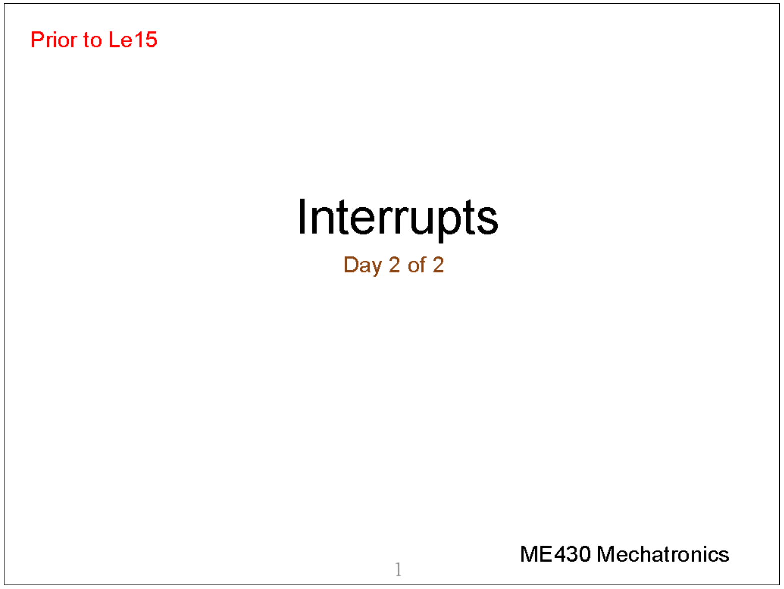
Panel 1



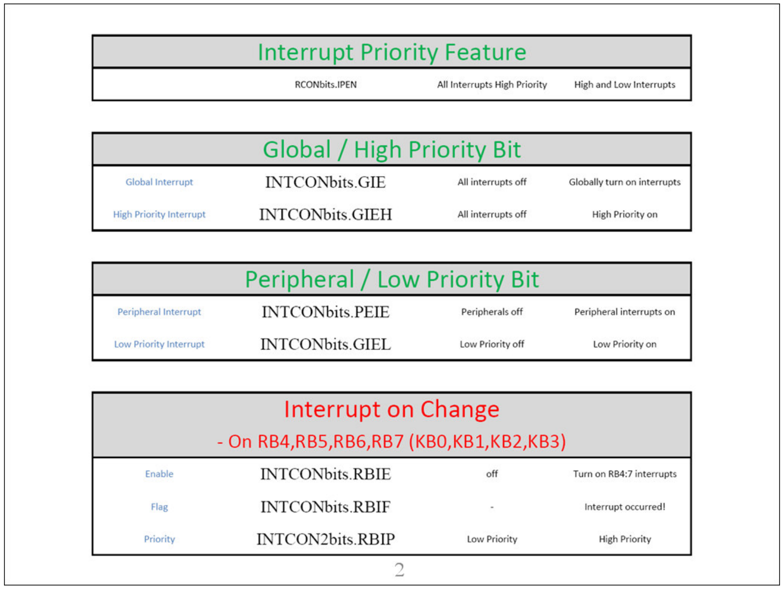
Prior to Le15

Interrupts

Day 2 of 2

ME430 Mechatronics

Panel 2



Interrupt Priority Feature

	RCONbits.IPEN	All Interrupts High Priority	High and Low Interrupts

Global / High Priority Bit

Global Interrupt	INTCONbits.GIE	All interrupts off	Globally turn on interrupts
High Priority Interrupt	INTCONbits.GIEH	All interrupts off	High Priority on

Peripheral / Low Priority Bit

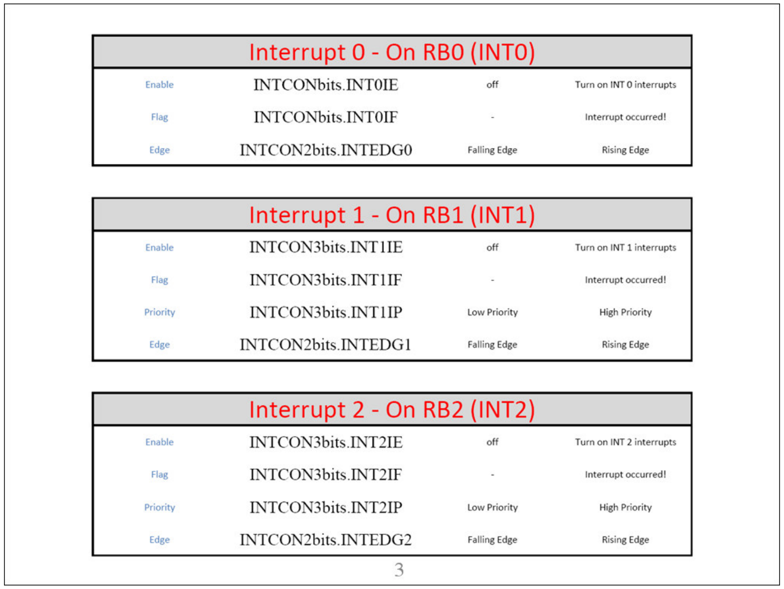
Peripheral Interrupt	INTCONbits.PEIE	Peripherals off	Peripheral interrupts on
Low Priority Interrupt	INTCONbits.GIEL	Low Priority off	Low Priority on

Interrupt on Change

- On RB4,RB5,RB6,RB7 (KB0,KB1,KB2,KB3)

Enable	INTCONbits.RBIE	off	Turn on RB4:7 interrupts
Flag	INTCONbits.RBIF	-	Interrupt occurred!
Priority	INTCON2bits.RBIP	Low Priority	High Priority

Panel 3



Interrupt 0 - On RB0 (INT0)

Enable	INTCONbits.INT0IE	off	Turn on INT 0 interrupts
Flag	INTCONbits.INT0IF	-	Interrupt occurred!
Edge	INTCON2bits.INTEDG0	Falling Edge	Rising Edge

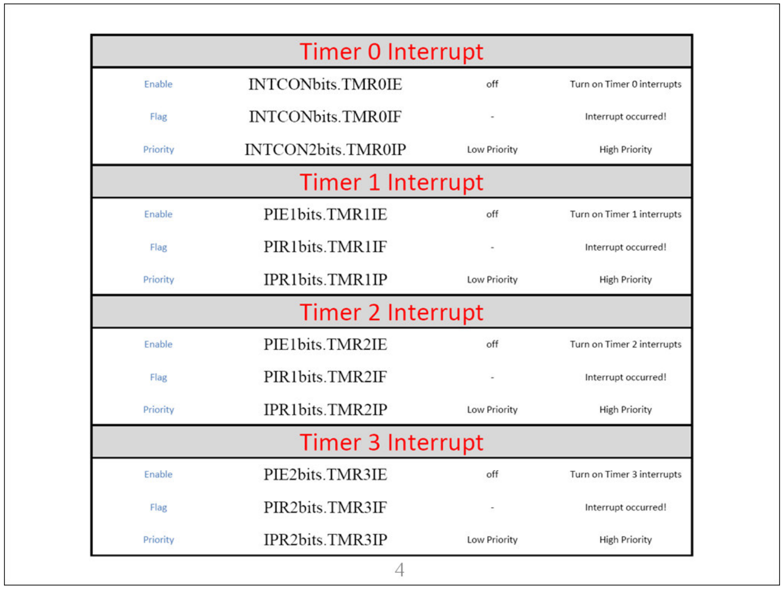
Interrupt 1 - On RB1 (INT1)

Enable	INTCON3bits.INT1IE	off	Turn on INT 1 interrupts
Flag	INTCON3bits.INT1IF	-	Interrupt occurred!
Priority	INTCON3bits.INT1IP	Low Priority	High Priority
Edge	INTCON2bits.INTEDG1	Falling Edge	Rising Edge

Interrupt 2 - On RB2 (INT2)

Enable	INTCON3bits.INT2IE	off	Turn on INT 2 interrupts
Flag	INTCON3bits.INT2IF	-	Interrupt occurred!
Priority	INTCON3bits.INT2IP	Low Priority	High Priority
Edge	INTCON2bits.INTEDG2	Falling Edge	Rising Edge

Panel 4



Timer 0 Interrupt

Enable	INTCONbits.TMR0IE	off	Turn on Timer 0 interrupts
Flag	INTCONbits.TMR0IF	-	Interrupt occurred!
Priority	INTCON2bits.TMR0IP	Low Priority	High Priority

Timer 1 Interrupt

Enable	PIE1bits.TMR1IE	off	Turn on Timer 1 interrupts
Flag	PIR1bits.TMR1IF	-	Interrupt occurred!
Priority	IPR1bits.TMR1IP	Low Priority	High Priority

Timer 2 Interrupt

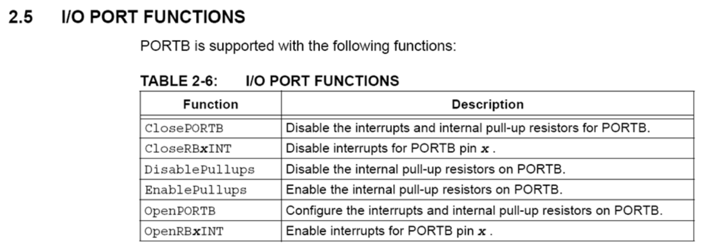
Enable	PIE1bits.TMR2IE	off	Turn on Timer 2 interrupts
Flag	PIR1bits.TMR2IF	-	Interrupt occurred!
Priority	IPR1bits.TMR2IP	Low Priority	High Priority

Timer 3 Interrupt

Enable	PIE2bits.TMR3IE	off	Turn on Timer 3 interrupts
Flag	PIR2bits.TMR3IF	-	Interrupt occurred!
Priority	IPR2bits.TMR3IP	Low Priority	High Priority

Panel 5

Please tell me there are some library functions to help me out…



2.5 I/O PORT FUNCTIONS

PORTB is supported with the following functions:

TABLE 2-6: I/O PORT FUNCTIONS

Function	Description
`ClosePORTB`	Disable the interrupts and internal pull-up resistors for PORTB.
`CloseRB`***x***`INT`	Disable interrupts for PORTB pin ***x*** .
`DisablePullups`	Disable the internal pull-up resistors on PORTB.
`EnablePullups`	Enable the internal pull-up resistors on PORTB.
`OpenPORTB`	Configure the interrupts and internal pull-up resistors on PORTB.
`OpenRB`***x***`INT`	Enable interrupts for PORTB pin ***x*** .

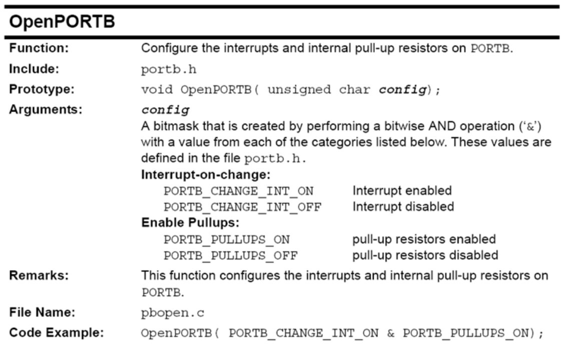
Only help you set the individual enable bits.

Don't help with global setting (GIE or IPEN)

Don't help with flag within ISR

Panel 6

The PORTB Interrupt on change function



OpenPORTB

Function: Configure the interrupts and internal pull-up resistors on `PORTB`.

Include: `portb.h`

Prototype: `void OpenPORTB( unsigned char` ***`config`*** `);`

Arguments: ***`config`***

A bitmask that is created by performing a bitwise AND operation ('`&`') with a value from each of the categories listed below. These values are defined in the file `portb.h`.

Interrupt-on-change:

`PORTB_CHANGE_INT_ON`	Interrupt enabled
`PORTB_CHANGE_INT_OFF`	Interrupt disabled

Enable Pullups:

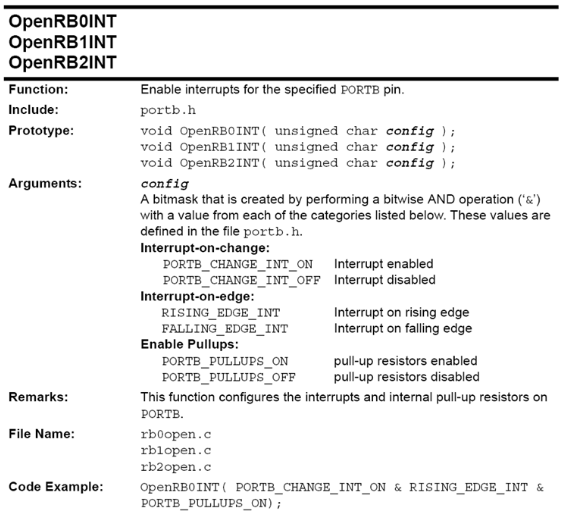
`PORTB_PULLUPS_ON`	pull-up resistors enabled
`PORTB_PULLUPS_OFF`	pull-up resistors disabled

Remarks: This function configures the interrupts and internal pull-up resistors on `PORTB`.

File Name: `pbopen.c`

Code Example: `OpenPORTB( PORTB_CHANGE_INT_ON & PORTB_PULLUPS_ON);`

Panel 7



OpenRB0INT
OpenRB1INT
OpenRB2INT

Function: Enable interrupts for the specified `PORTB` pin.

Include: `portb.h`

Prototype:

```
void OpenRB0INT( unsigned char config );
void OpenRB1INT( unsigned char config );
void OpenRB2INT( unsigned char config );
```

Arguments: ***`config`***

A bitmask that is created by performing a bitwise AND operation ('`&`') with a value from each of the categories listed below. These values are defined in the file `portb.h`.

Interrupt-on-change:

`PORTB_CHANGE_INT_ON`	Interrupt enabled
`PORTB_CHANGE_INT_OFF`	Interrupt disabled

Interrupt-on-edge:

`RISING_EDGE_INT`	Interrupt on rising edge
`FALLING_EDGE_INT`	Interrupt on falling edge

Enable Pullups:

`PORTB_PULLUPS_ON`	pull-up resistors enabled
`PORTB_PULLUPS_OFF`	pull-up resistors disabled

Remarks: This function configures the interrupts and internal pull-up resistors on `PORTB`.

File Name:

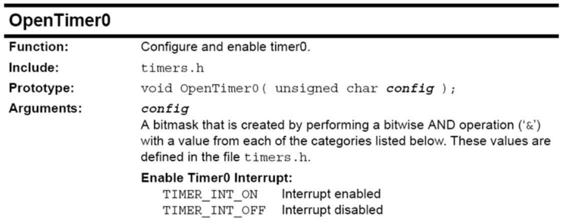
```
rb0open.c
rb1open.c
rb2open.c
```

Code Example:

```
OpenRB0INT( PORTB_CHANGE_INT_ON & RISING_EDGE_INT &
PORTB_PULLUPS_ON);
```

Panel 8

Timer library function to help set that interrupt



OpenTimer0

Function: Configure and enable timer0.

Include: `timers.h`

Prototype: `void OpenTimer0( unsigned char` ***`config`*** `);`

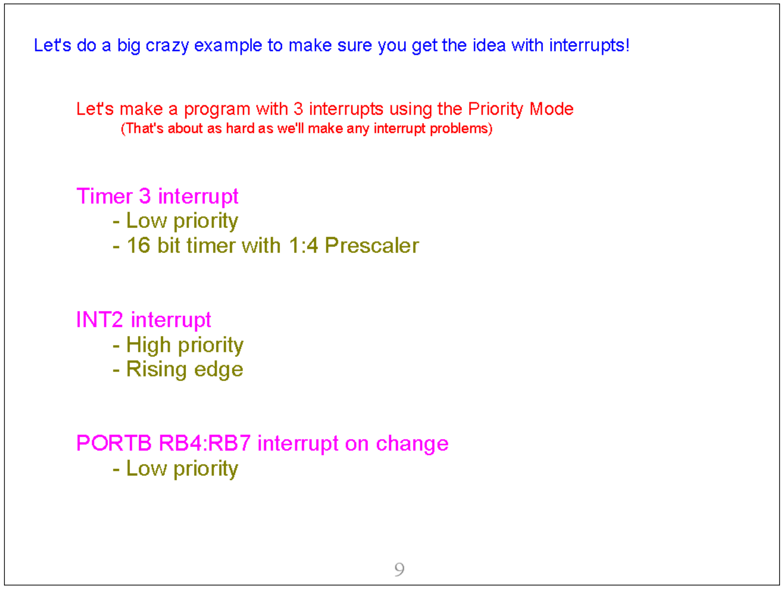
Arguments: ***`config`***

A bitmask that is created by performing a bitwise AND operation ('`&`') with a value from each of the categories listed below. These values are defined in the file `timers.h`.

Enable Timer0 Interrupt:

`TIMER_INT_ON`	Interrupt enabled
`TIMER_INT_OFF`	Interrupt disabled

Panel 9



Let's do a big crazy example to make sure you get the idea with interrupts!

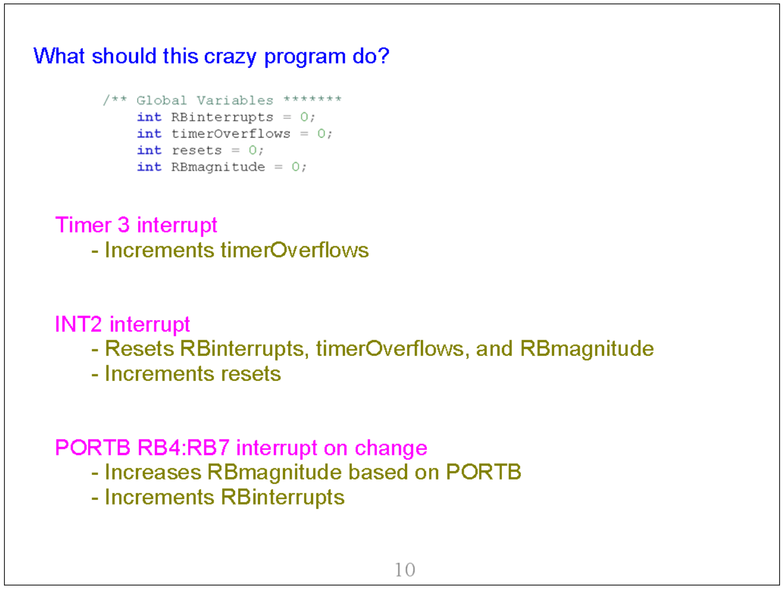
Let's make a program with 3 interrupts using the Priority Mode
(That's about as hard as we'll make any interrupt problems)

Timer 3 interrupt
- Low priority
- 16 bit timer with 1:4 Prescaler

INT2 interrupt
- High priority
- Rising edge

PORTB RB4:RB7 interrupt on change
- Low priority

Panel 10



What should this crazy program do?

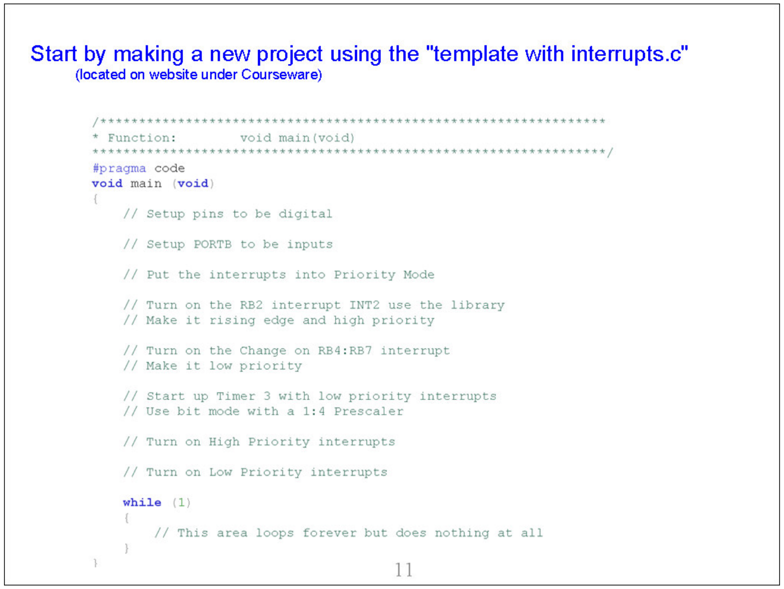
```
/** Global Variables *******
    int RBinterrupts = 0;
    int timerOverflows = 0;
    int resets = 0;
    int RBmagnitude = 0;
```

Timer 3 interrupt
- Increments timerOverflows

INT2 interrupt
- Resets RBinterrupts, timerOverflows, and RBmagnitude
- Increments resets

PORTB RB4:RB7 interrupt on change
- Increases RBmagnitude based on PORTB
- Increments RBinterrupts

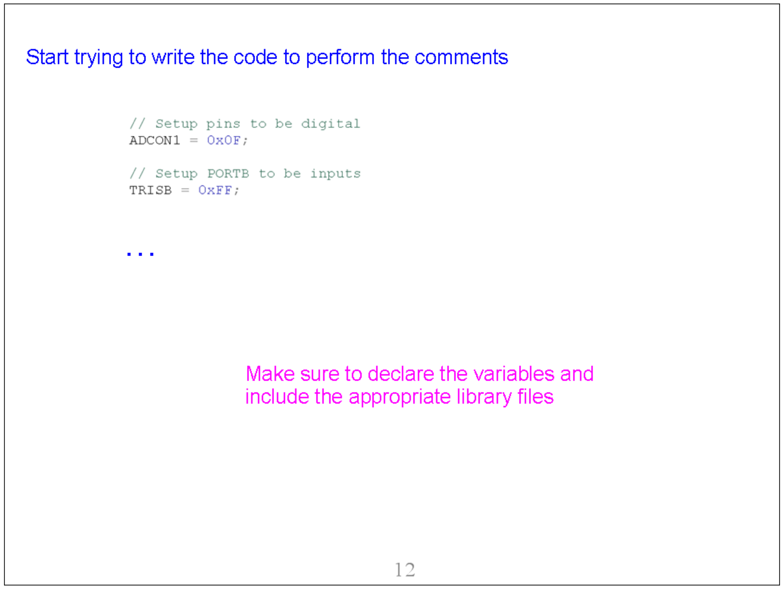
Panel 11



Start by making a new project using the "template with interrupts.c"
(located on website under Courseware)

```
/*******************************************************************
* Function:        void main(void)
********************************************************************/
#pragma code
void main (void)
{
    // Setup pins to be digital

    // Setup PORTB to be inputs

    // Put the interrupts into Priority Mode

    // Turn on the RB2 interrupt INT2 use the library
    // Make it rising edge and high priority

    // Turn on the Change on RB4:RB7 interrupt
    // Make it low priority

    // Start up Timer 3 with low priority interrupts
    // Use bit mode with a 1:4 Prescaler

    // Turn on High Priority interrupts

    // Turn on Low Priority interrupts

    while (1)
    {
        // This area loops forever but does nothing at all
    }
}
```

Panel 12



Start trying to write the code to perform the comments

```
// Setup pins to be digital
ADCON1 = 0x0F;

// Setup PORTB to be inputs
TRISB = 0xFF;
```

. . .

Make sure to declare the variables and include the appropriate library files

Panel 13

Copy in the comments for the High Interrupt Service Routine

```
/*********************************************************************
 * Function:        void high_isr(void)
 * Purpose:
 ********************************************************************/
#pragma interrupt high_isr
void high_isr(void)
{
    // High Priority Interrupt Service Routine (High ISR)
    // See if it was due to Interrupt 2 (it should be)
    // Reset the flag and counters, increment resets
}
```

Try to write the code for these

Panel 14

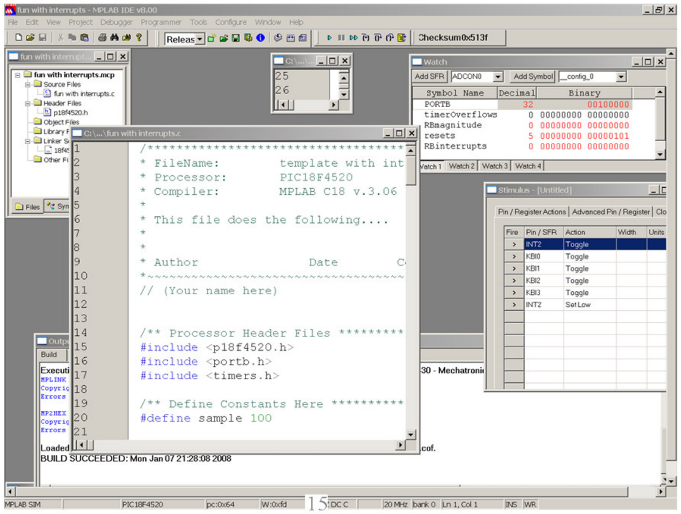
Copy in the comments for the Low Interrupt Service Routine

```
/*********************************************************************
 * Function:        void low_isr(void)
 * Purpose:
 ********************************************************************/
#pragma interruptlow low_isr
void low_isr(void)
{
    // Low Priority Interrupt Service Routine (Low ISR)
    // See if it's due to the timer overflow
    // If it is reset the flag and increment timerOverflows

    // See if it's due the change on RB4:RB7
    // If it is increase RBmagnitude, increment RBinterrupts, reset flag
    // Wierd note: Can only clear this flag AFTER reading PORTB

}
```

Try to write the code for these

Panel 15

Add the Watch window and Stimulus



```
25
26
```

```
/**********************************
* FileName:         template with int
* Processor:        PIC18F4520
* Compiler:         MPLAB C18 v.3.06
*
* This file does the following....
*
*
* Author                    Date        C
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// (Your name here)


/** Processor Header Files *********
#include <p18f4520.h>
#include <portb.h>
#include <timers.h>

/** Define Constants Here **********
#define sample 100
```

Symbol Name	Decimal	Binary
PORTB	32	00100000
timerOverflows	0	00000000 00000000
RBmagnitude	0	00000000 00000000
resets	5	00000000 00000101
RBinterrupts	0	00000000 00000000

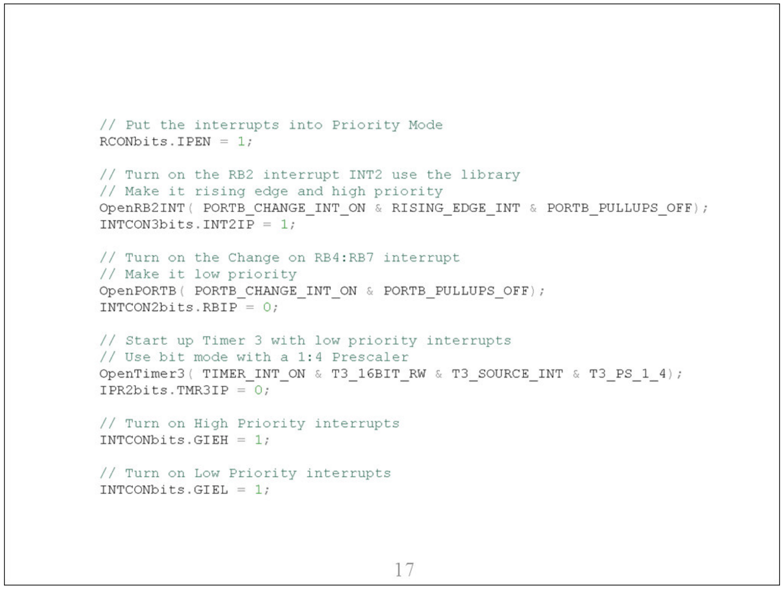
Fire	Pin / SFR	Action	Width	Units
>	INT2	Toggle		
>	KBI0	Toggle		
>	KBI1	Toggle		
>	KBI2	Toggle		
>	KBI3	Toggle		
>	INT2	Set Low		

```
Executi
MPLINK
Copyrig
Errors
MP2HEX
Copyrig
Errors
Loaded                                    .cof.
BUILD SUCCEEDED: Mon Jan 07 21:28:08 2008
30 - Mechatroni
```

Panel 16

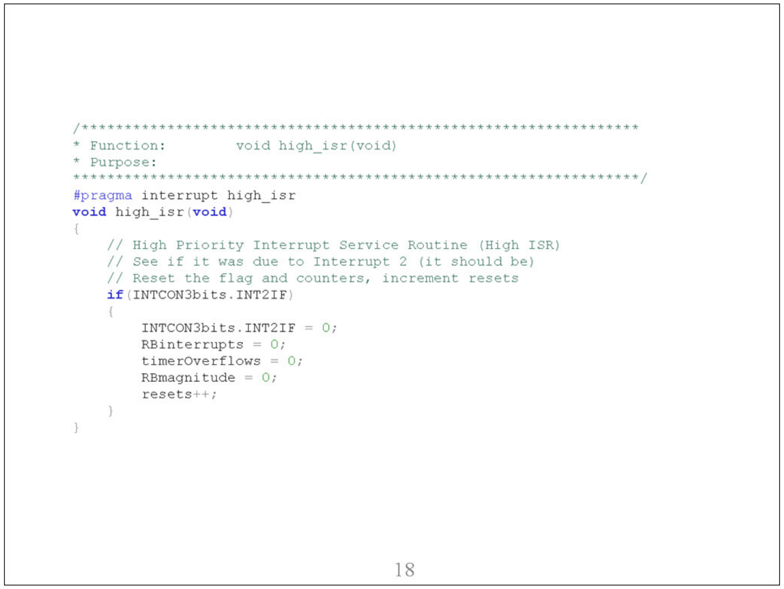
```
/** Header Files ********
#include <p18f4520.h>
#include <portb.h>
#include <timers.h>


            /** Global Variables *******
                int RBinterrupts = 0;
                int timerOverflows = 0;
                int resets = 0;
                int RBmagnitude = 0;



                                    // Setup pins to be digital
                                    ADCON1 = 0x0F;

                                    // Setup PORTB to be inputs
                                    TRISB = 0xFF;
```

Panel 17



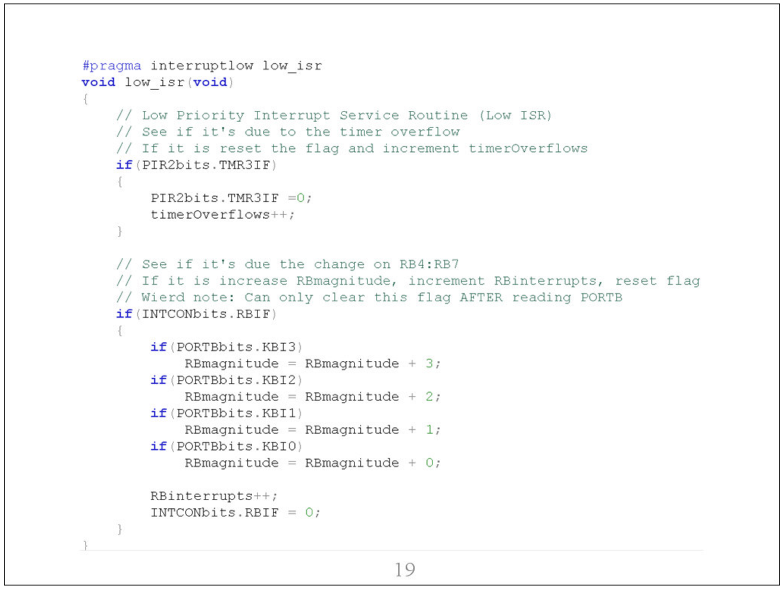
```
// Put the interrupts into Priority Mode
RCONbits.IPEN = 1;

// Turn on the RB2 interrupt INT2 use the library
// Make it rising edge and high priority
OpenRB2INT( PORTB_CHANGE_INT_ON & RISING_EDGE_INT & PORTB_PULLUPS_OFF);
INTCON3bits.INT2IP = 1;

// Turn on the Change on RB4:RB7 interrupt
// Make it low priority
OpenPORTB( PORTB_CHANGE_INT_ON & PORTB_PULLUPS_OFF);
INTCON2bits.RBIP = 0;

// Start up Timer 3 with low priority interrupts
// Use bit mode with a 1:4 Prescaler
OpenTimer3( TIMER_INT_ON & T3_16BIT_RW & T3_SOURCE_INT & T3_PS_1_4);
IPR2bits.TMR3IP = 0;

// Turn on High Priority interrupts
INTCONbits.GIEH = 1;

// Turn on Low Priority interrupts
INTCONbits.GIEL = 1;
```

Panel 18



```
/*******************************************************************
* Function:        void high_isr(void)
* Purpose:
********************************************************************/
#pragma interrupt high_isr
void high_isr(void)
{
    // High Priority Interrupt Service Routine (High ISR)
    // See if it was due to Interrupt 2 (it should be)
    // Reset the flag and counters, increment resets
    if(INTCON3bits.INT2IF)
    {
        INTCON3bits.INT2IF = 0;
        RBinterrupts = 0;
        timerOverflows = 0;
        RBmagnitude = 0;
        resets++;
    }
}
```

Panel 19



```
#pragma interruptlow low_isr
void low_isr(void)
{
    // Low Priority Interrupt Service Routine (Low ISR)
    // See if it's due to the timer overflow
    // If it is reset the flag and increment timerOverflows
    if(PIR2bits.TMR3IF)
    {
        PIR2bits.TMR3IF =0;
        timerOverflows++;
    }

    // See if it's due the change on RB4:RB7
    // If it is increase RBmagnitude, increment RBinterrupts, reset flag
    // Wierd note: Can only clear this flag AFTER reading PORTB
    if(INTCONbits.RBIF)
    {
        if(PORTBbits.KBI3)
            RBmagnitude = RBmagnitude + 3;
        if(PORTBbits.KBI2)
            RBmagnitude = RBmagnitude + 2;
        if(PORTBbits.KBI1)
            RBmagnitude = RBmagnitude + 1;
        if(PORTBbits.KBI0)
            RBmagnitude = RBmagnitude + 0;

        RBinterrupts++;
        INTCONbits.RBIF = 0;
    }
}
```