability in time-sensitive applications.	K4
CO5	Design and test reliable embedded interfaces using hardware timeouts and debounce logic for practical systems like switch interfaces and data acquisition setups.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	M	L	L
CO2	H	M	H	H	M	M
CO3	H	L	H	M	M	M
CO4	H	M	H	M	M	M
CO5	H	M	H	H	M	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	**Syllabus Content**	**Contact Hours**
Unit I	Real-Time and Embedded Systems: Introduction to real-time and embedded systems; software issues in embedded system development; software development process; requirement analysis techniques; use cases – definition, identification, and analysis; creation and interpretation of use case diagrams.	12
Unit II	Embedded Software Design and Testing: Architecture design of embedded systems; introduction to design patterns and detailed design methods; implementation aspects including programming languages (C, Embedded C, Assembly), compilers, runtime environments, and operating systems; software	12

	testing methodologies; writing and evaluating test cases.	
Unit III	Timing and Delay Mechanisms: Introduction to timing in embedded systems; creating hardware delays using Timer 0 and Timer 1; writing portable hardware delay routines; calculating time delays based on system clock; impact of timer configuration on delay accuracy and resolution.	12
Unit IV	Timeout Mechanisms: Understanding the need for timeout mechanisms in embedded systems; techniques for creating loop-based timeouts; testing and validating timeout functionality; practical considerations for real-time system behavior and reliability.	12
Unit V	Hardware Timeouts and Interfacing Techniques: Designing reliable switch interfaces using debounce and timeout techniques; implementing hardware-based timeout mechanisms; testing timeout routines through practical examples; developing robust embedded interfaces; conclusions and best practices in timing control.	12
	Total	**60**

TEXT BOOKS:

1. Embedded-C -MichaelJ.Pont,2ndEd., Pearson Education, 2008

REFERENCE BOOKS:

1. PIC micro MCUC-An introduction to programming, The Micro chip PIC in CCSC–Nigel Gardner.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	HARDWARE SOFTWARE CO-DESIGN	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand co-design issues including models, architectures, languages, and generic co-design methodologies; analyze hardware–software partitioning and distributed system co-synthesis algorithms.	K2
CO2	Evaluate prototyping and emulation techniques and environments; analyze architecture specialization techniques and system communication infrastructures for various embedded application domains such as control- and data-dominated systems.	K4
CO3	Examine modern embedded processor architectures, embedded software development tools, and compilation techniques suitable for performance-constrained systems.	K3
CO4	Apply principles of design specification, co-design computation models, concurrency management, and component interfacing; utilize verification techniques and tools for embedded system validation.	K4
CO5	Explore system-level specification and design using dedicated languages; apply heterogeneous specifications and multi-language co-simulation using tools such as Cosyma and Lycos.	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	H	M	H	M	L	M
CO2	H	M	H	H	M	H
CO3	H	M	H	H	M	H
CO4	H	M	H	H	M	M
CO5	H	M	H	H	M	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit I	Co-Design Issues: Co-Design Models, Architectures, Languages, A Generic Co-design Methodology. Co-Synthesis Algorithms: Hardware-software synthesis algorithms, hardware–software partitioning, distributed system co-synthesis.	12
Unit II	Prototyping and Emulation / Target Architectures: Prototyping and emulation techniques, environments, and future developments; architecture specialization techniques; system communication infrastructure. Target Architectures:	12

	Architecture specialization techniques, system communication infrastructure, target architecture and application system classes. Architectures for control dominated systems (8051), high-performance control architectures, architectures for data dominated systems (ADSP21060, TMS320C60), and mixed systems.	
Unit III	Compilation Techniques and Tools: Compilation techniques and tools for embedded processor architectures; modern embedded architectures; embedded software development needs; compilation technologies; practical considerations in compiler development.	12
Unit IV	Design Specification and Verification: Design and co-design concepts, co-design computational model, coordinating concurrent computations, interfacing components. Design verification, implementation verification, verification tools, and interface verification.	12
Unit V	Languages for System-Level Specification and Design: System-level specification and synthesis; design representation for system-level synthesis; system-level specification languages. Heterogeneous specifications and multi-language co-simulation; the Cosyma system and Lycos system.	12
	Total	**60**

Text Books:

1. Hardware/Software Co-Design Principles and Practice–Jorgen Staunstrup, Wayne Wolf – 2009, Springer.
2. Hardware/Software Co-Design-Giovanni De Micheli, Mariagiovanna Sami,2002, Kluwer Academic Publishers

Reference Books:

1. A Practical Introduction to Hardware/Software Co-design -Patrick R. Schaumont - 2010 Springer

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	MICRO ELECTRO MECHANICAL SYSTEMS (MEMS)	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the fundamentals of MEMS and NEMS, including definitions, classification, materials, and applications of micro- and nano-electromechanical systems across various domains.	K2
CO2	Explain and analyze MEMS fabrication technologies such as photolithography, etching, thin-film deposition, micromachining techniques (bulk, surface, LIGA), and microsystem packaging methods.	K2
CO3	Analyze the working principles and engineering mechanics behind MEMS-based microsensors such as acoustic wave sensors, resonant sensors, gyroscopes, and pressure sensors, with practical insights through case studies.	K4
CO4	Design and evaluate various micro actuators including thermal, electrostatic, piezoelectric, and shape memory alloy-based actuators, and understand micromechanical motors and pumps through real-world examples.	K5
CO5	Apply MEMS design methodologies considering constraints such as sensitivity and stability; explore interfacing techniques, control methods, simulation tools, and packaging/testing strategies for MEMS devices including RF applications.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	H	L	H	L	M	L
CO3	H	M	H	M	H	L
CO4	H	M	H	M	H	L
CO5	H	M	H	M	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit I	Overview and Introduction: New trends in Engineering and Science: Micro and Nano scale systems. Introduction to Design of MEMS and NEMS. Overview of Nano and Micro electromechanical Systems. Applications of	12

	Micro and Nano electromechanical systems. Micro electromechanical systems, devices and structures: Definitions. Materials for MEMS: Silicon, silicon compounds, polymers, metals.	
Unit II	MEMS Fabrication Technologies: Micro-system fabrication processes: Photolithography, Ion Implantation, Diffusion, Oxidation. Thin film depositions: LPCVD, Sputtering, Evaporation, Electroplating. Etching techniques: Dry and wet etching, electrochemical etching. Micromachining: Bulk Micromachining, Surface Micromachining, High Aspect-Ratio (LIGA and LIGA-like) Technology. Packaging: Microsystems packaging, Essential packaging technologies, Selection of packaging materials.	12
Unit III	MEMS Sensors: Design of Acoustic wave sensors, resonant sensor, Vibratory gyroscope, Capacitive and Piezo Resistive Pressure sensors – engineering mechanics behind these Microsensors. Case study: Piezo-resistive pressure sensor.	12
Unit IV	Micro Actuators: Design of Actuators: Actuation using thermal forces, shape memory alloys, piezoelectric crystals, electrostatic forces (Parallel plate, Torsion bar, and Comb drive actuators). Micromechanical Motors and Pumps. Case study: Comb drive actuators.	12
Unit V	Design & Interface: Design methods and design constraints for sensitivity and stability. Implementation of control methods for improving measurement sensitivity, linearity and reproducibility. Interface Electronics for MEMS. MEMS Simulation Techniques. MEMS for RF application. Bonding, Packaging and Testing of MEMS Devices.	12
	Total	**60**

Text Books:

1. MEMS Introduction and fundamentals, Mohammed Gad-El-Hak, Taylor & Francis.
2. Foundations of MEMS, Chang Liu,Pearson education India limited,2006.
3. MEMS and Microsystems Design and Manufacture, TaiRanHsu, TataMcrawHill,2002.

Reference Books:

1. Fundamentals of Micro fabrication, Marc Madou,CRCpress,1997.
2. Micro system Design, Stephen D. Senturia, Kluwer Academic Publishers,2001

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	PERVASIVE DEVICES AND TECHNOLOGY	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Relate to current trends in pervasive computing and develop a sense of their practicality	K2
CO2	Identify distinguishing features of the different mobile device categories, namely, Pocket PCs, Personal Digital Assistants (PDAs), and wireless phones.	K3
CO3	Recognize the difference between writing code for work stations and servers on one hand and for resource-constrained devices on the other hand.	K4
CO4	The learning process delivers insight onto building of sensor networks, communication in zigbee network and sensor netwoks protocols are studied.	K3
CO5	Design and develop a pervasive computing device for a specific need. And Develop a frame work for pervasive computing.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	M	H	M	M	L
CO2	M	L	M	L	L	L
CO3	M	L	H	L	M	L
CO4	H	M	H	H	M	M
CO5	H	M	H	M	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	**Syllabus**	**Contact Hours**
Unit I	Challenges for Wireless Sensor Networks, Characteristic requirements for WSN, WSN vs Adhoc Networks, Sensor node networking with commercial sensor nodes, Physical layer and transceiver design in WSNs, Applications of sensor networks	12
Unit II	Single-node architecture, Hardware components, Resource constraints, Energy consumption, Operating systems for WSNs, Network architecture, Sensor network scenarios, Optimization goals, Gateway concepts, Data dissemination (Flooding, Gossiping), Design principles for WSNs	12
Unit III	Wireless LAN – IEEE 802.11 system & protocol architecture, Services, AdHoc Networks, HiperLAN, Bluetooth, Wireless PAN, Wireless MAN,	12

	Wireless Backbone Networks, Wireless Access Technologies	
Unit IV	Wireless Sensor Network MAC protocols, Low duty cycle and wakeup concepts, Contention-based and schedule-based protocols, IEEE 802.15.4 MAC, Energy usage, Modulation schemes, SMAC, LEACH, Zigbee communication	12
Unit V	Wireless networking of devices, RF WPAN (802.15.1), Bluetooth protocol stack, Frame format, Link Manager Layer, Bluetooth piconet and applications	12
	Total	**60**

Text Books:

1. HolgerKarl, Andreas Willig,” Protocols & Architectures for WSN”,John Wiley,2012
2. Mark Ciampa, Jorge Olenewa,”Wireless Communications, Cengage Learning,2009.
3. Frank Adelstein, SandeepK. S Gupta etal,” Fundamentals of Mobile & Pervasive Computing, TMcHill,2010.

Reference Books:

1. Jaganathan Sarangapani, Wireless AdHoc & SensorN/Ws-Protocols & Control, CRC2007.
2. Kaveh Pahlavan, Prasanth Krishna moorthy, “Principles of Wireless Networks’ PHI/Pearson Education, 2003
3. Natalia Olifer and Victor Olifer, ”Computer Networks principles. Technologies and protocols for network design”, Wiley, 2015

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	REAL TIME SYSTEMS	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Knowledge about Schedulability analysis.	K2
CO2	Ability to learn Real-time programming environments.	K3
CO3	Knowledge about real time communication and databases.	K2
CO4	Ability to develop real time systems.	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	H	M	L
CO2	M	M	H	H	M	M
CO3	M	L	H	M	H	M
CO4	H	M	H	H	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit I	Introduction to Real-Time Computing: Introduction to real-time computing concepts. Examples of real-time applications. Structure of a real-time system. Characterization of real-time systems and tasks. Hard and soft timing constraints. Design challenges and performance metrics. Prediction of execution time: source code analysis, micro-architecture level analysis, cache and pipeline issues. Programming languages for real-time systems.	12
Unit II	Real-Time Operating Systems: Real-time OS concepts. Threads and tasks. Structure of microkernel. Time services. Scheduling mechanisms. Communication and synchronization. Event notification and software interrupts.	12

Unit III	Task Assignment and Scheduling: Task allocation algorithms. Single-processor and multiprocessor task scheduling. Clock-driven and priority-based scheduling algorithms. Fault-tolerant scheduling.	12
Unit IV	Real-Time Communication: Network topologies and architecture issues. Communication protocols: contention-based, token-based, polled bus, deadline-based protocols. Fault-tolerant routing. RTP and RTCP.	12
Unit V	Real-Time Databases: Transaction priorities. Concurrency control issues. Disk scheduling algorithms. Two-phase approach to improve predictability.	12
	Total	**60**

Text Book

1. C.M. Krishna, Kang G. Shin – “ Real Time Systems”, International Edition, McGraw Hill Companies, Inc., New York, 1997

Reference Books

1. Jane W.S. Liu, Real-Time Systems, Pearson Education India, 2000.
2. Philip A. Laplante and Seppo J. Ovaska, “Real-Time Systems Design and Analysis: Tools for the Practitioner” IV Edition IEEE Press, Wiley. 2011

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	EMBEDDED SYSTEM INTEGRATION	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to (Four to Six)

		Knowledge Level (K)#
CO1	Understand the principles of embedded computing, microprocessor-based systems, and formal design methodologies for complex embedded applications.	K2
CO2	Analyze processor architecture, instruction set design, and hardware abstraction techniques for efficient processor development	K4
CO3	Explore the ARM architecture, programmer's model, and development tools relevant for embedded system design	K4
CO4	Examine the support mechanisms in ARM systems including memory interfaces, AMBA protocols, debugging tools, and signal processing features.	K3
CO5	Evaluate memory hierarchy strategies and operating system support for embedded ARM-based applications	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	M	M	H	M	M	M
CO3	H	M	H	H	M	M
CO4	H	M	H	H	H	H
CO5	H	M	H	M	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus	Contact Hours
Unit I	**Embedded Computing and System Design**: Introduction to Embedded Computing, Complex Systems and Microprocessors, Embedded System Design Process, Formalisms for System Design, and Design Examples.	12
Unit II	**Fundamentals of Processor Design**: Introduction to Processor Design, Architecture and Organization, Abstraction in Hardware Design, Instruction Set Design, Processor Design Trade-offs, and RISC Architecture.	12

Unit III	**Overview of ARM Architecture**: Overview of ARM Architecture, Architecture Inheritance, ARM Programmer's Model, Development Tools for ARM, ARM Instruction Set, and Architectural Support for High-Level Languages.	12
Unit IV	**ARM System Development Support**: ARM Memory Interface, AMBA Bus Protocol, ARM Reference Peripheral Specifications, JTAG and Embedded Trace Support, Signal Processing Support, and ARM Processor Core Families.	12
Unit V	**Memory Hierarchy and OS Support**: Deals with Memory Size and Speed Considerations, On-Chip Memory and Caches, Memory Management Techniques, Memory Hierarchy Design, Architectural Support for Operating Systems, and Embedded ARM Applications.	12
	Total	**60**

Text Books:

1.DavidE.Simon,―An Embedded Software Primer, Pearson Education Asia, 2005

2.WayneWolf ―Computers as Components: Principles of Embedded Computing System

Reference Books :

1.Design‖, 3rdEditions, Morgan Kaufman Publishers, 2012.

2. Rajkamal, Embedded Systems Architecture, Programming and Design ,3rd Edition, TATA McGraw Hill, 2008.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	EMBEDDED NETWORKS AND PROTOCOLS	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Acquire knowledge on communication protocols of connecting Embedded Systems	K3
CO2	Master the design level parameters of USB and CAN bus protocols.	K2
CO3	Design Ethernet in Embedded networks considering different issues.	K5
CO4	Acquire the knowledge of wireless protocols in Embedded domain.	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	M	L	H	L	M	M
CO3	H	M	H	M	M	M
CO4	H	M	H	M	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	**Syllabus**	**Contact Hours**
Unit I	Embedded Networking: Introduction – Serial/Parallel Communication – Serial communication protocols - RS232 standard – RS485 – Synchronous Serial Protocols - Serial Peripheral Interface (SPI) – Inter Integrated Circuits (I2C) – PC Parallel port programming - ISA/PCI Bus protocols – Firewire.	12
Unit II	USB bus – Introduction – Speed Identification on the bus – USB States – USB bus communication Packets – Data flow types – Enumeration – Descriptors – PIC18 Microcontroller USB Interface – C Programs – CAN Bus – Introduction - Frames – Bit stuffing – Types of errors – Nominal Bit Timing – PIC microcontroller CAN Interface – A simple application with CAN.	12
Unit III	Elements of a network – Inside Ethernet – Building a Network: Hardware options – Cables, Connections and network speed – Design choices: Selecting components – Ethernet Controllers – Using the internet in local and internet communications – Inside the Internet protocol.	12

Unit IV	Exchanging messages using UDP and TCP – Serving web pages with Dynamic Data – Serving web pages that respond to user Input – Email for Embedded Systems – Using FTP – Keeping Devices and Network secure.	12
Unit V	Wireless sensor networks – Introduction – Applications – Network Topology – Localization – Time Synchronization – Energy efficient MAC protocols – SMAC – Energy efficient and robust routing – Data Centric routing.	12
	Total	**60**

TEXT BOOKS

1. Embedded Systems Design: A Unified Hardware/Software Introduction-Frank Vahid, Tony Givargis, John & Wiley Publications, 2002
2. Parallel Port Complete: Programming, interfacing and using the PCs parallel printer port-Jan Axelson, Penram Publications, 1996.

REFERENCE BOOKS

1. Advanced PIC microcontroller projects in C: from USB to RTOS with the PIC18F series –DoganI brahim, Elsevier 2008.
2. Embedded Ethernet and Internet Complete-Jan Axelson, Penram publications,2003.
3. Networking Wireless Sensors-Bhaskar Krishnama chari ,Cambridge press2005.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	RISC PROCESSOR ARCHITECTURE AND PROGRAMMING	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO 1	Describe the programmer's model of ARM processor and create and test assembly level programming.	K4
CO 2	Analyz evarious types of coprocessors and design suitable co-processor interface to ARM processor	K4
CO 3	IdentifythearchitecturalsupportofARMforoperatingsystemandanalyzethefunctionof memory Management unit of ARM	K3
CO 4	Students will develop more understanding on the concepts ARM Architecture, programming and application development	K2
CO 5	The learning process delivers insight into various embedded processors of RISC architecture /computational processors with improved design strategies	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	M	L	H	L	M	M
CO3	H	M	H	M	M	M
CO4	H	M	H	M	H	H
CO5	H	M	H	M	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
UNIT I	AVR MICROCONTROLLER ARCHITECTURE Architecture – memory organization – addressing modes – I/O Memory – EEPROM – I/O Ports – SRAM – Timer – UART – Interrupt Structure - Serial Communication with PC – ADC/DAC Interfacing	12
UNIT II	ARM ARCHITECTURE AND PROGRAMMING 'Arcon RISC Machine – Architectural Inheritance – Core & Architectures -- The ARM Programmer's model - Registers – Pipeline - Interrupts – ARM organization - ARM processor family – Co-processors. Instruction set – Thumb instruction set – Instruction cycle timings	12
UNIT	ARM APPLICATION DEVELOPMENT	12

III	Introduction to RT implementation with ARM – Exception Handling – Interrupts – Interrupt handling schemes - Firmware and bootloader – Free RTOS Embedded Operating Systems concepts – example on ARM core like ARM9 processor	
UNIT IV	MEMORY PROTECTION AND MANAGEMENT Protected Regions - Initializing MPU, Cache and Write Buffer - MPU to MMU - Virtual Memory - Page Tables - TLB - Domain and Memory Access Permission - Fast Context Switch Extension	12
UNIT V	DESIGN WITH ARM MICROCONTROLLERS Assembler Rules and Directives - Simple ASM/C programs - Hamming Code - Division - Negation - Simple Loops – Look up table - Block copy - subroutines - application	12
	Total	**60**

TEXT BOOKS:

1. Steve Furber,‘ARM system on chip architecture’,Addision Wesley
2. Andrew N.Sloss, Dominic Symes, Chris Wright, John Rayfield‘ ARM System
3. Developer’s Guide Designing and Optimizing System Software’, Elsevier2007

REFERENCE BOOKS:

1. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi‘ AVR Microcontroller and Embedded Systems using Assembly and C”, Pearson Education 2014.
2. ARM Architecture Reference Manual, LPC213xUser Manual
3. An Engineer’s Introduction To The LPC2100Series’ Hitex(UK) Ltd
4. www.Nuvoton.com / web sites on Advanced ARM Cortex Processors

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	EMBEDDED SYSTEMS LAB	L	T	P	C
		0	1	2	2

CO1	Demonstrate the ability to write and execute basic Embedded C programs on microcontroller platforms.	K4
CO2	Apply digital I/O interfacing techniques by programming ports to control and monitor external hardware.	K3
CO3	Implement timing-based operations using software and hardware delays, including loops and timers.	K4
CO4	Design embedded applications for real-time control scenarios such as traffic lights and alarms.	K5
CO5	Interface serial communication peripherals and measure real-time data over communication links	K4
CO6	Develop embedded software solutions for domain-specific applications such as industrial automation.	K5
CO7	Demonstrate the use of port headers and external devices (like LCDs and keypads) in an embedded system	K4

PART-A
List of Experiments by using Embedded C

1. Write a simple program to print " Hello World"
2. Write a simple program to show a delay
3. Write a loop application to copy values from P1 to P2.
4. Write a C program for counting the no of times that a switch is pressed & released.
5. Write a simple program to create a portable hardware delay.
6. Write a C program to test loop time outs.
7. Write a C program to test hardware based timeouts loops.
8. Illustrate the use of port header file (PORT M) using an interface consisting of a keyword and Liquid crystal display.
9. Develop a simple EOS showing traffic light sequencing.
10. Write a program to display elapsed time over RS-232 Link.
11. Write a program to drive SEOS Using Timer 0.
12. Develop software for milk pasteurization system.
13. Develop & implement a program for intruder alarm system

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

I Semester	SYSTEM DESIGN WITH EMBEDDED LINUX LAB	L	T	P	C
		0	1	2	2

CO1	Demonstrate the ability to interface sensors and actuators with microcontroller boards	K4
CO2	Develop applications using Raspberry Pi for real-time control of output devices and sensor monitoring	K3
CO3	Design embedded systems using BeagleBone board for basic input/output operations and display interfacing	K4
CO4	Interface input devices and sensors with Embedded Linux boards and develop basic human-machine interaction applications	K5
CO5	Integrate sensors, actuators, and communication interfaces to build real-time embedded applications	K4
CO6	Demonstrate debugging and testing skills for verifying sensor data, controlling actuators, and troubleshooting embedded systems	K5

Part-I: (Using Ardiuno Board)

1. Temperature and Humidity sensor
2. Soil moisture
3. Ultra sonic sound sensor to measure distance
4. IR Sensor

Part-II: (Using Raspberry PI)

1. Servo motor
2. MQ2 Gas sensor
3. LCD
4. Relay

Part-III: (Using beagle bone boards)

1. Led blinking
2. Seven segment display
3. LCD
4. Switch(buzzer)

Part-IV: (Using embedded Linux Board)

1. 4×4Matrix
2. Light dependent resistor

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	SYSTEM DESIGN USING EMBEDDED PROCESSORS	L	T	P	C
		3	1	0	4

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the fundamental concepts, architecture, and application areas of embedded systems along with development tools	K2
CO2	Explain the ARM Cortex-M3 architecture, its instruction sets, and internal registers relevant to embedded system programming	K2
CO3	Analyze exception handling mechanisms, Nested Vectored Interrupt Controller (NVIC), and interrupt behavior in Cortex-M3	K4
CO4	Develop embedded programs using C and assembly language with CMSIS support, including interrupt and memory protection handling	K4
CO5	Apply knowledge of STM32L15xxx microcontroller architecture and peripherals in designing, debugging, and implementing embedded system applications	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	H	M	M
CO2	M	L	H	H	M	M
CO3	H	M	H	H	M	L
CO4	H	M	H	H	M	M
CO5	H	M	H	H	M	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit 1	**Embedded Concepts:** Introduction to embedded systems, Application Areas, Categories of embedded systems, Overview of embedded system architecture, Specialties of embedded systems, Recent trends in embedded systems, Architecture of embedded systems, Hardware architecture, Software architecture, Application Software, Communication Software, Development and debugging Tools. **ARM Architecture:** Background of ARM Architecture, Architecture Versions, Processor Naming, Instruction Set Development, Thumb-2 and Instruction Set Architecture.	12
Unit	**Overview of Cortex-M3:** Cortex-M3 Basics: Registers, General Purpose	12

2	Registers, Stack Pointer, Link Register, Program Counter, Special Registers, Operation Mode, Exceptions and Interrupts, Vector Tables, Stack Memory Operations, Reset Sequence. **Instruction Sets:** Assembly Basics, Instruction List, Instruction Descriptions. **Cortex-M3 Implementation Overview:** Pipeline, Block Diagram, Bus Interfaces on Cortex-M3, I-Code Bus, D-Code Bus, System Bus, External PPB and DAP Bus.	
Unit 3	**Exceptions:** Exception Types, Priority, Vector Tables, Interrupt Inputs and Pending Behavior, Fault Exceptions, Supervisor Call and Pendable Service Call. **NVIC:** Nested Vectored Interrupt Controller Overview, Basic Interrupt Configuration, Software Interrupts and SYSTICK Timer. **Interrupt Behavior:** Interrupt/Exception Sequences, Exception Exits, Nested Interrupts, Tail-Chaining Interrupts, Late Arrivals and Interrupt Latency.	12
Unit 4	**Cortex-M3/M4 Programming:** Overview, Typical Development Flow, Using C, CMSIS (Cortex Microcontroller Software Interface Standard), Using Assembly. **Exception Programming:** Using Interrupts, Exception/Interrupt Handlers, Software Interrupts, Vector Table Relocation. Memory Protection Unit and Other Cortex-M3 Features: MPU Registers, Setting Up the MPU, Power Management, Multiprocessor Communication.	12
Unit 5	**Cortex-M3/M4 Microcontroller:** STM32L15xxx ARM Cortex M3/M4 Microcontroller: Memory and Bus Architecture, Power Control, Reset and Clock Control,STM32L15xxx Peripherals: GPIOs, System Configuration Controller, NVIC, ADC, Comparators, GP Timers, USART. **Development and Debugging Tools:** Software and Hardware tools like Cross Assembler, Compiler, Debugger, Simulator, In-Circuit Emulator (ICE), Logic Analyzer etc.	12
	Total	**60**

TEXTBOOKS:

1. The Definitive Guide to the ARM Cortex-M3, Joseph Yiu, Second Edition, Elsevier Inc. 2010.
2. Embedded/Real Time Systems Concepts, Design and Programming Black Book, Prasad, KVK.
3. David Seal “ARM Architecture Reference Manual”, 2001 Addison Wesley, England; Morgan Kaufmann Publishers

REFERENCES:

1. Steve Furber, “ARM System-on-Chip Architecture”, 2^{nd}Edition, Pearson Education
2. Cortex-M series-ARM Reference Manual
3. Cortex-M3 Technical Reference Manual (TRM)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	EMBEDDED REAL TIME OPERATING SYSTEMS	L	T	P	C
		3	1	0	4

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Illustrate real time programming concepts.	
CO2	Apply RTOS functions to implement embedded applications	
CO3	Understand fundamentals of design consideration for embedded applications	
CO4	Describe about the memory units and real time memory applications	
CO5	Discuss communication Common Design Problems	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	M	M	L	L
CO2	H	M	H	H	M	M
CO3	H	L	H	H	M	M
CO4	M	L	H	M	H	L
CO5	H	M	H	H	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
UNIT I	Introduction to Real-Time Operating Systems: Defining an RTOS, The scheduler, Kernel Objects and services, Key characteristics of an RTOS. Task: Defining a Task, Task States and Scheduling, Typical Task Operations, Typical Task Structure, Synchronization, Communication and Concurrency.	12
UNIT II	Semaphores: Defining Semaphores, Typical Semaphore Operations, Typical Semaphore Use. Message Queues: Defining Message Queues, Message Queue States, Message Queue Content, Message Queue Storage, Typical Message Queue Operations, Typical Message Queue Use. Pipes, Event Registers, Signals and Condition Variables.	12
UNIT III	Exceptions and Interrupts: Exceptions and Interrupts, Applications of Exceptions and Interrupts, Closer look at exceptions and interrupts, Processing General Exceptions, Nature of Spurious Interrupts. Timer and Timer Services:	12

	Real-Time Clocks and System Clocks, Programmable Interval Timers, Timer Interrupt Service Routines. I/O Subsystems: I/O Concepts, I/O Subsystems.	
UNIT IV	Memory Management: Dynamic Memory Allocation in Embedded Systems, Fixed-Size Memory Management in Embedded Systems, Blocking vs. Non-Blocking Memory Functions, Hardware Memory Management Units. Modularizing an Application for Concurrency: An Outside-In Approach to Decompose Applications, Guidelines and Recommendations for Identifying Concurrency, Schedulability Analysis.	12
UNIT V	Synchronization and Communication: Synchronization, Communication, Resource Synchronization Methods, Critical Section, Common Practical Design Patterns, Specific Solution Design Patterns. Common Design Problems: Resource Classification, Deadlocks, Priority Inversion.	12
	Total	**60**

Text Books

1. Qing Li, Caroline Yao (2003), "Real-Time Concepts for Embedded Systems", CMP Books.

Reference Books

1. Albert Cheng, (2002), "Real-Time Systems: Scheduling, Analysis and Verification", Wiley Interscience.
2. Hermann Kopetz, (1997), "Real-Time Systems: Design Principles for Distributed Embedded Applications", Kluwer.
3. Insup Lee, Joseph Leung, and Sang Son, (2008) "Handbook of Real-Time Systems", Chapman and Hall.Krishna and Kang G Shin, (2001), "Real-Time Systems", McGraw Hill.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	**INTERNET OF THINGS**	**L**	**T**	**P**	**C**
		3	**1**	**0**	**4**

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Analyze and compare various IoT hardware platforms and networking components including Linux-based configurations	K4
CO2	Understand the fundamentals of networking, OSI model, and data communication concepts essential for IoT systems	K2
CO3	Explain IoT architecture, communication patterns, and protocol stacks such as 6LoWPAN with security considerations	K2
CO4	Develop IoT applications using web technologies, databases, and mobile development tools with attention to data privacy	K5
CO5	Evaluate advanced IoT use cases, sensor node integration, and the role of big data and Industry 4.0 in smart systems.	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	M	H	L
CO2	M	L	H	M	H	M
CO3	H	M	H	M	H	M
CO4	H	M	H	H	H	M
CO5	H	M	H	H	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	**Syllabus Content**	**Contact Hours**
Unit 1	The IoT Networking Core: Technologies involved in IoT Development: Internet/Web and Networking Basics, OSI Model, Data transfer referred with OSI Model, IP Addressing, Point to Point Data transfer, Point to Multi Point Data transfer & Network Topologies, Sub-netting, Network Topologies referred with Web, Introduction to Web Servers, Introduction to Cloud Computing.	12
Unit 2	IoT Platform Overview: Overview of IoT supported Hardware platforms such as Raspberry Pi, ARM Cortex Processors, Arduino and Intel Galileo boards. Network Fundamentals: Overview and working principle of Wired Networking equipment – Routers, Switches; Overview and working principle of Wireless Networking equipment – Access Points, Hubs etc.	12

	Linux Network Configuration Concepts: Networking configurations in Linux, Accessing Hardware & Device Files interactions.	
Unit 3	IoT Architecture: History of IoT, M2M – Machine to Machine, Web of Things, IoT protocols. Applications: Remote Monitoring & Sensing, Remote Controlling, Performance Analysis. The Architecture: The Layering concepts, IoT Communication Pattern, IoT Protocol Architecture, The 6LoWPAN. Security aspects in IoT.	12
Unit 4	IoT Application Development: Application Protocols. Back-end Application Designing: Apache for handling HTTP Requests, PHP & MySQL for data processing, MongoDB Object type Database, HTML, CSS & jQuery for UI Designing, JSON library for data processing, Security & Privacy during development. Application Development for Mobile Platforms: Overview of Android / iOS App Development tools.	12
Unit 5	Case Study & Advanced IoT Applications: IoT applications in home, infrastructures, buildings, security, industries, home appliances, and other IoT electronic equipment. Use of Big Data and Visualization in IoT, Industry 4.0 concepts. Sensors and Sensor Nodes and interfacing using any embedded target boards (Raspberry Pi / Intel Galileo / ARM Cortex / Arduino).	12
	Total	**60**

TEXT BOOKS:

1. 6LoWPAN: The Wireless Embedded Internet, Zach Shelby, Carsten Bormann, Wiley
2. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, Dr. Ovidiu Vermesan, Dr. Peter Friess, River Publishers
3. Interconnecting Smart Objects with IP: The Next Internet, Jean-Philippe Vasseur, Adam Dunkels, Morgan Kuffmann

REFERENCES:

1. The Internet of Things: From RFID to the Next-Generation Pervasive Network ed Lu Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning
2. Internet of Things (A Hands-on-Approach), Vijay Madisetti, Arshdeep Bahga
3. Designing the Internet of Things, Adrian Mc Ewen (Author), Hakim Cassimally
4. Asoke K Talukder and RoopaR Yavagal,“Mobile Computing,” Tata Mc Graw Hill, 2010.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	**AI FOR EMBEDDED SYSTEMS**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the basics of TinyML and the current landscape of its applications and challenges	K2
CO2	Utilize TensorFlow for deep learning and TensorFlow Lite (TFLite) for TinyML	K4
CO3	Apply TinyML for gesture tracking with the Magic Wand project, including data collection, labeling, and training a CNN model.	K3
CO4	Learn various sensors such as accelerometers, gyroscopes, barometers, and magnetometers, and how to interface with sensor data	K4
CO5	Deploy TinyML models on power and performance-constrained devices to solve real word problems.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	M	M	M
CO2	M	M	H	M	H	H
CO3	H	M	H	H	H	M
CO4	M	L	H	M	H	L
CO5	H	M	H	H	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	**Syllabus Content**	**Contact Hours**
Unit I	Introduction: Introduction to TinyML, TinyML Landscape, Applications, and Challenges, TinyML Lifecycle and Workflow, Model Compression Techniques. Fundamentals of ML and TinyML, Pruning ML models, Quantization Aware Training (QAT) and Post Training Quantization (PTQ), Knowledge Distillation, Tiny Deep Learning, TensorFlow Lite (TFLite) for TinyML.	12
Unit II	TinyML for Keyword Spotting: Background on Keyword Spotting and Streaming Audio, Challenges and Constraints in Keyword Spotting, Keyword Spotting Architecture and Data Collection, Model Training, Evaluation Metrics, and Deployment.	12
Unit	TinyML for Visual Wake Words: Introduction to Visual Wake Words and Its	12

III	Challenges, Visual Wake Words Dataset, Mobile Nets, Transfer Learning for Visual Wake Words, Model Training, Evaluation Metrics, and Deployment.	
Unit IV	TinyML for Anomaly Detection: Background on Anomaly Detection and Signal Processing, Real and Synthetic Datasets, Unsupervised Learning, Threshold Choice, Model Training, Evaluation Metrics, and Deployment. Magic Wand: Gesture Tracking through Bluetooth, CNN for Magic Wand Sketch, Data Collection and Labeling, Model Training, Evaluation Metrics, and Deployment.	12
Unit V	TinyML for Predictive Maintenance: Background on Predictive Maintenance Solutions and Industry Applications, Sensors, Sensor Data, and Interface, Accelerometer, Gyroscope, Barometer, and Magnetometer, TinyML Framework for Predictive Maintenance, Model Training, Evaluation Metrics, and Deployment. Smart Lock Audio Recognition using TinyML: Audio classification for deploying sensitive smart lock model, Data processing on audio data, Generate, train, and test a TensorFlow model using the SensiML Python SDK, Compile and flash the model to the edge device and display the inferred classes in the SensiML Open Gateway user interface.	12
	Total	**60**

TEXT BOOK :

1. Pete Warden and Daniel Situnayake,TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'REILLY Publisher ,2020,1st Edition.

REFERENCES : 1. Gian Marco odice,TinyML Cookbook

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	EMBEDDED SYSTEM DESIGN USING FPGA	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Explain the architecture of embedded systems and identify the role of FPGAs and SoCs in modern VLSI-based platforms	K2
CO2	Develop and simulate digital circuits using VHDL/Verilog and design high-quality modular systems based on control flow graphs and abstraction principles	K4
CO3	Demonstrate the ability to select and integrate system software, cross-development tools, boot-loaders, and monitors in FPGA-based embedded platforms	K3
CO4	Analyze partitioning strategies and communication mechanisms to optimize performance, resource usage, and system scalability	K4
CO5	Apply principles of spatial parallelism and identify contemporary design issues to build efficient, high-performance FPGA-based solutions	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	M	L	H
CO2	H	M	H	H	M	H
CO3	H	M	H	H	M	H
CO4	H	M	H	H	M	H
CO5	H	M	H	H	M	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit I	Introduction to Embedded Systems and FPGA Platforms: Embedded System Overview: H/W-FPGA-Embedded SoC, Use of VLSI circuit technology, Platform FPGAs – Altera Cyclone, FPGA Platform, Components of platform FPGA systems, Adding custom compute cores, Assembling platform-based systems.	12
Unit II	Hardware Description and System Design: Hardware Description Languages: VHDL, Verilog, Other High-Level HDLs, HDL to Configuration Bit-stream generation. System Design using FPGA: Principles of system design, Design quality,	12

	Modules and interfaces, Abstraction and state, Cohesion and coupling, Design reuse strategies, Control flow graph, Origins of platform FPGA designs.	
Unit III	Software Design for FPGA Systems: Software Design Considerations: System Software Options, Root File System, Cross-Development Tools for Embedded Applications. Monitors and Boot-loaders: Role in platform-based development, Integration techniques.	12
Unit IV	Partitioning and Communication: Partitioning Overview: Partitioning Problem, Basic definitions, Expected performance gain, Resource considerations in partitioning, Analytical Approach to Partitioning. Scheduling and Communication: Invocation and coordination mechanisms, Transfer of state, Practical Issues in Profiling, Data structure design, Feature size manipulation.	12
Unit V	Parallelism and Contemporary Issues: Spatial Design Concepts: Principles of parallelism, Identifying parallelism in applications. Spatial Parallelism with Platform FPGAs: Within FPGA hardware cores, Across FPGA designs. Contemporary Issues in Embedded FPGA System Design: Trends, challenges, and emerging technologies.	12
	Total	**60**

Text Book(s):

1. Ron Sass, Andrew G. Schmidt, Embedded Systems Design with Platform FPGAs: Principles and Practices, First Edition, Tata McGraw Hill, India, 2011.

Reference Books:

1. Charles H. Roth Jr., Digital Systems Design Using VHDL, Reprint Edition, PWS Publishing Company (Thomson Books), USA, 2012.
2. V. A. Padroni, Circuit Design with VHDL, First Edition, MIT Press, Cambridge, England, 2011.
3. Wayne Wolf, FPGA Based System Design, First Edition, Prentice Hall, Modern Semiconductor Design Series, USA, 2011.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

		L	T	P	C
II Semester	**ASIC & SOC DESIGN**	**3**	**0**	**0**	**3**

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the types, architecture, and design flow of ASICs and their associated cell libraries.	K2
CO2	Analyze the construction of ASICs using design tools, including synthesis, floor planning, placement, and routing techniques	K4
CO3	Explain the System-on-Chip (SoC) design process, various design methodologies, specification techniques, and system-level challenges	K2
CO4	Explore the design principles, integration techniques, and performance issues involved in Multiprocessor System-on-Chip (MPSoC) architectures	K4
CO5	Apply verification strategies and methodologies for SoC, including system-level and hardware/software co-verification	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	H	M	H	M	M	L
CO3	H	M	H	H	M	L
CO4	H	M	H	M	M	H
CO5	H	M	H	H	M	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	**Syllabus Content**	**Contact Hours**
Unit I	Introduction to ASICs: Types of ASICs, Design Flow, Economics of ASICs, ASIC Cell Libraries, CMOS Logic Cell, Data Path Logic Cells, I/O Cells, and Cell Compilers.	12
Unit II	ASIC Library and Design Software: ASIC Library Design: Transistors as Resistors, Parasitic Capacitance, Logical Effort. Programmable ASIC Design Software: Design System, Logic Synthesis, Half Gate ASIC. ASIC Construction: Floor Planning and Placement, Routing.	12
Unit III	System-on-Chip Design Process: A Canonical SoC Design, SoC Design Flow – Waterfall vs. Spiral, Top-	12

	Down versus Bottom-Up Approaches. Specification Requirements, Types of Specifications, System Design Process, System Level Design Issues – Soft IP vs. Hard IP. Design for Timing Closure: Logic Design Issues, Physical Design Issues. Verification Strategy, On-Chip Buses and Interfaces, Low Power Strategies, Manufacturing Test Strategies.	
Unit IV	Multiprocessor SoCs (MPSoCs): Introduction to MPSoCs. Techniques for Designing MPSoCs, including Architecture, Integration, and Performance Optimization Techniques.	12
Unit V	SoC Verification: Verification Technology Options, Verification Methodology, Verification Languages and Approaches, Verification Plans. System Level Verification, Block Level Verification, Hardware/Software Co-Verification, Static Netlist Verification.	12
	Total	**60**

TEXT BOOKS:

1. "SoC Verification-Methodology and Techniques", Prakash Rashinkar, Peter Paterson and Leena Singh. Kluwer Academic Publishers, 2001.
2. "ReuseMethodologymanualforSystem-On-A-ChipDesigns",MichaelKeating, Pierre Bricaud, Kluwer Academic Publishers, second edition, 2001
3. Smith,"ApplicationSpecificIntegratedCircuits",Addison-Wesley,2006

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	ROBOTICS AND MACHINE VISION	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the basic concepts, structure, kinematics, and programming of industrial robotic systems	K2
CO2	Explain the principles of machine vision and differentiate between human vision and machine vision in industrial automation contexts	K2
CO3	Apply image acquisition and processing techniques, including various one-point and two-point transformations for machine vision tasks	K3
CO4	Analyze and implement edge enhancement and image analysis techniques using digital filters and feature detection operators	K4
CO5	Integrate robotics and machine vision systems for real-time industrial applications and evaluate system performance and challenges	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	M	L	H	L	M	L
CO3	H	M	H	M	M	L
CO4	H	M	H	M	M	M
CO5	H	M	H	H	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit I	Industrial Robots: Basic concepts of robotics, classification and structure of robotic systems. Kinematics analysis and coordinate transformations. Industrial applications of robots and introduction to robot programming.	12
Unit II	Introduction to Machine Vision: Principles of machine vision, vision in factory automation, comparison between human vision and machine vision. Economic considerations of implementing machine vision. Overview of machine vision systems. Image acquisition techniques including illumination, image formation, and focusing. Image detection fundamentals and types of cameras. Basics of image processing and presentation.	12

Unit III	Image Processing Techniques: Fundamental concepts of image processing including pixel, pixel location, gray scale, quantizing error, measurement error, and histograms. Basic machine vision processing operators: Monadic (one-point) transformations: identity operator, inverse operator, threshold operator, inverted threshold, binary threshold, inverted binary threshold, gray scale threshold, and inverted gray scale threshold. Dyadic (two-point) transformations: image addition, image subtraction, image multiplication. Convolution and spatial transformations.	12
Unit IV	Edge Enhancement and Image Analysis: Digital filtering techniques including low pass and high pass filters. Edge enhancement operators: Laplacian, Roberts gradient, Sobel, and other local operators. Image analysis techniques such as thresholding, pattern matching, edge detection, and the back-propagation algorithm.	12
Unit V	Integration and Applications: Integration of robotics and machine vision systems for industrial automation. Case studies of robotic vision applications. Design considerations in real-time implementation and challenges in deployment. Future trends in intelligent vision-guided robotic systems.	12
	Total	**60**

TEXT BOOKS:

1. Machine Vision and Digital Image Processing, by LouisJ. Galbiati, Jr.Prentice Hall, Englewood Cliffs, New Jersy.`
2. Robotics for Engineers, By, Yoram Koren, McGrawHill.
3. Robotics and Image Processing– an Introduction, by Janakiraman P.A., Tata McGraw Hill, New Delhi

REFERENCES:

1. Industrial Robotics–Technology, Programming and Applications, by Mikell P. Groover, Mitchell Wein, Roger N. Nageland Nicholas G.Odlrey, McGrawHill International Edistion.
2. Handbook OfI mage Processing Operators by Klette, Reinhard &Zamperoni, Piero; John Wiley & Sons Inc

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	NETWORK EMBEDDED APPLICATIONS	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand and differentiate various embedded communication protocols such as RS232, RS485, SPI, USB, and CAN used in embedded systems	K2
CO2	Analyze USB and CAN bus architectures, communication mechanisms, and interface methods using PIC microcontrollers	K4
CO3	Explain Ethernet fundamentals, hardware components, and the role of Internet Protocol (IP) in embedded networking	K2
CO4	Describe wireless embedded networking concepts including sensor networks, topologies, localization, and synchronization techniques	K3
CO5	Evaluate energy-efficient MAC and routing protocols like SMAC and data-centric routing strategies for robust wireless communication	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	H	L	H	M	M	M
CO3	H	M	H	M	M	L
CO4	H	M	H	M	H	M
CO5	H	M	H	M	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit I	Embedded Communication Protocols: Embedded Networking: Introduction, Serial/Parallel Communication, Serial Communication Protocols: RS232 standard, RS485, Synchronous Serial Protocols, Serial Peripheral Interface (SPI), USB, and CAN.	12
Unit II	Bus: USB Bus – Introduction, Speed Identification on the bus, USB States. USB Bus Communication: Packets, Data Flow Types, Enumeration, Descriptors. PIC 18 Microcontroller USB Interface Bus, Introduction, Frames, Bit Stuffing, Types of Errors, Nominal Bit Timing. PIC Microcontroller CAN Interface, A Simple Application with CAN.	12

Unit III	Ethernet Basics: Elements of a Network, Inside Ethernet. Building a Network: Hardware Options, Cables, Connections, and Network Speed. Design Choices: Selecting Components – Ethernet Controllers. Using the Internet in Local and Internet Communications – Inside the Internet Protocol.	12
Unit IV	Wireless Embedded Networking: Wireless Sensor Networks – Introduction, Applications, Network Topology, Localization, Time Synchronization.	12
Unit V	Energy Efficient MAC Protocols: SMAC – Energy Efficient and Robust Routing – Data Centric Routing.	12
	Total	**60**

TEXT BOOKS:

1. Embedded Systems Design:A Unified Hardware/Software Introduction - Frank Vahid, Tony Givargis, John & Wiley Publications, 2002.

2. ParallelPort Complete: Programming, interfacing and using the PCs parallel printer port- Jan Axelson, Penram Publications, 1996.

REFERENCE BOOKS :

1. Advanced PIC microcontroller projects in C: from USB to RTOS with the PIC 18F series- Dogan Ibrahim, Elsevier 2008.
2. Embedded Ethernet and Internet Complete -Jan Axelson, Penram publications,2003.
3. Networking Wireless Sensors-Bhaskar Krishnama chari ,Cambridge press 2005.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	GPU ARCHITECTURES AND PROGRAMMING	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand working proficiency with CUDA, algorithmic GPU programming and parallel computing	K2
CO2	Comprehend with classic scientific computing algorithms and problems	K3
CO3	Optimize GPU code and debug GPU code	K4
CO4	Analyze architecture specific details like memory access coalescing, shared memory usage, GPU thread scheduling	K4
CO5	Apply deep learning algorithms on embedded GPUs	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	M	M	H
CO2	M	L	H	L	M	H
CO3	H	M	H	M	M	H
CO4	H	M	H	M	M	H
CO5	H	M	H	H	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	**Contact Hours**
Unit I	GPU architectures: Introduction to the ideas of parallelism and the GPU programming model CPU vs GPU Parallelizing algorithms on paper, First CUDA program	12
Unit II	CUDA programming: Hardware of Graphics Processing Units and parallel communication patterns, Brief on GPU architecture, Basics of CUDA C, Floating point precision and support on GPUs	12
Unit III	Parallel primitives and algorithms on GPU: The CUDA programming language will be mastered while learning how to implement these algorithms., Matrix Operations, Stencil – Image Blurring, Filters, Gauss Jacobi-Finite difference updates for PDEs, Histogram, binning 1, Reduce – Maximum and Minimum – Summation, Prefix-sum (Scan) Algorithm – Radix Sort, Generating Cummulative Distributions , Complex algorithms – N-body solutions	12
Unit	Optimizing GPU Applications: Coalesced Memory Transactions, Grid Blocks,	12

IV	Thread Blocks, domain decomposition, Asynchronous Kernels and Multi streaming Possible Items: Libraries on GPU, cuBLAS Thrust, cuFFT, cuRAND, Multi-node GPU processing, Multi-GPU per node processing, CUDA in other languages (Python/Fortran), Scaling.	
Unit V	Deep learning on GPUs: Deep learning on GPUs, Combining graphics and compute, Display the results of computations– Interactive systems, Collision detection with voxelized solid (Gargoyle), Ray tracing in CUDA kernels, or ray tracing cores, Microsoft DXR (DX12 API), Vulkan, NVIDIA OptiX / RTX, NVIDIA Turing: “World‘s First Ray Tracing GPU”- Quadro RTX, Geforce RTX	12
	Total	**60**

TEXT BOOKS:

1. GPUs for Graphics: OpenGL 4.0 Shading Language Cookbook, 2nd Edition.
2. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Publisher: Addison-Wesley Professional,2013, 3 rd Edition.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	EMBEDDED COMPUTING SYSTEM	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand the fundamentals of circuit design, deep submicron challenges, and apply low-power and algorithm-level optimization techniques in embedded DSP systems	K2
CO2	Analyze and compare various embedded processor architectures and core-based ASIC design methods with a focus on power-efficient architectural techniques	K4
CO3	Apply compiler optimization strategies and power modeling techniques to improve performance-power trade-offs in embedded systems.	K3
CO4	Design and evaluate DSP algorithms considering A/D conversion limitations, finite precision effects, and implement efficient source and channel processing techniques	K5
CO5	Gain insight into networking principles for embedded systems, including addressing, routing, and wireless vs. wire-line trade-offs, and explore distributed OS design through real-world case studies like JINI.	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	H	M	H	M	M	M
CO3	H	M	H	M	M	M
CO4	H	M	H	M	M	L
CO5	H	M	H	M	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
Unit-I	Circuits and DSP Architecture: Circuit Design basics, Deep submicron issues, low power techniques, High level power models, algorithm transformation techniques, Dedicated architectures for embedded systems	12
Unit-II	Architectures Design: Embedded processor architectures, architectural techniques for low power design methods for core-based ASICs	12
Unit-III	Compiler and OS: Introduction to compiler optimization, power models for compiler optimizations, core size vs. performance/power trade off	12
Unit-IV	DSP Algorithm Design: A/D conversion and finite precision analysis, algorithms for embedded systems, source and channel processing, potable	12

	embedded code	
Unit-V	Networking: Networking Basics (addressing and routing), wireless vs. wire-line networking, distributed OS for networked embedded systems, Case study of JINI	12
	Total	**60**

TEXT BOOKS:

1. K.Hwang,―Advance Computer Architecture: Parallelism, Scalability and Programmability ‖, New York McGraw Hill Inc., 1993.
2. S.Y.Kung,―VLSIArrayProcessor‖,PrenticeHall,EnglewoodCliffs,NJ,1988

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	CRYPTOGRAPHY AND NETWORK SECURITY	L	T	P	C
		3	0	0	3

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Identify and utilize different forms of crypto graphy techniques.	K3
CO2	In corporate authentication and security in the network applications.	K2
CO3	Distinguish among different types of threats to the system and handle the same	K4
CO4	Analyze Public-Key (Asymmetric) Cryptography and message digest algorithms	K2
CO5	Discuss about Authentication and System Security	K2

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	M	L	H	L	M	L
CO2	M	L	H	M	M	L
CO3	H	M	H	M	M	L
CO4	M	L	H	M	H	L
CO5	H	M	H	H	H	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Unit	Syllabus Content	Contact Hours
UNIT I	Security: Need, security services, Attacks, OSI Security Architecture, one time passwords, Model for Network security, Classical Encryption Techniques like substitution ciphers, Transposition ciphers, Cryptanalysis of Classical Encryption Techniques.	12
UNIT II	Number Theory: Introduction, Fermat's and Euler's Theorem, The Chinese Remainder Theorem, Euclidean Algorithm, Extended Euclidean Algorithm, and Modular Arithmetic.	12
UNIT III	Private-Key (Symmetric) Cryptography: Block Ciphers, Stream Ciphers, RC4 Stream cipher, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES, RC5, IDEA, Linear and Differential Cryptanalysis.	12
UNIT IV	Public-Key (Asymmetric) Cryptography: RSA, Key Distribution and Management, Diffie-Hellman Key Exchange, Elliptic Curve Cryptography, Message Authentication Code, hash functions, message digest algorithms: MD4, MD5, Secure Hash algorithm, RIPEMD-160, HMAC.	12

UNIT V	Authentication and System Security: IP and Web Security, Digital Signatures, Digital Signature Standards, Authentication Protocols, Kerberos, IP security Architecture, Encapsulating Security Payload, Key Management, Web Security Considerations, Secure Socket Layer, Secure Electronic Transaction, Intruders, Intrusion Detection, Password Management, Worms, viruses, Trojans, Virus Countermeasures, Firewalls, Trusted Systems.	12
	Total	**60**

TEXT BOOKS:

1. William Stallings, "Cryptography and Network Security, Principles and Practices", Pearson Education, 3rd Edition.
2. Charlie Kaufman, Radia Perlman and Mike Speciner, "Network Security, Private Communication in a Public World", Prentice Hall, 2ndEdition

REFERENCE BOOKS:

1. Christopher M.King, Ertem Osmanoglu, Curtis Dalton, "Security Architecture, Design Deployment and Operations", RSA Press,
2. Stephen Northcutt, Leny Zeltser, Scott Winters, Karen Kent, and Ronald W. Ritchey, "Inside Network Perimeter Security", Pearson Education, 2ndEdition
3. Richard Bejtlich,"ThePracticeofNetworkSecurityMonitoring:UnderstandingIncidentDetection and Response", William Pollock Publisher, 2013.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	SYSTEM DESIGN USING EMBEDDED PROCESSORS LAB	L	T	P	C
		0	1	2	2

		Knowledge Level (K)#
CO1	Configure and program General Purpose Input/Output (GPIO) ports to control and monitor external devices such as LEDs using Embedded-C on ARM Cortex M3/M4	K3
CO2	Develop time-based control programs using hardware timers (polling and interrupt methods) for real-time LED blinking and toggling operations	K4
CO3	Implement serial communication protocols (USART) to transmit and receive data between ARM Cortex microcontroller and PC using polling and interrupt-based methods	K5
CO4	Utilize the ARM CMSIS and Standard Peripheral Libraries to configure hardware resources such as Timers, ADC, and USART for real-time applications	K3
CO5	Design and develop real-time embedded system applications for data acquisition and control using ARM Cortex M3/M4 peripherals	K5

- **Embedded-C Programming on ARM Cortex M3/M4 Microcontroller (any three experiments)**

1. Write a program to turn on green LED (PortB.6) and Blue LED(PortB.7)on STM32L-DiscoverybyconfiguringGPIO.
2. Write a program to toggle green LED (PortB.6) and Blue LED (PortB.7) on STM32L-Discovery by configuring GPIO and using software delays.
3. Write a program to toggle Blue LED (PortB.6) at a rate of 1sec.UseTimer3 in Polling method for delay generation.
4. Transmit a string "Programming with ARM Cortex" to PC by configuring the Registers of USART2. Use polling method.
5. Transmit a string "Programming with ARM Cortex" to PC by configuring the
    a. Registers of USART3.Use polling method.

- **ARMCortexM3/M4ProgrammingwithCMSIS (any five experiments)**

1. Write a program to toggle the LEDs at the rate of 1sec using standard peripheral library.UseTimer3forDelay.

2. Transmit a string “Programming with ARM Cortex” to PC by using standard peripherallibrarywiththehelpofUSART3.Usepollingmethod.

3. Receive the data send by PC, compare it with threshold and switch on the Green LED if below threshold and Red LED if above.

4. Write a program to read the analog input connected to ADC and compare with Threshold so as to control the Digital outputs (LEDs).Use standard peripheral Library and interrupt method.
5. Write a program to toggle Blue LED(PortB.6)at a rate of 1sec using Timer2 in Interrupt configuration.
6. Write a program to toggle Blue LED (PortB.6) at a rate of 1sec using Timer3 in Interrupt configuration.
7. Transmit a data to PC by using standard peripheral library with USART1. Use Interrupts method.
8. Receive a data sent by PC by using standard peripheral library with USART1. Use Interrupts method.

ARMCortexM3/M4Peripherals (any two experiments)

Design of a real-time data acquisition & control system using theSTM32Lxx ARM Cortex M3 Microcontroller

It is required to monitor and control the temperature in a boiler which ranges from 0C to 100C every **1second** using the STM32Lxx ARM Cortex M3 Microcontroller. The temperature has to be kept at a set-point of50C 2C. The temperatureismeasuredthroughanRTDsensorandistransmittedthrougha4-20 mA two wire transmitter. The 4-20mA is converted to 1 to 5V by 250 ohm terminatingresistor.1to5Visavailableattheanaloginputport.1Vcorresponds to 0□C and 5V corresponds to 100□C. An ON/OFF relay connected to A PIO Port bit is used to control the heater element. A PC is used as the monitoring and control station.

Read the data through ADC and send the data from 0V to 5V in steps of 0.1V.The same has to be repeated after reaching the maximum value of 5V.

1. The temperature has to be sent to the PC every 1 second in the following protocol format and the same has to be displayed using the LAS software in WISE-96 on the PC.

STX	**MSL**	**CMD**	**SCMD**	**DATA_LO**	**DATA_HI**	**ETX**
byte1	byte2	byte3	byte4	byte5	byte6	byte7

STX	:	Start of Text	02H
MSL	:	Message length, in bytes	
CMD	:	Command byte	90H
SCMD	:	Sub-command byte	00H(Channel no)
DATA_LO	:	Lower byte of data word	
DATA_HI	:	Upper byte of data word	
ETX	:	End of Text	03H

2. Provision should be given for receiving the set-point value of temperature from the PC, and the set point is to be framed in the above protocol format.
3. If the transmitter is switched off or if it sends invalid data, i.e, below 4mA, an error message packet similar to the above one with CMD byte set to 95H should be send to the PC, instead of the data packet.

REFERENCES:

1. Embedded/ Real Time Systems Concepts, Design and Programming Black Book, Prasad, KVK.
2. The Definitive Guide to the ARM Cortex-M3, Joseph Yiu, Second Edition, Elsevier Inc. 2010.
3. David Seal “ARM Architecture Reference Manual”, 2001Addison Wesley, England; Morgan Kaufmann Publishers
4. Andrew N Sloss, Dominic Symes, Chris Wright, “ARM System Developer's Guide- Designing and Optimizing System Software”, 2006, Elsevier.
5. Steve Furber,“ARMSystem-on-ChipArchitecture”,2^{nd} Edition, Pearson Education.
6. Cortex-M series-ARM Reference Manual
7. Cortex-M3Technical Reference Manual (TRM)
8. ARM Company Ltd. “ARM Architecture Reference Manual–ARMDDI0100E”
9. STM32L152xxARMCortexM3MicrocontrollerReferenceManual
10. ARMv7-M Architecture Reference Manual (ARMv7-MARM).

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

II Semester	EMBEDDED REAL TIME OPERATING SYSTEMS LAB	L	T	P	C
		0	1	2	2

Course Outcomes: At the end of the course, student will be able to

		Knowledge Level (K)#
CO1	Understand and implement task management operations in RTOS environments including task creation, scheduling, and synchronization	K2
CO2	Develop interrupt-driven applications using RTOS features for ARM-based embedded systems.	K4
CO3	Apply RTOS synchronization mechanisms such as semaphores, mutexes, monitors, and resource allocation algorithms to manage concurrent tasks	K3
CO4	Analyze and solve classic concurrency problems (e.g., Deadlock, Reader-Writer) in a multitasking RTOS environment	K4
CO5	Interface hardware peripherals (e.g., Display, ADC, DAC) with ARM Cortex microcontrollers using open-source RTOS frameworks.	K4
CO6	Design and simulate real-time applications like data logging and serial communication using RTOS features on ARM-based development boards	K5

<u>List of Experiments</u>:

Part-I: Experiments using ARM-926 with PERFECT RTOS

1. Register a new command in CLI.
2. Create a new Task.
3. Interrupt handling.
4. Allocate resource using semaphores.
5. Share resource using MUTEX.
6. Avoid deadlock using BANKER’S algorithm.
7. Synchronize two identical threads using MONITOR.
8. Reader’s Writer’s Problem for concurrent Tasks.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
R25 M.TECH EMBEDDED SYSTEMS COURSE STRUCTURE AND SYLLABUS

Part-II : Experiments on ARM-CORTEX processor using any open source RTOS.

(Coo-Cox-Software-Platform)

1. Implement the interfacing of display with the ARM- CORTEX processor.
2. Interface ADC and DAC ports with the Input and Output sensitive devices.
3. Simulate the temperature DATA Logger with the SERIAL communication with PC.
4. Implement the developer board as a modem for data communication using serial port communication between two PC's.

Lab Requirements:
Software:

- Eclipse IDE for C and C++ (YAGARTO Eclipse IDE), Perfect RTOS Library, COO-COX Software Platform, YAGARTO TOOLS, and TFTP SERVER.
- LINUX Environment for the compilation using Eclipse IDE & Java with latest version.

Hardware:

- The development kits of ARM-926 Developer Kits and ARM-Cortex Boards.
- Serial Cables, Network Cables and recommended power supply for the board.