

# Questions?

- Project

# Interrupts

- Key points
  - Two levels of interrupt
    - *Today only one level is considered: RCONbits.IPEN = 0;*
  - One general mask
    - *INTCONbits.GIE*
  - One mask for all peripherals
    - *INTCONbits.PEIE*
  - Each source as an Enable bit (PIE registers)
    - *PIE1bits.ADIE for A2D*
  - Each source as a Flag bit (PIR registers)
    - *PIR1bits.ADIF for A2D*
  - Upon interrupt, the execution is transferred at address 0x08 (high vector)
    - *Use Pragma code & interrupt to declare the handler*
    - *Check the Flags bits to see what triggers the interrupt*

# Using Interrupts for A2D

- To before the conversion:
  - Clear PIR1bits.ADIF
  - Set PIE1bits.ADIE
  - Set INTCONbits.PEIE
  - Set INTCONbits.GIE
- Upon conversion:
  - Read the value;
  - Clear PIR1bits.ADIF

# Timer0

- Key idea
  - A counter is incremented at each transition (such as clock cycle)
  - An interrupt is triggered when the counter reaches 0 (i.e. overflows)
- Setting:
  - Select the transition (T0CONbits.T0CS)
  - Select the prescaler mode (T0CONbits.PSA)
  - Set the prescaler value (T0CONbits.T0PS2:0)
  - Select the size (T0CONbits.T08BIT)
  - Load the timer value
    - *TMR0H first, then TMR0L*
  - Clear Timer0 interrupt flag (INTCONbits.TMR0IF)
  - Set Timer0 to trigger interrupt (INTCONbits.TMR0IE)
  - Start timer (T0CONbits.TMR0ON)

# Assignment

- Create a simple blinking light
  - .5s on, .5s off
- Change the A2D program so that
  - The LED is steady if  $LOW\_LIMIT \leq ADRESH \leq HIGH\_LIMIT$
  - The LED blink slowly if  $ADRESH < LOW\_LIMIT$
  - The LED blink rapidly if  $HIGH\_LIMIT < ADRESH$