Protocol APRS nodes PE1RXF

Version 1

Why a new protocol

APRS has a build in protocol for telemetry. It is well described and it is widely implemented. Websites like aprsdirect.com show the data in a neat graphical form. So why use something else?

The APRS telemetry protocol has several drawbacks. First, it uses a lot of bandwidth to send 5 analog and 8 logic values: four messages for PARAM, UNIT, EQNS and BITS and a message with the actual values. The analog values have a resolution of only 8 bits.

These five messages are send every time the values are send, which is great if you want every APRS user to be able to interpreted the messages on the fly. But the trade-off can be a very busy APRS channel.

I only want to send the telemetry from on site of the garden to the other, using as little bandwidth as possible, but still complying with the APRS protocol and the law. Than means I cannot encode my messages to make them smaller (not allowed by law, because it may be seen as encryption) and I have to use excising APRS messages techniques to comply with the APRS protocol. An APRS messages can contain up to 67 bytes of free text and has the capability to acknowledge messages.

This solves the second drawback: the standard telemetry messages may or may not arrive at the destination. Sending telemetry as an APRS message with acknowledgment makes sure a message is actually received by my server.

This protocol sends the actual data in one packet. It can send 11 values at once. The server can request additional information about the telemetry data such as names and units, much like described in the original APRS telemetry protocol. But this is optional, so less bandwidth is used.

Everyone can still interpreted the messages by reading this document, which is publicly available on my website.

Architecture

A typical network consists of a central server and a number of nodes. The central server has control over all the nodes. There can be a maximum of 14 nodes and only one server.

The server is named PE1RXF-10. The nodes are named PE1RXF-x, where x is 0-15, but not 10 (suffix of the server).

All communication is done via standard APRS messages. Therefore it respects the APRS protocol. The communication protocol itself is embedded in these APRS messages.

Node

A node is an APRS enabled data acquisition device with a maximum of 11 inputs and 5 outputs. The inputs can be analog or digital. The outputs are digital (eg. either on or off).

Commands

The server can send a command to a specific node by sending an APRS message to this node. This message contains the command. The node response by sending a message back to the server. This can either be an APRS acknowledge (ack***xx***) or a message as described in the command table below.

A command always starts with a question mark ‘?’ followed by a number between 00-99. Always include leading zero, as the nodes expect two digits, even when the number is below ten.

Commands which only set a parameter of the node include a message identifier. This way, the node can answer by sending an acknowledge. This follows the APRS message protocol.

Commands which expects a specific respond from the node (for example ‘send telemetry data’, see command table below) do not include a message identifier. The node does not send an acknowledge, as the defined response can be seen as an acknowledge.

It is possible to let the node send the telemetry data periodically. Intervals are 0 (off), 10, 20, 30, 40, 50 and 60 minutes. The telemetry format is the same as send after command 03. Some nodes can also use the APRS defined protocol for telemetry. The format used can be set by sending command 50 for the protocol described in this document or 51 for the standard APRS protocol.

If a command is not supported by the APRS node, it rejects the message. But only if the server sends an message identifier.

The server call, the node call and both suffixes are hard coded in the firmware and cannot be changed by a remote command. This is a safeguard to prevent someone changing these calls by illegally using the PE1RXF-10 call. This way the node cannot be hacked to use an other call sign. It is however possible by anyone to use the hard coded call sign to control the device. Using a call sign that is not yours is illegal and therefore not the responsibility of this protocol.

Never use this protocol for critical applications; it is only designed for experimentation purposes.

CMD	Description	Response	Example
00	Does nothing, this message is ignored	*<no response>*	
01	Get firmware version and information from node	V*xx,description*	V1,APRS node on roof
02	Send beacon once	APRS location beacon	5302.77N/00707.85EnAPRS node
03	Send telemetry data fields	Up to 11 values separated by commas. Values are numbers and can have decimal point and may start with a sign. Each field has a maximum length of 5 characters. Fields may have leading or trailing spaces.	+10.8,-99.9,12655,18,1, 36.8,1.056, 3,-77 ,-45.5,25.56,54345
04	Send description of data fields	Up to names of the data	Vin,Hum,Res,Temp1,Temp2,

		fields, separated by commas. Each description has a maximum length of 5 characters.	Vout,cnt1,degr,V1,V2,cnt2
05	Send units of data fields	Up to units of the data fields, separated by commas. Each description has a maximum length of 5 characters.	Volt,%,Ohms,C,C,Volt,#,deg, Volt,Volt,#
06	Send status of output pins	5 bits status of output pins	10011
07	Send description of output pins	Names of the 5 output pins, separated by commas. Each description has a maximum length of 10 characters.	Heater,Lamp,Pump,Bell,Gate
10{10	Set message path off	APRS acknowledge 10	ack10
11{11	Set message path to WIDE1-1	APRS acknowledge 11	ack11
12{12	Set message path to WIDE1-1, WIDE2-1	APRS acknowledge 12	ack12
13{13	Set message path 1 to WIDE1-1, WIDE2-2	APRS acknowledge 13	ack13
20{20	Set interval a telemetry message is send to <never>	APRS acknowledge 20	ack20
21{21	Set interval a telemetry message is send to 10 minutes	APRS acknowledge 21	ack21
22{22	Set interval a telemetry message is send to 20 minutes	APRS acknowledge 22	ack22
23{23	Set interval a telemetry message is send to 30 minutes	APRS acknowledge 23	ack23
24{24	Set interval a telemetry message is send to 40 minutes	APRS acknowledge 24	ack24
25{25	Set interval a telemetry message is send to 50 minutes	APRS acknowledge 25	ack25
26{26	Set interval a telemetry message is send to 60 minutes	APRS acknowledge 26	ack26
30{30	Set output 1 low	APRS acknowledge 30	ack30
31{31	Set output 1 high	APRS acknowledge 31	ack31
32{32	Set output 2 low	APRS acknowledge 32	ack32
33{33	Set output 2 high	APRS acknowledge 33	ack33
34{34	Set output 3 low	APRS acknowledge 34	ack34

35{35	Set output 3 high	APRS acknowledge 35	ack35
36{36	Set output 4 low	APRS acknowledge 36	ack36
37{37	Set output 4 high	APRS acknowledge 37	ack37
38{38	Set output 5 low	APRS acknowledge 38	ack38
39{39	Set output 5 high	APRS acknowledge 39	ack39
50{50	Send telemetry as described in command 03	APRS acknowledge 50	ack50
51{51	Send telemetry as described in original APRS specs	APRS acknowledge 50 APRS reject 50 when not supported by node	ack50 rej50