Environmental Monitoring as a Public Resource on the APRS Network

James Corum[1] and Devin R. Berg[1,*]

[1]Engineering and Technology Department, University of Wisconsin-Stout, Menomonie, WI 54751 USA
*Corresponding Author: bergdev@uwstout.edu

***Abstract*—This paper presents the proof-of-concept of an environmental data monitor for the collection and distribution of sensor data of interest to the amateur radio community and others via packet radio (AX.25 protocol) on the APRS network. Included is a discussion of suitable hardware and software to collect data, format data packets and transmit the data packets to the APRS network. The design is based on open source LightAPRS 2.0 tracker board, which is compatible with a variety of analog and digital sensors. The board is programmed in C++ using the Arduino IDE. Environmental data is encoded as an APRS telemetry packet and conforms to the APRS protocol. The protocol instructs the proper encoding and decoding for transmitting and receiving stations. Example packet formatting and a demonstration of the successful distribution of environmental data via the APRS network are provided.**

I. Introduction

Monitoring environmental data can provide useful, regional insights into various patterns such as changes in weather, ground temperature, lake or other waterway conditions, among many other possible measures. Distributed data collection via wireless sensor networks can provide adaptability and flexibility [1]. Such networks may be implemented using a variety of technologies such as LoRa, WiFi, 5G, or other similar options [1] [2] [3] [4]. The succes of such networks is dependent on connectivity for distribution of the collected data. Networks that provide mesh capabilities have an advantage in that any one node does not require a direct data connection back to a central hub or repository [5]. Instead, data can be propagated through the network until it reaches the desired destination. For data that is of public interest to the amateur radio community such as weather data or other environmental conditions such as ground temperature or solar flux, the Automatic Packet Reporting System (APRS) network could provide access to an existing network of wireless devices that are monitored by station operators and already in use for the distribution of regional information. The availability of inexpensive hardware both for collecting sensor data and for distribution via packet radio makes it possible to participate in this community owned network by sharing resources as an active contributor.

As a proof-of-concept, an example node was deployed in an outdoor setting to read environmental data, specifically air temperature. The data was transmitted on the standard North American APRS radio frequency in the two meter amateur radio band. The use of APRS radio repeaters operating on the same frequency relay received packets for a designated number of hops and some stations gate these packets to the internet. The following sections of this paper provide further detail on this proof-of-concept and demonstrate its use in transmitting sensor data across the APRS network.

II. The Automatic Packet Reporting System For Environmental Data Gathering

A. The Automatic Packet Reporting System

The APRS is a globally distributed packet radio network operated primarily by stations in the amateur radio service. This network distributes a variety of data types and is intended to provide information of local utility to a regional geographic area within range of radio communications. Stations may act as transmitters, receivers, or as repeaters, which receive packets and rebroadcast them across the network [6]. In this manner, these stations operate as a form of mesh network, enabling greater communication range than any single radio transmission can achieve. As a secondary function, some stations in the network provide gating of data packets to the internet. These stations provide access to the Automatic Packet Reporting System-Internet Service (APRS-IS) where they upload packets into a packet data stream, which may then be picked up by various websites and services that monitor this data stream. In these cases, the data transmitted by the packet can be inspected by anyone with access to one of these websites or services. Additionally, some APRS services retain and display data packets for a period ranging from hours to months. This is commonly done for packets containing weather data or other telemetry, which may then be compared to establish trends. Finnegan [6] provides a detailed explanation of the network's history and basic design, as shown in Figure 1. Any radio can broadcast data, however, the APRS standardizes communication between radio devices and structures the devices into a wireless network.

This work was partially supported by NSF SMA Award #1950289.

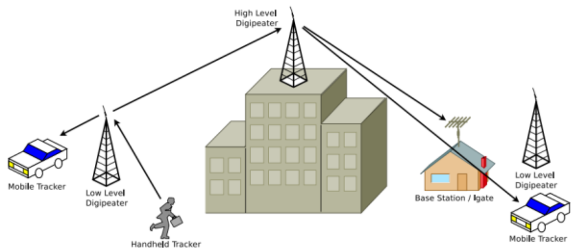

Figure 1. Example network arrangement of APRS nodes. [6]

B. Analyzing the APRS packet

The format of an APRS packet is primarily defined by the APRS Protocol Reference [7], which was last updated in the year 2000 [8]. Since the preparation of that document, several variations and revisions have been drafted but have not been merged back into the original Protocol Reference [8]. In this implementation, we attempt to provide minimal working examples using packet formats commonly used across the APRS network, but in some cases may not strictly adhere to the original Protocol Reference.

The transmission of sensor data on the APRS network can be done using a telemetry packet. The telemetry packet is one which simply contains the sensor data to be communicated. However, it is also useful to provide additional APRS message packets which instruct a receiving station how this telemetry data should be interpreted [9]. Additionally, position packets may be desired to provide a geographic location for where the sensor reading is being taken.

Each packet begins with a header. In the example below, the first eight characters include the call sign of the sending station and the station's secondary station id (SSID), which is included because some amateur radio operators will have several devices transmitting APRS packets. The > character indicates a separation between the source and the destination. In the example below, `APLIGA` is listed as a destination and is being used here to identify the make and model of the device that sent the message. The next two destinations, `WIDE1-1` and `WIDE2-1` are path configurations using the WIDEn-N paradigm which indicate how many times the packets should be repeated by another station [6]. These will be modified by any station which repeats the packet. Additionally, repeating stations may add their own station call sign to the path to indicate that they retransmitted the packet. The final character, the colon, separates the header from the packet's payload.

```
W9UWS-15>APLIGA,WIDE1-1,WIDE2-1:
```

An example position packet payload is shown below. The first character indicates the data type, here the `!` character indicates that this is a position packet for a fixed station. The next eight characters indicate the latitude in NMEA format with degrees, minutes, decimal minutes as well as the `N` character for North. Character 10 indicates which table, primary or alternate, should be used to select the symbol which will be displayed on graphical maps. Characters 11-19 provide the longitude, again in degrees, minutes, and decimal minutes. Character 20 specifies which symbol from the symbol lookup table should be displayed. The remaining characters allow for a comment which can be used to communicate additional information.

```
!3853.86N\07702.19W<comment
```

The actual sensor data is contained within the payload of a telemetry packet. Telemetry packets transmit five analog values and eight binary values. The format of the packet is dictated by the APRS protocol. However different APRS users or websites use different protocols. For example, a strict interpretation of the original APRS protocol only allows for 8-bit telemetry data [10]. By this interpretation, telemetry data sent on the APRS network must be mapped to integer values in the range 0–255. More recent APRS software allows for floating point data and will interpret such data packets correctly.

An example of the payload of a telemetry packet is provided below. The first character, `T`, indicates that the data type for the packet is telemetry. The three numbers after the `#` character are a zero-padded sequence number to uniquely identify individual packets. Incremented by one for each sent packet, the sequence indicates the order the packets were sent. The remaining comma-separated values contain the data, in this example, only the first is being used. The remaining zero values may be omitted.

```
T#176,4.6875,000,000,000,000,00000000
```

The telemetry packet on its own provides no mechanism to determine what the data means. Therefore, self-addressed message packets are used to inform the receiver how to interpret the data in the telemetry packet. For message packets, the packet data type is indicated by a colon, which is followed by the recipient of the message, in this case the call sign and SSID of the originating station. The recipient field is specifically nine characters in length. Therefore, if the callsign and station id are less than nine characters the remaining characters should be filled by spaces [7]. There are four standard packets used for proving context to the telemetry data.

A parameter packet contains the names of the five analog and eight digital telemetry data. There are no restrictions on the individual labels, but the overall character limit of the message is 197. In the example below, only one parameter is labeled, and it is given the label, “Temperature.”

```
:W9UWS-15 :PARM.Temperature
```

A unit packet contains the units for the telemetry values. As in the parameter packet, the unit label is shown here to be, “degC.”

```
:W9UWS-15 :UNIT.degC
```

An equation packet indicates the coefficients on a second-order function that provides mapping of the telemetry value to the real data. If the value contained in the telemetry packet is x, then the real data is calculated according to the equation $ax^2 + bx + c$, where the coefficients a, b, and c are delivered in the equation packet for each of the five analog values. Each group of three comma-separated numbers represents a, b, and c, respectively. The example packet below indicates that the receiver should multiply the first analog value in the telemetry packet by one. That is, $a = 0$, $b = 1$, and $c = 0$. The use of an equation packet may be less necessary if the telemetry packet is using floating point data rather than 8-bit numbers. If an equation packet is not transmitted, then most receivers assume a default as shown in the example.

```
:W9UWS-15 :EQNS.0,1,0
```

A Bits packet allows for a definition of the bit sense for each of the eight binary telemetry values sent by the telemetry packet. A value of `0` or `1` in the Bits packet when equal to the binary value from the telemetry packet indicates `true`, or if they are different, `false`. This packet can also be used to define a project title, which may be displayed by the recipient.

```
:W9UWS-15 :BITS.11111111,Title
```

C. Environmental Monitoring

Distributed sensors networks over wide geographical ranges for environmental monitoring can provide challenges in efficiently receiving sensor data. Mesh networks such as the APRS network can assist with the relay and recovery of data due to the density of nodes in the network which may already exist in a particular region. In the absence of sufficient node density, it is possible to distribute additional nodes, and even configure them as repeaters, with relatively low cost. A user can strategically place transmitters and repeaters throughout a location and achieve greater reliability in transmitting data across the APRS network. If it is desired that this data be accessible to the general public, then it may be necessary to ensure that an internet-enabled node is within the range of the mesh. Once data packets have reached an internet-enabled node, the data will be transmitted to the APRS-IS and become more broadly available.

The basic requirements for an individual environmental monitor node include a sensor or sensors, a microcontroller to process sensor data and prepare the packets, and a radio transmitter to broadcast the packets. Due to the nature of data transmitted by radio, there is limited bandwidth and frequent transmission of data packets can cause network congestion and interference with other transmitters. As the APRS network is maintained by amateur radio operators voluntarily providing their time and equipment as resources to the community, individuals must take care not to dominate the network with too frequent transmissions. Packet transmissions should only be transmitted as often as necessary based on the data type and should be spaced by at least several minutes. Position packets for stationary nodes and message packets need only be transmitted once or twice per day. Additionally, nodes should be configured to use a minimum number of hops using the WIDEn-N paradigm needed reach their destination. The station’s antenna, propagation conditions, and distance from other local stations will determine the packet’s effect on the APRS network as well.

III. Proof-of-Concept

A. Hardware

A basic station for environmental monitoring will consist of a sensor or sensors, a microcontroller, and a radio transmitter. An example implementation of such a station is described here. A minimal implementation procedure includes the following steps.

1. Attach the sensor to the microcontroller and read the data.
2. Place station in a location within range of other nodes in the network.
3. Construct the telemetry packet based on the APRS specification.
4. Transmit the packet on the appropriate radio frequency.

5. Monitor network traffic to ensure packets are reaching the desired destination.

In this proof-of-concept implementation of an environmental monitoring station, hardware was selected for both functionality and efficiency. The following components were used.

- The QRP-Labs LightAPRS 2.0 (https://qrp-labs.com/lightaprs2.html) is an Arduino based APRS transmitter for operation on the 2-meter Amateur Radio band. This board provides custom component integrations via I2C, SPI, or analog pins. The integrated radio transmitter outputs either 0.5 W or 1 W depending on the input voltage with a transmission range that is dependent on environmental conditions and line-of-sight.
- A waterproof 1-Wire DS18B20 digital temperature sensor collects temperature data.
- For power, a Voltaic Systems 10 W solar panel and 6,700 mAh battery pack with pass-through charging was used.
- An MFJ-1754 ¼ wave vertical antenna is connected to the LightAPRS 2.0 with coax cable.

As shown in Figure 2, the temperature probe is connected to the LightAPRS 2.0 using the 3.3V and GND pins for power, with the data line connected to in A2. The DS18B20 twelve-bit digital temperature sensor was used to provide example sensor data. This temperature probe requires only one data pin. In addition, the DS18B20 requires a pullup resistor.

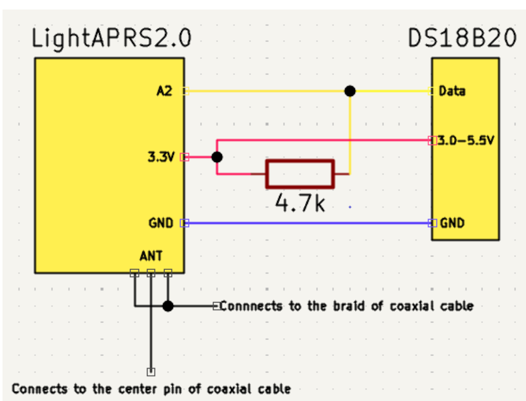

Figure 2. Circuit diagram showing the configuration of the temperature probe, LightAPRS 2.0, and antenna connection.

B. Software

Since the LightAPRS 2.0 is an Arduino-based device, it can be programmed with the Arduino IDE or other compatible software. Arduino IDE has importable libraries for the M0 microcontroller and much of the source files used in the project were written for Arduino. The DS18B20 requires the OneWire library to communicate with the microcontroller. Example code for the LightAPRS 2.0 are available from the developer.

C. Device Placement and Power

For optimal performance, the transmitter should be placed in a location with minimal interference. The fewer obstructions between the transmitter and nearby receiving stations, the greater the reliability of packet transmission.

The LightAPRS 2.0 has a current consumption between 450 mA and 750 mA during transmit and 7 mA when idle. The 6,700 mAh battery and 10 W solar panel should provide sufficient capacity even if the transmitter is operated at high duty cycles. During testing, the transmitter was set to transmit telemetry packets every two minutes

D. Data Display

Transmitted packets that successfully reach an internet-enabled station will be gated to the APRS-IS. From this data stream, services can read and process the telemetry packets. For example, aprs.fi is a website that receives and displays APRS packets in an easily accessible format. For telemetry packets, aprs.fi displays the data in graphical format grouped by originating station call sign and SSID, as shown in the example in Figure 3. This example shows temperature data collected over a twelve-hour period in Woodbury, MN.

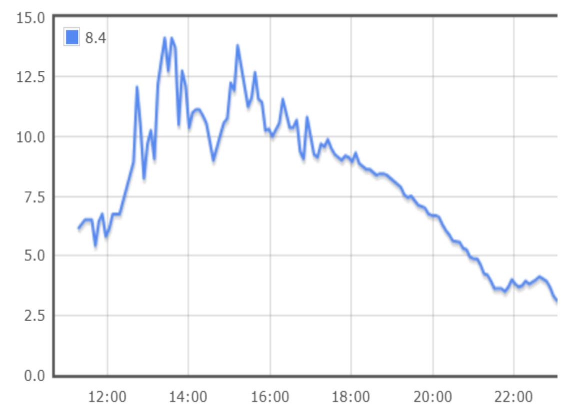

Figure 3. Temperature data transmitted by telemetry packets as it is displayed on aprs.fi.

Raw data can also be accessed for local processing. The aprs.fi website provides an application programming interface (https://aprs.fi/page/api) and packets can be similarly sourced directly from the APRS-IS data stream.

IV. Conclusion

The APRS environmental monitoring as a public resource leverages hardware and a community network infrastructure. It provides data not only useful for an individual, but also for people interested in the data for their own needs. In fact, many radio hobbyists have been using the network for years for this very function. This project describes an application broad scope by which any sensor data of interest to the community may be transmitted across the APRS network. The system used here provides an example of a low-cost monitoring station that can be deployed to collect and transmit data over a large area without dependence on proprietary hardware or networks. The use of community-based resources means the data generated should also be of community interest and open to the public. Care must be taken to not overload network capacity. Future work on this project could include further optimization of the hardware as well as added repeater functionality to individual nodes. We successfully demonstrated the operation of a single node for at least 24 hours. However, further testing is necessary to demonstrate its use in harsher weather conditions and for longer periods of time. The hardware cost could be further reduced by using components targeted at minimal viability. Transmitter power could be reduced in environments with greater node density. This project highlights the use of open-source technologies to generate and transmit data and give back to the community in a way that fosters additional research.

References

[1] L. Lombardo, S. Corbellini, M. Parvis, A. Elsayed, E. Angelini and S. Grassini, "Wireless Sensor Network for Distributed Environmental Monitoring," IEEE Transactions on Instrumentation and Measurement, vol. 67, no. 5, pp. 1214-1222, 2018.

[2] T. Lindquist, K. Chambers, N. Benish, D. R. Berg and A. Turkmen, "LoRa Wireless Mesh Network for Lake Biology Study," in 2024 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2024.

[3] W. Chanwattanapong, S. Hongdumnuen, B. Kumkhet, S. Junon and P. Sangmahamad, "LoRa Network Based Multi-Wireless Sensor Nodes and LoRa Gateway for Agriculture Application," in 2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C), Bangkok, Thailand, 2021.

[4] S. Chundu, C. Lundquist and D. R. Berg, "MenomiNet: A prototype network for real-time public lake data," in National Conference on Undergraduate Research, Eau Claire, WI, USA, 2023.

[5] L. M. Oliveira and J. J. Rodrigues, "Wireless Sensor Networks: A Survey on Environmental Monitoring," Journal of Communications, vol. 6, no. 2, pp. 143-151, 2011.

[6] K. W. Finnegan, "Examining Ambiguities in the Automatic Packet Reporting System," California Polytechnic State University, San Luis Obispo, 2014.

[7] The APRS Working Group, "APRS Protocol Reference: Protocol Version 1.0," 30 April 2000. [Online]. Available: https://files.tapr.org/software_library/aprs/aprsspec/spec/aprs101m/APRS101m.pdf. [Accessed 08 April 2025].

[8] J. W. Langner, "The APRS Documentation Project," February 2025. [Online]. Available: https://github.com/wb2osz/aprsspec. [Accessed 08 April 2025].

[9] K. W. Finnegan, "APRS Telemetry," December 2016. [Online]. Available: https://github.com/PhirePhly/aprs_notes/blob/master/telemetry_format.md. [Accessed 08 April 2025].

[10] K. W. Berger, "Sending APRS Analog Telemetry – The basics," September 2018. [Online]. Available: https://w4krl.com/sending-aprs-analog-telemetry-the-basics/. [Accessed 08 April 2025].