GUJARAT TECHNOLOGICAL UNIVERSITY

INSTRUMENTATION AND CONTROL (APPLIED INSTRUMENTATION) (03)
EMBEDDED SYSTEM FOR INSTRUMENTATION
SUBJECT CODE: 2710311
SEMESTER: I

Type of course: Core I

Prerequisite: Embedded Systems for Instrumentation

Rationale: This course provides an overview and fundamentals of embedded systems for real time applications which includes interfacing, programming and debugging. Also covers implementation of DSP applications on embedded platform.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks		Practical Marks				
				ESE (E)	PA (M)	PA (V)		PA (I)		
						ESE	OEP	PA	RP	
3	2#	2	5	70	30	20	10	10	10	150

Content:

Sr. No.	Topics	Teaching Hrs.	Module Weightage
1	**Introduction to ARM Cortex M Processors** Family of Cortex M processors, Advantages and applications, Resources available, Harvard and Von Neumann Architecture, General information, Features.	02	2%
2	**Introduction to Embedded Software Development** Development suites, Development boards, Debug adaptor, Software device driver, Software development flow, Compiling applications, Software flow, Data types in C programming, Inputs, outputs, and peripherals accesses, Microcontroller interfaces, Cortex microcontroller software interface standard (CMSIS).	02	3%
3	**Architecture of Cortex-M4 processor** Introduction, Programmer's model, Application program status register (APSR), Memory system, Exceptions and interrupts, System control block (SCB), Debug, Reset and reset sequence, Introduction to STM32F4xx architecture, Bus matrix, AHB & APB buses.	05	5%
4	**Instruction Set and ALP of Cortex-M4 processor** Moving data, Memory access, Arithmetic, Logic, Shift, Rotate, Data conversion, Bit-field processing, Compare and test, Program flow control, Saturation operations, Exception and sleep mode, Memory barrier, Miscellaneous and Cortex-M4 specific instructions, Barrel shifter, special instructions and special registers, Assembly language programming (ALP).	04	20%
5	**Memory System** Features, Memory map, Memory and peripherals interfacing, Memory requirements, Memory endianness, Data alignment and unaligned data	04	7%

	access support, Bit-band operations, Default memory access permissions, Memory access attributes, Exclusive accesses, Memory barriers, Memory system in a microcontroller.		
6	**Exceptions and Interrupts** Overview, Exception types, Overview of interrupt management, Definitions of priority, Vector table and vector table relocation, Interrupt inputs and pending behaviors, Exception sequence overview, NVIC registers, SCB registers, Special registers for exception or interrupt masking, Example procedures, Software interrupts, Exception handlers in C, Exception sequences, Interrupt latency and exception handling optimization.	03	10%
7	**Floating Point Operations and DSP Applications** Single-precision, half-precision, and double-precision floating point numbers, Floating point unit (FPU), Lazy stacking, Floating point programming in C, Hard-vfp and Soft-vfp, Special FPU modes, Floating point exceptions, DSP on a microcontroller, Dot product, Cortex-M4 DSP instructions, Writing optimized DSP code. Writing optimized DSP code	05	20%
8	**Embedded 'C' programming for STM32F4xx** Bit manipulations, addressing mechanism for memory mapped registers, Functions, Arrays, Pointers, structures and unions.	05	08%
9	**STM32F4xx Peripherals and programming** GPIO, Matrix keyboard, Seven segment display, Timers, PWM, LCD, ADC-DAC, UART, Ethernet, USB, Interrupt Mechanism (NVIC).	12	25%

Reference Books:

1. "The Definitive Guide to ARM® CORTEX®-M3 and CORTEX®-M4 Processors (Third Edition)", By *Joseph Yiu*, Newnes, Elsevier
2. "The Definitive Guide to the ARM® CORTEX®-M3 (Second Edition)", By *Joseph Yiu*, Newnes, Elsevier
3. "The insider's guide to the STM32 ARM based Microcontroller", www.hitex.com
4. Datasheet, programming and user reference manual of STM32F4xx (www.st.com)
5. "ARM System Developer's Guide: Designing and Optimizing System Software", By
    i. Andrew N. Sloss, Dominic Symes and, Chris Wright, 2004, Elsevier
6. "The Designer's Guide to the Cortex-M Processor Family: A Tutorial Approach", By Trevor Martin, Newnes, Elsevier

Course Outcome:

After learning the course the students should be able to

1. Understand how microprocessor, memory, peripheral components and, buses interact in an embedded system
2. Interface to on-chip and external peripherals
3. Program an embedded system in assembly and C
4. Design, implement and test a single-processor embedded systems for real-time applications
5. Optimizing embedded software for speed and size for industrial applications

List of Experiments:

1. Introduction to Integrated Development Environment (IDE)
2. Assembly language programs on manipulations of arrays
3. Programming of GPIO port with LED toggling and key interface
4. Programming of Base timer for accurate delays
5. Programming of Advanced Control Timers for edge aligned and center aligned PWM waveform

generation with programmable dead time
6. Programming of UART
7. Programming of Advanced control timer in capture mode (Frequency measurement)
8. Programming of on-chip ADC
9. Programming of on-chip DAC for wave form generation
10. Programming of LCD for embedded system output message display
11. Introduction to MATH & DSP library of STM32F4xx
12. Study and implementation of PID controller
13. Study and implementation of FIR and IIR filters
14. Study and implementation of 4-20 mA current loop
15. Study and implementation of Temperature measurement and control system
16. Introduction to auto code generation for STM32F4xx target using MATLAB Toolbox and Simulink
17. Study and programming the Cortex microcontrollers using NI LabVIEW for industrial application

Open Ended Problem: Solution of the open ended problem(s) in guidance of course instructor is mandatory. Few of the problems are specified as under.

1. Design and implementation of embedded system for industrial applications (e.g. instrumentation, control, automation but not limited to these) like real time data acquisition system, power factor monitoring and control, motor speed control, dish antenna positioning control system, automated vehicle collision control, automobile performance analyzer etc., using any of the 16-bit or 32-bit microcontroller available in the market.
2. Implement 64 bit x 64 bit multiplication using higher level language C and assembly language. What aspects of these programming did you like or dislike? Discuss on their performance parameters and timing optimization.
3. Implement N-point decimation-in-time (DIT) and decimation-in-frequency (DIF) radix-2 FFT algorithms using higher level language C and assembly language. Analyze their computational and timing complexity.