**Minnesota State University, Mankato**

**Curriculum Proposal**

Please type or select the requested information. Print completed forms, add appropriate paper attachments, and route through MSU’s curricular process for recommendations and decisions.

<table>
<thead>
<tr>
<th>(Check all that apply):</th>
<th>Proposal #</th>
<th>144</th>
</tr>
</thead>
<tbody>
<tr>
<td>College: Science, Engineering and Technology</td>
<td>Undergraduate</td>
<td></td>
</tr>
<tr>
<td>Department: Electrical and Computer Engineering and T</td>
<td>Graduate</td>
<td></td>
</tr>
<tr>
<td>Program: Electrical and Computer Engineering</td>
<td>CIP #</td>
<td></td>
</tr>
<tr>
<td>Type of Change: COURSE PROPOSALS</td>
<td>Effective Date of Change:</td>
<td></td>
</tr>
<tr>
<td>Proposed: Change in Course</td>
<td>Academic Year</td>
<td>06-07</td>
</tr>
<tr>
<td>Title Current: Design and Evaluation of Microprocessors</td>
<td>(For Office Use Only)</td>
<td></td>
</tr>
<tr>
<td>Title Proposed: Microprocessor I Laboratory</td>
<td>Course Designator and Number</td>
<td>EE344 1</td>
</tr>
<tr>
<td>24-Char. Abbrev: Micro II Lab</td>
<td>Credits</td>
<td></td>
</tr>
</tbody>
</table>

Include a course or program description for the Bulletin (30-40 words maximum for courses, 100 for programs):

Laboratory support for EE334. Use of development boards and C programming language to handle I/O devices, interrupts, and all peripheral functions. Multiple functions such as timers, A/D converters, I/O devices, interrupts, and serial modules will be used together to perform desired operations. Prerequisite: Concurrent with EE 334

Rationale or Justification for change:
The addition of EE234 and EE235 make it necessary to modify this course content.

---

### For General Education or Cultural Diversity Courses Only***

**General Education Course:**

<table>
<thead>
<tr>
<th>GE Category #</th>
<th>GE Category Name</th>
<th>(Maximum of 3 Categories)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

? For Writing Intensive Courses, attach a description of the kind and quantity of writing.

? For Upper Division Courses, include a description of the respects in which it is broad and general rather than narrow and specific, and so suitable as GE.

Attach paper copies of the following:

a. Syllabus or course outline.
b. Course’s student learning outcomes associated with each GE competency or CD designation.
c. List of strategies to be used to assess students’ achievement of each GE competency or CD designation.

---

### For New Courses***

<table>
<thead>
<tr>
<th>Instructional Type:</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course is an elective.</td>
<td></td>
</tr>
<tr>
<td>Course is required for program</td>
<td></td>
</tr>
<tr>
<td>Pre- or Co-requisites:</td>
<td></td>
</tr>
<tr>
<td>Course will be offered:</td>
<td></td>
</tr>
<tr>
<td>Fall Semester</td>
<td></td>
</tr>
<tr>
<td>Spring Semester</td>
<td></td>
</tr>
<tr>
<td>Summer Session</td>
<td></td>
</tr>
</tbody>
</table>

Course content or title is similar to courses in other departments. (Attach copy of letter of agreement with other program(s) contacted. Indicate the nature of the discussions and/or resolution of differences or potential conflicts.)

Attach paper copies of the following:

a. Syllabus or course outline.
b. Course’s student learning outcomes.
c. A list of resources required to offer and support this course.
d. A description of how teaching this course will affect department staffing.
e. If 400/500 level course, an explanation of added expectations of graduate students.

---

Revised September 2002
EE344 Microprocessor II Laboratory Syllabus

Instructor: Dr. Han-Way Huang (email: hanwayh@gmail.com)
Classroom: TN 382
Office: TE 324
Phone: 389-1121
Textbook: "Using the MCS-51 Microcontroller" by Han-Way Huang; Oxford University Press

Contents:
- Use of software development tools to enter, compile C programs and download them onto demo board for execution.
- Interrupt service routine in C
- Interrupt-driven input and output with DIP switches, seven-segment displays, and LCD.
- Waveform generation with D/A converter and C language
- MCS-51 interrupt handling in C
- Timer functions including input capture, compare, periodical interrupt, pulse width modulation, and their applications with C language.
- Using the PWM function to generate waveforms with specified frequency and duty cycle
- The function of the SPI module and its application in I/O interfacing.
- The I²C protocol and the implementation of the I²C in MCS-51 microcontroller.
- The A/D converter and data acquisition.
- The CAN protocol and the implementation in MCS-51 microcontroller and the programming in C language.

Objectives

After taking this course, you will be able to

1. Enter and compile C programs using the development tools
2. Write modular programs by dividing a large program into subroutines
3. Use of C language to drive parallel I/O ports and I/O devices such as LCD, seven-segment displays, keypad, D/A converter, stepper motors, and interrupt-driven I/O.
4. Write C program and use the timer function to perform signal measurements, waveform generations, and use the generated waveform to carry out a wide variety of control functions such as motor control, sound generation, light dimming control, and so on.
5. Use of the C language and the SPI function to interface a wide variety of peripheral devices such as seven-segment display driver chip, matrix displays, and so on.
6. Write C programs that use the I²C function and several functions to interface with a wide variety of peripheral devices such as thermostat, real-time clock chip, serial EEPROM and so on.
7. Write C programs that drive the A/D converter (on-chip and off-chip) to perform data acquisition.
8. Write C programs that control the CAN module to carry out data communications between many CAN nodes on the same bus.

Lab Requirements

1. Lab assignment will be made weekly or bi-weekly depending on the difficulty of the assignments.
2. Each lab assignment must be demonstrated to work in order to earn full score
3. A lab report must be typed and submitted within one week after the lab was demonstrated to work.
4. The lab report should include the following:
   (1) problem description
   (2) Algorithm procedure
   (3) Program source code with adequate comments
   (4) TA or instructor signature proving that the lab assignment is working
   (5) Conclusion

Grading methods

- A: > 90%
- B: from 80% to less than 89%
- C: from 70% to less than 79%
- D: from 60% to less than 69%
- F: below 60%
Upon completion of this course, students will have the ability to:

1. Use software development tools to enter, compile, and download C programs onto demo board for execution.
2. Perform C program debugging using software and hardware tools.
3. Write interrupt-driven C programs.
4. Develop C programs that use parallel ports to drive input and output devices such as LEDs, switches, keypads, seven-segment displays, LCD, and stepper motors.
5. Write C programs that use the microprocessor’s timer functions to measure waveform characteristics such as pulse width, period, frequency, and duty cycle and then display the result in appropriate I/O devices such as LCD or LED. The program may display messages that prompt the user to set up data and use interrupt signal to inform the MCU to read the value.
6. Write C programs that use A/D converter to measure voltage, temperature, and other physical quantities such as humidity, pressure, and light intensity. The program is required to use timer function to control the frequency of A/D conversion.
7. Use a D/A converter to generate waveforms (or generate sirens using a speaker).
8. Write C programs that use SPI interface to interface with I/O devices such as a matrix display and real-time clock (RTC). The program will need to use interrupt, input and output devices, and timer functions in the same program.
9. Write C programs that use I²C interface to drive I/O devices such as an EEPROM chip.
10. Write C programs that use CAN bus to transmit data frame over a CAN bus, request other CAN nodes to transmit data frame of certain identifier, and receive data frame from CAN bus.
11. Configure system clock to the desired frequency.
12. Erase and program the on-chip program flash memory.
13. Use multiple peripheral functions in the same laboratory projects.
**Resources Needed to Teach EE344 (1)**

1. Microprocessor demo boards (already in place)
2. Software development tools (already in place)

**Impact to the Department Staffing**

This is an existent course. It does not cause additional teaching load to the department.
Attachment added October 27, 2006
Laboratory Topics
EE 344 (1 credit) periods 110 minutes

Week 1 – Lab Introduction – Course requirements, documentation standards, equipment use
Week 2 - Enter and compile C programs using the development tools
Week 3 – Creation of modular programs and hardware libraries
Week 4-5 - C language to drive parallel I/O ports and I/O devices such as LCD, seven-segment displays, keypad, D/A converter, stepper motors, and interrupt-driven I/O (including hardware interfaces).
Week 6 - C programs and use of timer hardware to perform signal measurements, waveform generations, key debounce, and data acquisition
Week 7 - Control functions and interfaces to control motors and light dimming
Week 8 – Control functions and interfaces for sound generation (in C)
Week 9-10 - Use of the C language and necessary hardware to create SPI functionality
Week 11 Hardware interface and necessary code to create I²C functionality to peripheral devices such as thermostats and real-time clock chips.
Week 12 – Addition of a external high speed A/D converter (on-chip and off-chip) to perform data acquisition.
Week 13 – Creation of a CAN interface to support data communications between other CAN nodes on the same bus.
Week 14 – 15 Student projects