



Una estación remota compartida para modos digitales

F. Tusell
EA2BEE



Introducción

Las dificultades en algunos entornos urbanos para erigir buenos sistemas radiantes, el creciente ruido radioeléctrico en las ciudades y la difusión, mayor velocidad y abaratamiento del acceso a internet son razones que explican el interés creciente por la instalación de estaciones de radioaficionado remotas para ser operadas vía internet.

Este trabajo documenta la instalación de una estación remota para (algunos) modos digitales. Más que proporcionar una receta concreta, que puede no convenir a muchos, se detallan varias alternativas, examinando alternativas de conexión vía internet. La motivación arranca de la lectura de un hilo en el foro Técnico de la URE, iniciado el 4 de septiembre de 2022 («Control remoto de estación de radio. ¿Existe solución para *router* con IP condicionada por CG-NAT?»). Aunque lo que a continuación se describe es una instalación más simple que la contemplada en dicho hilo, se abordan algunas de las cuestiones que allí se tocaban. Un artículo posterior describirá el procedimiento de instalación de una estación remota con Remote Rig, que incorpora algunas dificultades adicionales.

El proyecto: motivación y materiales

URV-ABRA, Sección local de URE en Vizcaya, mantiene tres ubicaciones en los montes Oiz, Garbea y Arnotegi, dando servicios de repetidor y DMR a buena parte de la provincia y más allá. De modo natural surgió la idea de aprovechar alguno de esos excelentes emplazamientos para otros cometidos; inicialmente pensamos en una estación basada únicamente en un receptor SDR, accesible remotamente mediante una *interface* como OpenWebRX (<https://www.openwebrx.de/>).

Este objetivo evolucionó a otro más ambicioso: una estación que permitiera hacer radio remotamente. Pensamos en especial en aquellos socios que pueden tener dificultad para instalar antenas de gran dimensión en sus domicilios urbanos, o que sufren un ruido que hace difícil o imposible hacer radio.

Para implementar dicha idea con un presupuesto limitado, nos fijamos en el transceptor para modos digitales QDX, ofrecido tanto en kit como montado por QRP Labs (<https://qrp-labs.com/>). Es un aparato que en kit puede obtenerse bien por debajo de 100 euros, algo conveniente ante el riesgo de robo o destrozo vandálico que corren equipos en ubicaciones de montaña. Tiene poca potencia (algo menos de 5 W) y permite trabajar modos digitales FSK de tono único, incluyendo todo lo que ofrece el programa Wsjt-x, JS8Call y algunos otros modos disponibles en el programa fldigi, como RTTY y Olivia. (No se dispone en cambio de modos multi-tono, como PSK31.)

Lo hay en dos versiones para las bandas entre 20 m y 80 m o entre 10 m y 20 m. La versión escogida por nosotros fue la primera, consecuentes con el objetivo de facilitar hacer radio en las bandas que requieren antenas de mayores dimensiones.

Para pilotar dicho transceptor, hace falta un ordenador. Por razones de precio y para mantener el consumo de energía en valores asumibles descartamos el uso de un PC convencional, optando por una Raspberry (<https://www.raspberrypi.com>). Hay alternativas, como la gama OrangePi (<https://orangepiweb.es>). Es importante escoger un modelo con capacidad suficiente, pues la decodificación de FT8, por ejemplo, puede exceder de las prestaciones de algunos de los modelos más bajos de la gama. Una Raspberry Pi 4 o equivalente es suficiente; nosotros hemos empleado una Raspberry 400, ante las escaseces que hicieron imposible adquirir una Raspberry 4 en el momento de acometer este proyecto.

Implementación

Describimos a continuación la interacción entre el servidor (la Raspberry) controlando el QDX (lado derecho del esquema figura 1) y la estación cliente (lado izquierdo del esquema). Abordaremos primero los detalles comunes a todas las formas de conexión, para detenernos luego en las variantes precisas según se emplee en el lado remoto una conexión cableada de fibra óptica o cobre, con dirección IP pública, fija o no; o una conexión basada en un *router* inalámbrico 4G, que casi invariablemente nos proporcionará solo una dirección IP no pública.

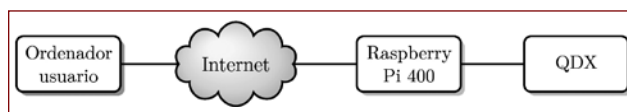


Figura 1. Esquema de la instalación. Se hace abstracción del modo de conexión de ambos extremos a internet

Conexión en la parte remota

La conexión entre la Raspberry y el QDX se reduce a un cable USB con conectores de tipo A y B. No es preciso hacer uso de la entrada de sonido de la Raspberry, el QDX incorpora todo lo necesario. Fuero de ello, solo es preciso proporcionar alimentación a ambos aparatos. Para la Raspberry servirá un cargador de tipo telefónico capaz de proporcionar 2A a 5V. El QDX puede construirse alternativamente para ser alimentado a 9V o 12V, de manera que habremos de prever una segunda alimentación del voltaje preciso.

La conexión de la Raspberry a internet puede hacerse por Wi-Fi o cable: el modelo concreto empleado incorpora Wi-Fi, si fuera preciso usarla. Finalmente, conectaremos el QDX a la antena, para lo que dispone de un BNC en su parte trasera.

Software en la parte remota

La Raspberry soporta diferentes sistemas operativos, en general variantes de Linux. Poco importa cual se emplee: nosotros hemos utilizado Debian (versión 11, apodada *bullseye*), pero puede utilizarse otra. No es éste el lugar para describir una instalación desde cero: hay abundantes tutoriales en internet y *Radioaficionados* ha publicado también en los años 2021 y 2022 una serie de artículos de Anthony Le Cren, F4GOH, sobre la utilidad de la Raspberry Pi en el cuarto de radio. En particular, la entrega de dicha serie correspondiente a marzo de 2022 describe una instalación de solo recepción con bastantes similitudes a la que detallamos a continuación.

Seleccionaremos un conjunto de *software* mínimo al instalar la Raspberry: omitiremos cosas como servidores de impresión, programas de ofimática, o navegadores. Necesitaremos un *software* para establecer conexiones remotas seguras, como SSH; normalmente forma parte de una instalación standard. Entre las cosas que no se instalan por omisión seleccionaremos un emulador de terminal como minicom (hay quien prefiere Putty), el programa WSJT-X y x2go (paquetes x2go-server y sus dependencias). Solo la “mitad servidor” hace falta en la estación remota, y no requiere ninguna configuración: toda se hace en la “mitad cliente” a la que nos referiremos más abajo.

Por lo que hace a WSJT-X, cada usuario debe configurarlo de acuerdo con sus preferencias (indicativo de cada uno, colores, ventanas visibles, etc.). Algunas opciones sin embargo vienen condicionadas por el *hardware* y se describen a continuación.

► En los ajustes, la pestaña “Radio” deberá quedar como ilustra la figura 2.

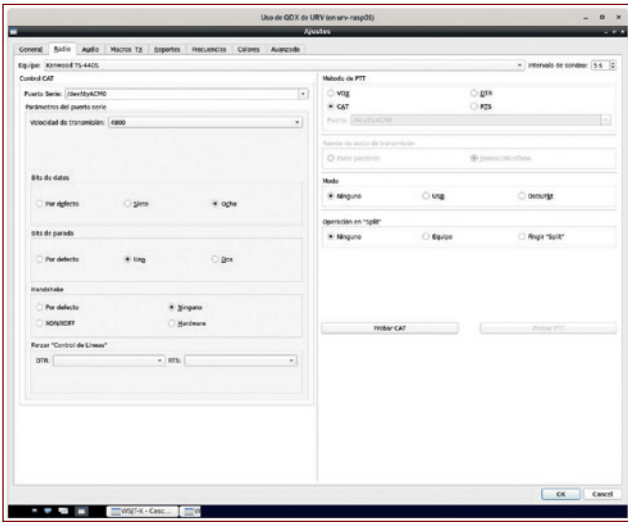


Figura 2. Configuración WSJT-X para el QDX (I)

El QDX emula a un Kenwood TS-440, que puede señalarse como Equipo. (Versiones modernas de Wsjt-x, a partir de la 2.6.1, ya incluyen no obstante el QDX como equipo, por lo que podremos seleccionarlo directamente. Las versiones incluidas con la mayoría de las distribuciones de Linux son relativamente antiguas y se requerirá señalar la emulación.) El nombre del puerto serie puede cambiar dependiendo del tipo de Raspberry que empleemos.

- ▶ La pestaña “Audio” deberá quedar como en la figura 3. Seleccionaremos como dispositivos de entrada y salida los que nos aparezcan en las opciones que incluyan “QDX” en su nombre; de nuevo, la denominación exacta puede variar.

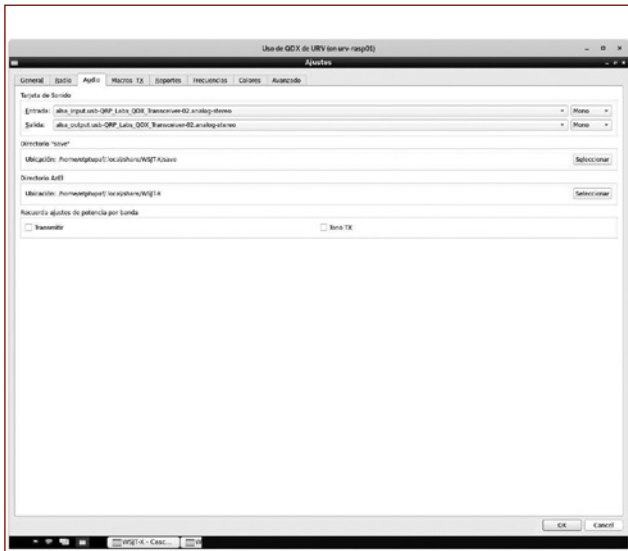


Figura 3. Configuración WSJT-X para el QDX (II)

Software en la estación local

Todo lo que necesitamos es un emulador de terminal gráfico, capaz de conectar a la Raspberry remota y ejecutar allí Wsjt-x. Utilizaremos en el ordenador local el cliente de X2Go. Si la máquina local es Linux, instalaremos el paquete x2go-client o de nombre similar de la distribución que empleemos (Debian, Ubuntu...). Si empleamos otro sistema operativo, como Windows, podemos obtener el programa de <https://wiki.x2go.org/doku.php>. Describimos a continuación la configuración (muy simple) que requiere, señalando las variantes que introducen los diferentes modos de conexión más abajo.

Enlace con la estación remota

El programa x2go cliente nos permite definir características de sesiones sobre ordenadores remotos. Una vez hecho, establecer la conexión se reduce a un “clic”.

Dependiendo de la infraestructura disponible, la dificultad de establecimiento de la conexión oscila entre trivial y moderadamente compleja. La modalidad de enlace que se emplee afecta sólo a una de las pantallas de configuración de x2go. Describimos las diferentes posibilidades a continuación.

La primera pantalla de configuración se muestra en la Figura 4. Solo es necesario lo que se señala: la dirección IP de destino y la puerta de acceso. Podemos, desde luego, hacer uso de acceso mediante SSH y clave pública, que nos exime de teclear contraseñas, pero no es preciso.

El ejemplo que se presenta en dicha Figura presupone la Raspberry detrás de un router con dirección IP pública y conocida, y SSH escuchando en el puerto 22. La Raspberry se hará accesible desde el exterior abriendo en el router el puerto 22 y dirigiéndolo a la Raspberry: es el único que es preciso abrir. Puede también colocarse en la “zona desmilitarizada” (DMZ) pero suele desaconsejarse por cuestiones de seguridad.

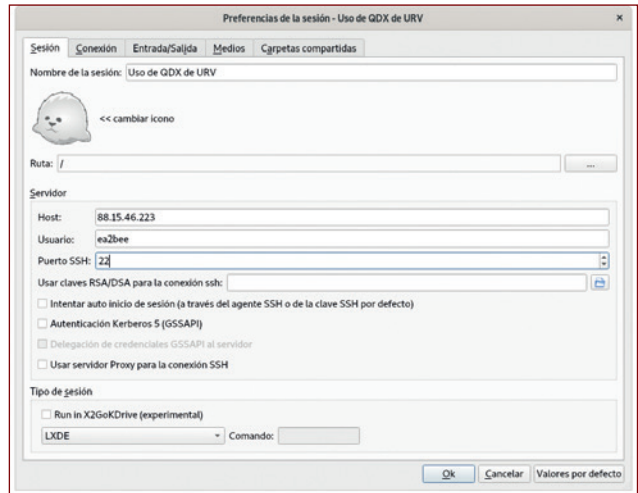


Figura 4. Configuración x2go para acceder al QDX (I)

La figura 5 permite configurar la velocidad de la conexión, desde muy lenta (módem) a muy rápida (LAN). Será preciso experimentar aquí para encontrar la que mejor se adapta a la velocidad de nuestro enlace. En el número de colores, señalar 64k es más que suficiente. Señalar un elevado número de colores puede reducir la velocidad.

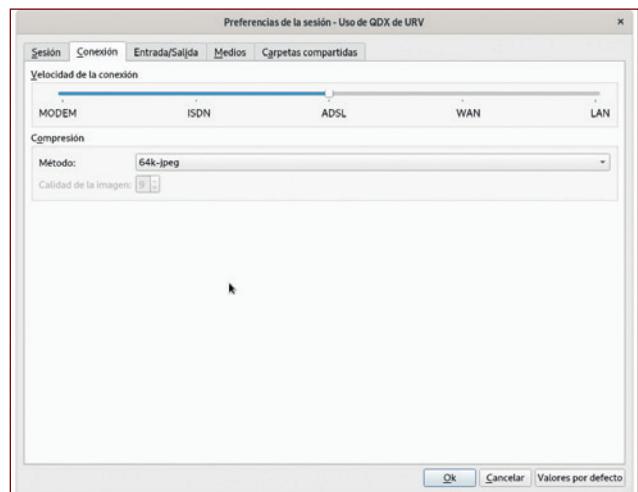


Figura 5. Configuración x2go para acceder al QDX (II)

La figura 6 muestra la configuración de entrada salida. Las dimensiones de pantalla que se han consignado son las mínimas aconsejables para el uso de un programa como Wsjtx. Si el tamaño de nuestra pantalla lo permite, seleccionaremos mayor número de píxeles en ambas dimensiones horizontal y vertical.

Podemos permitir al programa que autodetecte nuestro teclado, lo que habitualmente hace. Si es preciso, podemos seleccionar uno nosotros, de acuerdo con la máquina que utilicemos: 'pc105' y lenguaje 'es' será probablemente una buena elección.

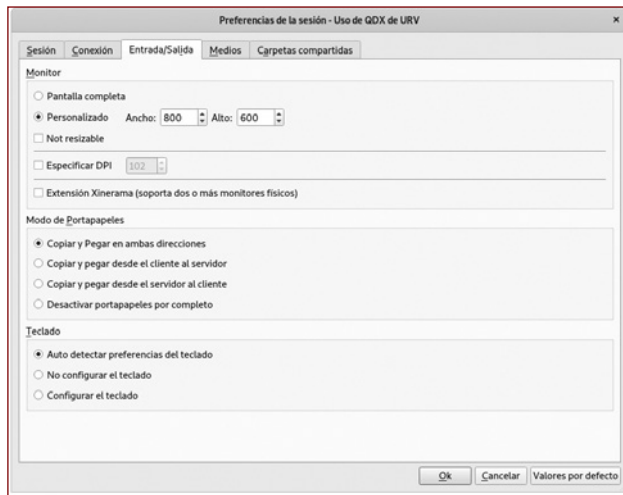


Figura 6. Configuración x2go para acceder al QDX (III)

En la siguiente pantalla (figura 7) es importante que desactivemos el sonido: no queremos que sea enviado a nuestra máquina local desde la remota, sino que sea tratado allí por el programa de modos digitales que empleemos, como wsjtx. Podemos retener o no la capacidad de copiar y pegar e imprimir entre nuestro ordenador y la Raspberry remota.

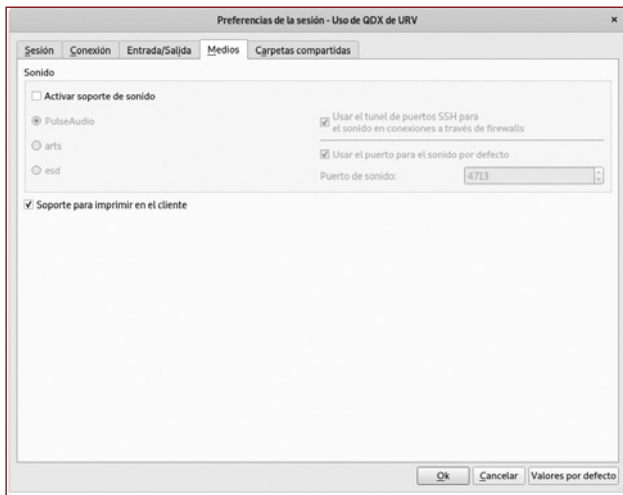


Figura 7. Configuración x2go para acceder al QDX (IV)

Finalmente, podemos dejar la pantalla “Carpets compartidas” como muestra la figura 8. El programa permite “montar” carpetas locales en la Raspberry remota, lo que puede tener utilidad dependiendo del programa de log que empleemos para anotar directamente en nuestra máquina local los QSO que realicemos. De otro modo, dejaremos la pantalla en blanco, tal cual se muestra.

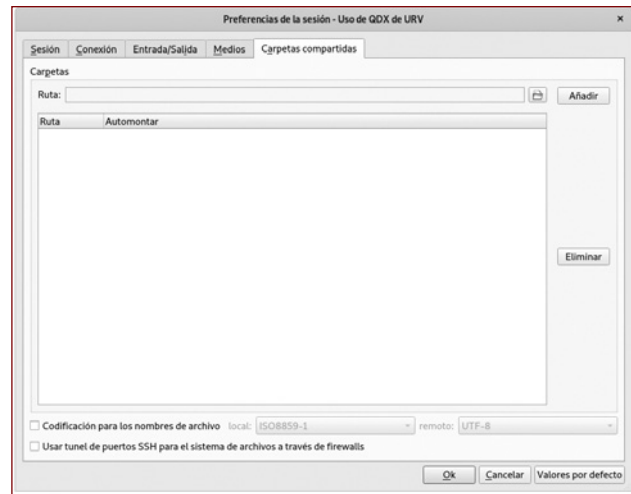


Figura 8. Configuración x2go para acceder al QDX (V)

Examinamos a continuación las variantes precisas según el modo de conexión.

La estación remota dispone de conexión con dirección IP pública y fija

Este será (frecuentemente) el caso si la ubicación remota dispone de internet mediante fibra o ADSL, o (más raramente) si el acceso es mediante un módem 4G/5G. En este último caso son pocos los proveedores de Internet que proporcionan tarjetas con IP fija y pública y normalmente lo hacen con un considerable sobreprecio.

En este caso, basta señalar dicha dirección IP en la pantalla mostrada en la figura 4.

Para no tener que recordar la dirección IP numérica, suele recurrirse al servicio de resolución de nombres de Internet, o DNS. Si somos propietarios de un dominio, haremos que se resuelva a la dirección IP en cuestión y escribiremos dicho dominio en la casilla correspondiente en lugar de la dirección IP.

La estación remota dispone de conexión con dirección IP pública pero no fija

Actuaremos como en el apartado anterior: aquí la dirección IP es pública y accesible desde cualquier lugar de internet, pero cambiante: típicamente, permanece inalterada durante largos periodos, pero cambia ocasionalmente y cuando reiniciamos el router. De nuevo, si disponemos de un dominio, nos interesará hacer que apunte a la dirección que en cada momento tenga nuestro router. El servicio DDNS (o servicio dinámico de nombres) facilita esto: nuestro *router* contactará regularmente al prestador del servicio DDNS notificando su dirección IP. Si no disponemos de un dominio de nuestra propiedad, podemos hacer uso de alguno de los proveedores de DDNS que nos prestarán este servicio de modo gratuito o mediante pago, con un nombre de dominio facilitado por ellos.

Si el *router* que empleamos no permite contactar al proveedor de DDNS de nuestra elección, podemos encargarnos a la Raspberry que lo haga. (En Debian, hay un paquete especializado en esto, *ddclient*.)

La única modificación que tenemos que hacer en este caso es señalar en la pantalla de la figura 4 no una dirección IP, sino un nombre de dominio que se resuelva a la dirección pública correcta en cada momento.

Conviene aclarar que para un uso compartido, la IP pública (fija o no) es casi un requisito. Carecer de ella obliga a instalar una *software* adicional a cada usuario e integrarse en una red privada virtual. Para un uso individual, o de un reducido número de usuarios, es sin embargo factible el uso de internet con IP no pública en el lado remoto, como se indica a continuación.

La estación remota no dispone de conexión con dirección IP pública

Este es el caso habitual cuando el extremo remoto está detrás de un *router* 4G/5G, inaccesible tras una capa de CG-NAT. La dirección IP que recibe el *router* es compartida con otros y no resulta accesible a conexiones entrantes. ¿Hemos de abandonar por ello toda esperanza de conectar a un equipo remoto para el que solo contamos con un *router* 4G? No, como vamos a ver.

En primer lugar, la IP de la tarjeta no puede recibir conexiones iniciadas externamente, pero sí puede iniciarlas. Si nos basta acceder a la estación remota desde un solo ordenador con IP pública, podemos hacer que sea la estación remota la que inicie la conexión a este ordenador. Este tipo de soluciones se implementan mediante túneles SSH. Hay amplia documentación en internet.

Alternativamente, podemos hacer que la máquina remota se conecte a una red privada virtual (VPN) a la que también conectaremos nosotros desde el extremo local. Hay abundante *software* para crear VPNs, siendo algunos de fuente y uso libre como OpenVPN y SoftEther; pero para hacer uso de dichas VPN hemos de contar con una máquina “en el exterior” con dirección IP pública de modo que al menos en un primer momento la Raspberry remota pueda establecer una conexión. Esta máquina puede ser nuestro cliente local, en el caso de que tenga dirección pública.

Cuando incluso esto falla, hay servicios, como ZeroTier, <http://zerotier.com/es/>, gratuitos para redes de hasta 25 usuarios y un sólo administrador, que permiten la creación de una VPN en unos pocos clics. Esta VPN ya cuenta con servidores externos, no necesitamos preocuparnos de ello. Basta crear una cuenta, instalar un módulo en todas las máquinas involucradas (que pueden Windows, Linux, Mac, incluso Android) y añadirlas a la VPN. Este es un proceso fácil y visual que se realiza desde una página web tras registrarnos. y no lleva más allá de unos minutos.

Cuando la máquina remota (la Raspberry en nuestro caso) se conecta a ZeroTier, inicia una conexión al servidor: esto la hace accesible detrás del CG-NAT. Otras máquinas en la misma red virtual pueden ya acceder a ella.

Las direcciones en una VPN creada mediante ZeroTier son persistentes: podemos simplemente escribir la dirección de la Raspberry remota en la casilla correspondiente de la pantalla de configuración de x2go y tendremos un fácil acceso que atraviesa el CG-NAT. La velocidad no es la misma que obtendríamos sobre una conexión directa, pero sí suficiente en las pruebas que hemos hecho.

Pruebas y rendimiento

En el momento de escribir, la estación remota ha funcionado por espacio de dos meses en una ubicación temporal, con IP pública. Los socios que lo han deseado han recibido un usuario y contraseña en la Raspberry remota; lógicamente, han de cerciorarse de que no hay otros usuarios en la máquina haciendo uso del QDX, pues sólo uno puede hacerlo a la vez.

El *software* y la configuración se ha revelado estable y confiable, sin problemas reseñables; y aunque 4 W de salida es una potencia pequeña, incluso para modos como FT8, a veces sorprende. Un socio —EA2DR— realizó en los primeros días de pruebas un QSO con Australia, además de otros muchos con estaciones menos distantes. El que esto escribe ha recibido una contestación de ZL llamando con 4 W, aunque sin alcanzar a completar el QSO. Con 4 W y un simple dipolo para 40 m es fácil trabajar USA en FT8 por caminos oscuros. Realmente es sorprendente lo que se puede lograr en QRP con paciencia, buenas condiciones de propagación y antena, dedicación, y un poquito de suerte.

Como posibles cautelas, el QDX es poco tolerante a antenas mal adaptadas. Dado que el cambio de antena a distancia añade coste y complejidad a un proyecto como el nuestro, es aconsejable emplear una antena multibanda. Puede configurarse el QDX de manera que solo transmita en bandas predeterminadas, pero es engorroso si se cambia de antena y banda con frecuencia.●