Timers, Interrupts, and PWM
Timers/Counters

- **Timer** – generate accurate time-based periodic interrupts
- **Counter** – count external pulses (ex. Encoder)

- $TMRx = 16$ bit number
- Increments $TMRx$
Interrupts

• When interrupt is generated (flag), program jumps to the Interrupt Service Routine (ISR)
  – Timer, RS232, I2C, external, etc.
• ISR – section of code outside the main function
• Timer Based Interrupt depends on
  – Period Register (PRx)
  – Timer Prescaler
  – Peripheral Bus Clock
• Priorities
• Example
PWM

- OCx – 5 Pins for PWM Output

// init OC1 module
OpenOC1( OC_ON | OC_TIMER2_SRC | OC_PWM_FAULT_PIN_DISABLE, 0, 0);

// init Timer2 mode and period (PR2) // produces 1ms period
OpenTimer2( T2_ON | T2_PS_1_2 | T2_SOURCE_INT, 0x9C3F);

mT2SetIntPriority( 7); // set Timer2 Interrupt Priority
mT2ClearIntFlag(); // clear interrupt flag
mT2IntEnable( 1); // enable timer2 interrupts
Lab 4 – PIC Control of a DC Motor

• Includes, global variables and constants
• Main
  – Initializations
  – Infinite loop ➔ Empty!
• 1kHz Interrupt Service Routine
  – Read the encoders
  – Calculate position error
  – Set PWM
• Serial Interrupt
  – Receive keystrokes from PC as input
  – Send Encoder data back to PC for analysis