ENGR 170 Mechatronics: Introduction to Microcontrollers, Cuyamaca College

Fall Semester 2009, Section 4593

Saturday 9:00 am – 1:10 pm, Room F301

This course is an introduction to the use of microcontrollers to build "smart" devices. You will learn to program microcontrollers to operate lights, motors, and other electromechanical devices in response to inputs from mechanical, optical, and electrical sensors. We will use BASIC Stamp microcontrollers, which are designed to be easy to learn with. This course is the first of a 4 course sequence. ENGR 171, 172, and 173 respectively cover introductory robotics, intermediate microcontrollers, and intermediate robotics.

Instructor
Dr. Duncan McGehee
Office: F303
Phone: 619-660-4242
email: duncan.mcgehee@gcccd.edu, Website: www.cuyamaca.edu/duncan.mcgehee
Office Hours: MW 5 - 6 pm, T 6 - 7 pm, W 2 – 3 pm, Th 12 – 1 pm, or by appointment

Units and Prerequisites
1.5 units. No prerequisites.

Required Equipment
Required: *What's a Microcontroller*, A. Lindsay, ISBN 1-928982-02-6, Parallax, Inc., 2004. Available as free download from www.parallax.com, or for sale, either at the Cuyamaca College bookstore or directly from Parallax..
Strongly recommended:
Basic Stamp Discovery Kit (USB), Item 27807, Order directly from Parallax Inc: www.parallax.com. $159.99 + tax and shipping.

Other Required Supplies
USB flash drive. Any size will suffice.

Grading
A: 90 - 100
B: 80 - 89.9999
C: 70 - 79.9999
D: 60 - 69.9999
F: < 60

Lab activities and projects	85%
Final project	15%
	100%

ENGR 170 is "project-based", meaning that as you complete projects, the instructor reviews and signs off on them. Each project is worth a certain number of points, and when you complete a lab you submit it for grading. In addition to the signatures, certain projects require schematics and program listings. If you do not include the schematic you will lose a point, and if you do not include the program listing, or if it is poorly commented or badly formatted, you'll lose another point. These points can be recovered if you resubmit the lab with corrections.

Course Objectives (Expected Student Learning Outcomes)

By the end of the class, you will be able to:

1) Design control systems using flowcharts and pseudocode
2) Write programs in a high-level programming language like BASIC to control a microcontroller
3) Use a microcontroller to:
   a. Detect inputs from mechanical switches, potentiometers and optical sensors, and use the inputs to control the microcontroller
   b. Measure variable electrical inputs and respond in a programmed way
   c. Control LEDs to flash as programmed and in response to inputs
   d. Control servo motors as programmed and in response to inputs
   e. Store and retrieve data using non-volatile memory (EEPROM)
   f. Produce sounds as programmed and in response to inputs
   g. Communicate with other integrated circuits such as solid-state potentiometers
4) Integrate the elements of #1-3 above to create an electromechanical device to achieve a desired goal

Policies
1) This is a lab class, and you are encouraged to help each other on the textbook activities. With the creative projects, although you may discuss your ideas with friends, each project should be unique and original.

Important Dates
4 September: Final day to add classes, or to drop without a 'W'.
25 September: Last day to switch to pass/no pass (In my opinion the best way to take the class).
12 November: Final day to drop classes.

subject to minor changes

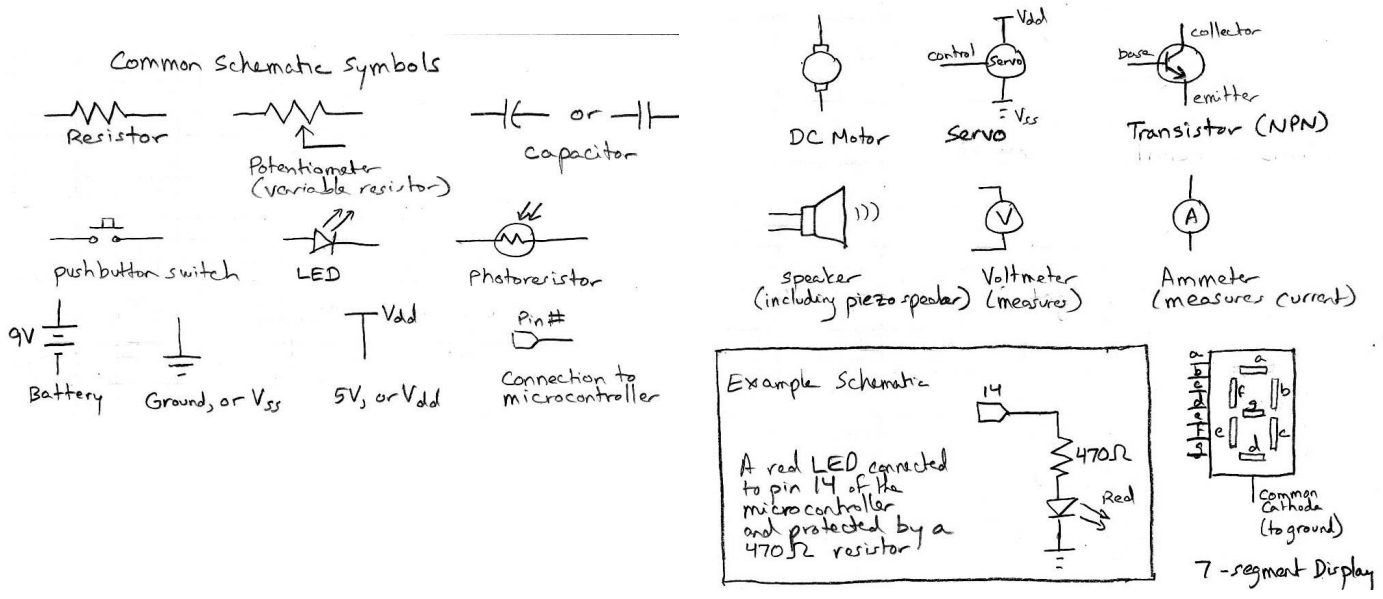
ENGR 170 Mechatronics: Introduction to Microcontrollers, Cuyamaca College

Fall Semester 2009, Section 4593

Saturday 9:00 am – 1:10 pm, Room F301

Tentative Schedule

Week	Dates	Topic	Reading
1	29 Aug	Introduction, what is a microcontroller, BASIC programming, communicating with the microcontroller, ASCII, basic electricity, diodes and LEDs	Ch 1, skipping Activities 1 & 2)
2	5 Sept	Number systems, Binary numbers, basic elements of programming, Timing, Binary output signals, LEDs	Chapter 2
3	12 Sept	Digital I/O (input and output): Pushbuttons and LEDs	Ch 3
4	19 Sept	Digital I/O (input and output): Pushbuttons and LEDs	Ch 3
5	26 Sept	Motion control: Servo Motors	Ch 4
6	3 Oct	Motion control: Servo Motors	Ch 4
7	10 Oct	Intro to Simple A/D (Analog to Digital) **ALSO THIS IS ROBO ED EXPO weekend at Cuyamaca**	Ch 5
8	17 Oct	Simple A/D (Analog to digital): Resistance measurements	Ch 5
9	24 Oct	Hexadecimal, 7-segment displays, Indexed Arrays	Ch 6
10	31 Oct	Hexadecimal, 7-segment displays, Indexed Arrays	Ch 6
11	7 Nov	Light measurements, EEPROM, Subroutines	Ch 7
12	14 Nov	**Veterans' Day Holiday (no class)**	
13	21 Nov	Light measurements, EEPROM, Subroutines	Ch 7
14	28 Nov	**Thanksgiving Weekend (no class)**	
15	5 Dec	Sound Production, System Integration	Chs 8, 9, 10
16	12 Dec	Final Projects	



This course adheres to policies outlined in the Cuyamaca College Catalog. For further information, please see the section of the catalog entitled *Academic Policies*.

subject to minor changes

ENGR 170 Mechatronics: Introduction to Microcontrollers, Cuyamaca College

Fall Semester 2009, Section 4593

Saturday 9:00 am – 1:10 pm, Room F301

Good Programming Practices for ENGR 170

1. Always give a title to your program and a brief description of what it does in the first lines. Also, always provide your name and the date you wrote the program. If you modify the program, add modification dates as well:

Example:

```
' Program FirstProgram.bs2
' This program demonstrates communication between the PC and the Basic Stamp via
' the DEBUG terminal
' Written 6/15/06 Duncan McGehee
' Modified 8/20/09 DM
etc.
```

2. Use plenty of comments to explain what your program does. Some people write their programs first using only comments, then go in and fill in the operations. Note that the single quote mark (aka the apostrophe) indicates that what follows is a comment, and should not be interpreted as an instruction to the Basic Stamp. Also note that you can place a comment on the same line as an instruction.

Example:

```
DEBUG "Hello, world"    'This command sends the words "Hello, world" to the
                        ' DEBUG terminal
```

3. Any loop structure or conditional (IF-THEN-ENDIF) should be formatted by indenting everything inside the loop or condition, as shown in the examples below. This improves readability and makes it easier to navigate and debug a program. Don't ask the instructor for help figuring out why your program doesn't work if you haven't done this. Loop structures include among other things Do-loops and FOR-NEXT loops.

```
'Example 1
DO
    HIGH 14
    PAUSE 500
    LOW 14
    PAUSE 500
LOOP
```

```
'Example 2
IF (IN3 = 1) THEN
    FOR counter = 1 to 5
        HIGH 14
        PAUSE 100
        LOW 14
        PAUSE 100
    NEXT counter
ELSE
    HIGH 14
    PAUSE 500
    LOW 14
    PAUSE 500
ENDIF
```