An APRS-IS Solar-Powered Telemetry System

Karl W. Berger, W4KRL
IoT Kits
PO Box 1864
Centreville, VA 20122
W4KRL@ARRL.net
https://w4krl.com

Abstract

APRS-IS is used as a free cloud service between Wi-Fi connected solar-powered sensors and a dedicated remote color TFT display. Weather station energy consumption is reduced by placing the microcontroller in a low power state between updates. The remote display uses an APRS SSID to filter and acknowledge messages of interest.

Key Words: APRS-IS, Solar Power, Telemetry, ESP8266, Arduino

Introduction

Solar power and wireless connectivity are particularly useful features for a telemetry transmitter as they permit placing the station in locations that might be difficult for wiring and battery maintenance.

The system posts weather and station information to both APRS-IS and the free Internet of Things service ThingSpeak[1]. A user must have a valid license to post data to APRS-IS.

Solar energy collected by a small photovoltaic panel is stored in a lithium polymer (LiPo) cell. The device has worked more than a year with no attention.

Programming the ESP8266[2] System on Chip (SoC) with the Arduino Core[3] provides a flexible, low-cost Wi-Fi enabled platform for connection to APRS-IS. The Core opens the entire Arduino ecosystem including the friendly, cross-platform Integrated Development Environment (IDE) as well as many hardware and data analysis libraries to developers at no cost.

While the ESP8266 is surface mount device with 1.5 mm pitch pins, it is supplied in a variety of development kits with 0.1-inch (2.54 mm) pitch pins making it easy to use with conventional through-hole construction. The widely available WeMos D1 Mini[4] is ideally suited for solar-powered applications through its use of a voltage regulator and USB-converter with very low parasitic loads. With the D1 Mini “deep sleep” mode total system current draw is reduced to about 80 µA at 5 Vdc. Contrast this with awake current averaging around 80 mA with peak draws up to 300 mA.

The recent availability of a low-cost, full-color display “shield” compatible with the D1 Mini form was the stimulus to develop a dedicated remote display with APRS-IS as the telemetry link.

Weather Station

The weather station is comprised of a D1 Mini, two sensors, a solar panel, LiPo cell, and charge controller. Small slide switches control system power and the program/run mode of the ESP8266.

A BME280[5] sensor measures barometric pressure, temperature, and humidity and a BH1750FVI[6] provides ambient light sensing. Both are supplied on 0.1-inch pitch breakout boards. The BME280 draws less than 4 µA when active and 0.1 µA when idle. The BH1750 draws 190 µA when active and 1 µA asleep. Both are 3.3-volt I^2C devices easily interfaced to the D1 Mini.

The light intensity measurement is handled as a telemetry channel rather than as weather data because the APRS weather protocol does not include light intensity. Two additional non-weather telemetry channels report Li-Po cell voltage and the Wi-Fi Received Signal Strength Indication (RSSI) as measured by the ESP8266.

When active, the D1 Mini connects to Wi-Fi, reads the sensors, posts the data to both APRS-IS and ThingSpeak, and then enters the “Deep-Sleep” mode for a fixed period. Only the RTC is active in Deep-Sleep. At the end of the sleep period the RTC output pin asserts a reset of the ESP8266 causing it to wake up in an initialized state.

In normal operation, all active tasks generally complete in less than seven seconds. Failure to connect to the Internet or to successfully read the sensors may add a second or two to the awake time. No special recovery actions are taken in the event of a failure, the unit simply goes to sleep for the specified time and tries again on the next wake cycle. This approach was taken after an early version of the firmware repeatedly attempted to retry the Internet connection draining the LiPo cell during an Internet outage.

Solar Energy Considerations

The selection of the solar panel and LiPo cell are subject to several engineering design choices. Time asleep largely establishes the average energy consumption and determines the minimum size of the solar panel. The period that the unit can operate without solar energy input sets the LiPo cell capacity. These are interrelated parameters that are influenced by the commercially available ratings of solar panels and LiPo cells and the actual measurement of active and asleep power draws. The calculations are simplified by using milliampere-hours (mAh) as the basis of energy consumption rather that Watt-hours. This is a commonly accepted approximation in battery-powered designs.

It was arbitrarily assumed that: 1) there are four hours per day of sunshine, 2) the unit should be able to operate seven days without sun, and 3) the sleep duration is five minutes.

Active Mode:	80 mA × 7 seconds =	560	mAs
Sleep Mode:	0.08 mA × 300 seconds =	24	mAs
Energy per cycle:	24 mAs + 560 mAs =	584	mAs
Average current:	584 mAs ÷ (300 + 7 seconds) =	1.9	mA
Energy per day:	24 hours × 1.9 mA =	46	mAh
Solar panel rating:	46 mAh/day ÷ 4 h/day =	12	mA
LiPo cell capacity:	7 days × 46 mAh/day =	320	mAh

Manufacturer ratings for LiPo cells are notoriously optimistic, a factor of two is not uncommon. A cell rated at 600 mAh cell was selected. A 55 x 80 mm 6-volt solar panel rated at 0.6 W should deliver 100 mA, a factor of 8 greater than needed but the cost and size are so small as to allow the overdesign.

A TP4056[7] linear charge controller set at 1 A prevents overcharge of the cell. The cell is directly connected to the voltage regulator input on the D1 Mini. Operation has been observed down to 3.0 V input. Power for the I^2C sensors is supplied from the regulated 3.3V terminal of the D1 Mini.

Figure 1 shows cell voltage over a four-day period. Cell voltage ranges between 4.0 and 4.1 Vdc corresponding to a nearly steady 100% charge.

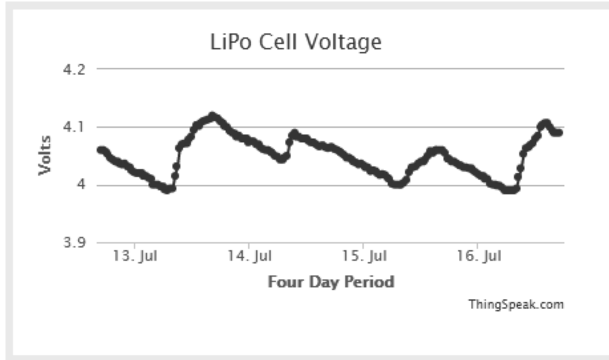

Figure 1 LiPo Cell Voltage over a Four-Day Period

Telemetry Sequence Numbers

According to Chapter 13 of the APRS Protocol Reference[8] each telemetry report must be preceded with a three-character sequence number, presumably 000 to 999. Since the ESP8266 awakens in an initial state, the firmware cannot directly increment a sequence number. A counter could be stored in the built-in flash memory but that has a finite number of lifetime cycles and would quickly wear out if a short sleep period is used.

The initial approach taken in the telemetry station was to obtain a timestamp from a Network Time Protocol (NTP) server. The time in seconds from January 1, 1900 is first divided by the sleep duration in seconds to get an interval number. Then taking modulo 1000 of the interval generates a value from 0 to 999 that is padded with leading zeros as required by the APRS protocol. This worked quite well in practice.

Since developing this sequence numbering method, an ESP8266 API[9] has become available that enables access to an unused portion of the RTC memory. Because RTC memory stays active in Deep Sleep, it is possible to store an incrementing sequence number there. This eliminates the time needed to logon to an NTP server saving valuable energy during the awake period.

Telemetry Transmitter Construction

Since all active components are available as 0.1-inch pitch breakout boards, it was easy to prototype the circuit on a solderless breadboard. This same approach was carried over to a formal double-sided printed circuit board for both practical and economic reasons: Through-hole construction is much easier for most hams in contrast to surface mount components and, oddly, breakout boards are usually cheaper than the cost of the main component. Figure 2 shows the single-board version of the station.

A “stacked shield” style transmitter compatible with the D1 Mini format has also been designed. One shield mounts the sensors, and another mounts the charge controller and supporting components. The charge controller shield uses female headers and has screw holes for mounting. The D1 Mini is the middle board with stackable headers and the sensor board is on top with male headers. The stack is shown in Figure 3.

The shield version is easily adaptable for other projects. Figure 4 shows a “dual shield” board mounting a sensor board and an OLED display stacked on a D1 Mini for use as an indoor, USB-powered temperature display.

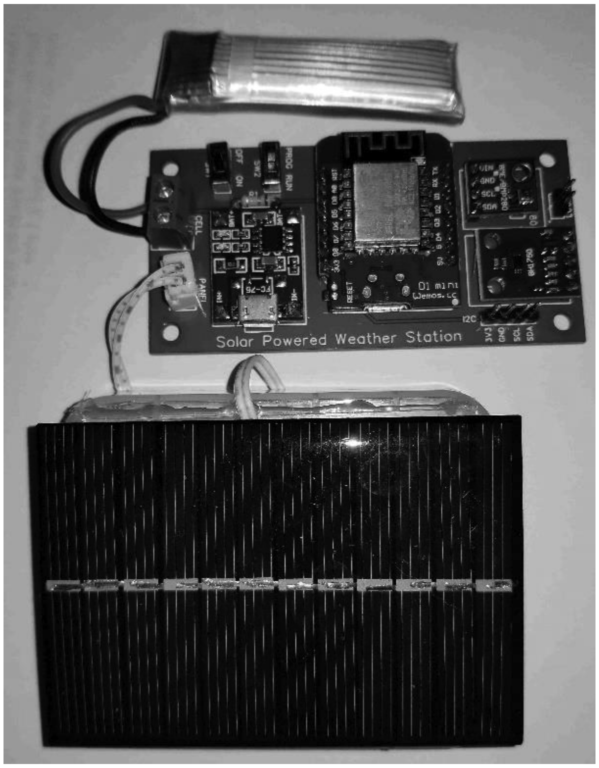

Figure 2 Single Board Version with Solar Panel and LiPo Cell

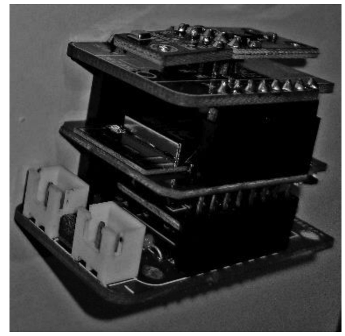

Figure 3 Stacked Shield Version

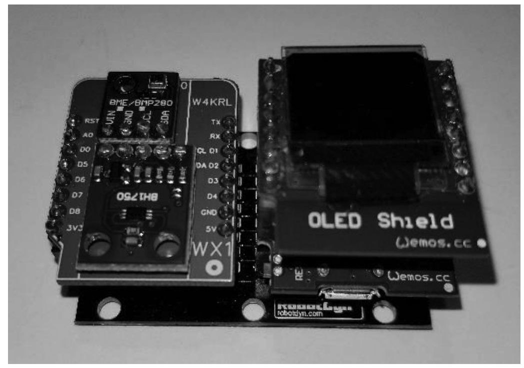

Figure 4 Dual Shield Variant with OLED display

APRS and ThingSpeak Compared

Formatting weather and telemetry data for the APRS protocol requires a lot of careful manipulation as the number of characters and range of values are strictly prescribed for each parameter. Barometric pressure, humidity and other measurements all have different formats requiring separate coding for each parameter including those not used in this project such as wind velocity and rainfall.

ThingSpeak, as with other Internet of Things (IoT) services, is flexible in terms of data formatting. String representation of any numerical parameter is permitted. No scaling is required. There is a limit of eight data fields per channel, but multiple channels can be bound together and organized with tags.

While APRS data is available through APIs from services such as aprs.fi and findu.com, ThingSpeak provides very powerful MATLAB[10] data analysis and visualization tools. For example, the MATLAB histogram plot in Figure 2 clearly shows the consistency of the Wi-Fi signal strength. A chart like this could be correlated with humidity, for example, to see how propagation is affected by weather. It could be the basis of an interesting science fair project.

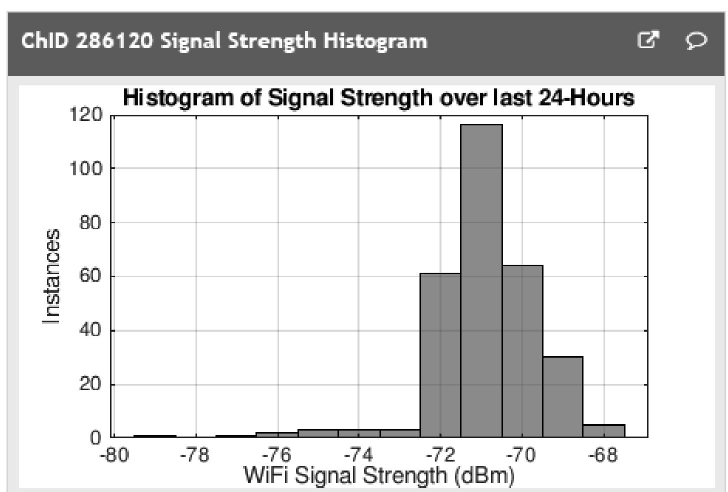

Figure 5 Incidence of Wi-Fi signal strength over one day

In general, it is easier to incorporate ThingSpeak data into other projects. Of course, only licensed radio amateurs may post data to APRS leaving IoT services the only option for non-hams.

The great feature of APRS is accessibility by radio. Though a VHF connection was not used in this project, it should be possible to extend all the features, including solar power, to a full, stand-alone VHF APRS station.

Remote Display

A dedicated display completes the telemetry system. The display uses a 1-inch square full color shield in the D1 Mini format. Only three components are needed: 1) a D1 Mini, 2) the display shield, and 3) a pushbutton. Power is supplied through a Micro USB cable from any USB port or 5-Volt adapter.

Normally the display shows local and UTC times derived from an NTP server in a sequence of digital and analog frames as shown in Figure 6. The firmware constantly listens to a filtered APRS-IS Tier 2 server looking for the SSID of the solar powered weather station. It adds the weather and telemetry data to the sequence of display frames as soon as they are received.

By providing the display with its own APRS SSID, it is possible to direct APRS messages to the remote display from any APRS or APRS-IS connected device. The received message will remain on the display until acknowledged by pressing the pushbutton. The display then returns to the sequential display of time, weather and telemetry data.

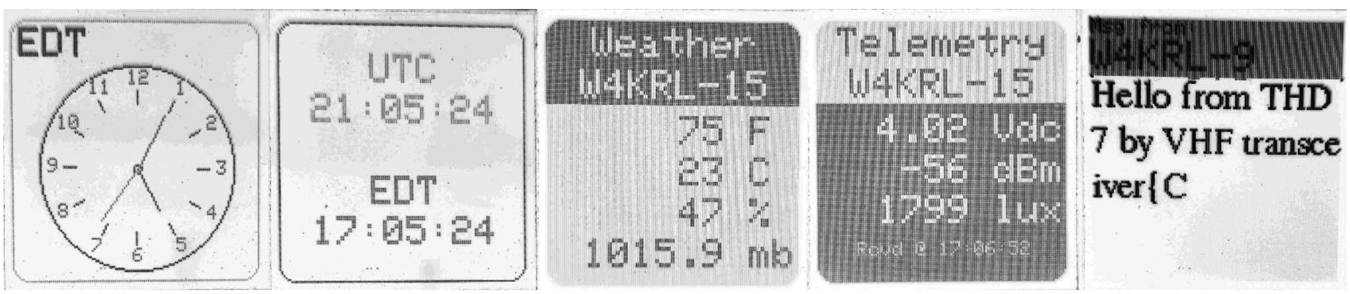

Figure 2 Composite photograph of display sequence with APRS message

Conclusion

APRS-IS is a practical backbone for telemetry stations. Wi-Fi connectivity combined with solar-power enable flexible siting of the stations, eliminates wiring, and minimizes maintenance. Solar power has proven to be practical, reliable, and easily achievable at low cost with small component volume. For licensed radio amateurs there is the possibility of using a VHF connection to APRS.

Development kits based on the ESP8266 System on Chip provide wireless Internet connectivity and computational power at very low cost. All development tools and an immense software library are available for free thanks to the use of the Arduino Core. Subsystem hardware based on 0.1-inch pitch breakout boards provides easy constructability with simple soldering tools by any builder.

A dedicated remote display terminal is a convenient way to view the transmitted weather and telemetry data and to receive directed APRS messages.

All software, instructions, and a kit of parts are available from the author at https://w4krl.com and W4KRL@ARRL.net.

Data streams are visible at https://thingspeak.com/channels/286120 and at https://aprs.fi/W4KRL-15.

References

[1] The MathWorks, Inc. (n.d.). Understand Your Things. Retrieved July 18, 2018, from https://thingspeak.com/

[2] Espressif Systems (Shanghai) Pte., Ltd. (n.d.). ESP8266 Overview. Retrieved July 18, 2018, from https://www.espressif.com/en/products/hardware/esp8266ex/overview

[3] ESP8266 Community Forum (Ed.). (n.d.). Arduino core for ESP8266 WiFi chip. Retrieved July 18, 2018, from https://github.com/esp8266/Arduino

[4] WEMOS.CC. (n.d.). D1 mini. Retrieved July 18, 2018, from https://wiki.wemos.cc/products:d1:d1_mini

[5] Bosch Sensortec GmbH. (n.d.). BME280 Integrated Environmental Units. Retrieved from https://www.bosch-sensortec.com/bst/products/all_products/bme280

[6] Rohm Semiconductor. (2011, November). Digital 16bit Serial Output Type Ambient Light Sensor IC BH1750FVI. Retrieved July 18, 2018, from http://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf

[7] Nanjing Top Power ASIC Corp. (n.d.). TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8. Retrieved July 18, 2018, from https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf

[8] Wade, I., G3NRW (Ed.). (2000, August 29). APRS Protocol Reference Version 1.0.1. Retrieved July 18, 2018, from http://www.aprs.org/doc/APRS101.PDF

[9] ESP8266 Community Forum (Ed.). (n.d.). ESP-Specific APIs. Retrieved from http://esp8266.github.io/Arduino/versions/2.3.0/doc/libraries.html#esp-specific-apis ESP-specific APIs

[10] The MathWorks, Inc. (Ed.). (n.d.). Documentation: Prepare and Analyze Data. Retrieved July 18, 2018, from https://www.mathworks.com/help/thingspeak/matlab-analysis-and-visualization.html