Roadmap for *knight-rider*  program

At each diode the program beeps with a frequency between 200 and 15kHz.

The diodes are swept up and down(19 steps) at most every half a second.

Along with the diodes the frequency of the beeper is increased exponentially to sweep

From min(poti) to max(poti) in a linear way in log(freq). This will sound best to the ear.

This gives a minimum of 10 iterations and a minimum of 660 instructions at opposite ends of the spectrum.

Approach:

Use tmr2 to time the oscillations for the beeper.

Continually find the 10 iterations average of the 3 potis.

Functions:

* Intialise TMR2 with the Values corresponding to the diode (1-10)
* Recalculate these values in many iterations

IF the item is >10 do the analogue measurements instead

Work with static variables

1. recalculate the increase factor
2. recalculate the current period
3. recalculate the number of cycles
4. find the prescaler value
   1. find the largest postscaler value

22 the remainder is PR2

At the end of the routine check if you have used up more than half the time to rollover

If not, do the next iteration

Item >10 start the A2D for the first poti

Check each time if A2DIF? Is set. YES -> take the value and add it to the sum

All three sums done? Divide by 8

// there should be plenty of time for this. No need to check proc-time or // iterations

Procedure:

LOOP:

Intialise your diode with by checking C0-7 & B4-5

Intialise TMR2 with the appropriate values by calling a subroutine

Start counting down your cycles in a separate variable & loop until both this is down and

* recalc is done

Flop the beeper

Call the recalculation routine

Wait for the TMR2IF to appear, (clear it)

Count down a cycle

Loop

Count your diode up to 19