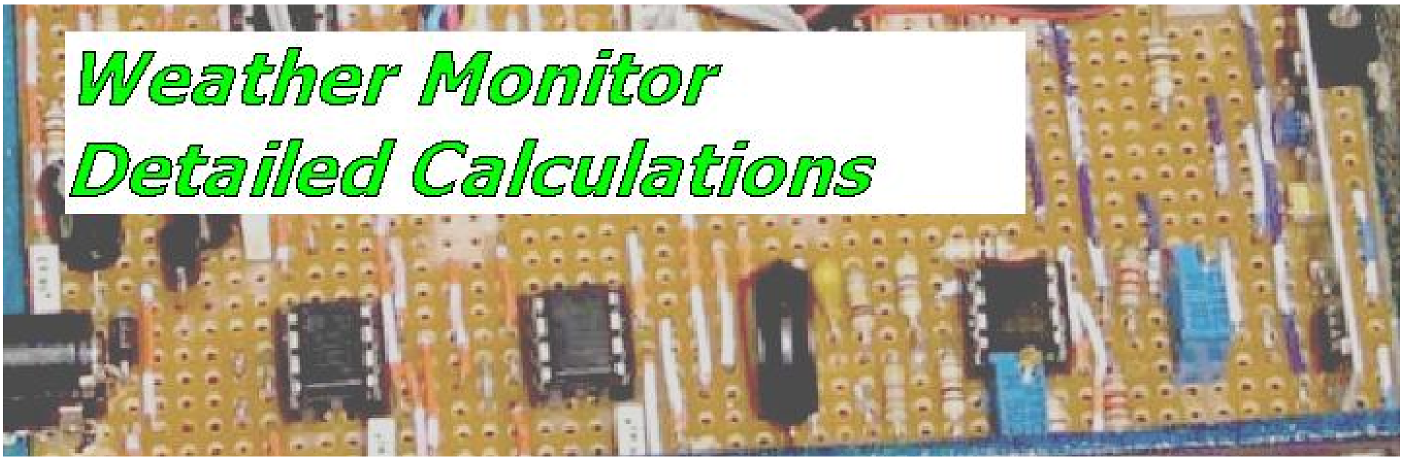


Author: Jeremy Leach Email: ukc802139700@btconnect.com June 2006

Introduction

This document is a supplement to the main Weather Monitor article. It describes the code calculations relating to each sensor in detail.

It should be remembered that these calculations are from a theoretical viewpoint. In reality component tolerances will affect accuracy. However the theory is still a good starting point !

The pressure calculations are pretty complex, which are mainly a result of the relatively simple Op-amp circuitry chosen. But in theory the pressure sensor circuitry should only need calibrating once so don't let the maths put you off. Also it's usually the *trend* in pressure that's important, not the absolute value, and this circuit allows pressure to be tracked to around one tenth of a Millibar resolution.

❖

In this article

Acknowledgements

- PICAXE is a trademark of Revolution Education Ltd.
- Bray's 1-wire barometer: http://www.davidbray.org/onewire/barometer.html
- Site to calculate adjustment of pressure with height: http://hyperphysics.phy-astr.gsu.edu/Hbase/kinetic/barfor.html#c1

Humidity Maths

The HIH3610 sensor output is directly fed into a PICAXE ADC input. See Figure 1.

From the datasheet VOut = Vsupply ((0.0062 * SensorRH) + 0.16)

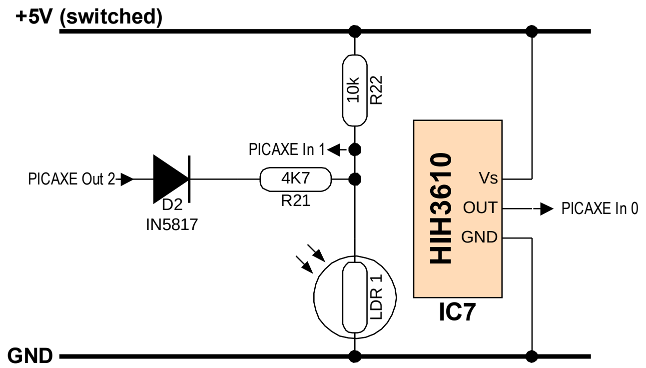
Therefore:

VOut/Vsupply = (0.0062 * SensorRH) + 0.16 **EQN1**

Also considering the ADC input:

VOut/Vsupply = ADC/1023 **EQN2**

Figure 1 : Humidity and Light sensors



Substituting Equation2 in Equation 1:

ADCValue/1023 = (0.0062 * SensorRH) + 0.16

Therefore SensorRH = (ADCValue/1023) * (1/0.0062) - (0.16/0.0062)

SensorRH = (ADCValue * 0.1576) - 25.80646

Modifying this into a usable form for the PICAXE:

64xSensorRH = 10 * ADCValue - 1652

Also from the datasheet:

TrueRH = SensorRH / (1.0546 - 0.00216T)

This is not presently used in the PICAXE code as temperature only has quite a small affect on the Humidity value, and this formula is difficult to implement.

Note:

- VOut = output of the Humidity sensor
- SensorRH = The unadjusted Relative Humidity that the sensor is reading
- TrueRH = The relative Humidity reading adjusted for the affect of temperature
- T = Temperature in Degrees Celsius
- ADCValue = Analogue to Digital Reading (max value is 1023 for 10 bit reading)

Light Maths

LDR response

I was not able to find a datasheet for the miniature LDR I used. However I found information on the Net that indicated that the resistance at 10 Lux should be approximately 20Kohm, and the resistance at 100 Lux approximately 2Kohm.The Dark resistance is approximately 1 MegOhm.

The resistance of LDRs are inversely proportional to the Lux, therefore:

R_{LDR} = C/Lux , where C is a constant and R_{LDR} the resistance of the LDR.

To give the 10Lux and 100Lux values mentioned C must be 200k. Therefore:

R_{LDR} = 200k/Lux **EQN1**

Calculating Lux readings

Figure 1 shows the circuit used. R21 is effectively switched in and out of use by the PICAXE output.

When R21 is not in use, there is a simple relationship:

(ADCValue/1023) * V_{Supply} = (R_{LDR} /(R_{LDR} + 10K)) * V_{Supply}

Therefore 1023 * R_{LDR} = ADCValue (R_{LDR} + 10K)

Therefore 1023* R_{LDR} = ADCValue * R_{LDR} + ADCValue * 10K

Therefore R_{LDR} (1023 - ADCValue) = ADCValue * 10K

Using EQN1:

(200K/Lux)(1023 - ADCValue) = ADCValue * 10K

Therefore ADCValue * Lux = 20460 - 20*ADCValue

Therefore Lux = (20460 - 20*ADCValue)/ADCValue

Therefore:

For the scenario where R21 is not activated:

Lux = (20460/ADCValue) - 20

When R21 is activated, things get more difficult!

D2 is a Schottky diode, which typically has a forward voltage drop of around 0.6V.

If we assign I_1 to be the current flowing through the diode and the 4K7 resistor, I_2 as the current flowing through the 10K resistor and I_3 as the combined current flowing through the LDR, and V_{LDR} the voltage across the LDR, and assume a 5Volt supply :

$I_1 = (5 - V_{LDR} - 0.6)/4K7 = (4.4 - V_{LDR})/4K7$

$I_2 = (5 - V_{LDR})/10K$

$I_3 = V_{LDR} / R_{LDR}$

Therefore $V_{LDR} / R_{LDR} = ((4.4 - V_{LDR})/4K7) + ((5 - V_{LDR})/10K)$

Therefore $(10K*V_{LDR})/ R_{LDR} = 2.127(4.4 - V_{LDR}) + (5 - V_{LDR})$

$= 9.3588 - 2.127*V_{LDR} + 5 - V_{LDR}$

$= 14.3588 - 3.127 * V_{LDR}$

Therefore $10K * V_{LDR} = 14.3588 * R_{LDR} - 3.127 * R_{LDR} * V_{LDR}$

Therefore $V_{LDR} * (10K + 3.127 * R_{LDR}) = 14.3588 * R_{LDR}$

Therefore:

$V_{LDR} = (14.36 * R_{LDR})/(10K + 3.127 * R_{LDR})$ **EQN2**

And $V_{LDR}/5$ = ADCValue/1023

Therefore:

V_{LDR} = ADCValue / 204.6 **EQN3**

Substituting EQN3 into EQN2:

ADCValue / 204.6 = $(14.36 * R_{LDR})/(10K + 3.127 * R_{LDR})$

Therefore ADCValue * 10K + ADCValue * $3.127 * R_{LDR}) = 2938 * R_{LDR}$

Therefore R_{LDR}(2938 - 3.127 * ADCValue) = 10K * ADCValue

Using EQN1:

(200K/Lux)* (2938 - 3.127 * ADCValue) = 10K * ADCValue

Therefore Lux = 20 * (2938 - 3.127 * ADCValue) / ADCValue

Therefore:

For the scenario where R21 is activated:

Lux = (58760/ADCValue) - 62.5

Temperature Maths

The DS18B20 sensor readings are obtained in all cases using the Basic ReadTemp12 Command. A Celcius reading is obtained by multiplying the raw value by 1/16, i.e 0.0625.

Pressure Maths

The output from the MPX4115

From the datasheet, the pressure equation becomes :

$V_{Out} = V_{Supply}$ (P * 0.0009 - 0.095) **EQN1**

where P is the atmospheric pressure in Millibars.

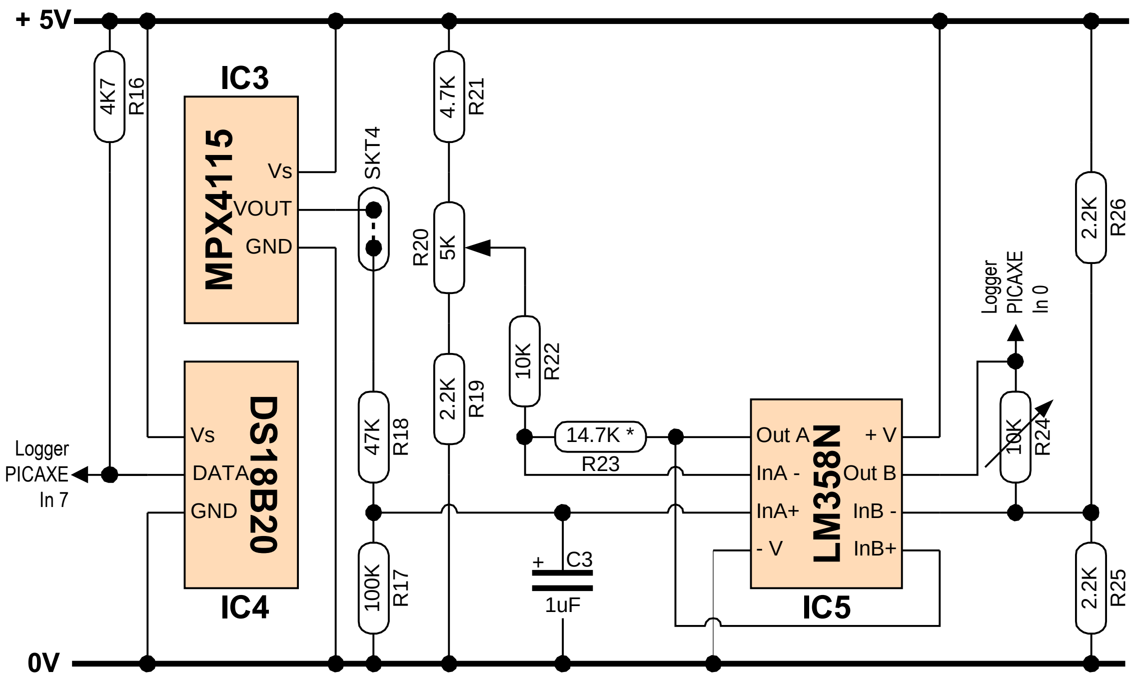
Normal Sea-Level pressure range is 104 to 97.2 kPa = 104000 to 97200 Pa = 1040 to 972 Millibars (Note: 1bar = 10^5 Pa, 1mbar = 100Pa).

So absolute pressure range for a location can be calculated using this useful site:

http://hyperphysics.phy-astr.gsu.edu/Hbase/kinetic/barfor.html#c1

The height of my house is 65m, or 213feet above sea level. So, for me, this will give a voltage range for V_{Out} from MPX4115, assuming 5 Volt supply, of :

Figure 2 : Local temperature and pressure sensors



5 * (1032 * 0.0009 - 0.095) to 5 * (965 * 0.0009 - 0.095) = 4.169V to 3.8675V

V_{Out_High} = 4.169V

V_{Out_Low} = 3.8675V

Conditioning by the resistor divider

Figure 2 shows the circuit I've used. The output from the MPX4115 is fed via a resistor divider R17 and R18 into OpAmpA.

The gain of the resistor divider = R17/(R17 + R18) = 100K / 147K = 0.6802

Therefore the range of input voltage to InA+, for the normal pressure range is(0.68 * 3.8675V) to (0.68 * 4.169V) = 2.630V to 2.835V.

The midpoint of this range = 2.937V.

Single Supply OpAmp maths

In order to work out the maths for both OpAmp stages, it's first necessary to have a detour about bias-voltage.

Both OpAmps in this circuit use potential dividers to supply a bias voltage and are configured as in Figure 3. This single-supply design is complex to calculate because the voltage at the centre of the potential divider varies.

A simplification of the scenario (with resistor R1b = 0) is shown in Figure 4 Analysis of this simple circuit shows:

$I_3 = I_2 + I_1$ **EQN2**

Also,

$(V_1 - V_2) / R_1 = I_1$

and

$(V_{Supply} - V_2)/R_2 = I_2$

and

$V_2/R_3 = I_3$

Using Equation 2:

$V_2/R_3 = (V_{Supply} - V_2)/R_2 + (V_1 - V_2) / R_1$

Which reduces to:

$V_1 = V_2[(R_1R_2 + R_1R_3 + R_2R_3)/ R_2R_3] + [R_1R_3 * V_{Supply}/ R_2R_3]$

Therefore:

$V_1 = V_2[(R_1R_2 + R_1R_3 + R_2R_3)/ R_2R_3] + [R_1 * V_{Supply}/ R_2]$

This is a straight-line equation of the form:

Figure 3 : Single supply circuit

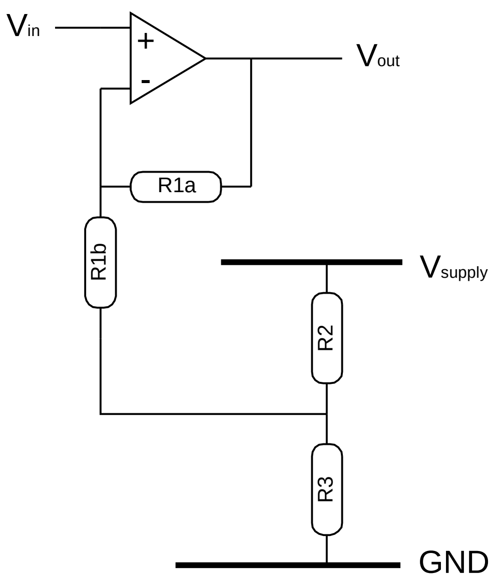
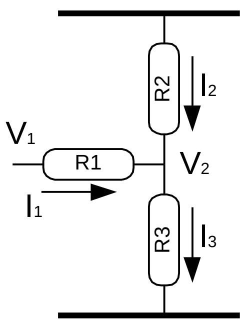


Figure 4: Simplified scenario



$V_1 = MV_2 - C$ **EQN3**

Where

$M = (R_1R_2 + R_1R_3 + R_2R_3)/ R_2R_3 =$

$M = R_1/R_3 + R_1/R_2 + 1$ **EQN4**

And

$C = R_1/R_2 * V_{Supply}$ **EQN5**

It's fairly obvious that because this is a straight-line equation, the amplifier stages will have a linear response to input voltage.

Maths for OpAmpA

Figure 2 shows that this OpAmp has a more complex arrangement than in Figure 4, because R1b is not zero. However it is straightforward to analyse:

Remembering that when an OpAmp is used as an amplifier, the input voltages are essentially equal, due to the high internal gain. Therefore the voltage at the junction of R1a and R1b = V_{in}. And R1a + R1b = R1.

So:

$V_{in} - V_2 = R_{1b} / R_1 * (V_{out} - V_2)$

Therefore:

$(V_{in} - V_2)* (R_1 / R_{1b}) + V_2 = V_{out}$

Or:

$V_{out} = V_{in}* (R_1 / R_{1b}) + V_2* (1 - R_1/R_{1b})$ **EQN6**

Using Equation 3 to eliminate V_2 eventually gives this equation relating V_{out} to V_{in}:

$\mathbf{V_{out} = [(V_{in}* R_1) - (C * R_{1a}/M)]/ (R_{1b} + R_{1a}/M)}$

Maths for OpAmpB

Figure 2 shows that R1b is zero in this case, and so :

$\mathbf{V_{Out} = MV_{in} - C}$

Overall effect of the OpAmp stages

These forumlae are used in the *BarometerSetup.xls* workbook available on my website. This calculates the predicted voltages at each stage of the circuit for different variable resistance values.

A pair of variable resistor values can be chosen to give the desired output voltage range. The real voltages can then be measured and compared to the theoretical values and the pots adjusted accordingly.

The Barometer output should then be compared to local weather station output, and the variable resistor values fine tuned. Please note that at the time of writing I have not actually done this final step. It may be quite tricky ! Further information on the general principle is covered on David Bray's site mentioned in the Acknowledgements.

Wind Speed Maths

At present I've no calibrated my anemometer, so I've just put an initial guess at wind-speed in my code. This is obviously going to be wrong ! I need to calibrate my anemometer by, say, driving along at fixed speeds with it held out of the window. Then have a lookup table in code, or a formula if the relationship is linear.

Anyway, here's the basis of my calculations at present:

The cups of my anemometer lie on the circumference of a circle with radius = 5cm. Circumference is therefore 2 * Pi * 5 = 31 cm.

If there was no drag at all then we could assume the cups rotated at the speed of the wind. Therefore:

Windspeed = RevsPerMinute * 31 cm per minute

= RevsPerMinute * 0.3 * 60 metres per hour

= RevsPerMinute * 18 / 1609 miles per hour (Note:1 mile = 1609 metres)

But there will be drag. So assume that the real windspeed is 20% faster than the cup rotation speed.

Therefore:

Windspeed = RevsPerMinute * (12 * 18) / 16090

= RevsPerMinute * 216 / 16090   mph

And we want to display to 1 decimal place, therefore:

WindSpeed10 = RevsPerMinute * 216 / 1609 mph

Which is approximately:

WindSpeed10 = RevsPerMinute * 13 / 100 mph