Interrupts

- Life blood of real-time systems
  - What are the other units up to?
  - Trigger on specific events
  - Only way to do hard real-time
- Interrupts stop regular control flow and jump to an Interrupt service routine (ISR)
- Involuntary transfer of control (Context switch)
Interrupts

- Setting up an interrupt
  - Setup jump in the interrupt vector table
  - Enable interrupt for device (e.g., pin changed, Rx ready)
- Enable global interrupts
  - sei() and cli()
- Most of the time execution starts as soon as the event happens
  - Ignore scheduler
- Priorities
  - Lower priority interrupts can not interrupt higher priority ones
  - CPU has lowest priority
Interrupts

- Interrupt running thread/task
- Critical sections
  - Synchronization
  - Read/Write conflicts
- Disable all or some interrupts to avoid data corruption
- Interrupt running interrupt handler
  - Priority inversion
  - Scheduling, resources
Interrupts

- Allow system to recognize event
  - Minimize time that interrupts are disabled
- Prevent other tasks from running
  - Minimize time in interrupt
- Example: Receiving a character from the serial port
  - Enable Rx IRQ
  - ISR
    - Fetch character and put into queue, set flags, rti
  - Main program:
    - Check flags, process character
- Only very simple tasks should be done in the ISR
- Transmit Interrupt Handler
  - Load next character to transmit
Interrupts and Stacks

- 4KByte RAM only
- Subroutines calls should be minimized
- Stacks
  - ATMega128 uses only one stack pointer
  - More complex systems use separate stacks for IRQs for security
  - Stack space on the ATMega128 is limited
Interrupt Vectors

- Interrupt priorities are fixed
- Based on location in interrupt table
- Lower location have higher priority
- Page 59 in ATMega128 Datasheet
Interrupts and avr-gcc

- avr-gcc does a lot of the low level handling
- avr/signal.h
  - INTERRUPT (starts with IRQs enabled, deprecated)
  - SIGNAL (starts with IRQs disabled, deprecated)
  - ISR (starts with IRQs disabled, new code)
- avr/iom128.h
  - SIG_PIN_CHANGE0, SIG_OVERFLOW1, ...
- avr/interrupt.h
  - cli() clear I Bit in SREG -> disable all interrupts
  - sei() enable all interrupts