NEW YORK CITY COLLEGE OF TECHNOLOGY
THE CITY UNIVERSITY OF NEW YORK

Department: **Electrical & Telecommunications Engineering Technology**

Course Code &
Course Title: EET 3112
Advanced Microcontroller and Embedded System Design

Courses Description: This course will introduce in-depth of microcontroller systems, including architecture, interface, and applications. It is a "hands-on" course which will use PIC microcontrollers from Microchip Inc. to implement embedded systems. Students will be expected to spend a considerable amount of time in the lab experimenting with the microcontroller. They will gain significant experience in using a PIC-based microcontroller to input information from user, via buttons and switches, and output information using LEDs and LCD displays. Both assembly and C/C++ languages are used during the lab and assignments. Toward the end of the semester, students will know how to design and develop an embedded system in details for different applications.

Prerequisite: AAS degree in EET or department approval

Corequisite: EET 3102

Textbook: PIC32 Microcontrollers and the Digilent chipKIT:
Introductory to Advanced Projects, By Dogan Ibrahim,
ISBN-13: 9780080999340
ISBN-10: 0080999344
Publisher: Newnes, 2015

Reference books: Advanced PIC Microcontroller Projects in C
ISBN-13: 978-0750686112
ISBN-10: 0750686111
Publisher: Newnes, 2008

Course Objectives/
Course Outcomes:
(ETAC/ABET Criteria 3, Program Criteria)

Upon completion of this course the students will be able to:

1. Understand differences between general purpose microprocessor and single chip computer – microcontroller (ETAC/ABET Criteria 3a,3b).
2. Understand the main building blocks of a microcontroller and digital minicomputer (ETAC/ABET Criteria 3a,3b).
3. Learn the MPLAB IDE XC32 Compiler. Create, build, compile, debug, and run a project in C/C++ program in the MPLAB IDE (ETAC/ABET Criteria 3a, 3b, 3c, 3e, 3g, 3f, 3h, 3k).

4. Learn the differences between general purpose C/C++ programming and C/C++ programming for embedded system (ETAC/ABET Criteria 3a, 3b).
5. Follow the logic and flow of information in a program. How to create functions and classes (ETAC/ABET Criteria 3a, 3b, 3c, 3d, 3f, 3k).
6. Write, run and analyze C/C++ programs for embedded devices (ETAC/ABET Criteria 3a, 3b, 3c, 3d, 3e, 3f, 3g, 3k, PC a).
7. Transmit data from serial port to PC. Handle the interrupt on PIC. Learn the functions of CCP in depth (ETAC/ABET Criteria 3a, 3b, 3c, 3d, 3e, 3f, 3g, 3k, PC a, PC c).
8. Use A/D and D/A converter and Interfacing microcontroller with sensors. Output data on LCD. Learn the functions of sensors and PMODs (ETAC/ABET Criteria 3a, 3b, 3c, 3d, 3e, 3f, 3g, 3k, PC a, PC b, PC c).
9. Communicate with peripheral sensors (ETAC/ABET Criteria 3a,3b, 3c, 3d, 3e, 3f, 3g, 3h, 3k, PC a, PC b, PC c).

CLASSHOURS: 5 (2 Lecture hours & 3 Lab hours)

CREDITS: 3

Course Coordinator: Prof. X. Wei
Email: xwei@citytech.cuny.edu
Phone: 718-260-5934

Prepared by: Prof. X. Wei
Spring 2020

Grading Procedure:

Class Participation:	10%
Homework:	15%
Lab Report:	15%
Midterm Exam	30%
Final Exam	30%

Instructional Objectives and Assessment

Instructional Objectives:	Assessment:
A. Understand differences between general purpose microprocessor and single chip computer - microcontroller. B. Understand the main building blocks of a microcontroller and digital minicomputer.	Using the PIC32 development board the student should be able to: Understand the block diagram of the microcontroller (hardware/software) and its difference with Inter or AMD CPU; Identify the chips and the general wiring diagram that make up the PIC32 board; Know the functions of the Microcontroller (ALU, CPU Registers, Clock); Know the memory organization; Draw a schematic diagram of the microcontroller; Draw a logic diagram of the microcontroller.
C. Learn the MPLAB IDE XC32 Compiler. Create, build, compile, debug, and run a project in C/C++ program in the MPLAB IDE. D. Learn the differences between general purpose C/C++ programming and C/C++ programming for embedded system.	Use MPLAB IDE to create a project and C/C++ source code program. Learn how to build the executable code and transfer it from PC to microcontroller, know how to Use MPLAB ICD3 or chipKIT3 to perform software debugging for the PIC32 microcontroller, learn to Use MPLAB IDE simulator in perform software debugging for the PIC32 microcontroller. Compare the styles of two types of C/C++ programming from sample programs.
E. Follow the logic and flow of information in a program. How to create functions and classes.	Understand the structure of C/C++ source. Get familiar with logical expression and bit operation. Learn the rules to create a function and class. Understand the scope of local variables and global variables. Testing the codes in MPLAB IDE with XC32 compiler
F. Write, run and analyze C/C++ programs for embedded devices.	Learn more differences between embedded C/C++ and general C/C++ program. Understand the head file, C/C++ source file and their relationships. Learn more library functions in XC32 C++ compiler.
G. Transmit data from serial port to PC. Handle the interrupt on PIC. Learn the functions of CCP in depth.	Understand the four aspects of the EIA232 standard, explain the operation of a USART module, wire the USART pins to the EIA232 connector, and program the USART module to perform data communications. Learn the functions of Capture/Compare/PWM modules in XC32 Compiler and their applications.
H. Use A/D and D/A converter and Interfacing microcontroller with sensors. Output data on LCD. Learn the functions of sensors and PMODs. I. Communicate with peripheral sensors.	Explain the A/D conversion process, describe the resolution, the various channels, and the operation modes of the PIC32 A/D converter, interpret the A/D converter for the application, Interface PIC32 with different sensors and networking Pmods from Digilent Inc.

Course Outline:

Week	***Lecture Topic***	***Laboratory***
1	**1. Embedded systems in today's world** What is an embedded system? CISC and RISC, Some example embedded systems **2. Microcontroller Systems** Instruction sets, Memory types, Organizing memory, Timer, Watchdog, Interrupts, PWM, CAN, ZigBee, Ethernet, JTAG, USB interface **3. PIC Microcontroller Features** PIC Microcontroller families, Microchip and the PIC microcontroller, PIC microcontrollers today **4. What others do –** Other microcontrollers in industry and latest electronics at CES show.	1. Installation of Microchip MPLAB IDE development system. 2. Demo of the PIC16, PIC18, PIC24 & PIC32 hardware systems. 3. Microchip Microcontroller development IDE. 4. Introduce the function of picKIT3 and ICD3.
2	1. Minimum Systems and the PIC Family: A family overview, an architecture overview of the PIC16, PIC18, PIC24, and PIC32. A review of memory technologies. 2. Input/Output Ports and configuration. 3. General Introduction of MPLAB IDE 4. Starting to program – an introduction to embedded C/C++ program. 5. Objects and Classes; Object-Oriented Analysis and Design; Basic I/O in C++ using IOStreams. 6. The process of writing in C/C++ 7. The project development procedure.	1. Create a project in Microchip MPLAB IDE and type the C/C++ code in MPLAB IDE; 2. First example code in the PIC IDE; 3. Choose device and compiler in MPLAB IDE. 4. Build the project and run the program. 5. Introduction of Arduino Project #1
3	**1. Introducing Digilent's chipKIT hardware board, MX3, MX4, & MX7.** **2.** Introducing MPLAB IDE: The elements of MPLAB, The MPLAB project files. The MPLAB user interface. **3.** Operator overloading; Global operators vs. class member operators; Defining member functions in a class definition; Defining member functions in separate .cpp files. **4.** Building C/C++ programs **5.** Introducing logical instructions: for loop and while loop; **6.** Introduction of MPLAB IDE XC8 and XC32 C++ compiler **7.** Introduction of the MPLAB SIM.	**1. An in depth introductory of MPLAB IDE.** Creating a project, Entering source code, build the project. **2. An introduction to simulation** Getting started; Generating port inputs; Viewing microcontroller features; Resetting and running the program. **3.** Downloading the program to a microcontroller with picKIT3 or ICD3. **4.** Run the program with SIM. **5.** Discussion and development on Arduino Project #1

4	1. **Introudction of Peripheral Modules**: PmodCLP, PmodKYPD, PmodSD, PmodBTN, PmodSTEP, PmodTMP3, PmodDA1, PmodAMP2. 2. Working with chipKIT MX3 board: Power, LEDs, Reset, CPU clock, I2C, SPI, and UART interfaces. Pmod connectors. 3. Constructors and destructors; Memory mapped hardware access. 4. Programming C/C++ on MX3 board.	1. Setting the configuration bits within the program. 2. Check the features of picKIT3 and ICD3. 3. Programming and testing temp sensor and light sensor. 4. Discussion and development on Arduino Project #1.
5	**1.** Representing inheritance in a class, Accessing base class members from a derived class, Public, private and protected inheritance, Inheritance and pointers to class instances, Overriding and hiding methods, Virtual functions, dynamic polymorphism. **2. From switches to keypads;** The keypad, Design example: use of PmodKYPD in MX3 board **3. LED & LCD displays;** Display the information on the LED and LCD with PIC board. **4. Introduction of Liquid crystal displays** **5. Some simple sensors for PIC.**	1. Input/Output of Ports. 2. Tristate Register 3. Actuators: motors and servos 4. Interfacing to sensors. 5. Frequency (speed) control and measurement. 6. Arduino Project #1 report and Arduino project #2 introduction.
6	**The CCP in PIC:** **1. The capture/compare/PWM (CCP) modules;** A capture/compare/PWM overview, Capture mode, Compare mode. **2. The capture/compare/PWM (CCP) modules:** The principle The control registers, Capture mode, Compare mode, Pulse width modulation. **3. Implementing synchronous serial I/O in the microcontroller** **4. Enhancing synchronous serial, and the Inter-Integrated Circuit bus** **5. The MSSP configured for I2C** **6.** Introduction to the standard template library(STL), Template functions, Template classes. **7.** Programming Pmod for DC Brush Motor control.	1. Shift operator 2. XOR operator 3. Software delay 4. Loops 5. Arduino Project #2 discussion and development

7	**1. Data acquisition and manipulation;** The A/D converter, Signal conditioning – amplification and filtering, The analog multiplexer, Data acquisition in the microcontroller environment. **2. The PIC32 ADC module;** Controlling the ADC, The analog input model, Calculating acquisition time. **3. Applying the ADC in the Pmod light sensor;** Configuration of the ADC, Acquisition time, Data conversion. **4. Some simple data manipulation techniques;** Fixed- and floating-point arithmetic, Binary to Binary Coded Decimal conversion, Multiplication. **5. The comparator module** **6.** Object-oriented approaches to multitasking and task scheduling, Object oriented approaches to inter-process communication **7. Programming on the PIC32 ADC.**	1. Interfacing with Ports 2. Running LEDs 3. Direction of Ports 4. Interrupt handling. 5. Arduino Project# 2 discussion and progress report Demo of Interfacing with DC Motor with Pmod H-bridge. Demo of project on Stepper Motor Drive with PmodSTEP. Arduino Project #2 discussion and development
8	**Midterm Exam**	
9	1. The PIC peripherals 2. The parallel ports; The PIC parallel slave port. 3. The timers; Timer 0, Timer 1, Timer 2, Timer 3, The Watchdog Timer. 4. Use of XC32 libraries, and the Standard Library. 5. What are exceptions, Throwing an exception, The try block, Catching an exception 6. Programming on the Timer and parallel ports.	1. Using Timer 0 and Timer 1. 2. Using interrupts, and the ISR action. 3. Interfacing with Pmod RTCC-Real-time clock/calendar. 4. Example with interrupt on overflow – flashing LEDs. 5. Demo programs on the Digilent's MX7 board.

10	1. Function calls and data passing in C /C++ and the embedded environment; 2. Sequence, Data Containers, Iterators and Algorithms, Efficient use of the STL. 3. Delay functions in XC32 Library. 4. Using the timer and PWM peripherals. 5. Introduce PmodMaxSONAR-Ultrasonic Range Finder and PmodGYRO 3-Axis Digital Gyroscope 6. Formatting data for display 7. Using library functions for data formatting	1. Binary to Hexadecimal decoder 2. Display on seven segments LEDs. 3. An example program: using pointers, arrays and strings 4. Arduino Project #2 report
11	1. Effective use of templates, C/C++ Interfacing with assembly language code. Controlling memory allocation ;Memory allocation pragmas, Setting the Configuration Words, Interrupts, The Interrupt Service Routine , Locating and identifying the ISR. 3. Storage classes; Scope, Duration, Linkage, Working with PIC Series memory, Storage class examples. 4. Processor-specific header files; SFR definitions 5. Introduce to Pmod GPS Receiver	1. CCP module: Comparator 2. Compare two analog inputs 3. Demo of Interfacing with Pmod GPS receiver 4. Demo of project on Pmod CMPS 3-Axis Digital Compass. 5. Arduino Project #3 discussion and progress report .
12	1. C++ Considerations for Safety-Critical and Real-Time Systems, Overview of the MISRA C++ Guidelines. 2. Techniques of Connectivity and Networking 3. Networking and connectivity 4. Introduction of 802.11 WiFi protocols 5. The IrDA and the PIC microcontroller	ADC module 1. Analog to Digital Conversion 2. Precision representation of ADC 3. Arduino Project #3 discussion and progress report

13	PIC: Connectivity and Communication 1. Radio connectivity 2. Bluetooth and PmodBT2 interface 3. Zigbee and the PIC microcontroller 4. Controller Area Network (CAN) 5. Local Interconnect Network (LIN)	CCP Module 1. An example of software generated PWM 2. PWM used for digital-to analog conversion 3. An example of PWM used for digital-to-analog conversion. 4. Design example: use of LCD display Term project #3 discussion and progress report
14	1. Embedded systems and the Internet 2. Introduction of Arduino & Raspberry Pi 3. Internet of Things (IoT) with Microcontroller and sensors. 4. Artificial Intelligence of Things for Embedded System. 5. Future of Embedded System: IoT vs AIoT. 6. More applications of Embedded System in Industry.	Implement C/C++ code and display the information on LCD modules. Projects #3 demonstrations & presentations. Review
15	**Final Exam**	

New York City College of Technology Policy on Academic Integrity:

Students and all others who work with information, ideas, texts, images, music, inventions, and other intellectual property owe their audience and sources accuracy and honesty in using, crediting, and citing sources. As a community of intellectual and professional workers, the College recognizes its responsibility for providing instruction in information literacy and academic integrity, offering models of good practice, and responding vigilantly and appropriately to infractions of academic integrity. Accordingly, academic dishonesty is prohibited in The City University of New York and at New York City College of Technology and is punishable by penalties, including failing grades, suspension, and expulsion. The complete text of the policy on Academic Integrity may be found in the college catalog.