ENGR 170 Mechatronics: Introduction to Microcontrollers, Cuyamaca College

Fall Semester 2010, Section 4593

Saturday 9:00 am – 12:55 pm, Room F301

Mechatronics is the combination of mechanical, electronic, and computer engineering to create automatic "intelligent" devices. Microcontrollers offer an easy and flexible way to do this. ENGR 170 introduces the use of microcontrollers to operate motors, lights, and other electromechanical devices in response to sensors. You will learn about microcontrollers through a series of projects of increasing sophistication. You will also encountered fundamental concepts in robotics to let you develop autonomous robots that interact with their surroundings. You will build a basic robot, then use it as a test platform to experiment with these ideas.

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, Th 12 – 1 pm, Sat 1 – 3 (in F301), or by appointment

Units and Prerequisites
1.5 units. No prerequisites.

Required Equipment
EITHER: Boebot robot kit, Item 28832 for $159.99 PLUS "What's a Microcontroller?" kit, Item 28152 for $69.99, both from www.parallax.com
OR: Basic Stamp Activity Kit, Item 90005 for $99.99 PLUS Boebot parts and text kit, Item 28154 for $89.99, both from www.parallax.com. This second option is cheaper, but uses a slightly lower quality experimental board.

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	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.

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
3 September: Final day to add classes, or to drop without a 'W'.
24 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.

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 such as BASIC or C to control a microcontroller
3) Use a microcontroller to:
   a. Detect inputs from sensors, and use the inputs to control the microcontroller
   b. Control LEDs, servo motors, speakers, and integrated circuits in response to inputs and programming
   c. Store and retrieve data using non-volatile memory (EEPROM)
4) Integrate the elements of #1-3 above to create an electromechanical device to achieve a desired goal
5) Design an autonomous robot that can survive in an uncertain environment by building up complex behaviors from a combination of simple and robust responses to stimuli.

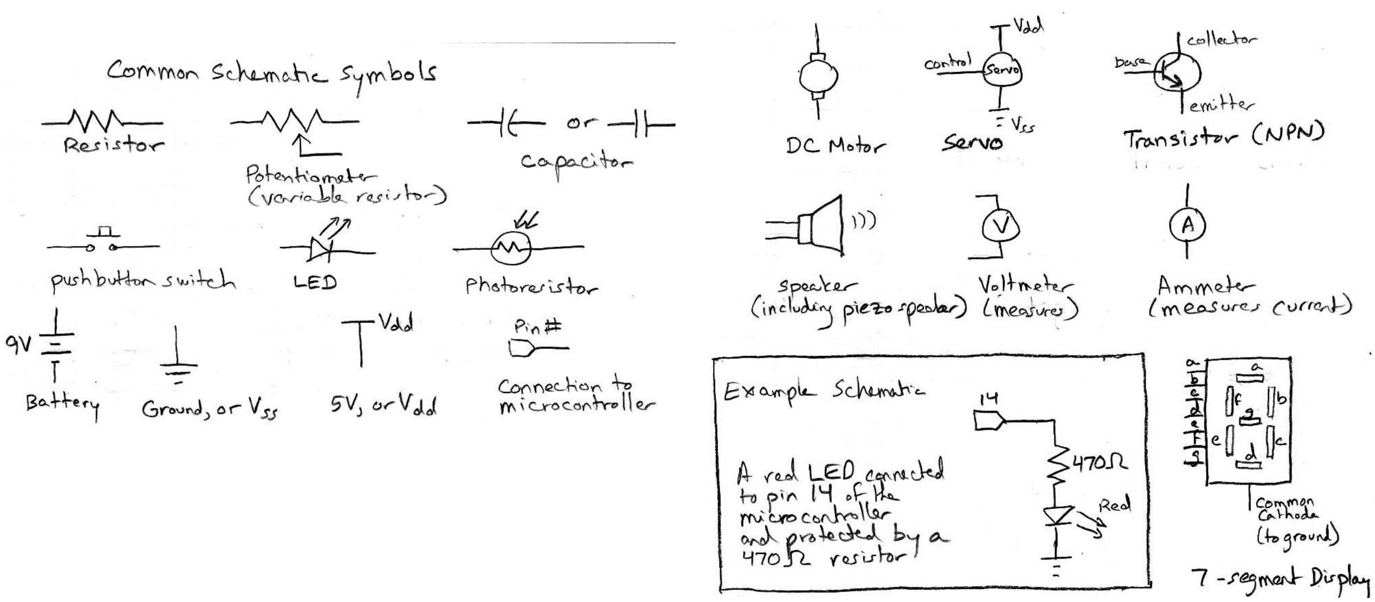
ENGR 170 Mechatronics: Introduction to Microcontrollers, Cuyamaca College

Fall Semester 2010, Section 4593

Saturday 9:00 am – 12:55 pm, Room F301

Tentative Schedule

Week	Dates	Topic	Reading
1	28 Aug	Introduction, what is a microcontroller, BASIC programming, communicating with the microcontroller, ASCII, basic electricity, resistors and LEDs	*What's a Microcontroller?* (WaM) Ch 1, skipping Activities 1 & 2), Ch 2
2	4 Sept	Digital I/O (input and output) and conditionals	WaM Ch 3
3	11 Sept	Motion control: Servo Motors	WaM Ch 4
4	18 Sept	Simple A/D (Analog to digital): Resistance measurements	WaM Ch 5
5	25 Sept	Hexadecimal, 7-segment displays, Indexed Arrays	WaM Ch 6
6	2 Oct	Light measurements, EEPROM, Subroutines	WaM Ch 7
7	9 Oct	Sound Production, SELECT-CASE structure Expanded Control: Transistors and integrated circuits	WaM Ch 8 Ch 9
8	16 Oct	Robo Education Expo	
9	23 Oct	Continuously rotating servo motors, Robot assembly, Robotic Motion	*Robotics with the Boebot* (RwB) Ch 2, 3, and 4
10	30 Oct	Robotic navigation 1: tactile sensors	RwB Ch 4, Ch 5 part 1
11	6 Nov	Robotic "behavior"	RwB Ch 5 part 2
12	13 Nov	Robotic navigation 2: Light sensors	RwB Ch 6 part 1
13	20 Nov	Light sensitive response, "Behavior"	RwB Ch 6 part 2
14	27 Nov	**Thanksgiving Weekend (no class)**	
15	4 Dec	Robotic navigation 3: infrared sensors	RwB Ch 7
16	11 Dec	Final Project	
	18 Dec	Final Project	



subject to minor changes

ENGR 170 Mechatronics: Introduction to Microcontrollers, Cuyamaca College

Fall Semester 2010, Section 4593

Saturday 9:00 am – 12:55 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.

<table>
<tr>
<td>

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

</td>
<td>

```
‘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
```

</td>
</tr>
</table>

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