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.

(All that apply): Proposals

Proposal # 145
Effective Date of Change:
Academic Year 60-07
(For Office Use Only)

Course Designator
Number of
Credits

Includes a course or program description for the Bulletin (30-40 words maximum for courses, 100 for programs):
A more advanced study of microprocessors and microcontrollers in embedded system design. Use of C language in programming, interrupt interfaces such as SPI, I2C, and CAN. External memory design and on-chip program memory protection are also studied. Prerequisite: EE234 and EE 295

Rationale or Justification for Change:
This course is evolved as the result of the addition of EE 234 and EE 235 to the electrical and computer engineering curriculum.

***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***

(Check all that apply): Instructional Type: Lecture

Instructional Type: Lecture

Grading Format: Grade [ ] P/N

Course will be offered:

- Fall Semester
- Spring Semester
- Summer Session

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.
### Signature Page

<table>
<thead>
<tr>
<th>Department</th>
<th>( \checkmark ) Recommended</th>
<th>Category/Courses</th>
<th>( \checkmark ) Not Recommended</th>
<th>Category/Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>____</td>
<td>____</td>
<td>____</td>
<td>____</td>
<td>____</td>
</tr>
</tbody>
</table>

**Comments:**

<table>
<thead>
<tr>
<th>College Curriculum Committee</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Recommended</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**Committee Chair:**

**Date:** 11/2/06

**Comments:**

<table>
<thead>
<tr>
<th>College Dean</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Recommended</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**Dean:**

**Date:** 10/6/06

**Comments:**

<table>
<thead>
<tr>
<th>General Education Subcommittee</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Recommended</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**General Education Subcommittee Chair:**

**Date:** __/__/06

**Comments:**

<table>
<thead>
<tr>
<th>Undergraduate Curriculum and Academic Policy Committee</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Recommended</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**UCAP Faculty Chair:**

**Date:** 11/8/06

**Comments:**

<table>
<thead>
<tr>
<th>Faculty Association Graduate Committee</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Recommended</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**Faculty Association Graduate Chair:**

**Date:** __/__/06

**Comments:**

<table>
<thead>
<tr>
<th>Graduate Dean</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Recommended</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**Graduate Dean:**

**Date:** __/__/06

**Comments:**

<table>
<thead>
<tr>
<th>Academic Affairs Council</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Recommended</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**Assistant Vice President:**

**Date:** 4/20/07

**Comments:**

<table>
<thead>
<tr>
<th>Senior Vice President and Vice President for Academic Affairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \checkmark ) Approved</td>
</tr>
<tr>
<td>____</td>
</tr>
</tbody>
</table>

**Sr. Vice President/VICE Pres. Academic Affairs:**

**Date:** 4/20/06

**Comments:**
EE334 Microprocessor II Syllabus

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

Contents:
- Architecture of the MCS-51 microcontroller
- C language tutorial
- Software and hardware development tools
- MCS-51 interrupt handling in C
- Features of parallel ports and characteristics of a wide variety of parallel I/O devices
- Timer functions including input capture, compare, periodical interrupt, pulse width modulation, and their applications with C language.
- Function and applications of the UART function
- 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 features and its programming.
- The CAN protocol and the implementation in MCS-51 microcontroller and the programming in C language.
- Program memory protection
- External memory expansion
- Multi-tasking without operating system

Objectives
After taking this course, you will be able to

1. Use C language to program embedded applications
2. 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.
3. Use C language and 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.
4. Write modular programs by dividing a large program into subroutines
5. Understand interrupts and be able to write interrupt-driven programs
6. Use parallel ports to perform input and output operations (including D/A conversion)
7. Use timer function in measuring frequencies, pulse widths, periods, and generating waveforms.
8. Use of the C language and the SPI function to interface a wide variety of peripheral devices.
9. Use of the C language and the I2C function to interface a wide variety of peripheral devices.
10. Use of the C language and the A/D converter (on-chip and off-chip) to perform data acquisition.
11. Use of the C language and the CAN module to carry data communication between many CAN nodes on the same bus.
12. Perform program memory protection.
Tentative Schedule

Week 1: The architecture of the MCS-51 microcontroller, tutorial of C language.
Week 2: Interrupt and reset handling in C.
Week 3: Parallel percts, interfacing with LEDs, seven-segment displays, LCDs, keypads.
Week 4: D/A converters and stepper motors, Timers 0 through 4, Capture/Compare/PWM modules 1 to 5.
Week 5: Using capture mode to measure pulse width, signal period. Use compare mode to generate waveforms and create time delays. Use compare mode to make sound and play songs. Operation and configuration of PWM mode in C language.
Week 6: Use PWM to control DC motors, the operation of USART module, applications of synchronous and asynchronous modes of USART.
Week 7: SPI module configuration, interface with shift register 74HC595, seven-segment display driver MAX7221, digital temperature sensor TC72.
Week 8: use SPI to interface with real-time clock DS1306.
Week 9: PC protocol, implementation of PC, master and slave modes, interface with digital thermometer DS1631A using PC.
Week 10: Interface with EEPROM 24LC08B and matrix display driver MAX6953.
Week 12: Use A/D converter to measure temperature, humidity, and barometric pressure.
Week 13: CAN protocol, 8051 CAN module and configuration using C language.
Week 14: CAN programming in C and data communication.
Week 15: on-chip EEPROM, flash memory and system configuration.

Exams
Exams will be given at the time scheduled. If you cannot make that time, talk to the instructor ahead of time to make alternate arrangement. Unexcused absence will result in zero credit. No electronic devices will be allowed during the exams including (but not limited to): calculators, phones, PDAs, MP3/CD players.

- Exam 1:
- Exam 2:
- Final: university scheduled date

Homework
Assignments are due at the beginning of class on the day stated. In the event a hardcopy cannot be turned in at that time, an e-mail containing your completed and finished assignment as a word document will suffice until you can turn in the hardcopy.

Grading methods

Exam 1 and Exam 2: 20% each, Homework: 25%, Final Exam: 35%

- A: > 85%
- B: from 70% to less than 85%
- C: from 55% to less than 69%
- D: from 40% to less than 55%
- F: below 40%
EE334 Microprocessor Engineering (3) – Proposed Outcomes Fall 2006
Upon the completion of this course, students will have the ability to:
1. Understand the architecture of a microprocessor including the CPU registers, memory organization, and addressing modes.
2. To prepare logically organized C programs that will run on a microprocessor or microcontroller.
3. Write modular programs in C and be able to logically separate large programs into appropriate smaller program segments
4. To integrate C and assembly code in programs
5. Appropriately design and document C programs
6. Perform interrupt and reset handling programming in C
7. Use parallel I/O ports to interface with I/O devices such as DIP switches, keypad/keyboard, seven-segment display, LCD, D/A converter, and stepper motor and use C and assembly language where appropriate to carry out all the I/O programming task.
8. Perform interrupt-driven I/O in C language.
9. Use both internal and external timer functions to establish timing information on an input signal. Software to support this will be prepared predominately in C.
10. Use both internal and external timer functions to generate time dependent output signals. Software to support this will be prepared predominately in C.
11. Understand the UART serial interface and use it to interface with I/O devices and communicate with a PC or other microprocessors. Software to support this will be prepared predominately in C.
12. Use the serial peripheral interface (SPI) to interface with I/O devices such as display drivers, A/D and D/A converters, matrix displays, digital potentiometers, serial memory devices such as EEPROM and SRAM, and many other I/O devices with SPI interface. Software to support this will be prepared predominately in C.
13. Use both integrated A/D and off chip A/D converters to measure analog data such as temperature, humidity, and pressure. Conversion of the signal will be accomplished used software written in C.
14. Understand and use the Inter-Integrated Circuit (I2C) to communicate with I/O devices such as display drivers, A/D and D/A converters, matrix displays, digital potentiometers, serial memory devices such as EEPROM and SRAM, and many other I/O devices with I2C interface.
15. Understand and use the Controller-Area Network (CAN) to communicate with other microprocessors and exchange data or perform control operations.
16. Understand the operation and programming (may include erasure) of on-chip memory including flash memory and EEPROM.
17. Learn to add external memory to the microprocessor.
18. Perform protection to the on-chip program (stored in flash memory) and on-chip data.
Resources Needed to Teach EE334 (3)

1. No special equipment is needed to teach this course.

Impact to the Department Staffing

This is an existing course. It does not cause additional teaching load to the department.