1 Interrupt Programming tips

1.1 Tip: Initiallize in the proper order

1. configure and enable individual interrupts (but NOT global!)

2. clear flags

3. enable interrupts globally (This MUST be LAST!)

When using interrupts, make global interrupt enabling be the LAST thing in your initiallization section.

1.2 Tip: Check to make sure it's the right flag

When debugging ISRs, be sure to check that the flag you expected is set when you enter the ISR

(Sometimes, entering the ISR clears the flag, but often this is not the case. You want to be sure you're responding to the event you thought you were.)

1.3 Tip: Don't waste time

Interrupt routines should be as quick as possible

Most processing should be in main program

waiting should NEVER happen in ISR!!!

1.4 Tip:Clean up after yourself!

(i.e. Leave everything the way you found it) Since interrupts can happen in between ANY two lines of code, an interrupt routine must not change any important register.

1.5 Tip:Clear flags before returning

In interrupt service routines, make sure to clear flags before returning from interrupts

2 Communication between ISRs and main

2.1 Flags

Problem: The main program needs to know if an interrupt has occurred.

Solution:

```
main clears flag, then loops checking for flag set
ISR sets flag
if main sees flag set, interrupt has occurred.
```

You can have flags for each level of priority, and have main branch depending on which (if any) is set.

2.1.1 Prioritized

```
eg. main:
if 1 flag
jsr process1
jump main
if 2 flag
jsr process2
jump main
:
:
```

2.1.2 Non-prioritized (i.e. round-robin)

```
eg. main:
if 1 flag
jsr process1
if 2 flag
jsr process2
:
:
jump main
```

2.2 Buffers

Problem: The ISR needs to provide data for the main program.

Solution:

```
use flag as above

ISR writes data to TAIL of a buffer (queue);

increment TAIL;

set flag;

main;
loop:clear flag;
 if buffer TAIL<>HEAD, then process
increment HEAD as each value
processed
when TAIL=HEAD, main reads flag
if flag set, go to loop
else reset HEAD and TAIL to initial value
```

There is a slight possibility of a problem here; consider that interrupts can occur anytime; including BETWEEN setting HEAD to its initial value and setting TAIL to its initial value.

Think carefully about how to fix this.

You DON'T want to prevent the interrupt routine from working; you CAN vary what the main program does.

General Programming tip

3 Tip: Never make any block of code more than 1 page long

Use macros, subroutines, etc. to break things down so you can see all of any one "chunk" on the screen at the same time.

As soon as you have to scroll, you lose track of details.