ELEC A223: EMBEDDED CONTROL SYSTEMS

Item	Value
Curriculum Committee Approval Date	12/02/2020
Top Code	093410 - Computer Electronics
Units	4 Total Units
Hours	108 Total Hours (Lecture Hours 54; Lab Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Grading Policy	Standard Letter (S)

Course Description

An advanced course designed to develop hands-on skills and knowledge of microchips used in embedded systems. Emphasis is on the ATmega microprocessor as implemented on the Arduino Development Platform. Students will learn how to develop circuits, integrate sensors, control outputs, write code in C, and compile and flash programs to the microprocessor. This course includes an introduction to Real Time Operating Systems (RTOS) commonly used in embedded control systems. ADVISORY: ELEC A111. Transfer Credit: CSU.

Course Level Student Learning Outcome(s)

1. Students will become proficient at authoring code in C that is functional, readable, and properly documented using industry standards.
2. Students will be able to interface various analog and digital sensors as well as other microchips with a microcontroller.
3. Students will be able to use diagnostic tools such as oscilloscopes and multimeters to visually analyze microcontroller signals.

Course Objectives

- 1. Explain the basic components of a microcontroller.
- 2. Demonstrate the use of an oscilloscope to visualize microcontroller output waveforms.
- 3. Demonstrate the ability to author, compile, and flash a custom program to a microcontroller using common methods.
- 4. Differentiate between analog and digital signals.
- 5. Demonstrate the ability to interface digital signals to the microcontroller.
- 6. Demonstrate the ability to read analog signals into a microcontroller program.
- 7. Demonstrate the ability to produce PWM signals.
- 8. Use interrupts to monitor a button.
- 9. Differentiate between the different types of memory used on a microcontroller.
- 10. Demonstrate the ability to program a microcontroller to utilize serial communication protocols.
- 11. Demonstrate the ability to interface a IC sensor to a microcontroller using an i2c bus.

Lecture Content

Microcontroller Design Basic parts of a microcomputer Bus structure Specifications of the processor used in lab Diagnostic Equipment Review Oscilloscope Multimeter Programming Methods in C Programming Languages Libraries Development Environments Compliers Methods of USB Serial ICSP Program Structure Setup Loop Interrupts User Functions Digital Inputs and Outputs Capabilities Interrupt Basic I/O PWM i2c Serial Assigning functions and characteristics Pull Up Resistors Triggers Polling Interrupts Writing using library functions Writing using registers Update Frequency Sensors Reed Switches Buttons Limit Switches Microcontroller Memory Flash Memory Bootloader Program Memory RAM Applications Registers Addressing Variables and Size Limitations EEPROM Applications Addressing Write Cycles External Storage Analog I/O Analog to Digital Resolution Reference Voltage Scale and Shift Digital to Analog Step Size Filtering Techniques Analog Sensors Potentiometers Force Sensitive Resistors Photoresistors Temperature Sensors I2c Devices Interfacing and Addressing Using Libraries Reading Datasheets Parsing Values

Lab Content

Microcontroller Familiarization Exercise Connecting to the IDE “Hello World” Program via Serial Toggling an LED Test Equipment Familiarization Use an Oscilloscope to test the accuracy of the microcontrollers timers and PWM capabilities. Use a benchtop multimeter to test the accuracy of the microcontrollers Use an Oscilloscope to visualize the Digital to Analog converters step size. Serial communication Exercise Connect two microcontrollers and utilize their serial communication capabilities to pass text based messages back and forth. Polling vs. Interrupt Exercise Implement a button using polling Implement a button using interrupts Troubleshoot button “bounce” Traffic Signal Program Develop flowchart or pseudocode. Use LEDs and buttons to represent a traffic light and road sensors. Program the microcontroller in C to emulate a standard “priority and on demand” traffic signal. Analog I/O Exercise Develop a program to read a potentiometer value into the microcontroller. Use the value to control the speed and direction of a motor driven by a PWM signal. Stepper Motor Exercise Develop a program to drive a stepper motor based on a potentiometer position. Interface a stepper motor driver with the microcontroller. I2c Sensor Integration (Final Project) Students choose a sensor to integrate with their microcontroller. The sensor value must be used to either drive an output or displayed using some visual means as well as serial. Example- a temperature sensor reports the temperature over serial and illuminates LED s based on defined ranges.

Method(s) of Instruction

- Lecture (02)
- DE Live Online Lecture (02S)
- DE Online Lecture (02X)
- Lab (04)
- DE Live Online Lab (04S)
- DE Online Lab (04X)

Instructional Techniques

Lecture Detailed whiteboard lectures with opportunity for student engagement. Discussion of media provided and assigned via Canvas. Demonstration of tactile skills. Demonstration of computer programming skills via projector or teleconferencing. Lab Students build and test electronic circuits interfaced with microcontrollers. Lab projects

reinforce lecture topics and are paced to coincide or lag the lecture content. Lab projects generate content that students use to generate reports and documentation, enhancing writing and critical thinking skills.

Reading Assignments

Students will spend approximately 2 hours per week on reading instructor created handouts provided via the LMS

Writing Assignments

Exams Technical reports Engineering journals Approximately 2 hours per week.

Out-of-class Assignments

Students will spend approximately 2-3 hours per week on out-of-class assignments such as the following, in addition to reading/writing assignments: Assignments following lecture content designed to engrain core concepts. Technical reports for each project assigned.

Demonstration of Critical Thinking

Students will design, build, and test several electronic circuits interfaced with a microcontroller. Students will use the C programming language to program the microcontroller to conduct specific tasks. Students will troubleshoot common problems with electronic circuits and microcontrollers

Required Writing, Problem Solving, Skills Demonstration

The students are required to keep a portfolio of lab projects. Students will demonstrate the skills necessary to build circuits around microcontrollers. Each lab project requires a technical report consisting of the following Circuit Schematic Program Flow Chart/Pseudocode Fully commented and documented code. Written explanation of how the circuit functions.

Eligible Disciplines

Electricity (electrical power distribution): Any bachelors degree and two years of professional experience, or any associate degree and six years of professional experience. Electromechanical technology (industrial mechanical technology): Any bachelors degree and two years of professional experience, or any associate degree and six years of professional experience. Electronic technology (radio, television, computer repair, avionics): Any bachelors degree and two years of professional experience, or any associate degree and six years of professional experience.

Other Resources

1. Material provided through LMS.