Milwaukee School of Engineering


# CE-2800 Embedded Systems I

**Description**

This course presents a typical embedded microcontroller and assembly language programming as an efficient and direct means of programmatically controlling the hardware. Topics covered include the addressing modes, register file, and instruction set of a microcontroller; subsystems such as timers and analog to digital conversion; and interrupts. Software control of hardware is stressed. In the laboratory, students design software to demonstrate proficiency in these areas.

**prereq: SE-1011 or experience with a procedural programming language**
**coreq: CE1900**

**Required Materials**

- *Notebook computer required*
- *Microcontroller and subsystem documentation*

**Course Learning Outcomes**

Upon successful completion of this course, the student will:

- recognize the role of assembly language programming
- state the programmer's model of a typical embedded processor
- break down the instruction set of a typical embedded processor, recognizing load/store, arithmetic, conditional branch, and unconditional branch instructions
- construct assembly language programs by using and reusing subroutines
- apply memory addressing and various addressing modes
- understand the concept and usage of interrupts
- given proper documentation, be able to configure and use common microcontroller subsystems such as timers, uart, ADC

**Course Topics**

- Introduction to microcomputer/microcontroller structure from a programmer's perspective
- Programmer’s model of the microcontroller
- Addressing modes and memory types
- Tool usage (assembler, downloader, simulator, debugger)
- Microcontroller instruction set
- Assembly language program structure, including comparisons to high-level languages
- I/O port configuration and usage
- Timer subsystem
- A/D conversion
- Interrupts, including their use related to the timer and external sources such as pushbuttons

- Tests and review

Prerequisites by topic
- Programming fundamentals including functions with arguments
- Good program design techniques

Laboratory topics
- Tools familiarization: assemble, download, run, and simulate a program given to the student
- The first student-written program: assemble, download, run, and simulate a program written by the student.
- Simple I/O program, Button I/O
- Bit banging the LCD display
- Keyboard scanning program
- Timer subsystem program
- A/D program
- Interrupt-driven program

Course Structure 3 - 3 - 4 (class hours/week, lab hours/week, credit hours)

ACCE Content (credit hours):

| General Education | Math/Science | Business/Mgmt | Construction | Construction Science |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |

ABET-EAC Content (credit hours):

| Engineering Topics | Design | Gen Ed | Math/Science | Other |
|---|---|---|---|---|
| 4 | No | 0 | 0 | 0 |

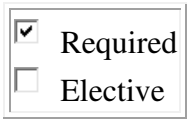
ABET-TAC Content (credit hours):

| Communications | Math/Science | SS/HU | Technical Content | Other |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |

Coordinator: William Barnekow, Professor

Last Review: William Barnekow, Professor on Apr. 25, 2012

Last Update: William Barnekow, Professor on Apr. 25, 2012
William Barnekow, Professor on March 19, 2013



# Weekly Lecture Topics

- **Week 1**

  Lecture Topics:

  Day 1

  - Introduction to the course, course policy, grading policy, review number systems decimal, binary, hexadecimal

  Day 2

  - Binary arithmetic, two's complement addition, embedded system model, CPU, memory, I/O types of memory, flash, eeprom, sram, memory size vs. address bits

  Day 3

  - Introduction to AVR architecture, types of memory and their uses, registers, status flags, program counter

  Lab Topics:

  Day 1

  - Introduction to software tools

- **Week 2**

  Lecture Topics:

  Day 1

  - Introduction to instruction set, arithmetic instructions, branch instructions, data transfer instructions, addressing modes, immediate addressing, direct addressing, pointer addressing

  Day 2

  - Continue addressing modes, program example, adding numbers in a loop

  Day 3

  - Hand simulation of program example, parallel I/O ports, PORTn, DDRn, PINn

  Lab Topics:

  Day 1

  - Basic output and debugging

- **Week 3**

  Lecture Topics:

  Day 1

  - Continue parallel I/O, timing loops, calculate duration of timing loop

  Day 2

  - Review timing loops~~,~~
  - Introduce ~~s~~tack pointer, stack operation, subroutines

  Day 3

  - Review subroutine operation, saving and restoring registers, program example, discuss subroutine documentation

  Lab Topics:

  Day 1

  - Basic output and looping techniques

- **Week 4**

  Lecture Topics:

  Day 1

  - Shift and rotate instructions, bit manipulation operations, set/clear bit in port, branch based upon bit in register clear/set, examples

  Day 2

  - Continue bit manipulation operations

  Day 3

  - LCD interface, 8-bit interface, command write procedure, status read procedure, converting binary to ASCII

  Lab Topics:

  Day 1

  - Timing loops

- **Week 5**

  Lecture Topics:

  Day 1

  - LCD interface, 8-bit interface, data write procedure, data read procedure, converting binary to ASCII

  Day 2

  - ADC fundamentals

  Day 3

  - Midterm Exam

  Lab Topics:

  Day 1

  - LCD interface

- **Week 6**

  Lecture Topics:

  Day 1

  - ADC polling examp;le, Begin interrupt handling

  Day 2

  - Interrupt handling, example

  Day 3

  - Edge triggered interrupt from parallel port

  Lab Topics:

  Day 1

  - Binary to ASCII conversion and display to LCD

- **Week 7**

  Lecture Topics:

Day 1

- Introduction to timers

Day 2

- Periodic interrupts

Day 3

- Introduction to PWM

Lab Topics:

Day 1

- Analog to Digital conversion

- **Week 8**

Lecture Topics:

Day 1

- Continue PWM and PWM example

Day 2

- Introduction to keypad

Day 3

- Keypad encoding method

Lab Topics:

Day 1

- Interrupts

- **Week 9**

Lecture Topics:

Day 1

- Serial communications basics, asynchronous serial data format, start and stop bits, baud rate definition, parity

Day 2

        - Registers needed, transmit function example(polling), receive function example(polling)

    Day 3

        - Begin serial interrupts

Lab Topics:

    Day 1

        - Keypad interface

- **Week 10**

Lecture Topics:

    Day 1

        - Continue Receive function(interrupts), transmit function(interrupts)

    Day 2

        - Continue examples

    Day 3

        - Review for final

Lab Topics:

    Day 1

        - TBA

- **Week 11**

Lecture Topics:

    Day 1

        - Final exam