Curriculum Committee Approval: 02/06/18

Lecture Contact Hours: 32-36; Homework Hours: 64-72; Total Student Learning Hours: 96-108
Laboratory Contact Hours: 48-54; Homework Hours: 0; Total Student Learning Hours: 48-54

CUYAMACA COLLEGE
COURSE OUTLINE OF RECORD

ENGINEERING 176 – MECHATRONICS: PROTOTYPE DESIGN

2 hours lecture, 3 hours laboratory, 3 units

Catalog Description
This course focuses on electromechanical product development. Control of single chip microcontrollers including memory-mapped I/O (Input/Output), direct access to registers, and fine control of timing. Development of custom circuits including manufacture of printed circuits. Control of DC and AC motors and stepper motors. Development of mechanisms and transmissions. Introduction to manufacturing techniques. This course includes a capstone design project. *Also listed as CS 176. Not open to students with credit in CS 176.*

Prerequisite
"C" grade or higher or "Pass" in CS 175 or ENGR 175 or equivalent

Entrance Skills
Without the following skills, competencies, and/or knowledge, students entering this course will be highly unlikely to succeed:
1) Understanding of what a microcontroller is.
2) Basic knowledge of programming techniques including the use of flowcharts, pseudocode, and subroutines.
3) Basic knowledge of a high-level computer language such as C++ that can be used with microcontrollers.
4) Ability to use a microcontroller to take inputs from sensors such as switches, potentiometers, and photoresistors.
5) Ability to use a microcontroller to output signals to devices such as LEDs, servo motors, and DC motors.
6) Ability to integrate a microcontroller into a larger circuit.

Course Content
1) Breadboarding circuits using single-chip microcontrollers
2) Use of compilable high-level languages such as C; concepts of source code and machine code (hex files)
3) Assembly language programming
4) In-circuit programming of microcontrollers
5) Memory-mapped I/O (Input/Output)
6) Registers and register operations
7) Interrupt and exception handling
8) Energy conservation and low-power techniques
9) Theory and practice of sensor calibration
10) Reading and understanding integrated circuit (IC) specifications, especially microcontroller specifications
11) Printed circuit board (PCB) design
12) Custom making PCBs in the lab: board manufacture, populating and soldering, testing
13) Control of 120V AC (Alternating Current)
14) DC motor control: Theory of operation, power consumption vs. speed, pulse width modulation, H-bridge operation

15) Stepper motor control: Theory of operation, power consumption, wave step patterns, half-stepping, two-phase stepping
16) Mechanisms and transmissions
17) Physical prototyping and manufacturing
    a. 3D printing
    b. Laser cutters
    c. Investment casting
    d. Vacuum forming
18) Capstone project including a microcontroller, a printed circuit, and mechanical components.

Course Objectives

Students will be able to:

1) Read and interpret specifications for an unknown microcontroller.
2) Design and build custom microcontroller-based circuits.
3) Write and compile programs in a high-level language such as C.
4) Write programs and subroutines in Assembly language
5) Use microcontrollers to:
    a. Detect inputs from mechanical switches, potentiometers and optical sensors, and use the inputs to control the microcontroller.
    b. Control LEDs, servos, and 7-segment displays.
    c. Control DC motors.
    d. Control stepper motors.
    e. Control high power (AC) circuits.
6) Use physical prototyping tools including 3D printers, laser cutters, investment casting, and vacuum forming to construct a physical object of arbitrary complexity.
7) Integrate the elements of #1-6 above to create an electromechanical device to achieve a desired goal.

Method of Evaluation

A grading system will be established by the instructor and implemented uniformly. Grades will be based on demonstrated proficiency in subject matter determined by multiple measurements for evaluation, one of which must be essay exams, skills demonstration or, where appropriate, the symbol system.

1) Periodic quizzes that evaluate student learning and retention of underlying theory. Questions are mostly word problems, but with some short answer conceptual questions. An example would be a problem in which students would be given the data sheet for a new microcontroller, asked to predict its worst-case ability to drive a set of LEDs, and propose alternative techniques for driving large numbers of LEDs.
2) Lab projects of increasing complexity that measure students' ability to solve problems using a high-level programming language like C, subroutines, assembly language, generic programmers, solderless breadboards, lab-made printed circuit boards, and the use of LEDs, switches, servo motors, DC motors, and relays (for example). An example would be a circuit that switches on and off a 120 VAC fan based on temperature input from a thermistor.
3) Lab projects of increasing complexity that measure students' ability to design and build physical prototypes using a 3D printer, a laser cutter, a vacuum former, and a casting lab. An example would be a simple transmission that transmits power from a DC motor to a wheel.
4) Final capstone project in which students integrate and demonstrate the various topics and techniques covered in class. An example would be a robot built using a lab-made printed circuit board and a non-traditional transmission built using a 3D printer.

Special Materials Required of Student

Microcontroller kits as specified by instructor

Minimum Instructional Facilities

1) Lab with large flat work tables, computers, storage cabinets, computer projection system, overhead projector
2) Computer lab running 3D solid modeling software (e.g., SolidWorks) and a microcontroller programming environment (e.g., MPLab)
3) Electronics lab including soldering stations and printed circuit manufacturing capability
4) Physical prototyping lab including 3D printers, laser cutters, casting facility (mold-making, wax injection, burnout oven, kiln), drill press, and assorted hand tools

Method of Instruction

1) Lecture and discussion
2) Lab demonstration and assignments
3) Design exhibitions
4) Guest speakers

Out-of-Class Assignments

1) Weekly homework including reading and writing assignments
2) Final project
3) Field trips to local manufacturers

Texts and References

1) Required (representative examples):
   a. Ibrahim, D. *PIC Microcontroller Projects in C, Basic to Advanced*. 2nd edition. Newnes, 2014.
   b. Sclater, N. *Mechanisms and Mechanical Devices Sourcebook*. 5th edition. McGraw-Hill, 2011.
2) Supplemental (representative examples):
   a. Clark, Dennis and Michael Owings. *Building Robot Drive Trains*. McGraw-Hill, 2003.
   b. Roberts, D. *Making Things Move, DIY Mechanisms for Inventors, Hobbyists, and Artists*. 1st edition. McGraw-Hill, 2011.

Exit Skills

Students having successfully completed this course exit with the following skills, competencies and/or knowledge

1) Read and interpret specifications for an unknown microcontroller.
2) Design and build custom microcontroller-based circuits.
3) Write and compile programs in a high-level language such as C.
4) Write programs and subroutines in Assembly language.
5) Use microcontrollers to:
   a. Detect inputs from mechanical switches, potentiometers and optical sensors, and use the inputs to control the microcontroller.
   b. Control LEDs, servos, and 7-segment displays.
   c. Control DC motors.
   d. Control stepper motors.
   e. Control high power (AC) circuits.
6) Use physical prototyping tools including 3D printers, laser cutters, investment casting, and vacuum forming to construct a physical object of arbitrary complexity.
7) Integrate the elements of #1-6 above to create an electromechanical device to achieve a desired goal.

Student Learning Outcomes

Upon successful completion of this course, students will be able to:

1) Read and interpret specifications for an unknown microcontroller.
2) Design and build custom microcontroller-based circuits.
3) Write and compile programs in a high-level language such as C.
4) Write programs and subroutines in Assembly language.

5) Use microcontrollers to:
   a. Detect inputs from mechanical switches, potentiometers and optical sensors, and use the inputs to control the microcontroller.
   b. Control LEDs, servos, and 7-segment displays.
   c. Control DC motors.
   d. Control stepper motors.
   e. Control high power (AC) circuits.
6) Use physical prototyping tools including 3D printers, laser cutters, investment casting, and vacuum forming to construct a physical object of arbitrary complexity.
7) Integrate the elements of #1-6 above to create an electromechanical device to achieve a desired goal.