# OWL2pe and BASIC Stamp to anemometer




**Contents (updated 6/03/2004)**



### Annemometer (rotating cups with reed relay signal) top

Here is data from the popular Davis/Digitar unit:

- The Digitar 7911 or 7913 or produces one pulse per revolution of the cups.
- The cups turn 26 rpm (26 pulses per minute) for each mph of windspeed...
- 0.4333 hz (pulses per second) per mph.
- sixty hertz at 138.5 mph windspeed.
- Starting threshold is 2-4 mph

The annemometer is connected to a general purpose i/o pin on the OWL2c with a pullup resistor (2k) to +5.12 volts switched. Use the swiched supply so that the current won't add up if the cups happen to stop right at the point where the reed switch is closed. Windspeed counting is only done when the OWL 2c is awake and the power supply is turned on. Another possibility is to tie the 2k resistor to another i/o pin, instead of to the power supply. That way, the main power does not have to be turned on for the frequent measurements needed for windspeed averaging.

We'd like to count the number of pulses in a time interval, which is directly proportional to windspeed. Counting for 2.308 seconds will give a result directly in units of mph,

```
60hz * 2.308seconds = 138.5mph
```

Here is how to implement this on the BS2, OWL2c data logger.

```
 wind var byte
 count wpin,2308,wind
```

Usually the data needed will be the average wind speed for some period of time, and the maximum over that same time, or other statistics. The following routine has one part that executes frequently, and another part that executes only when data is logged into the file:

```
      wind     var   byte
      windacc  var   word    ' accumulation for average
      windmax  var   byte    ' maximum
      Nsmpl    var   word    ' number of samples
     ' frequently, e.g. once per 10 seconds
       count wpin,2308,wind
       windmax = wind min windmax ' maintains maximum windspeed
       windacc=windacc + wind     '
```

```
      Nsmpl=Nsmpl+1               ' one more sample

    ' infrequently, only when it is time to log data
      result=windacc/Nsmpl        ' compute average
      gosub logdata               ' log it
      result=windmax              ' fetch maximum
      gosub logdata               ' log it
      windavg=0                   ' reinitialize
      Nsmpl=0
      windmax=0
```

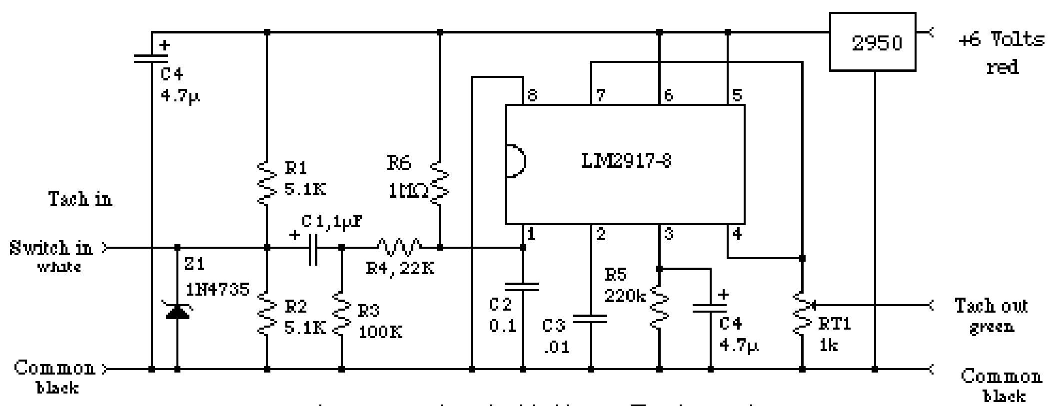
Here are a couple of other conversion factors, for other units and resolutions:

```
    count wpin,1154,wind      ' for 2 miles per hour
    count wpin,4616,wind      ' for 0.5 miles per hour

    count wpin,1032,wind      ' for 1 meter per second
    count wpin,2064,wind      ' for 0.5 meter per second

    count wpin,1858,wind      ' for 2 kilometer per hour
    count wpin,3716,wind      ' for 1 kilometer per hour
```

Windspeed tends to be sporadic, and the distribution highly skewed. Infrequent samples (e.g. hourly) of windspeed tend to be quite low, compared to the average or maximum. It is best to sample the windspeed frequently.

### Wind interface using an analog Tachometer

Here is a circuit that can convert the output of the Davis anemometer (Or any anemometer that uses a switch-type output), into a voltage for input to an analog to digital converter. The output of this circuit can be calibrated to give 1 volt output at 100 miles per hour by adjusting RT1. The network of resistors and capacitors at the input is there to turn the on-off switch action into a series of bipolar pulses for input into the LM2917 tachometer chip (National Semi). Also, the circuit at the input protects the circuit from lightning induced surges. Anemometers are available that have a AC generator, and these produce an AC sine wave signal. Those anemometers can also use this circuit, but remove R1, R3 and R6 and replace C1 with a jumper.




Anemometer to Voltage Tachometer.
Tracy Allen, EME Systems, (c) 1992

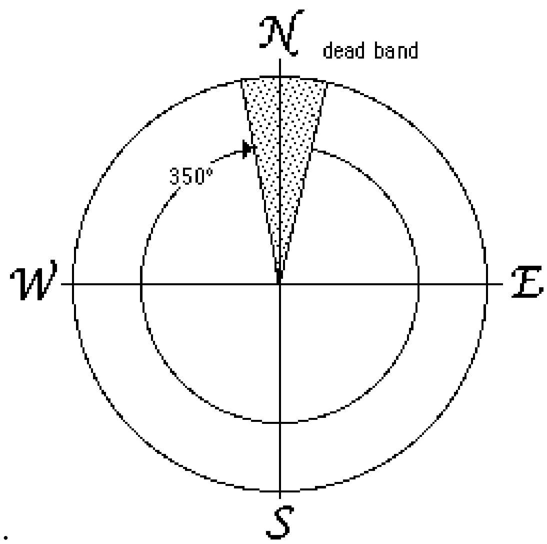
---

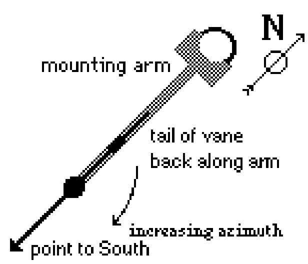
### Wind direction vane

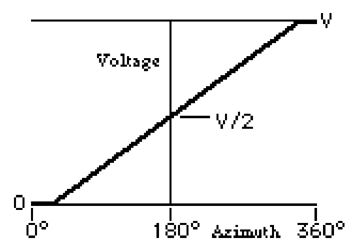
top

The wind vane pot connects between +5.12 volts switched and common, and the wiper goes into one of the OWL2c input channels. The potentiometer produces a voltage between 0 and 5.12 that the software converts to azimuth. The figure shows the recommended mounting, with the wind vane arm pointing south out from the mast. The vane is installed so that the voltage reading is V/2 when the vane is pointed due south, with the tail of the vane back along the mounting arm. There may be physical constraints or factors having to do with the previaling winds that demand a different mounting direction. If so, the offset may be easily supplied in software as discussed below.

The potentiometer in the wind vane is free to move through 360 degrees, but there is a dead band of about 10 degrees at the crossover point in the potentiometer travel. As shown in the graph, the output signal is zero volts for 5 degrees before the signal starts its linear rise, and the signal is maximum voltage through 5 degrees at the other extreme of the travel. The dead band varies some from unit to unit. If you are looking for the greatest possible accuracy, then you will need to find out exactly what the dead band is. You can also find out exactly what the dead band is by cutting a hole and a slit in a sheet of polar graph paper and putting it around the vane, and taking readings of w1 as a function of position of the vane in degrees. If you do not need great accuracy (5%) then simply use the typical value, 10 for the dead band calibration factor.




The following formula in the OWL2, BASIC Stamp, converts the reading from millivolts to azimuth:

```
WDead   con 10                  ' dead band in degrees
WDead2  con WDead/2             ' half the dead band in degrees
WDfs    con 360-WDead           ' full scale span
gosub ADread                    'result is 0-5120 millivolts
azimuth = result */ WDfs / 20 + WDead2
```

The formula for azimuth produces in readings that go from 5 degrees to 355 degrees. The reading in the dead band is 5 degrees from zero to 5 degrees azimuth, and 355 degees for 355 to 360 degrees azimuth.

{The */ operator is described in the document on math operations. Here the millivolt reading is being multiplied times the fraction WDfs/5120, where 5120 millivolts is full scale. The BS2 keeps the calculation in range by breaking it up into WDfs/256 * 1/20. The */ simultaneously performs the multiplication by WDfs and division by 256, without integer overflow. Then comes the division by 20 to finish up.}

If you only need 16 compass points, then you can use a simpler formula. Wind vanes often don't require much accuracy, and in such cases, 16 compass points may suffice. In studies of microclimate and drift, however, the accuracy needs to be 1% or better.

```
compass = result + 160 // 5120 / 320
```

16 compass points come out as 320 millivolts per compass point. The "+160" offsets the result, so that compass point zero comes out at 22.5 degrees on each side of north, and so on, the compass points are centered on the 16 cardinal directions.

When you put the vane up on a pole, you may have to orient it with the dead band pointing in a different direction. Then you only need to enter that direction in the calculation as an offset::

```
WDead   con 10                       ' dead band in degrees
WDead2  con WDead/2                  ' half the dead band in degrees
WDfs    con 360-WDead                ' full scale span
WDoff   con 105                      ' offset=105 degrees (example)
gosub ADread                         'result is 0-5120 millivolts
azimuth = result */ WDfs / 20 + WDoff +WDead/2 // 360
```

In this example, the azimuth goes from 110 degrees, smoothly through 360 to zero at North azimuth, and then back to 100 degrees at the other side of the dead band, and jumps back to 110 degrees at the midpoint of the dead band.

---