COMP 4550
Analog - Digital Conversion

Autonomous Agents Lab, University of Manitoba
jacky@cs.umanitoba.ca
http://www.cs.umanitoba.ca/~jacky
http://aalab.cs.umanitoba.ca
Analog to Digital Conversion

- Convert an analog voltage to a digital value
- Schematics
  - Flash converter (directly conversion)
  - Cascading converter (pipeline conversion)
  - Weighing converter (successive approximation)
  - Sigma Delta converter
- Most common is dual-slope counting converter because of high accuracy and low cost
- Requires a certain amount of time
AD Converter: Flash Converter

2-Bit Flash Analog to Digital Converter

Input Quantization (Comparison)  Bubble Error Correction  Digital Encoding
AD Converter: Successive Approximation

SAR: Successive Approximation Register
AD Conversion: Sigma Delta

Flip flop stores comparator output and adds to voltage on 1st comparator
Clocked at very high frequency (e.g., 64 times oversampling)
ATMega 128 AD Conversion

- 10 bit resolution
- 13 – 260 uSec. conversion time
  - 50KHz to 1 Mhz ADC clock selection
- 8 Multiplexed input channels
  - Can only convert one channel at a time
- VCC, 2.56V Reference voltage
- Free running or single conversion mode
- Interrupt driven conversion
- Interrupt can be set when conversion complete
ATMega 128 AD Converter

- ADMUX
  - select input channel
- REFS0/1
  - select reference voltage
- ADLAR
  - left shift 10 bit result
- MUX4..0
  - select input voltage (32 possibilities)
- 8 single ended voltages
- 8 difference voltages
ATMega128 AD Converter

- **ADCSRA**
- **ADEN**
  - ADC enable
- **ADCSC**
  - Start conversion
  - First conversion takes 25 clock cycles
- **ADATE**
  - ADC Autotrigger enabled
- **ADIF**
  - ADC Interrupt flag
- **ADIE**
  - ADC Interrupt enable flag
- **ADPS[2:0]**
  - ADC prescalar
  - 50 - 200 kHz for full resolution
  - Higher clock rate with less resolution
ATMega128 AD Converter

- **ADCL/H**
  - 16 bit result (may be left shifted)
- **Read L first, then H**
- **ADCSRB**
- **ADTS[2:0]**
  - Autoconversion trigger source
  - Free, ADC, Timer
- **DIDR0**
  - disable digital input to reduce power consumption
- **PRR**
  - Enable ADC in power reduction reg.
ATMega128 AD Routine

- Initialization
  - Setup ADMUX, ADCSRA, ADCSRB
  - Enable MUX input
    - 0..7 is single ended PINF 0..7
- Reference voltage, AREF = VCC
  - No left shift, no interrupts
- Dummy read to stabilize result
- Reading a value
- Polling mode
  - Start conversion
  - Wait for ADIF flag to go low
  - Want to average the result over several readings
ADCSRA = ADCSRA | ( 1 << ADSC );
while( ! ADCSRA & ( 1 << ADIF ) );
uint16_t val = 0;

for( i=0;i<3;i++) {
    ADCSRA = ADCSRA | ( 1 << ADSC );
    while( ! ADCSRA & ( 1 << ADIF ) );
    
    tmp = ADCL;
    tmp = tmp | ( ADCH << 8 );

    val = val + tmp;
}
return val/3;
Taibotics Educational Robot
Light Sensors

- Three light sensors connected to the Taibotics Educational Robot
  - Left: Port F 1
  - Middle: Port F 2
  - Right: Port F3
- Implement an interrupt driven AD conversion for the light sensors so that you can detect the line
- In automatic conversion mode, the conversion will already have started when the ISR is called
- Changing ADMUX in the ISR will take effect in the next interrupt