- Time Interfacing

- design example

- design a measurement system for a robot

- we will count the number of rotations for a wheel

- we will count on the rising edge

- we will use timer 0A

- the priority of the interrupt will be 2

- desired resolution

- 1/32 of a turn

- Activate the port (this is for LM3S811)

void initialize\_timer(){

// PD4 is timerA0

SYSCTL\_RCGC2\_R |= 0x08;

GPIO\_PORTB\_DIR\_R &= ~(0x10);

GPIO\_PORTB\_AFSEL\_R |= 0x10;

GPIO\_PORTB\_DEN\_R |= 0x10;

// We have to activate the timer as well, this is using a different register to

// enable. Timer 0 is bit 16

SYSCTL\_RCGC1\_R |= 0x1000;

// timer1=0x2000, timer2=0x4000, timer3=0x8000

// We need to disable timer during setup , Set the TAEN bit to 0 in the CTL register

// The TAEN bit is bit 0

TIMER0\_CTL\_R &= ~(0x01);

// Set the mode for the timer, in this case we are setting it into 16 bit mode

/\* 0x0 = 32 bit config

\* 0x1 = 32 bit real time clock (RTC) counter config

\* 0x2 = 0x3 = Reserved

\* 0x4 - 0x7 = 16 bit timer configuration, function is controlled by bits 1:0 of

\* GPTMTAMR (GPTMTA Mode Register) and GPTMMTBMR

\*/

TIMER0\_CFG\_R = 0x07;

/ GPTMTAMR

// TAAMS, bit 3, if 0 capture mode, else 1 PWM mode

// TACMR, bit 2, capture mode, if 0 edge count mode, else 1 edge time mode

/\* TAMR, bit 1 - 0

\* 0x0 - Reserved

\* 0x1 - One Shot Timer

\* 0x2 - Periodic Timer

\* 0x3 - Capture

\*/

TIMER0\_TAMR\_R = 0x3;

/\* CTL

\* Edge: 0x0 = rising, 0x1 = falling, 0x3 = Both

\*/

// Make the edge we are triggered don a rising edge

TIMER0\_CTL\_R = 0x0;

// Load a start value, in our case we choose the highest 16 bit value

// our timer starts from 0xFFFF and goes down to 0x0000

TIMER0\_TAILR\_R = 0x0000FFFF;

// Enable capture match interrupt

TIMER0\_IMR\_R |= 0x1;

// Clear out any existing interrupts

TIMER0\_ICR\_R = 0x1;

// Now we enable the timer!

TIMER0\_CTL\_R |= 0x01;

// Set the priority for the interrupt

// Timer A is bits 31 - 29 in PRI4

NVIC\_PRI4\_R = 20000000;

// Enable the interrupt. Bit 19 in EN0

NVIE\_EN0\_R = 0x80000;

}

// This function is automatically called whenever there is a rising edge.

void Timer0A\_Handler(){

// Clear out the interrupt

TIMER0\_ICR\_R = 0x01;

// Increase our count

}

count++;

}