Ultra-low-cost Logging Anemometer for Wind Power Generation Feasibility Surveys

Michael Cosgrove, Bruce Rhodes and Jonathan Scott
Department of Engineering, The University of Waikato
Private Bag 3105, Hamilton, 3240, New Zealand
Email: jbs@waikato.ac.nz

***Abstract*— We describe the design and implementation of a cup anemometer capable of logging average wind speed, maximum wind speed and seconds with wind above a criterion speed, on individual weeks. The instrument will operate for at least 54 weeks on two AA dry cells. The intention is that this instrument be deployed to ascertain whether it would be economical to install a wind generator, at a fraction of the cost of a multipurpose weather data logging station. It is designed around a $2 microcontroller that provides non-volatile memory. Provision has been made for using either a magnetic reed switch or the motor from a discarded hard disk drive to sense rotation. The latter enables the use of an otherwise worthless bearing and drive assembly to provide an especially frugal solution.**

I. Introduction

Small-scale wind generators are now available in shops with prices ranging from a few hundred to a few thousand dollars. With the exception of especially inaccessible or known high-wind locations, it is not clear in advance of most installations whether a generator would be cost effective in comparison to grid connection or solar panels. A logging weather station can be used to carry out a survey, but may represent a substantial fraction of the cost of deploying a small wind generator in the first place, and these units are not generally designed to be used in the absence of power and connectivity. Global databases of prevailing wind conditions are rarely useful in the case of small installations, since microclimates and local terrain alter wind exposure of small-scale sites a great deal, while surveys target commercial-sized installations that use wind at heights from 10 to 200 metres.[1], [2]

There are a number of professional logging air-speed meters on sale. They typically cost $400 or more, and at least another $100 with proof of traceability if required.[4] A low-cost logging anemometer has been described, using similar technology to the design in this manuscript, but it is not designed for free-standing, self-contained operation, nor is it designed to run for extended periods on small dry cells.[3] That design logs direction as well as speed from the same cup-type sensor, using an ingenious asymmetry in the cups and cyclic variation in shaft speed. No establishment of its accuracy or traceability is presented.

Our solution to this problem is a small anemometer of the cup type commonly seen on domestic weather stations and yacht masts.[5] The design is based on a Microchip PIC12F683 microcontroller costing a little over US$1.[1] [6] The design uses only readily-available electronic and mechanical parts. The whole device is small, will operate for at least 1 year on a pair of AA dry cells, and can store enough data within the microcontroller in non-volatile EERAM to permit assessment of the economic viability of power generation from wind. While an anemometer of the thermal or ultrasonic type with no moving parts might be easier to construct, other designs consume significant power, and we require low power consumption. Additionally, we allow for the possibility of constructing the mechanical bearing assembly using parts from a discarded hard disk drive (HDD) and sensing rotation using the signal from the motor incorporated with the excellent bearings in these devices. Given the large number of failed and obselete HDDs found these days, and the high-quality bearings in the platen drive, this alternative makes the design especially accessible to the handyman, although power consumption is greater when using this sensing mechanism.

An analog wind speed meter has been described that uses a small motor-bearing assembly in generator mode as the sensor. Although not explicitly stated, it must be intended that HDDs be used as the source of the Permanent-Magnet AC (PMAC) motor that features in this design.[7] The version described in [7] cannot log data.

II. Hardware

The circuit diagram is shown in figure 1. The power supply is a pair of dry cells, giving a nominal 3 V supply. The circuit will operate with a cell endpoint voltage of 1 V and with typical all-inclusive operating current in the region of 300 μA in modest to low wind conditions. Power consumption rises to 750μA in high wind speeds. Power consumption is sufficiently low that it is possible to run for 1 year using Alkaline cells.[2]

[1]The Microchip PIC16F684 can be substituted with only minor code changes. The '684 is identical except that it has 14 rather than 8 pins. Although the 6 extra pins are unnecessary, the larger device is supported by the free PICC-Lite C compiler from Hitech. Pins 6, 7 & 8 are substituted by pins 12, 13 & 14 on the '684. Some frills need to be dispensed with, as the free compiler can use only half the memory space. This substitution may be of use to the frugal experimenter.

[2]Readers with experience in dry cell powered electronics will be aware that prediction of dry cell capacity for load currents below a milliamp or so is difficult. Apart from the impact of shelf life, ambient temperature, and variations in chemistry that may not be reflected in cell labels, manufacturers do not specify performance for low currents yet capacity increases with decreasing current for any given endpoint. For this reason we are not numerically specific, but capacities of several Ampere-hours are typical.

The circuit senses cup rotation by one of two methods, either the closure of a reed switch by a rotating magnet or through the millivolt signal emitted from a permanent magnet motor. In the former case, one port of the microcontroller is configured as a digital input, and another as a digital output to provide pull-up when required, to save power. In the latter case, one port is disabled and the other acts as an analog input.

A light-dependent resistor (LDR) allows the circuit to sense daylight, and so to detect dawn. Operation is diurnally synchronised, removing the need for precision timekeeping. The LDR circuit can also detect a button-push allowing a user to trigger calibration, or temporarily activate analog wind speed readout during setup. Although it is intended that logged data be retrieved through the in-circuit serial programming interface (ICSP), the push-button switch could also be used to trigger readout of the log.

The circuit uses an LED to signal a user during calibration and setup, to provide a heartbeat flash every 10 seconds, and the same circuit allows the circuit to sense battery voltage through measuring the LED voltage with respect to the supply voltage. The heartbeat rate halves when the battery begins to reach the end of its life. Finally, through pulse-width modulation of the LED, the circuit can control LED brightness and provide approximate analog readout on an attached analog panel meter or digital multimeter (DMM).

A. Prototype Mechanical Hardware

A picture of a prototype PCB using a PIC12F683 and a reed switch sensor is shown in figure 2, while figure 3 shows a version that uses an old HDD spindle motor and can accomodate either a PIC12F683 or a PIC16F684. When using a reed switch, it is important to mount the switch so as to have a single closure per revolution of the cups, or to ensure closures are equispaced. If the magnet is very strong, the switch should be mounted so that it cuts the axis of rotation, resulting in one closure occurring as the magnet sweeps past each end of the switch. The speed sensing algorithm allows for such inequality if it occurs. However, extremely unequal delays to closure could fool the debouncing algorithm at higher wind speeds.

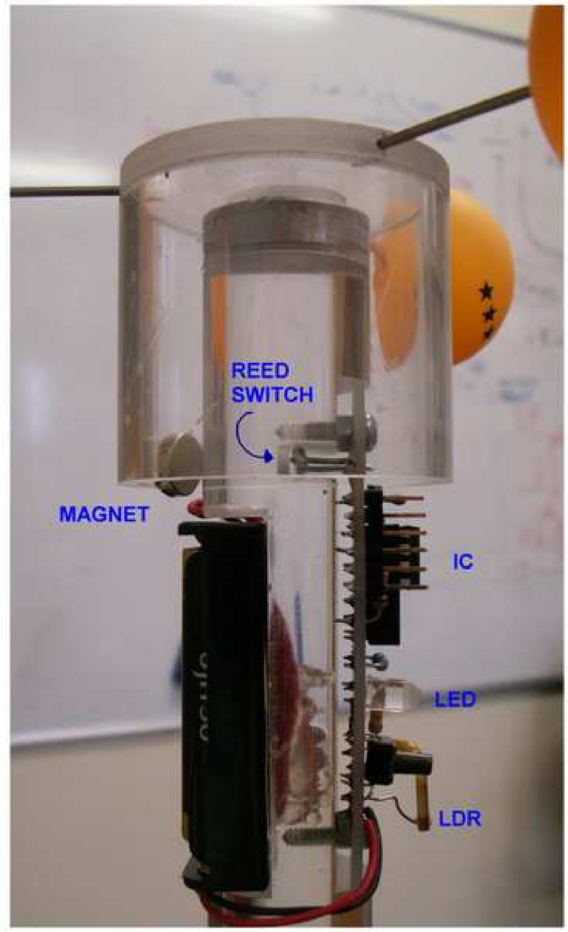

Fig. 2. Photograph of a prototype constructed in transparent acryllic and using a magnetic reed switch. The PIC12F683, LDR and LED can be seen on the PCB. Note also the ICSP header for programming and data retrieval. The reed switch is mounted below the PCB, in a channel milled in the central column so as to intersect the axis of rotation and deliver two equispaced closures per rotation of the cups. The magnet is a commercial button type, mounted on the shroud that carries the cups.

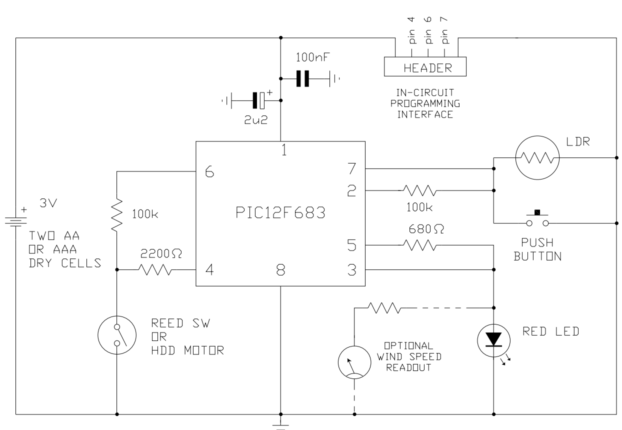

Fig. 1. Circuit diagram of the logging anemometer.

Our “typical” prototype has been constructed using standard plumbing parts, a commercial ball-bearing, and ping-pong balls. It is cheap, durable, and repeatable. Figure 4 shows a prototype mounted on a vehicle for testing purposes. The housing is standard 50mm PVC pipe. The rotating cap is a plumbing fixture intended to cap 65mm pipe. The external ring of a 25mm ball-bearing force-fits into a plumbing adapter that is in turn inserted into the 50mm pipe, and the end-cap is mated to the inner race of the bearing by means of a small post. The reed switch is attached to the PCB inside the 50mm pipe, and a thin magnet is attached to the rotating end cap. The cups are halves of ping-pong balls glued on stainless steel wires with the centres of the cups 65mm from the bearing centreline. The unit in figure 4 has an inordinately long pipe for ease of mounting in the wind stream above turbulent layers.

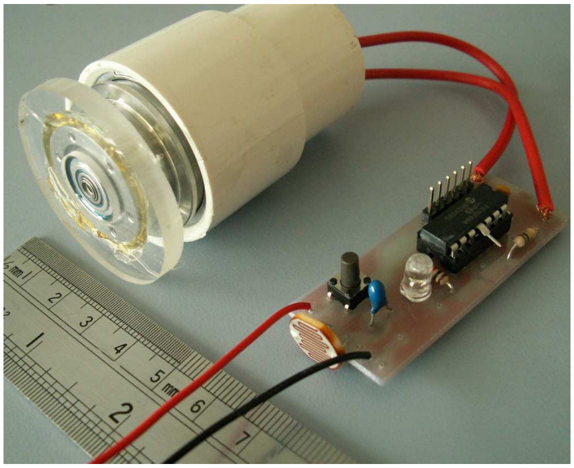

Fig. 3. Photograph of a version using an HDD motor as the sensor bearing and pickup unit, and a printed circuit board that can accomodate either a PIC12F683 or a PIC16F684. The unit is shown with the latter version installed: Note the lifted pin that allows the same socket to accomodate the PIC12F683 directly.

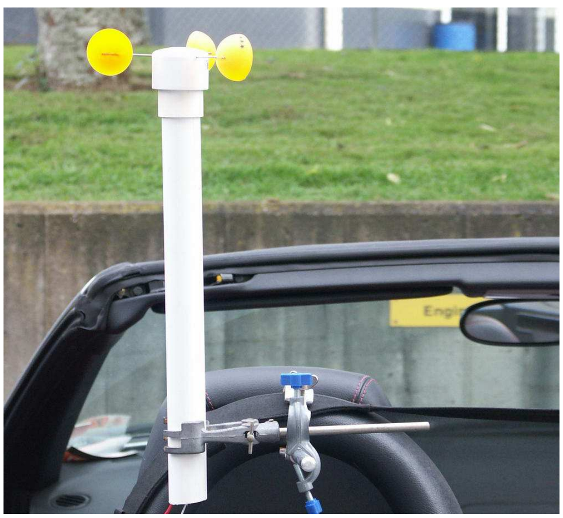

Fig. 4. Prototype anemometer with extended-tube housing mounted by means of retort clamps on a vehicle for testing. Turbulence error was reduced when mounted more to the front of the vehicle.

III. Capabilities and Usage

It is known that annual mean wind speed is a decent indicator of the wind energy potential of a site.[1] However, wind power flux is proportional to the cube of wind speed, and power availability is governed by short-term wind activity. These considerations imply that to be more useful, a wind survey should have more than simply a mean value over the annual cycle.

The logging anemometer described here will record the peak wind speed, the average wind speed, and the number of seconds for which wind speed exceeded a preset criterion, for each of the preceding 54 weeks. It also records the same data for each day in the most recent week of operation, for short-term situations. The first committment of data to NV memory occurs at dawn after the first partial day of operation. Detection of dawn requires at least two hours of operation in darkness. The program also records the mean and mean-cube of wind velocity over the integral number of days in the whole logging period.

The speed criterion value is set at programming time. Typically this value is set to suit a particular wind generator, and will be in the range of 8–15 kmh. For example, one popular make of wind turbine is specified to generate power with wind speeds exceeding 7.5 mph, or 12 kph.[9] The default calibration factor that relates wind speed to anemometer rotation frequency is set at programming time. This is suitable for a given hardware situation. However, the calibration factor can be reprogrammed in situ, to allow for the use of uncalibrated hardware. The method is described in section V.

In normal operation the logger need simply be mounted in a suitable position and left for 1 year. At the end of that time it is demounted and the batteries removed. The NVRAM can then be queried and the data extracted. The same cheap programmer used with the microcontroller can read the EERAM data.[6] Since the logger records 54 weeks of data plus the last few days of data beyond an integral number of weeks, there is some freedom in the required demounting time while still guaranteeing that a number of units will present comparable data. In other words you can be a few days late in picking up some units at the end of the survey and still get concurrent data.

IV. Firmware

The anemometer code was written in C. It is beyond the scope of this manuscript to describe the program in detail. However, the heart of an embedded real-time system lies in the use of timers and the interrupt handling, and this can be described. Figure 5 presents the flow chart for the Interrupt Service Routine (ISR) when shaft rotation is sensed using a magnetic reed switch. Switch closures generate interrupts.[3] One timer is used to generate regular interrupts for timekeeping. Another timer generates timeout interrupts. The timeouts guarantee that low rotation speeds are correctly handled as the interval between switch closures approaches the modulo value of the counter of fractions of seconds. A third timer controls the PWM to drive the analog meter without generating interrupts.

The clock speed is dynamically changed to optimise power consumption without risking the process becoming dominated by interrupt handling. Appropriate constants change to maintain good timekeeping. The processor can slow down by a factor of up to 256 with power consumption falling almost proportionally. This reduces wasted power at times when the processor is required to do little more than while away the idle milliseconds awaiting an interrupt.

[3] Switch openings also generate interrupts, since the microcontroller responds to changes on port pins. We choose to discard the opening interrupts, since they have less propensity to bounce and this aids our software debounce algorithm. This level of detail is not presented in the flowcharts.

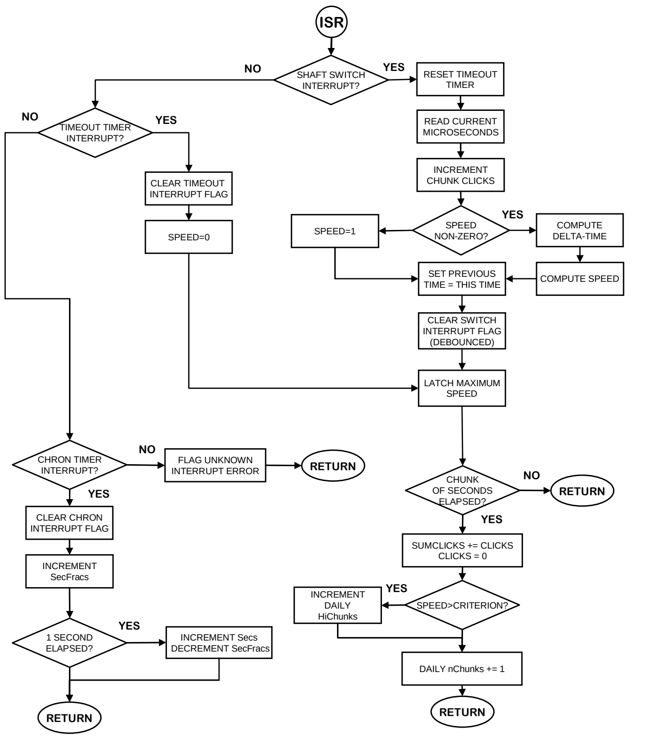

Fig. 5. Flowchart of the Interrupt Service Routine (ISR) for the switch-sensing method. Note the use of a timeout for detecting effective-zero speed.

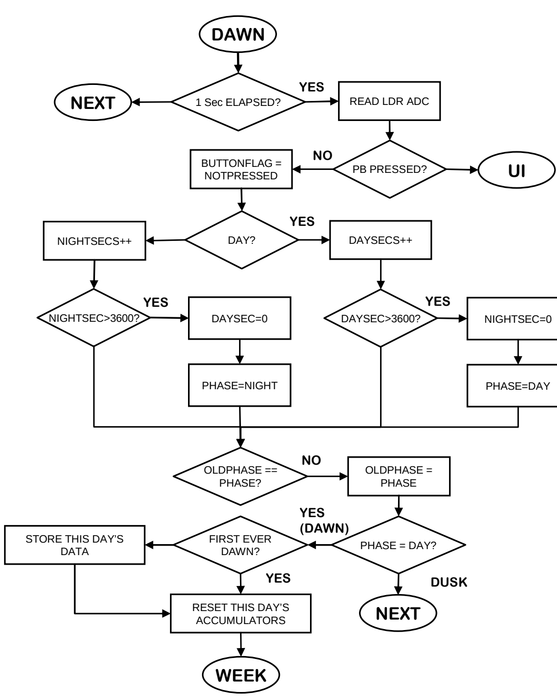

Fig. 6. Flowchart of the program reading input from pin 3 and detecting dawns for diurnal synchronisation.

Most mainline code comes into play only at initialisation, or upon the "detection of dawn", when diurnal synchronisation occurs, results are compiled, and non-volatile memory is updated. Figure 6 shows the structure of this part of the code.

Error handling is included. Errors are recorded in the non-volatile memory. The occurrence of an error typically suggests that the logged data may be untrustworthy. Table I lists the errors that are recognised and their meaning.

Error	Description
UNKINT	Unknown source caused interrupt. Should never happen.
SWDBNC	Shaft signal occurred $> 300\mu$s but so soon that wind speed would exceed 250kmh. Possible switch failure.
NODAWN	More than 30 hours elapsed without detecting dawn. Possibly LDR covered or failed.
CALERR	Cal factor out of range. Possibly not 40kmh during cal, or code compiled with incorrect POLES/CLOSURES value.
VIRGIN	NV memory not empty at first dawn. Spurious reset or device previously deployed. (Resumed.)
LDR_SW	Resistance of PB high or LDR low.
HISTOP	Speed fell to immeasurable level from a high rate. Possible rotor jam, bearing drag, or cup interference.
ERECUR	More than 8 errors have occurred.

TABLE I

LIST OF THE ERRORS AND POSSIBLE CAUSES.

When a multipole motor is substituted for the reed switch "closures" are simulated by looking at sequential measurements of motor voltage. The ADC measures motor voltage as frequently as possible, and in the regular chron part of the ISR median filtering and noise rejection are used to detect transitions that are treated like switch closures. The flow for this is given in figure 7.

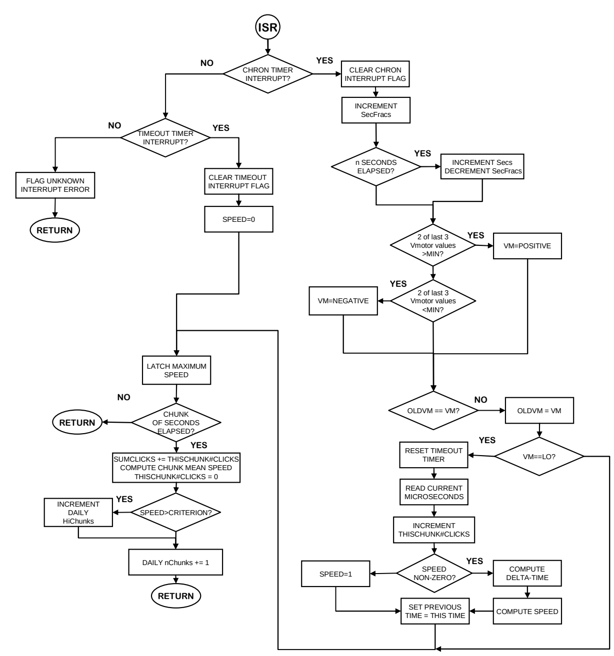

Fig. 7. Flowchart of the ISR for the HDD-motor sensing method.

V. Calibration

Where a given fixed cup assembly is used, the units can be pre-programmed with the calibration factor relating wind speed and shaft sensor signals. In our prototype we measured the rotation speed against known wind speeds provided by a moving vehicle in still air, and checked with instruments such as another anemometer or a GPS receiver. The prototype used ping-pong balls as the source of cups as these are available and come in a standard size. They were mounted at a precise distance of 65mm from the axis of rotation. The results of measurements are shown in figure 8. This gives a good idea of the rate at which signals will have to be handled by the microcontroller. Our prototypes gave about 0.285 rotations per second for each kilometre per hour of wind speed, or 3.5 kph for each Hertz of signal with a single switch closure per shaft revolution, measured using an Agilent U1252A DMM.

With two switch closures per revolution, our target maximum wind speed of 128 kph corresponds to almost 75 Hz.[4] Common HDD motors are permanent-magnet types with stator coils, and have 6 or fewer poles, typically 3 or 5. This implies that the anemometer will see an input signal of $<$225 Hz. Our design is capable of handling these frequencies with a calibration accuracy of better than $\pm$1 kph +1% in average and $\pm$1 kph +4% error in peak-speed measurement. It should be noted that our hardware is untested above wind speeds of 80kph, but the electronic design limits remain.

We discovered in wind tests that errors between our hardware and a GPS-based reference increased at lower speeds. We attribute this to a combination of stiction and air turbulence. We also observed that repeatability of our prototype's rotation speed was affected by the proximity of large objects. We believe that turbulent air from nearby objects of significant size has a greater impact as a fraction of speed at lower speeds. Descriptions of calibration in [3] imply similar findings. These problems were avoided by extending mounting standoffs and through calibrating at no less than 40 kph.

The anemometers are configured to allow calibration in the field. The method requires a known wind speed of 40kmh. If the anemometer is exposed to an air speed of 40kmh and the user holds the push-button down for more than 15 seconds, the calibration factor is set to the value required for the hardware to read 40kmh in response to the rotation speed in effect for the last few seconds of the press. It should be noted that this calibration method is to be avoided if possible, as noise in the reading at the moment of calibration impacts all further data. It is more reliable to make several measurements and obtain the calibration factor by regression as shown in figure 8.

For purposes of confirming helthy operation of both speed sensing and light sensing the firmware is configured to continuously modulate the LED or an attached meter in response to light for 6 seconds, and speed for 3 minutes, after 2-second press of the push button. This proves very helpful in confirming correct operation of each unit.

[4]The ping-pong-ball construction tends to fail at wind speeds around 100 kph as a result of centrifugal force, but we allow for stronger alternatives.

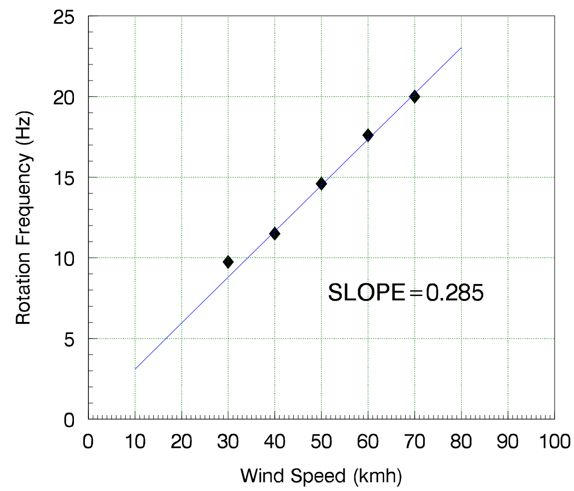

Fig. 8. Plot of shaft frequency against wind speed of a prototype anemometer. Data points are marked as diamonds. A linear regression, ignoring data below 40kmh, yields the straight line shown. The slope of the line $m = 0.285$ Hz/kmh.

VI. Initial Measured Data

A simple program translates a dump from the PIC data EEPROM into labelled text and decimal numbers, or a suitable format for reading into any data display package. Figure 9 presents some data logged during a very windy week in October by a one prototype anemometer. Figure 10 shows data extracted from the log of a commercial weather station over the same period. Data for 2 days, October 14 and 15, may be inaccurate because our prototype mount buckled and the anemometer was not vertical.

The two anemometers were mounted to the same post atop a 4-story building on a higher area of Hamilton, within 2 metres of each other. Both were mounted about 300mm spaced from the same vertical mounting post, though not at the same angle. Both were at least 1m clear of other obstructions mounted on the same roof (antennas, exhaust vents, etc). The prototype unit was tested to confirm the calibration before it was installed, ran without unexpected errors, and was functional when recovered.

The most striking observation comparing the two sources of data is that it is hard to believe that they could be exposed to the same environment. No constant calibration error can account for the differences. Comparing two days' data from the two sources it is possible to see one system reporting higher gust speeds and lower mean speeds compared to the other system on one day, and the reverse on the next day. In other words our anemometer might report higher average speed and lower gust speed than the commercial system today, but lower average speed and higher gust speed than the commercial system on the next day.

The commercial station logged averages at 10-minute intervals in contrast to our prototype using 10-second intervals. This could have a significant impact on the Working Fraction estimates if wind was gusty, showing a large standard deviation within an interval, but this should have no effect upon average

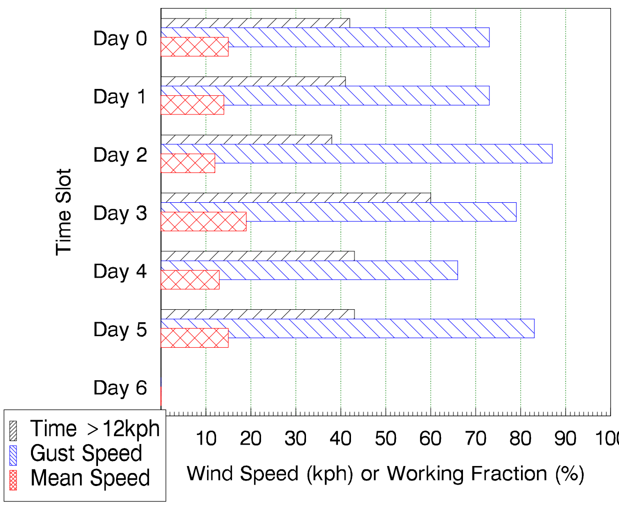

Fig. 9. Graphical presentation of data logged over the week up to the current day by a prototype. The "Working Fraction" is the fraction of the seconds in the day that the wind speed exceeded the criterion for generator output, a measure of the time above 12kph in this example, recorded in 10-second chunks. Day 5 corresponds to October 17^{th}, "DoM 17" in figure 10.

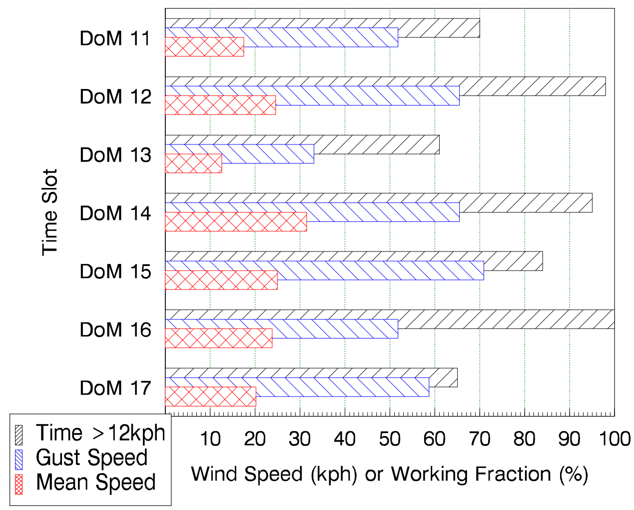

Fig. 10. Graphical presentation of data logged over several days in October by a commercial weather station. Display format is similar to figure 9. The days are labelled using the Day of the Month as the OS provides actual date and time. The working fraction is computed from logged data with the default logging interval of 600 seconds.

or gust (peak speed) data. The day-to-day trends do not match, deprecating this explanation.

We tested the same prototype in a "constant wind" generated in the lab by a fan, and we obtain average results as expected. Gust speeds match average for days wholly spent in the fan test. Gust speeds obtained from vehicle tests agree with test conditions. We have reasonable confidence in our logged data.

At time of writing we have not explained the differences. We are not even sure which system inspires the highest confidence. The commercial system suggests on October 16 that gust speeds reached only 52 kmh, the average was 23 kph, and yet the wind speed did not ever average less than 12 kph in the entire 24 hour period, a situation that seems unusual. We intend to collect data over a longer period, and to correlate this with the performance of a nearby wind generator attached to a data logger.

VII. Conclusion

The instrument as described should find application wherever a small-scale wind generator is considered, and can potentially save considerable wasted investment. It can also be used wherever the logging of wind speed alone is required. It has been designed to be

- very cheap to build,
- able to operate for a full year on AA dry cells,
- readily adapted for a particular wind generator, and
- easily adapted to a variety of sensor assembies.

Initial use highlights

- the impact of minor variations in location, altitude and terrain, and
- the difficulty of comparing data sets representing statistical wind speed data.

References

[1] P. J. Edwards, *Small scale wind power guide for New Zealand, Australia and the South Pacific*, NZERDC Report 136, November 1986.
[2] V. Smyth, *Wind energy resource survey of New Zealand. Phase 1: National survey using existing data*, NZERDC publication P114, January 1987.
[3] "DIY Rotorvane Anemometer" from *REALTIME CONTROL*, see `http://home.alphalink.com.au/~derekw/ane/anemain.htm`.
[4] "CFM Vane Anemometer Datalogger", *Extech*, `http://www.extech.com`.
[5] Wikipedia anemometer entry at `http://en.wikipedia.org/wiki/Anemometer`.
[6] Datasheet and programmers for the PIC12F683 available from Microchip Inc., `http://www.microchip.com`.
[7] "Homemade Easter Egg Anemometer", *Otherpower.com*, `http://otherpower.com/anemometer.html`.
[8] W. K. Widger, "Estimations of wind speed frequency distributions using only the monthly average and fastest mile data", *Journal of applied Meterology*, vol 16, 1977, pp224–247.
[9] Air-X specification, Southwest Windpower, `http://www.windenergy.com` or `http://www.renewablestore.com.au`.