Course Syllabus

The Mechatronics Revolution: Fundamentals and Core Concepts

Professor: Dr. Jonathan Rogers

Course Description

The Mechatronics Revolution is upon us. Never before has it been easier to build robotic devices and computer-controlled machines. The confluence of low-cost embedded computers, sensors, and actuators has made experimentation and prototyping easier than ever. In this course, students will learn to harness the power of microcontrollers, sensors, and actuators to build useful and interesting robotic devices. Course modules will explore the mechatronics design process, fundamentals of microcontrollers, embedded system programming, actuator control, and sensor interfacing. Through several hands-on labs, students will solidify their knowledge of core mechatronics concepts and gain real-world experience building their own mechatronic systems. Upon successful completion of the course, students will be equipped with the fundamental knowledge they need to harness the Mechatronics Revolution.

Prerequisite

- Familiarity with basic circuits and circuit components
- Familiarity with Programming in either C, MATLAB, or Python

Course Goals

Students who take this course can expect to achieve the following goals:

- Recognize and describe the elements of a microcontroller as well as the operating principles of common motors, sensors, and circuits commonly used in mechatronic devices
- Create basic programs for microcontrollers using interrupt-driven programming
- Design and implement microcontroller programs that read from sensors and achieve appropriate actuator commands
- Implement a mechatronic device using actuators, sensors, and a microcontroller command processor

Grading Policy

- There will be six lab assignments. Lab assignment 1-5 will be worth 15% of the final grade. The final lab will be worth 25% of the final grade.

- Some labs will require students to answer a series of questions, while others will require students to upload recorded sensor data. These submissions will be graded by course teaching assistants.

Course Timing and Assignment Due Dates

- All assignments are self-paced and there are no due dates. Students are encouraged to complete the course modules and assignments in a timely way. Since each of the assignments build on each other, this will ensure that skills and knowledge acquired early in the course are not forgotten later on.

Plagiarism Policy

- Plagiarism is considered a serious offense. You are not allowed to copy and paste or submit materials created or published by others, as if you created the materials. All materials submitted and posted must be your own.

Student Honor Code

All Audit and Verified course participants (myself, teaching assistants, and learners) are expected and required to abide by the letter and the spirit of the edX honor code.

- Review the edXHonor Code:
- You are responsible for completing your own work.
- Any learner found in violation of the edX Honor Code will be subject to any/all of the actions listed in the edX Honor Code.

Communication

- All learners should ask questions, and answer their fellow learners' questions, on the course discussion forums. Often, discussions with fellow learners are the sources of key pieces of learning.
- Verified learners can also ask questions of the instructor and teaching assistants via the course discussion forums. For special cases such as failed submissions due to system errors, missing grades, failed file uploads, emergencies that prevent you from submitting, personal issues, etc., a special email address will be provided in a discussion forum for you to directly contact the instructor and teaching assistants.

Netiquette

- Netiquette refers to etiquette that is used when communicating on the Internet. When you are communicating via email, discussion forums or synchronously (real-time), please use correct spelling, punctuation and grammar consistent with the academic environment and scholarship.

- We expect all participants (learners, faculty, teaching assistants, staff) to interact respectfully. Learners who do not adhere to this guideline may be removed from the course.

Course Structure

The table below contains a course topic outline.

Modules	Lesson Topics

Module 1: Overview of Microcontrollers	• Topic 1: What is Mechatronics?
	  • Lesson 1: Mechatronics as the Interface of Actuators, Sensors, and Computers
	  • Lesson 2: The Role of Mechatronics in Robotics
	  • Lesson 3: The Mechatronics Design Process
	• Topic 2: Introduction to Microcontroller Technology
	  • Lesson 1: Microcontrollers, Microprocessors, and Embedded Computers
	  • Lesson 2: Microcontroller Specifications
	  • Lesson 3: The MSP432 Microcontroller

Module 2: Circuits and Electrical Components	• Topic 1: Basic Circuit Concepts
	  • Lesson 1: Circuit Elements and Constitutive Relationships
	  • Lesson 2: Circuit Analysis and Voltage Regulation
	  • Lesson 3: Amplifier Circuits
	• Topic 2: Semiconductor Circuit Elements
	  • Lesson 1: Diodes, LEDs, and Photodiodes
	  • Lesson 2: Bipolar Junction Transistors (BJTs)
	  • Lesson 3: Metal-Oxide Semiconductor Field Effect Transistors (MOSFETs)
	• Topic 3: Example Circuits for Mechatronic Devices
	  • Lesson 1: H-Bridges and Motor Driver Circuits
	  • Lesson 2: Signal Conditioning Circuits

Module 3: Microcontroller Fundamentals	• Topic 1: Integer Math
	  • Lesson 1: Binary Numbers
	  • Lesson 2: Two's Complement and Integer Types
	  • Lesson 3: Hexadecimal Numbers
	• Topic 2: C Programming for Microcontrollers
	  • Lesson 1: Binary, Assembly, and Compilation Workflow
	  • Lesson 2: Variable Types and Scope
	  • Lesson 3: Functions and Flow Control

	• Lesson 4: Arrays and Structs • Lesson 5: Macros and Header Files • Lesson 6: ASCII and Strings • Lesson 7: Bitwise Operations • Topic 3: General-Purpose Input Output • Lesson 1: GPIO Pins and Ports • Lesson 2: Reading and Writing Digital Inputs and Outputs • Lesson 3: MSP432 Driver Library • Lesson 4: Switch, Button, and LED Interfacing • Topic 4: Interrupt-Driven Programming • Lesson 1: Overview of Interrupt-Driven Programming • Lesson 2: Configuring Interrupts on the MSP432 • Lesson 3: GPIO Interrupts • Topic 5: Clocks and Timers • Lesson 1: Microcontroller clocks • Lesson 2: Microcontroller timers
Module 4: Motors and Servos	• Topic 1: Actuators for Mechatronic Systems • Lesson 1: Actuator Types and Tradeoffs • Topic 2: Brushed DC Motors • Lesson 1: Introduction to Brushed DC Motors • Lesson 2: Torque vs Speed Characteristics of DC Motors • Lesson 3: Brushless DC Motors • Lesson 4: Specifications of DC Motors • Topic 3: Pulse Width Modulation and Servo Control • Lesson 1: PWM Waveform • Lesson 2: PWM Control of Mechanical Devices • Lesson 3: Generating PWM Signals on the MSP432 • Lesson 4: Servo Operation and Control
Module 5: Sensors and Microcontroller Interfacing	• Topic 1: Sensors for Mechatronic Systems • Lesson 1: Basic Sensor Specifications • Lesson 2: Resistive Displacement Sensors • Lesson 3: Encoders • Lesson 4: Proximity Sensors • Topic 2: Analog-to-Digital Conversion • Lesson 1: Purpose and Operation of an ADC • Lesson 2: ADC Range, Resolution, and Precision • Lesson 3: Configuring the ADC on the MSP432 • Topic 3: Universal Asynchronous Receiver-Transmitter (UART) • Lesson 1: TTL Serial Communication

	• Lesson 2: Serial Interfaces and UART Configuration • Lesson 3: UART Operation • Lesson 4: Using UART on the MSP432 • Topic 4: I2C and Serial Peripheral Interface (SPI) • Lesson 1: I2C Communication Protocol • Lesson 2: SPI Communication Protocol • Lesson 3: Using I2C on the MSP432 • Lesson 4: Using SPI on the MSP432 • Topic 5: Examples of Sensor-Microcontroller Integration • Lesson 1: Temperature Sensor Data Acquisition – Part 1 • Lesson 2: Temperature Sensor Data Acquisition – Part 2 • Lesson 3: Interfacing with and Ultrasonic Proximity Sensor – Part 1 • Lesson 4: Interfacing with and Ultrasonic Proximity Sensor – Part 2
Module 6: Building Mechatronic Systems	• Topic 1: Building a Hexapod Robot • Lesson 1: Hexapod Robot: Mechatronic Design • Lesson 2: Hexapod Robot: Software Design • Topic 2: Building a Line Following Robot • Lesson 1: Line Following Robot: Mechatronic Design • Lesson 2: Line Following Robot: Finite State Machine • Lesson 3: Line Following Robot: Software

Course Materials

- All content and course materials can be accessed online
- There is no textbook for this course
- There are four main documents that you will find useful for this course. They are available for free online on the Texas Instruments website. Links are provided for each document; however, these links may change as the documents are updated. These documents are:

1. **MSP432P4xxx SimpleLink Microcontrollers Technical Reference Manual**. Provides a detailed description of the MSP432P4xxx microcontroller and operation of all the peripherals. (https://www.ti.com/lit/ug/slau356i/slau356i.pdf)
2. **MSP432P401R, MSP432P401M Datasheet.** Provides a description of the electrical characteristics of the MSP432 microcontroller. (http://www.ti.com/lit/ds/symlink/msp432p401r.pdf)
3. **MSP432 Peripheral Driver Library User's Guide.** Provides documentation for the MSP432 Driver Library (driverlib) functions and data structures. (http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSP432_Driver_Library/latest/

exports/driverlib/msp432_driverlib_3_21_00_05/doc/MSP432P4xx/MSP432_DriverLib_Users_Guide-MSP432P4xx-3_21_00_05.pdf)

4. **MSP432P401R SimpleLink Microcontroller LaunchPad Development Kit User's Guide.** Provides instructions and specifications for using the LaunchPad development kit. (http://www.ti.com/lit/ug/slau597f/slau597f.pdf)

Students are highly encouraged to have these documents handy and refer to them throughout the course for more information when needed.

Technology/Software Requirements

- Internet connection (DSL, LAN, or cable connection desirable)
- Adobe Acrobat PDF reader (free download; see https://get.adobe.com/reader/)
- Texas Instruments Code Composer Studio. One of the Lab Assignments will provide instructions on how to download and install this software.