Final Exam

• When: 8:00-10:00 am Thursday, May 10th
• Location: here

• 1/3: midterm material
  – See lecture notes for midterm preparation
• 2/3: material since midterm

• 1 page of personal notes
• No electronic devices/books/other notes
Final Preparation

• Exam discussion on D2L
  – Post sample questions (and answers)
  – Some may appear on the exam

• Look to homework assignments and exams from last year (both the midterm and final) for the types of questions
  – Note that class coverage in previous years has been different
Pre-Midterm Material

- Number Representations (binary, hex)
  - Two’s complement for signed numbers
- Arithmetic: incrementing, decrementing and shifting
- Bit-wise operators
- Analog to digital conversion (and vice versa)
- Microprocessor components
- Digital I/O on the Atmel Mega processors
- Basic circuits: LEDs, analog comparators, resistors
Key Microprocessor Components

• Data bus
• Data memory (RAM)
• Program memory (EEPROM in our case)
• General- versus special-purpose registers
  – Instruction register
  – Program counter
• Instruction decoder
• Arithmetic Logical Unit
Microcontroller I/O

- Special purpose registers in the Atmel Mega processors:
  - DDRx: data direction
  - PORTx: control pin state
  - PINx: read pin state

- Relationship to C code
New Material

- Pulse-width modulation (PWM)
- H-bridges
- Serial communication and the ASCII representation
- Proportional-derivative control
- Timer/counters
- Interrupts and interrupt service routines
- Finite state machines
Timer/Counters

- Prescalers
- Counters (hardware)
  - Timer0, 2: 8-bit
  - Timer1, 3, 4, 5: 16-bit
- Interrupts on timerX overflow

- Computing timerX count frequencies/periods
- Computing timerX interrupt frequencies/periods
Interrupts

• What are they?
• Interrupt service routines. Examples:
  – Pulse Width Modulation (PWM) generation (see slides)
  – Producing digital signals of various frequencies (e.g., can introduce software counters, too)
Finite State Machines

• Definition
  – States
  – Events
  – Transition function
  – Outputs
  – State transition diagrams

• FSMs for control
C Code

• Be prepared to read (and possibly fix) simple C code
• If any, you will not write more than a few lines of code
• Look to lecture discussions of code and your projects as you prepare