A Parallax P2/ESP8266/ESP32 Wireless Weather Station – Current Status

Richard J. S. Morrison, Melbourne, AUSTRALIA March 26th, 2021

The weather station described here is controlled by a Parallax P2 chip, whose multiple cores work team up to acquire wind speed and wind direction data from an ultrasonic anemometer having 4 transducers. In addition, the instrument records data from a suite of temperature, pressure, humidity, light and rain rate/rain total sensors as well as acquiring date and time stamp information.

A requirement of this instrumentation was that it be capable of field deployment, reporting wirelessly back to a base station located some distance away. The base station developed here incorporates an ESP32 chip, an ILI9341 OLED display and an IR keypad receiver/controller that enables the user to visualize the acquired data in both textual and graphical forms. The base station also provides an SD card slot for data logging and subsequent post-processing of weather readings.

The wireless link employs Espressif's ESPNOW protocol. ESPNOW is a communications protocol enabling seamless wireless communication between two or more Espressif chips such as the ESP8266 and its newer cousin, the ESP32. Two-way communication links are easily constructed between these devices, allowing individual packets of data (typically up to 250 bytes in size) to be transferred back and forth. Much larger data buffers can be easily accommodated by performing multi-packet transfers in quick succession.

Consequently, at the weather station end, sensor data is transferred via a serial link from the P2 to an inexpensive ESP-12S (ESP8266-based) module, which then transmits ESPNOW data packets at ~ 10Hz to an ESP32-WROVER-B module on the base station PCB. Each of these packets amount to ~80 bytes of data.

The photos below show the 3 PCB's that make up this weather station ecosystem. Fig. 1 shows the P2 base board which breaks out the P2's full complement of I/O pins to double row headers at the top and bottom of the PCB. In Fig. 2, we see the weather station PCB (with its on-board ESP-12S module) plugged into the P2 PCB and Fig. 3 shows the ESP32/ILI9341 base station without the ILI9341 OLED display mounted. Later photos in Figs. 6 and 7 show the ILI9341 display during operation.

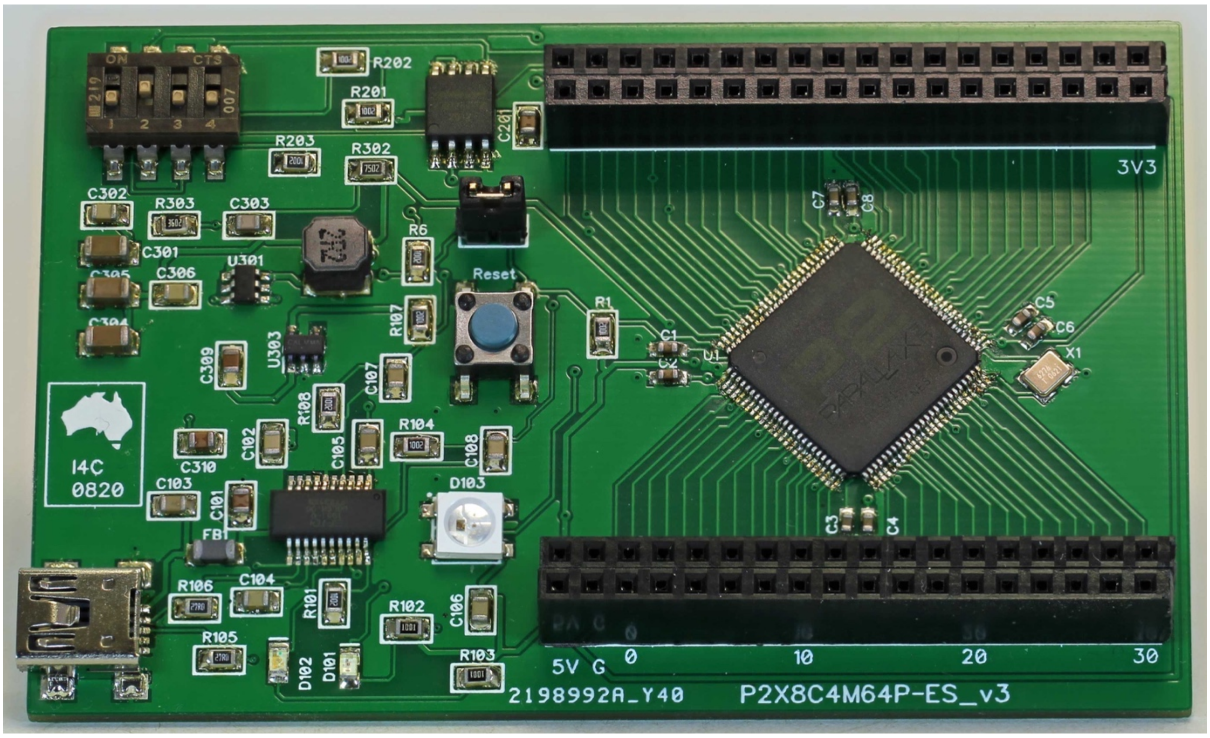

Fig. 1 The P2 base board.

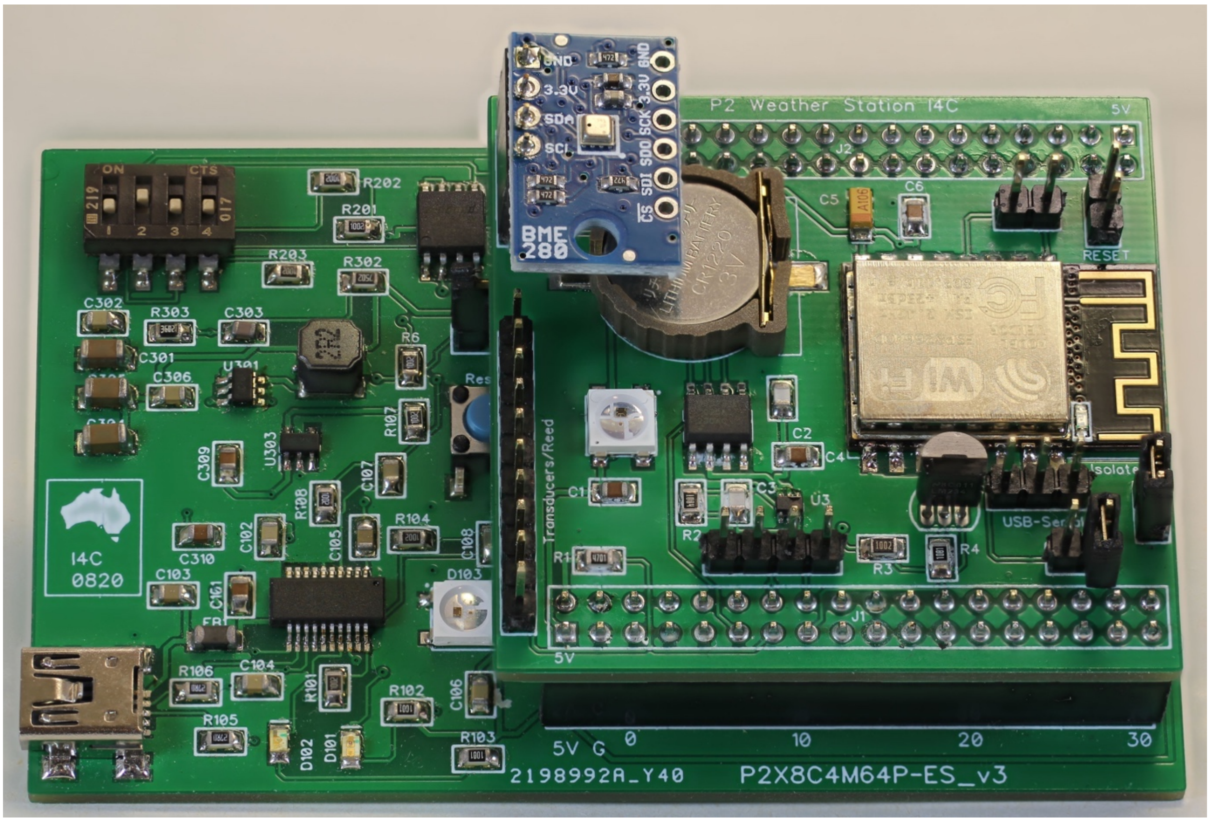

Fig. 2 The weather station PCB mounted atop the P2 base board.

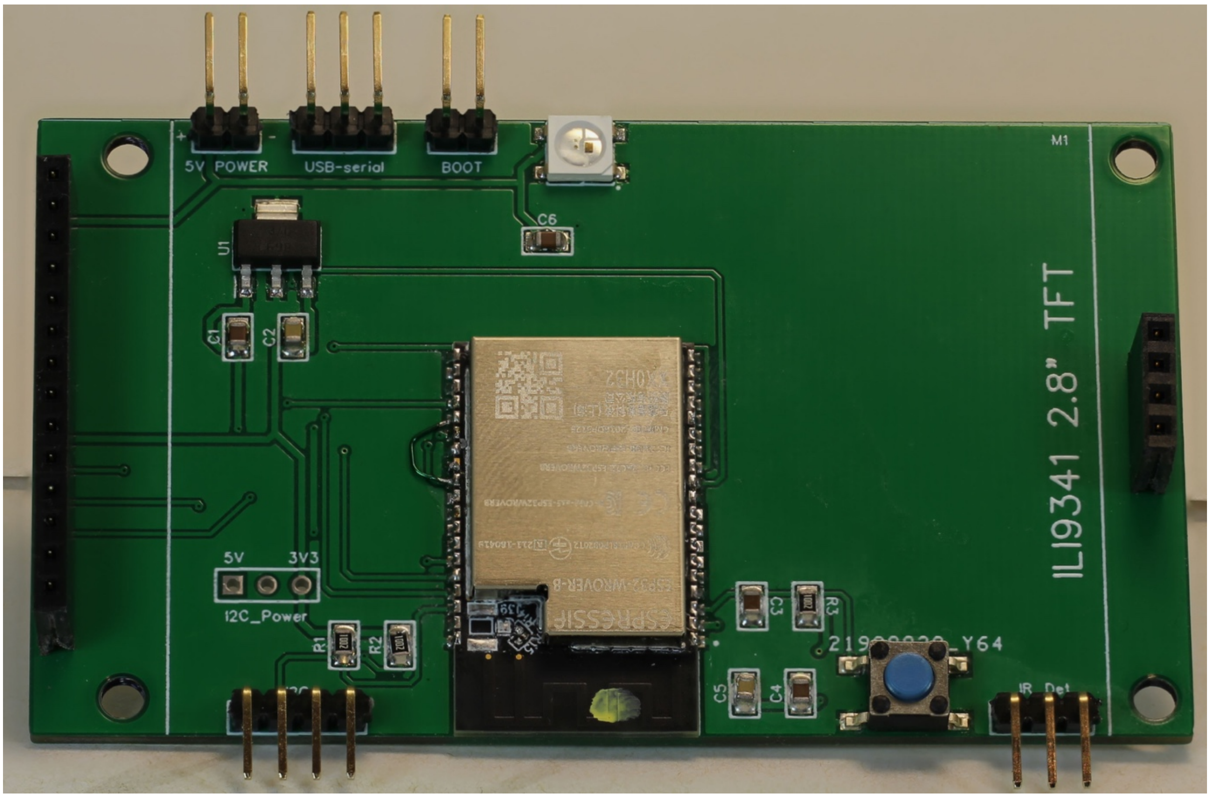

Fig. 3 The base station PCB. A 2.8” ILI9341 OLED display (which also has an SD card socket) plugs into this PCB while the ESP32 module handles receipt of ESPNOW sensor data packets from the weather station. For further details refer to the text.

Weather Station Components

The chipset on-board the weather station PCB includes (in addition to the ESP-12S module) a DS3231 real-time clock and associated coin cell battery, a header that accepts a Bosch BME280 temperature, pressure and humidity sensor module, a MAX9636 op-amp configured as a transimpedance amplifier, an LM234 adjustable current source and a WS2812B RGB LED.

The LM234 adjustable current source drives a small current (~70 uA) through a series-wired precision resistor and thermistor to ground (the latter’s resistance is nominally 10k @ 25C). 2 of the P2’s smartpins configured as ADC’s measure the voltage at the precision resistor and at its junction with the precision thermistor, allowing the actual current flow, and hence the thermistor resistance to be determined. The latter can then be converted to a temperature reading using the Steinhart-Hart equation with the manufacturer’s published coefficients.

Although the BME280 chip measures temperature it was felt advantageous to include provision for a thermistor in the system to realize a more accurate temperature measurement. In practice there is some heat generated by the P2 base board and both the BME280 module (I2C interface) and thermistor would normally need to be situated at some distance from it to give correct readings of the ambient temperature (see also Fig. 6 below).

The instrumentation also measures light via the small current generated by a low cost red LED that is connected to the input of the transimpedance amplifier, whose output is digitized by another P2 ADC smartpin. This allows sunny/cloudy/heavily overcast conditions to be continuously monitored and recognized.

The aforementioned LED and thermistor connect to the weather station PCB via a 4-pin header that can be seen just to the left of bottom center of the PCB.

Reed relay closures from a tipping bucket rain gauge (see later) are also monitored on another P2 I/O pin, allowing both rain rate and total rainfall data to be accumulated.

Without doubt the most interesting part of this weather station project is it's wind speed measurement system. This harnesses the P2's unique Goertzel capabilities to both excite an ultrasonic transducer at 40 kHz (using a pair of DAC+/DAC- pins) and to measure the phase of the signal received by a second (downstream) transducer connected to another ADC pin. The idea for doing so came from a suggestion by Chip Gracey on the Parallax forums - this method resulting in a perfectly viable wind measurement solution using no external components apart from the 4 transducers !

The instrument contains two such pairs of transducers – one for each of the orthogonal directions (N-S) and (E-W). The 8 leads from the four transducers connect to a 10-pin header that can be seen at the lower left of this PCB; the other two pins on this header connect to the external reed relay mentioned earlier.

The current mounting scheme for the transducers is shown in Fig. 4 below. In the next iteration the sensors will be mounted inside the 4 top elbows rather than being tethered to them via cable ties as can be seen in this image.

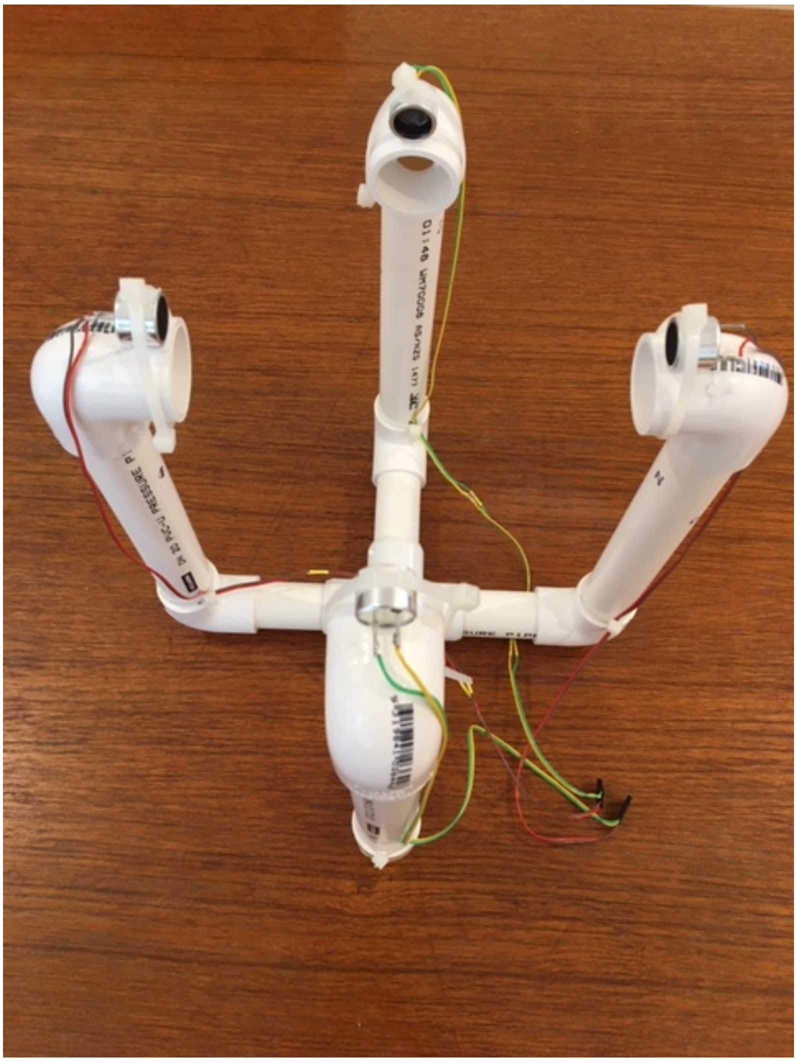

Fig. 4 The ultrasonic anemometer's transducers are mounted using a 4-way cross, some 90^0 elbows and piping – all standard PVC fittings.

Finally, an on-board WS2812B was also included as an aid during the development/debugging phase, but also to provide colourful visual cues during normal operations. I have made this a practice in all my recent designs using both the P2 and Espressif chips.

Software

The discussion here can be conveniently divided into three sections – what goes on in the P2, the ESP-12S and then the ESP32.

P2

On the P2, the *main COG* launches a number of data acquisition COG's that continuously read data from the various sensors and store the results to assigned locations in HUB RAM. A serial COG picks up these sensor readings and passes them to the ESP-12S for wireless transmission.

The main COG also launches a small serial monitor program that accepts commands from LabVIEW, allowing a means to upload the sensor data directly to a PC. This was a key step during development to check that data was being correctly acquired.

The LabVIEW control screen in Fig. 5 below displays results during a typical experiment. Here an oscillating fan was used to blow air through the anemometer at an angle of approximately 45^0. The lower two plots at the left of screen show raw and calm (no wind) corrected Goertzel phase readings, respectively, while the upper traces show the resolved N-S and E-W wind speeds as well as the total wind speed. The other traces to the right of screen show the other sensor readings recorded by the instrument.

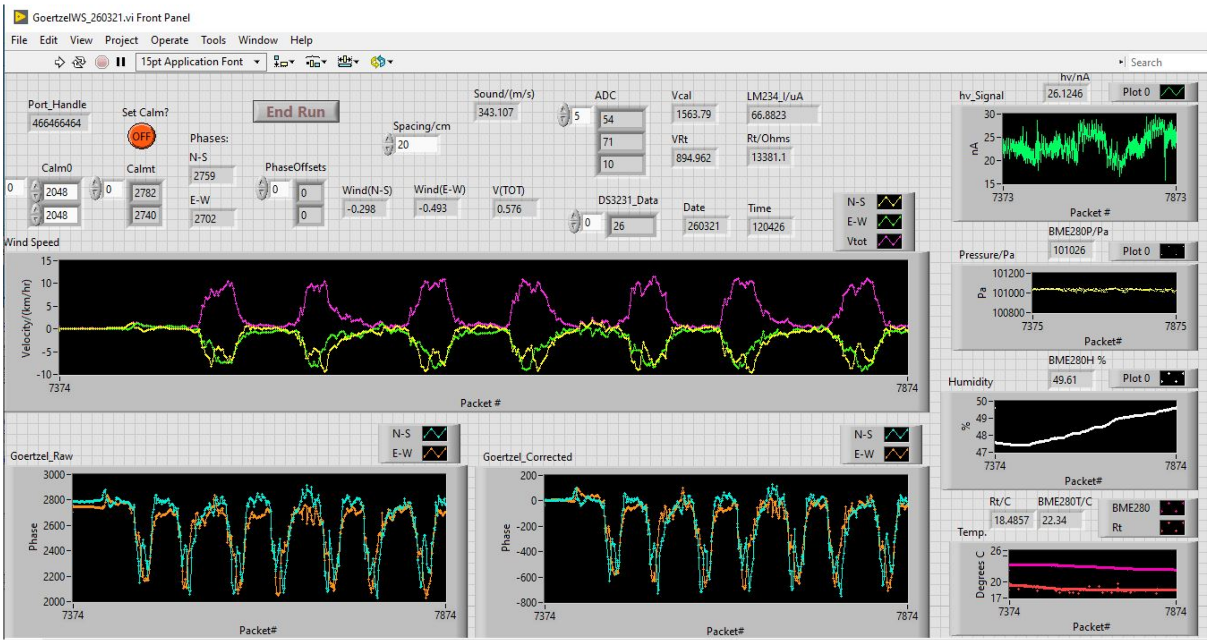

Fig. 5 The weather station LabVIEW vi that was developed to test the P2 data acquisition part of this project.

Here is some brief information on the functions of the other COG's deployed in this application -:

Two Goertzel COG's are used to continuously excite and read phase values back from each pair of ultrasonic transducers. The excitation frequency is 40 kHz. Differential outputs on P0 and P1 connect to one 40 kHz transducer with an ADC input coming on P4 from its opposite partner. In the other direction P2/P3 and P5 are used.

An *ADC COG* continuously reads data from 3 analog pins, capturing an optical signal on P9 and the voltages at a calibration resistor and thermistor on P10 and P11.

An *I2C COG* is dedicated to managing I2C transactions controlling the BME280 and DS3231 chips on-board with SCL on P6 and SDA on P7.

A *Rain pulse detector COG* counts pulses on P8 (handling relay bounce) from a tipping bucket rain gauge. This gauge has a reed relay that closes each time the bucket tips; on each closure a pulse gets generated and this COG increments the rain counter accordingly.

A *serial COG* is used to send data every 100 msec from the P2 to the ESP-12S module. Pin P12 is allocated for this purpose. The baud rate for these transfers is currently 115,200 bps but this could be increased if necessary.

In summary, the weather station interface uses a total of 13 pins on the P2.

ESP-12S

The role of the ESP-12S module is very simple. As already mentioned, an I/O pin on the P2 is connected to the Rx pin on the weather station’s ESP-12S module. After an interval of 100 msec, a new data packet is assembled on the P2 and transferred serially to the ESP-12S, which then broadcasts this packet to the base station. The process then repeats.

ESP32

The base station employs an ESP32-WROVER-B module. It’s job is to receive the ESPNOW packets from the ESP-12S and decode them into a full set of meteorological readings. This does involve a reasonable amount of processing, particularly for data received back from the BME280 module.

Each individual BME280 chip stores it’s own set of calibration coefficients and these are used along with the actual sensor readings to realize a valid set of T, P and H values.

Goertzel phase values from each of the (N-S) and (E-W) directions are 12-bit numbers. Since the transducers are being driven sinusoidally at 40 kHz, one full cycle or period (phase values of 0-4095) then represents 25 microseconds of flight time. Code on the ESP32 stores a pair of phase values recorded under still conditions and then uses the real-time changes in these values to compute the wind speeds in each direction. These calculations also require the speed of sound in air - which depends on the temperature (and also very slightly on the humidity) - as well as the transducer spacing.

An algorithm incorporated into the ESP32 code was required to handle phase roll-over conditions that inevitably occur so that the measurements remain correct throughout the full range of possible wind conditions.

As mentioned earlier, data from a number of ADC channels are acquired by the P2 using it's SINC2 ADC smartpin mode. Conversion of these readings to voltages is a trivial step.

The ESP32 has three other important functions apart from crunching the sensor data. It has an SPI connection to an ILI9341 display that makes use of a very nice open source graphics library (https://github.com/Bodmer/TFT_eSPI). This allows data to be displayed on screen and this can be done either textually or graphically (see Figs. 6 and 7 for examples).

In order to switch between these modes and control what is being displayed, an IR receiver and IR control keypad are implemented via a small module connected to the 3-pin header at the bottom right of the PCB shown in Fig. 3. Button presses on the keypad then allow the user to see all the measurements or alternatively, to see a graph of individual measurements as they are being received.

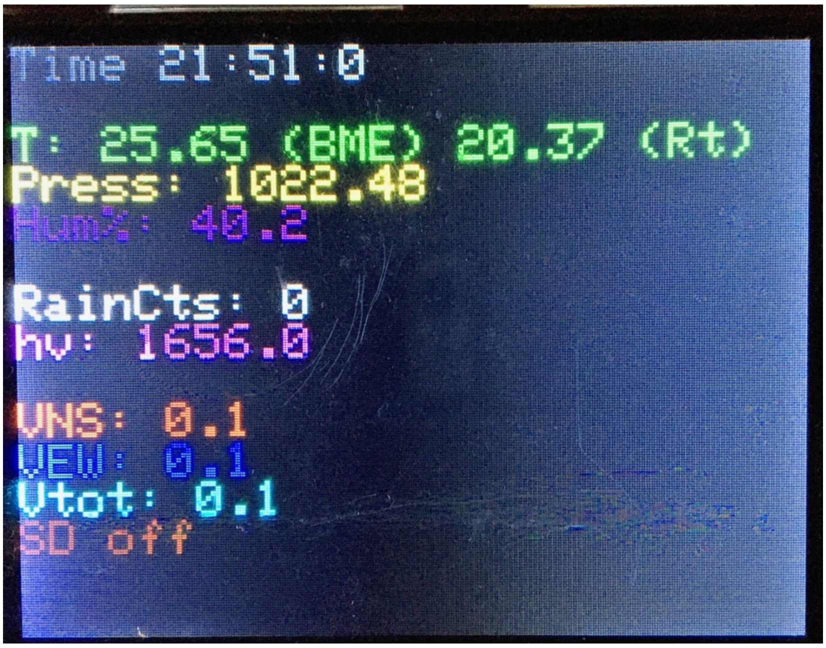

Fig. 6 ILI9341 textual display of weather station data in calm conditions. There are two temperature sensors on board – the one internal to the BME280 is affected by slight heating from the P2 below and hence it's reading differs from the thermistor reading; the latter correctly reflects the ambient temperature.

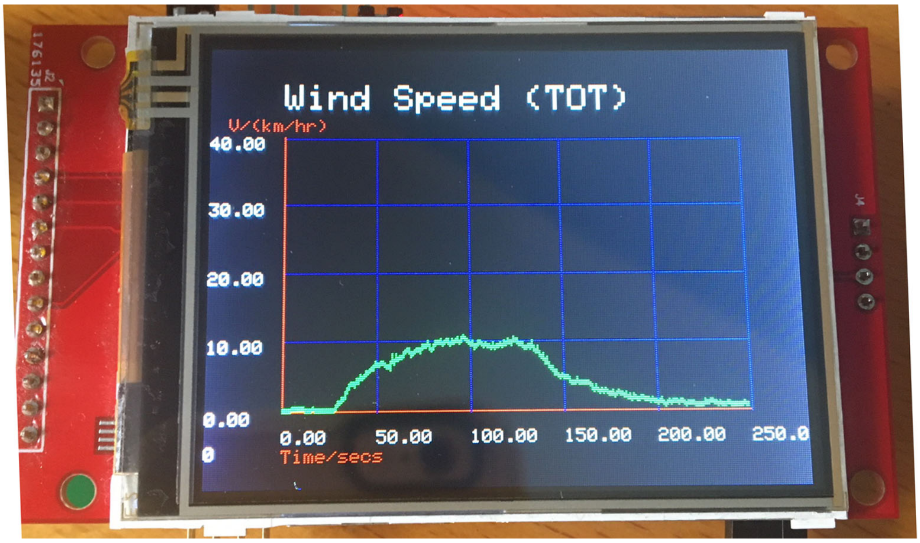

Fig.7 ILI9341 graphical display of wind speed. A fan is switched on briefly, then switched off. The horizontal axis is incorrectly labelled and should read packet number. Total run time was approximately 30 seconds (100 ms packet interval, plus some serial transfer time per packet) .

Finally, the ESP32 code allows data to be stored onto the SD card; this mode can also be enabled/disabled via the keypad control.

Results

Of all the variables being recorded by this weather station, only the wind speed data is likely to change in a dynamic way over a short timeframe. Fig. 8 shows representative wind data logged to the SD card. This data was recorded at the top of an open hill over a period of just over 1 hour.

As seen from the plots, the prevailing conditions during the run were a steady predominantly westerly wind of around 5 km/hr but with gusts exceeding 20 km/hr.

My estimation of the noise in the data displayed here is a few tenths of a km/hr. Currently the leads to the transducers are unshielded and it remains to be seen whether using shielded leads might slightly improve performance. Some further work is still required to characterize/understand system noise.

What one can say for certain is that low wind speeds are likely to be much more reliably recorded with this system than with a traditional rotating cup-based anemometer.

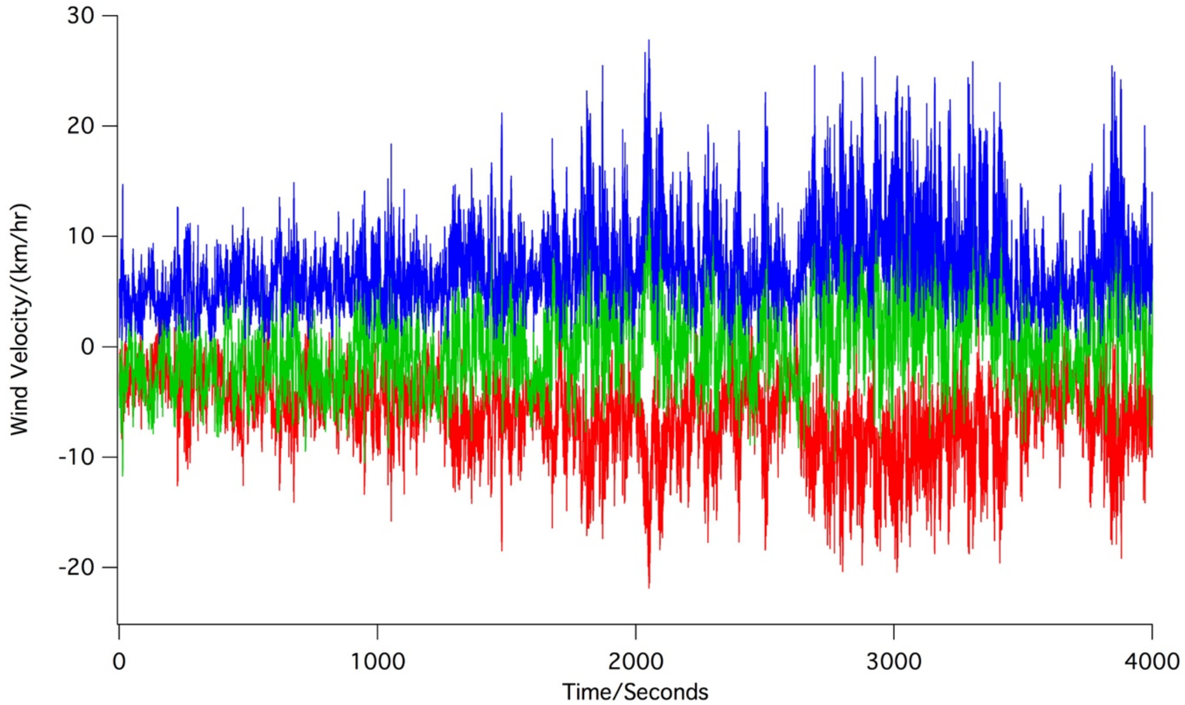

Fig. 8 Some sample wind speed data recorded by the instrument. Here the green and red traces are the wind speeds in the N-S and E-W directions and the blue trace represents the total wind speed obtained from the magnitude of their vector sum.

Future work

(a) The transducers in this instrument are not waterproof and so they will need to be replaced by fully sealed types. These will produce a much lower signal than do the current ones, but the P2's ADC circuitry has higher sensitivity modes than are currently being used - so this is not likely to be an issue.

(b) A waterproof 3D printed enclosure will also need to be developed for this weather station and some means of battery powering (equipped with a means for solar charging) will also be required; I do have some experience in doing something similar in an unrelated field-based instrument.

(c) A tipping bucket rain gauge has been 3D-printed but its operation has not been trialled yet, although the P2 code correctly handles the reed relay contact bounce issue. A conventional rain gauge has been installed at the intended measurement site for eventual comparison with readings from this system once real rain data becomes available.

(d) The range of ESPNOW transmissions is limited to perhaps a few hundred meters under the most favourable of conditions - and 50 meters or so is more typical for reliable transmissions. In previous work experimenting with 2.4 GHz YAGI antennas I found that this range can be greatly extended by connecting the Espressif PCB trace antenna to the driven element of a YAGI. This leads to a very simple yet interesting possibility. The weather station's ESP-12S module could send its data packet to a nearby YAGI-fitted ESPxxxx chip which would then merely act as a relay station, on-sending that data packet to the ESP32. This will be easy to try out and would mean that the weather station could potentially send data a long way to a house where the base station needs to be located.

(e) Finally, it would be trivial to make the weather data available on the internet so that the information can be accessed remotely from afar. This capability will certainly be added down the track.

Concluding remarks

While not everyone will be interested in the details of data acquisition and logging of meteorological information, this project does show what is possible by teaming the P2 up with Espressif chips.

I think what is remarkable is how much heavy lifting can be done with very little extra in the way of additional components. Therein lies the real strength of the P2 – plenty of I/O, smartpins, ADC's, DAC's and its 8 cores. Hopefully, this report has demonstrated that quite sophisticated remote instrumentation projects can be realized at very low cost. I have a number of other instrumentation projects currently underway using the P2 that similarly leverage its many powerful features.