

# Telemetrino, telemetría fácil para digi-repetidores de APRS



equipo de cacharreo.es,  
Ourense

## 1. Primeros pasos

Hace algunos años ya, navegando por la red en busca de algún desarrollo que permitiese decodificar el tráfico de APRS sin necesidad de un PC conocimos el Antitracker [1, 2]. Este circuito, conectado a la salida de audio de un receptor, decodifica tramas de APRS a 1200 bps y, para cada estación recibida, genera un *waypoint* que envía en formato serie a través de un puerto RS-232 a un navegador GPS, en cuya pantalla podemos visualizar el *waypoint* generado.



Foto 1

Tras programar los PIC y montar el circuito en una placa de prototipo (foto 1), pudimos comprobar su correcto funcionamiento en un PC mediante un programa típico de terminal como puede ser el hyperterminal (Windows) o minicom (Linux).

Por aquel entonces el *tracking* móvil en APRS era dominado por el TinyTrak fabricado por Byonics, un dispositivo muy sencillo pero eficiente basado en PIC que en ese momento únicamente permitía el envío de tramas de posición [3]. Fue en esa época cuando otro dispositivo llamó nuestra atención: el Opentracker+ de Argent Data Systems [4, 5]. El OpenTracker+, al igual que el TinyTrak, permite el envío de tramas de posición, pero además incorpora la capacidad de decodificar las tramas de APRS del Antitracker. Todo ello implementado en un único microcontrolador.

Pero la cosa no terminaba ahí. A diferencia de TinyTrak, el OpenTracker era fácilmente reprogramable para convertir el dispositivo en un TNC KISS, una estación meteorológica o incluso una baliza de CW (AFSK). Este proyecto había sido desarrollado por el radioaficionado N1VG sobre la filosofía del código abierto [6]. Las fuentes del código estaban y están disponibles en la red. Enseguida fuimos conscientes del enorme potencial que nos ofrecía el Opentracker+, lo cual me animó a realizar la traducción de su manual de uso al español, para facilitar su difusión y uso entre la comunidad de radioaficionados de habla hispana [7].

## 2. La telemetría

Otra de las capacidades que incluye el Opentracker+ es el envío de datos de telemetría de acuerdo al protocolo de APRS. Hasta aquel momento, y salvo los desarrollos iniciales de los compañeros

del Digigrup-EA3 [8, 9], no encontramos más implementaciones de esta funcionalidad para APRS. El manejo de telemetría abre un interesante campo de experimentación y desarrollo para el radioaficionado, además de resultar también de gran utilidad práctica.

Con el objeto de familiarizarnos con el funcionamiento de la telemetría en APRS, y sirviéndonos del Opentracker+ que ya implementa esa capacidad, pusimos en marcha hace ya más de cuatro años nuestro propio proyecto de telemetría: el "proyecto TRITON" [10]. Para sacarlo adelante, desde el equipo de cacharreo.es, establecimos en su momento acuerdos de colaboración con diferentes entidades más allá del limitado círculo de la radioafición.

El protocolo APRS contempla el envío de telemetría de manera codificada. Pueden enviarse hasta cinco valores analógicos y ocho binarios. Los valores analógicos tienen una resolución de ocho bits y pueden abarcar cualquier magnitud medible, como tensión, temperatura, presión, etc. Los valores binarios pueden adoptar los valores 0 o 1, representando el estado de un interruptor o cualquier codificación binaria que nosotros establezcamos. Como mencionábamos anteriormente, esta información se envía en AX25 con una codificación propia. Para recuperar los valores originales de cada magnitud es preciso decodificarlos. Partiendo de la documentación original disponible en inglés [11] y de nuestra propia experiencia, realizamos un tutorial en donde se explica en detalle cómo se maneja la telemetría en APRS [12].

## 3. El Arduino

No vamos a extendernos mucho hablando de él porque en Internet hay suficiente información al respecto, comenzando por la página oficial en español [13]. El Arduino es una plataforma de desarrollo basada en un potente microcontrolador Atmel de 8 bits creada con fines educativos sobre la filosofía de *software* y *hardware* abierto. Reúne una serie de interesantes cualidades que representan un potencial enorme todavía por explotar para la comunidad de radioaficionados. Por ejemplo: entorno de programación muy sencillo y multiplataforma (Windows, Linux y Mac), lenguaje de programación de alto nivel -lo que supone que una mayor facilidad para comenzar a programar-, gran cantidad de accesorios hardware disponibles, así como multitud de librerías de programación que permiten manejar directamente pantallas LCD, servos, sensores, interfaces Ethernet etc. Además de contar con el soporte de una amplia comunidad de usuarios y desarrolladores por todo el mundo.

Actualmente miles de personas están usando para algo un Arduino y compartiendo sus proyectos en la red, incluso entre la comunidad de radioaficionados.

## 4. El Telemetrino

La idea inicial consistía en utilizar un Arduino para ampliar las opciones de telemetría del Opentracker+, enviando también a la red de APRS las tramas con la información necesaria para permitir la decodificación automática de la telemetría.

Esta idea evolucionó rápidamente hasta convertirse en un sistema independiente que dota a cualquier digirrepetidor -corriendo el UI-Digi en versión 1.8 o superior- de la capacidad de enviar telemetría. La primera versión del código fuente se programó en un Arduino montado sobre una placa de prototipos. Esta sencilla configuración se presentó en la edición de la Ham-Radio del

Salnés del año 2010 [14]. A pesar de la prisa con la que preparamos el prototipo, este funcionó a la perfección (foto 2).

El siguiente paso fue mejorar el prototipo inicial optimizando el código y montando un circuito autónomo en su propia PCB (foto 3), para realizar un ensayo completo sobre un digi real en funcionamiento. Para realizar la primera prueba de campo en la red contamos con la colaboración de los compañeros de URS, la sección de URE en Salamanca. Desde el primer momento acogieron con entusiasmo la idea de participar en los ensayos del Telemetrino instalándolo en uno de sus digis de la red de APRS. El lugar elegido, el EA1V-3, en la emblemática Peña de Francia a más de 1.700 metros SNM, nos ha servido para evaluar a lo largo del tiempo el correcto funcionamiento del programa. Desde su puesta en marcha ha estado suministrando, a través de la red, información que permite supervisar de manera remota los principales parámetros de funcionamiento de la instalación desatendida. En este caso las tensiones y temperaturas de operación, no solo del digi de APRS sino también del repetidor de fonía.

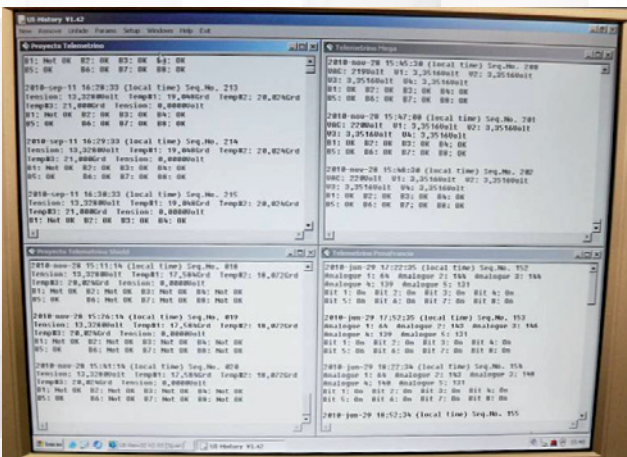


Foto 2

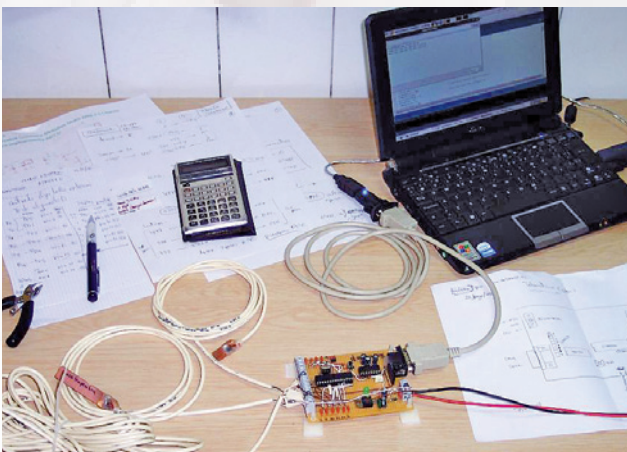


Foto 3

Además de los valores instantáneos, es posible revisar los gráficos que forman los datos históricos recogidos en alguno de los servidores de APRS existentes en la web [15]. Con estas herramientas los usuarios y administradores pueden monitorizar a distancia el funcionamiento de la instalación en todo momento (foto 4), con lo que ello representa en cuanto al ahorro de dinero y tiempo en el mantenimiento, pudiendo incluso llegar a prevenir posibles fallos de funcionamiento.

Una vez verificado el exitoso funcionamiento de nuestro prototipo, diseñamos un Telemetrino-*shield* compatible con la *pin-out* de la placa Arduino (foto 5). Un *shield* viene a ser una placa *sandwich* que se "enchufa" directamente sobre el Arduino, manteniendo su

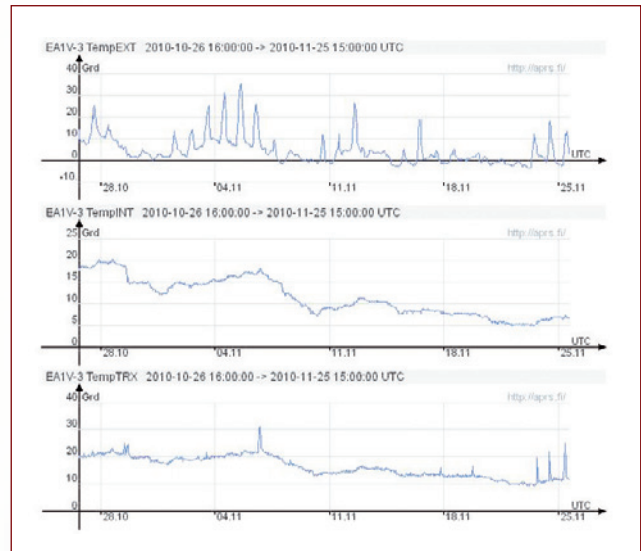


Foto 4



Foto 5

misma disposición de pines. El *shield* del Telemetrino se diseñó como un PCB configurable y permite adaptar las entradas del Arduino para conectar diferentes tipos de sensores o actuadores acorde con el funcionamiento del programa. También incorpora una interface RS-232 para enviar la información ya procesada hacia el puerto serie del TNC.

#### 4.1. ¿Cómo funciona?

Podemos decir que el Telemetrino es un sistema compuesto por un *shield* y un programa, es *plug&play*. Funciona como una "mochila" enchufable al puerto serie del TNC de un digi en activo. Periódicamente realiza las medidas de los parámetros que se le indiquen en el programa, los codifica según el protocolo APRS y proporciona al TNC una trama con el formato correcto para ser enviada a la red de APRS. Además genera los mensajes necesarios para que cualquier estación a la escucha del digi pueda decodificar correctamente la telemetría enviada. El TNC tan solo tiene que inyectar entre el tráfico de la red la información que el Telemetrino le suministra a través del puerto serie. No es preciso realizar ningún cambio en la programación del TNC.

#### 4.2. ¿Qué podemos llegar a hacer con él?

Esta versión del Telemetrino implementa funciones de telemetría básica y proporciona también a la red de APRS toda la información necesaria para decodificar y visualizar correctamente los datos. No obstante, la potencia de computación del Arduino permite extender esta capacidad hasta límites insospechados. Es posible recoger o medir multitud de variables del entorno así como de la propia estación de radio en la cual se instale, procesar esas medidas, evaluarlas y generar informes en texto plano con los resultados. Estos informes podrían ser inyectados en la red a modo de boletines periódicos o bien como mensajes ocasionales de alerta dirigidos a una estación concreta, por ejemplo el responsable del mantenimiento de la instalación. La telemetría permite evaluar de manera remota todos los parámetros importantes que determinan el buen funcionamiento de cualquier estación desatendida. Por ejemplo, podríamos monitorizar la ROE de las antenas (foto 6), conocer el estado de carga de las baterías o las fluctuaciones de la tensión de red, vigilar la tempera-

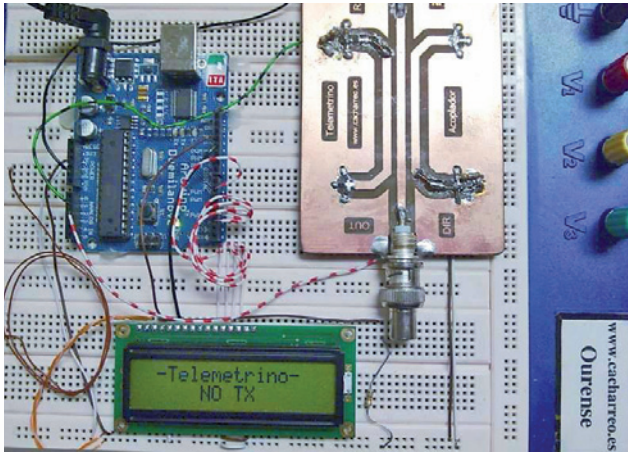


Foto 6

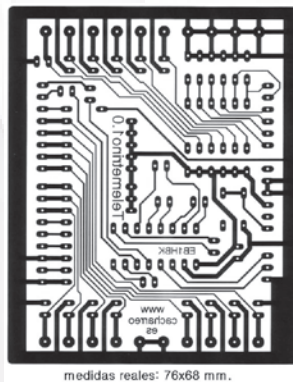


Foto 7

tura de los transmisores, realizar medidas meteorológicas, generar estadísticas de uso, etc.

Es posible inyectar toda esta información entre el tráfico habitual de la red de APRS o incluso como trama comprimida en forma de burst en la misma "cola" de un repetidor de fonía, activable a voluntad mediante el clásico tono de audio al hacer PTT. Incluso es factible hacer que el Telemetrino devuelva en la telemetría el nivel de señal con

el que llegaba al repetidor la última portadora que lo activó, permitiendo que las estaciones que lo utilizan disponer de una información objetiva e inmediata del nivel de señal con el que les escucha el repetidor. Estos son algunos ejemplos del uso que podemos darle al Telemetrino, tan solo hay que indicarle como hacerlo modificando adecuadamente el programa.

Hasta el momento hemos considerado el envío de telemetría como un camino unidireccional, esto es, el Telemetrino realiza una serie de medidas, codifica la telemetría y genera las tramas para que el TNC las envíe a la red, pero sus posibilidades son aún mayores. Basta ajustar el programa para simular un comportamiento inteligente a la hora de manejar la estación remota. Por ejemplo no solo puede informar del estado energético de las baterías, podría ocuparse totalmente de su gestión y administrar con eficiencia la reserva de carga, o conmutar a una antena secundaria si la principal falla. Incluso proceder a la desconexión total de los equipos de la instalación ante la inminencia de una tormenta eléctrica. No debemos olvidar que Arduino es en el fondo un completo computador de 8 bits con capacidad mas que suficiente para ejecutar todas estas tareas de manera eficiente y autónoma, basta con programarlo para ello.

En Orense, el equipo de cacharreo, hemos dado unos tímidos pasos en este sentido, pero no dejamos de sorprendernos de todo lo que, como radioaficionados, podemos llegar a obtener de algo aparentemente tan sencillo como un Arduino. Por ello desarrollamos el Telemetrino siguiendo la filosofía de código y hardware abierto [16]. El diseño del circuito impreso del *shield* (foto 7) y el código fuente del programa están disponibles en nuestra web para su descarga [17]. De este modo todo aquel que lo desee puede construir su propio Telemetrino y contribuir también a mejorar su funcionamiento. La placa del *shield* responde a un diseño flexible y está concebida para usarse también como placa de E/S de propósito general para otras aplicaciones con Arduino.

El Telemetrino puede usarse libremente de acuerdo con las

condiciones de la licencia original de Arduino. En concreto el programa se acoge a la licencia GPL versión 3 [18], y el diseño del *shield* tiene licencia Creative Commons-Attribution-Share-Alike [19].

Este artículo fue escrito por EB1HBK y revisado por EB1WM, el texto y las fotos tienen licencia Creative Commons-BY-NC-ND [20]

## Agradecimientos

A este proyecto han aportado sus contribuciones, por orden alfabético, los siguientes colegas: EA1GDH (Alberto Vila), EB1AJP (Antonio Díaz), EB1HBK (Javier Moldes) y EB1WM (Marcos X. Álvarez). Quisiéramos expresar nuestro reconocimiento a todos los que nos han prestado su ayuda y apoyo en este proyecto, especialmente a los compañeros de URSA, por su actitud abierta y entusiasta con el Telemetrino, incorporándolo a su red de APRS para evaluar su funcionamiento. Al CAM por su colaboración en el proyecto TRITON -germen del Telemetrino-. Y por supuesto al resto de componentes habituales del equipo de cacharreo.es: EA1AY, EA1DBB, EA1DG y EA1HVT.

## Referencias

- [1] Descripción del Antitracker por sus desarrolladores: <http://www.radio-active.net.au/web/tracking/antitracker.html>
- [2] Nuestras impresiones sobre el Antitracker: <http://www.cacharreo.es/antitracker>
- [3] TinyTrak3 de Byonics: <http://www.byonics.com/tinytrak/>
- [4] Especificaciones del OpenTracker+: <http://www.argentdata.com/products/otplus.html>
- [5] Nuestras reexiciones sobre el OpenTracker+: <http://www.cacharreo.es/opentracker>
- [6] Comparativa entre el OpenTracker+ y el TinyTrak3: [http://www.cacharreo.es/es/comparativa\\_OT+\\_TT3](http://www.cacharreo.es/es/comparativa_OT+_TT3)
- [7] Manual del OpenTracker+ traducido al español: [http://www.cacharreo.es/Manual\\_es\\_OT+](http://www.cacharreo.es/Manual_es_OT+)
- [8] Codificador para telemetría APRS de EA3CNO: <http://www.comunicacio.net/digigrup/aprs/ea3cno.htm>
- [9] Telemetría y el programa Ui-View: [http://www.comunicacio.net/digigrup/aprs/telemetr%EDa\\_y\\_ui-view.htm](http://www.comunicacio.net/digigrup/aprs/telemetr%EDa_y_ui-view.htm)
- [10] Transmisor Informativo de Telemetría de OureNse: <http://www.cacharreo.es/ProyectoTriton>
- [11] APRS Telemetry System. Using the Micro.Interface.Module (MIM): <http://www.aprs.net/vm/DOS/TELEMETRY.HTM>
- [12] APRS+: Tutorial de Telemetría en APRs en español: <http://www.cacharreo.es/APRS+Telemetría>
- [13] Arduino: plataforma abierta de desarrollo de prototipos electrónicos: <http://www.arduino.cc/es/>
- [14] Ham-Radio Salnés 2010: [http://www.cacharreo.es/Ham\\_Radio\\_Salnes\\_2010](http://www.cacharreo.es/Ham_Radio_Salnes_2010)
- [15] Telemetrino en la Peña de Francia: <http://aprs.fi/telemetry/a/EA1V-3?range=week>
- [16] Arduino FAQ: <http://arduino.cc/es/Main/FAQ>
- [17] Diseño del Telemetrino: <http://www.cacharreo.es/Telemetrino>
- [18] Licencia Pública General GNU: <http://www.viti.es/gnu/licenses/gpl.html>
- [19] Creative Commons: Attribution-Share Alike: [http://creativecommons.org/licenses/by-sa/3.0/deed.es\\_ES](http://creativecommons.org/licenses/by-sa/3.0/deed.es_ES)
- [20] Creative Commons: Attribution-Noncommercial-No Derivative Works: [http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es\\_ES](http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es_ES)