```
Data_Dump:                      'Dump all 255 Memory locations
for b11 = 1 to 255              'For Each Memory Location
read b11, b12                   'Read the EEPRM brain cell
sertxd (#b11,44,#b12,13,10)     'Print it on the F8 or F9 Screen
toggle 2                        'Make a tick sound
next b11                        'Play it again Sam
low 2                           'Finish up leaving pin 2 in a low state

Read_Voltage:                   'Battery Check when starting up each time
 calibadc10 w1                  'Read a 1.024 volts internal refference voltage.
 w1 = 10230 / w1                'Rough calc b3 to 10'ths of a volt
 goto Output_Data               'Go and Blink out the battery voltage as Volts and 1/10th of
Volt

Data_Logger:                    'Main Data Blinking and Storage Starts Here
for b11 = 0 to 255              'For each memory location
for b10 = 0 to 10               'Read the same value 10 - 25 times.  Repeating increases
duration
                                'AND allows the present value to be observed at a
reasonable rate
                                '10x is good for 'over night (VERY approx).
                                '25x = 24 hours
                                '100x could be for a school week etc
Time_Delay:
nap 8                           'Save lots of power by sleeping between measuring 7=Fast
8=Medium 9=Slow

Input_Readings:                 'Reading Data starts here

Read_ADC_Pin_1:                 'Take a reading
 high 4                         'Enegise the sensor ONLY when needed to save power

 readadc 1, w1                  'Measure voltage ratio on pin 1 into Variable brain cell w1
 input 4                        'DE-Energise sernsor to save power

Output_Data:                    'Sending out the data starts at this point

Debugger:                       'Debug is INVLUABLE if chosen to demonstrate binary /
CS aspects
'debug                          'LOOK at the gears and wheels, bits and bytes on the F6 Debug
Screen
                                'IF enabled and IF programming lead is plugged in. Disable

otherwise
```

```
Math:                           'H+D+U Number crunching time (good example of
Pic-Math
b21 = w1 /  100                 'Hundi Calculate Hundreds, place result into b21 RAM
brain cell
b22 = w1 /  10 // 10        'Tens: Shift right, MOD Divide result by 10 to get remainder of /10
b23 = w1 // 10              'Ones: MOD Divide by 10 to get remainder of /10 I.e. Ones

Sound_Beep_Pitch:               'Set up an audible (shifting pitch) proportional to the data
b0 = w1 // 64 + 64          'b0 is mod divided to get a 0 <> 64 range annd then shifted UP by
64

Hundreds:
do until b21 = 0
sound 2,(b0,50)
dec b21
nap 4
loop
nap 6

Tens:
do until b22 = 0
sound 2,(b0,25)
dec b22
nap 4
loop
nap 6

Ones:
do until b23 = 0
sound 2,(b0,5)
dec b23
nap 4
loop
nap 6

next b10

Store_Data:
write b11,w1
sertxd (#b11,9,#b12,9,#w1,13,10)

next b11
```

end