**IR Remote Questions**

1. Will an output pin trigger a pin change interrupt?

Yes it will, disable interrupts when pin change interrupt occurs

1. Can I do timing/delays by doing comparisons on previous TCNT0 values?

**Theory of Operation**

First the I/O registers will be set in the reset routine, then the MCU will be put into sleep mode. When a pin change occurs, this causes an interrupt which will execute the scan routine. This routine will poll the pins to determine which button is pushed and which signal needs to be transmitted. When the button pushed is determined, the output routine will be executed. So, the signal transmits a 9ms start pulse, a preamble code, then the signal code 3 times, followed by a stop bit. Each bit in the preamble and signal code is pulse width encoded, and then the full signal is modulated with a 38kHz carrier wave.

One way to create this signal would be to send each pulse (start pulse and each positive pulse) to a sub-routine that modulates it with the carrier wave. Only positive pulses need to be sent to the modulating routine. The modulating routine must only be executed for the length of the delay (mark delay, one space, zero space). To achieve this, the modulating routine can be executed and stopped by a timer interrupt that occurs when the delay is reached.

Also, another way might be to execute the modulating routine and check for a timer overflow flag during each pass, and exit when there is one instead of using an interrupt.

The carrier wave could also be put within a nested for loop and the loop can be exited when the number of instructions (delay) has been executed.

**Delay Options**

1. Use nested for loops, refer to 4X4 matrix application note.
2. Use timer overflow interrupts, max frequency is 4687.5 Hz. (213.3µs)
3. Use timer overflow interrupts BUT initialize timer value register to a different value (ex. 100), this would increase the frequency of overflow interrupts.
4. Read the value of the timer register and compare it to previous value to check if delay has passed.

I have to pick a solution that would work for 5 different delay values. Maybe use a few?

**Delay option answers:**

1. If I use nested loops, I could change the value for each loop counter corresponding to the delay needed. This nested loop can be in a delay function and take the loop counter values as arguments. However, the carrier has to be modulated onto any high bit/pulse, so this may add lots of instructions within the loop
2. This isn’t possible for the 38kHz needed for carrier wave. I think this is only ideal for really long delays.
3. *If* this even works, the instructions used to initialize the timer value would have to be taken into account.
4. Would have to account for comparison instructions in delay, won’t be as precise because the delay would be greater than or equal to the value wanted.

Nested Loops

* Account for the time to call and return from delay function

**Things to test**

1. To achieve this, the modulating routine can be executed and stopped by a timer interrupt that occurs when the delay is reached.
2. Also, another way might be to execute the modulating routine and check for a timer overflow flag during each pass, and exit when there is one instead of using an interrupt.
3. The carrier wave could also be put within a nested for loop and the loop can be exited when the number of instructions (delay) has been executed.