

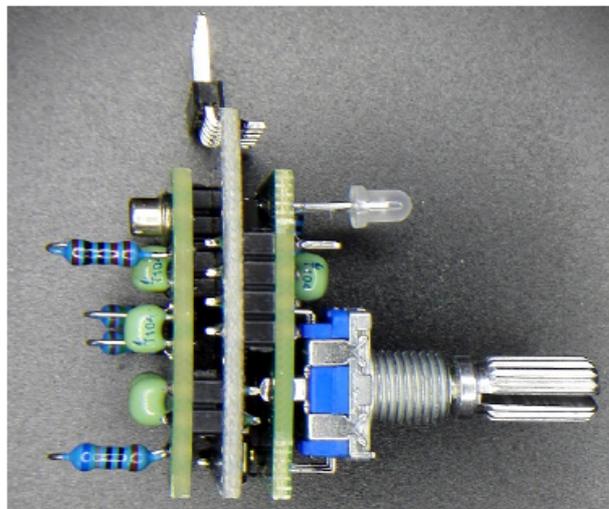
Manual de montaje del OFV digital Sandwich de CRKITS

Revisión C, 13 de abril de 2017

<http://crkits.com>

Traducción del original de Adam Rong BD6CR por Jon Iza, EA2SN

El OFV digital Sandwich es un interesante complemento de su radio KN-Q7A, CS-series u otro equipo de SSB. Al añadir el Sandwich Vd. puede acceder a toda la banda y dispone de un led indicador. La estabilidad de frecuencia es igual que la del cristal, con una pequeña deriva térmica si transmite de forma continua. El OFV digital Sandwich no produce tantas espurias (birdies) como un DDS. Permite generar la señal de OFB lo que le permite operar en USB y en LSB. También puede calibrar el cristal oscilador para que la frecuencia generada sea exacta en cada paso de 1 kHz o 100 Hz.



Lista de componentes

Para las placas de Control y Oscilador

- R1, R3, R4, R5, R6 = 1 k Ω
- R2, R7, R8 = 330 Ω
- C1 a C7 = 100 nF (104)
- X1 = cristal de 27 MHz
- D1 = Led de dos colores, 3 mm, vea en la siguiente página la identificación de patillas
- IC1 = Si5351A, montado ya en la placa
- SW1 = codificador rotativo con pulsador, 20 pulsos por revolución
- JP10 = Puente para seleccionar los límites de banda del OFV, OFB y frecuencias de operación iniciales, también conocido como Puente de la Región ITU. Vea en la página siguiente la información sobre su funcionamiento

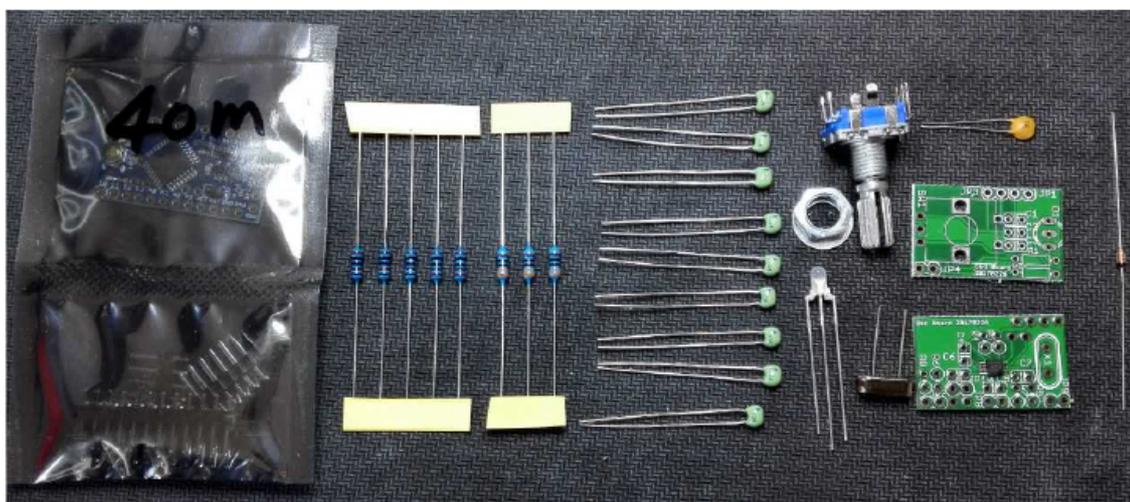
Componentes adicionales para el montaje en el transceptor:

- 2 condensadores 100 nF (104) para cablear al OFV y OFB
- 1 condensador 15 pF para filtrado adicional del OFV
- 1 diodo 1N4148 para evitar la corriente reversa en la alimentación (puede que se elimine del kit)

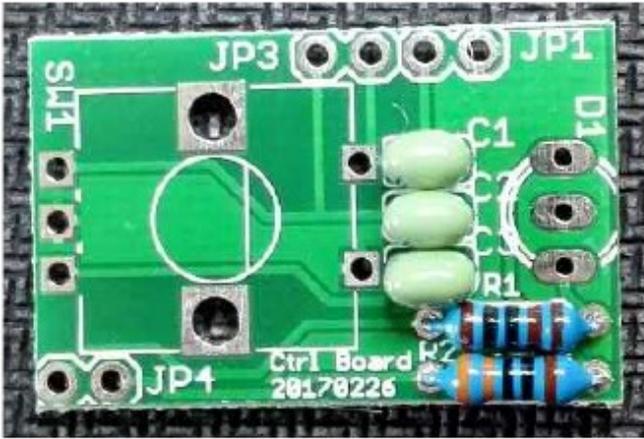
Arduino Pro Mini 3.3V/8MHz con el firmware pre-programado (opcional)

Programador USB-Serie FTDI con el circuito integrado FT232RL, tensión por defecto

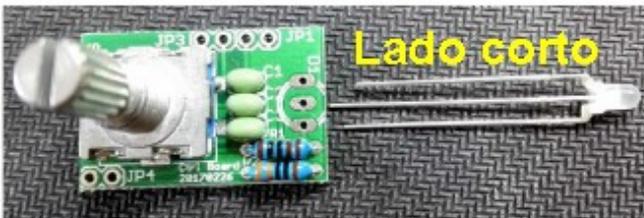
V_{CC} = 3,3 V (opcional)



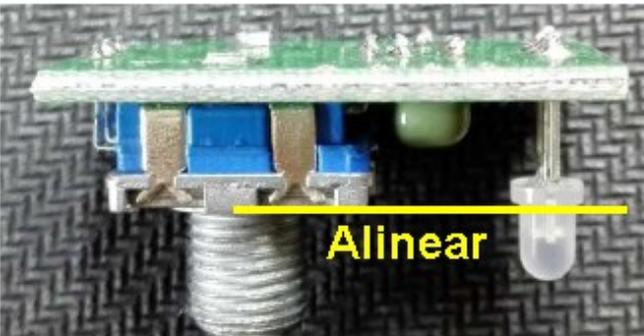
Montaje del kit



Comience con la placa de Control.
 Instale $R1 = 1\text{ k}\Omega$ y $R2 = 330\ \Omega$.
 Instale $C1 = C2 = C3 = 100\text{ nF}$ (104).
 Corte las patillas sobrantes a ras.



Instale SW1 = codificador rotativo con pulsador (encoder).



Instale D1 = led bicolor de 3 mm. La patilla más corta va en el agujero junto a la marca D1 (arriba).

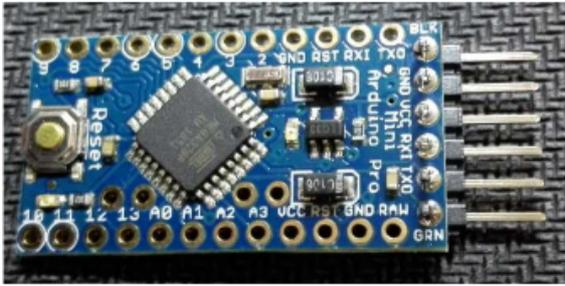
Deje 6 mm entre el cuerpo del led y la placa de circuito impreso. Así, el led quedará alineado con el encoder.
 Corte las patillas sobrantes a ras.



Continúe con la placa del Oscilador.
 El circuito integrado viene ya montado.
 Compruebe si hay puentes entre patillas o alguna soldadura fría..
 Instale $C4, C5, C6, C7 = 100\text{ nF}$ (104).
 Instale $R3, R4, R5, R6 = 1\text{ k}\Omega$.
 Instale $R7, R8 = 330\ \Omega$.
 Instale X1 = cristal de 27 MHz.
 Revise la tabla siguiente y, si es necesario, instale JP10 utilizando un rabillo sobrante.
 Corte las patillas sobrantes a ras.
 Se recomienda soldar los resistores tal como en la foto para facilitar su uso como puntos de prueba.

JP10	80 metros	40 metros	20 metros
Abierto (ITU R1,R3)	límite superior 3800 kHz frecuencia 3700 kHz OFB 8467,2 kHz	límite superior 7200 kHz frecuencia 7100 kHz OFB 8467,2 kHz	límite superior 14350 kHz frecuencia 14300 kHz OFB 4096,0 kHz
Cerrado (ITU R2)	límite superior 3900 kHz frecuencia 3700 kHz OFB 8467,2 kHz	límite superior 7300 kHz frecuencia 7200 kHz OFB 8192,0 kHz	límite superior 14350 kHz frecuencia 14200 kHz OFB 4194,3 kHz

Nota: esto aplica solo al KN-Q7A. El CS-Series usa 8467,2 kHz como FI en todas sus versiones: vea su manual y use sus "sketch" específicos para Arduino.



El kit completo viene con un Arduino Pro Mini ATmega328 3.3V/8MHz ya programado. Si Vd quiere actualizar el firmware, es un buen momento para instalar un conector en “peine” en L insertándolo por la cara inferior. Si no va a actualizar el firmware se puede saltar este paso. Conecte los cables al programador opcional FTDI 3.3V tal como muestra la fotografía.



Para la actualización (o la programación) necesitará el entorno de desarrollo integrado (IDE) de Arduino. Si no lo tiene instalado, descárguelo para su sistema operativo desde:

<https://www.arduino.cc/en/Main/Software>

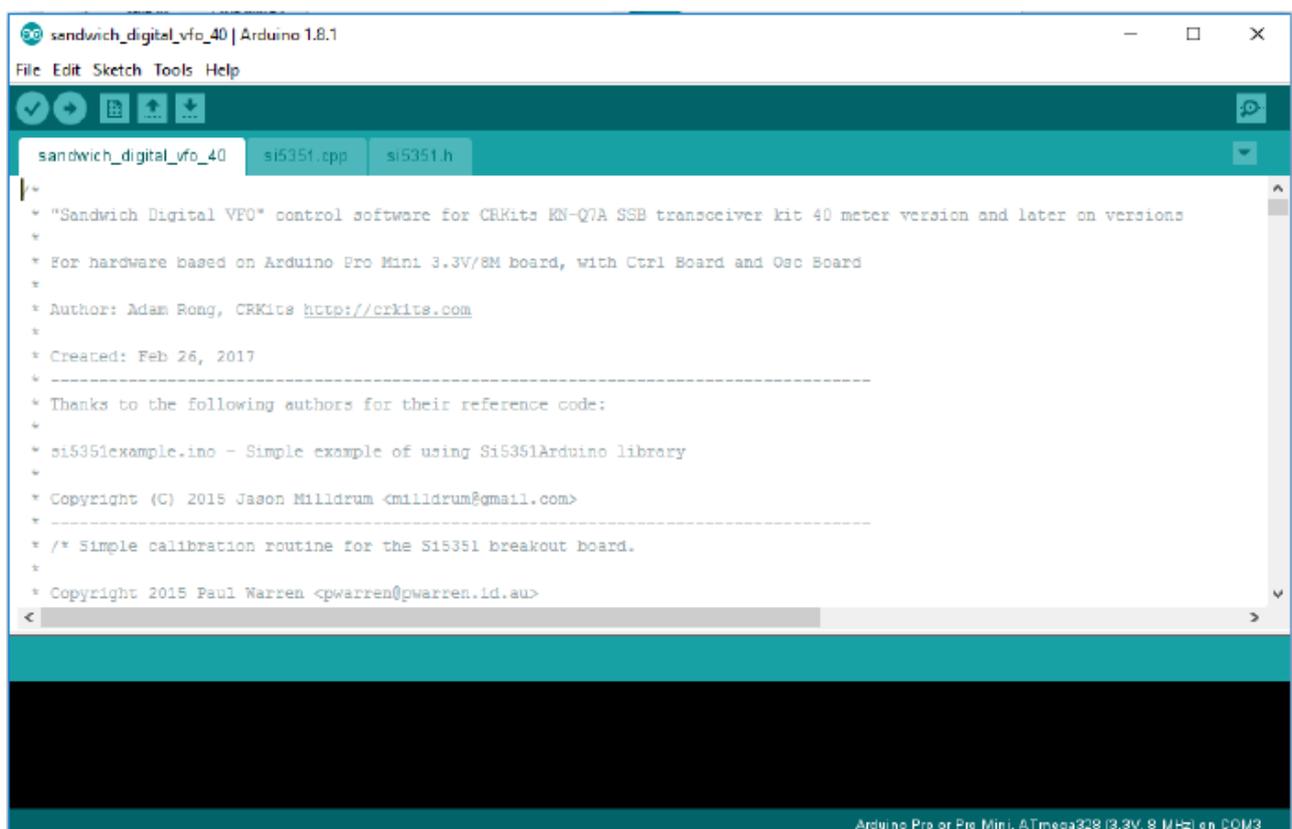
Conecte el programador FTDI a una puerta USB de su computador. Si necesita el driver, descárguelo e instálelo desde:

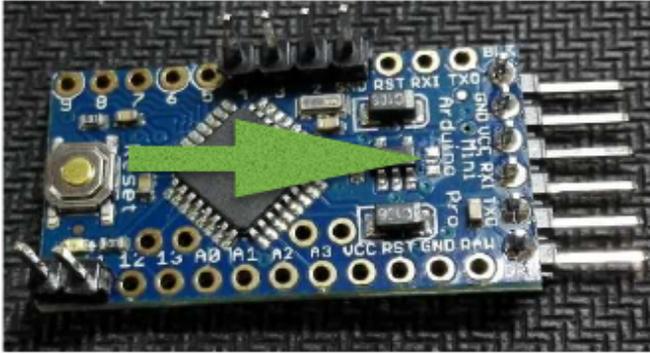
<http://www.ftdichip.com/Drivers/VCP.htm>

Descargue el “sketch” de Arduino (código fuente: elija la banda de su equipo) desde:

<http://crkits.com>

Abra el zip y haga doble clic en el archivo con la extensión “.ino”. Esto abrirá el IDE Arduino y cargará los archivos automáticamente. En la pestaña Herramientas (Tools), elija la Placa (Board): “Arduino Pro or Pro Mini” y, como procesador (Processor): “ATmega328 (3.3V, 8MHz)”. Elija también la puerta de conexión USB al programador serie, que será la que se ha creado al conectarlo. Una vez conectada su placa de Arduino Pro Mini al programador, haga clic en el segundo icono por la izquierda, arriba, (flecha hacia la derecha) para compilar y “subir” el firmware a su placa.





Instale un conector de 4 patillas desde GND hasta 4 en la cara superior.
 Instale un conector de 2 patillas entre las patillas 11 y 12, en la cara superior.
 Asegúrese de que están perpendiculares a la placa y bien asentados, y suéldelos.
 [Opcional] Quite el resistor SMD indicado por la flecha para reducir el consumo en 1,5 mA.
 Es un indicador de encendido y no es necesario en esta placa.



Instale un conector de 4 patillas desde GND hasta A3 por la cara inferior.
 Instale un conector de 2 patillas entre las patillas A4 y A5 por la cara inferior.
 Instale un conector de 2 patillas entre las patillas 6 y 7 por la cara inferior.
 Asegúrese de que están perpendiculares a la placa y bien asentados, y suéldelos.

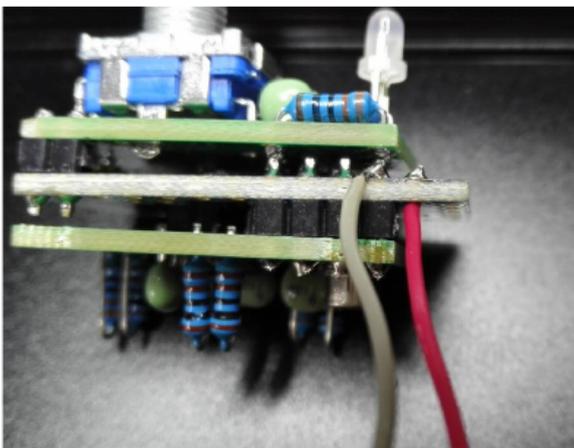
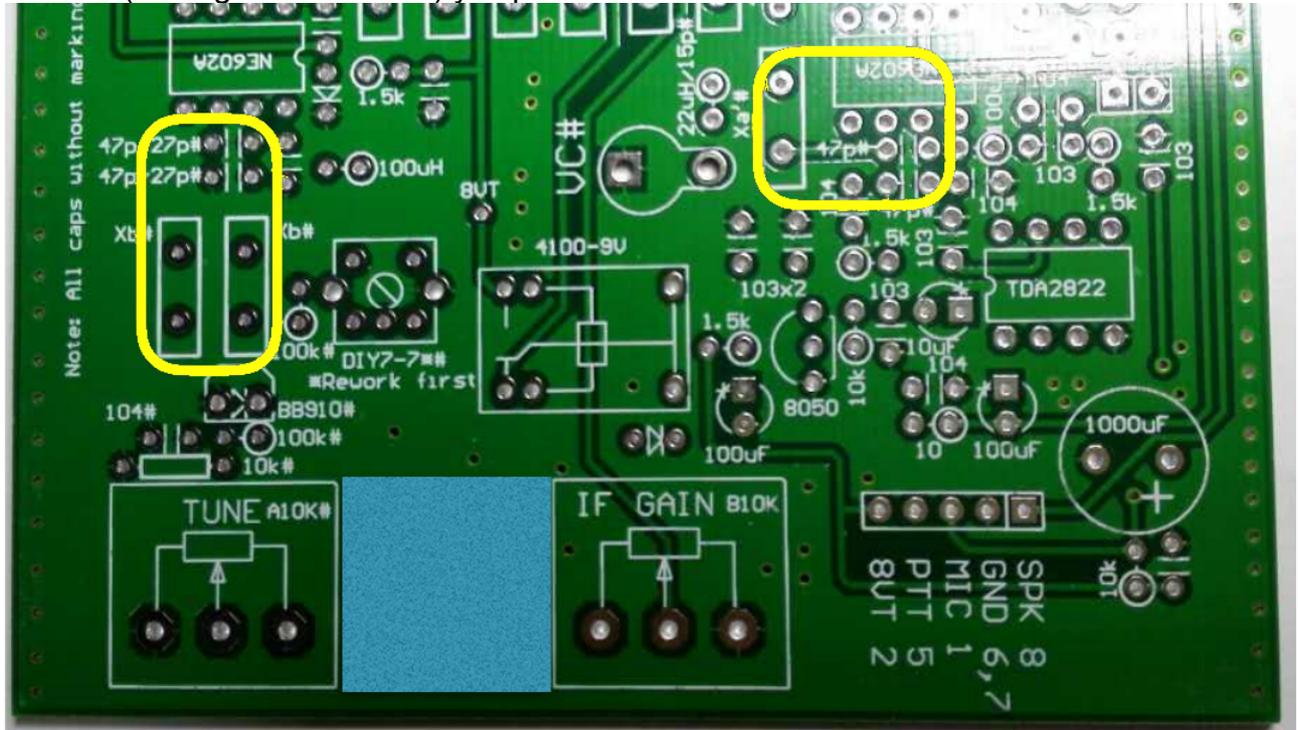


Enchufe la placa de Control en la cara superior de la placa de Arduino.
 Compruebe que no hay cortocircuitos entre las placas.
 Asegúrese de que están bien asentadas y suéldelas.
 Corte las patillas sobrantes a ras.

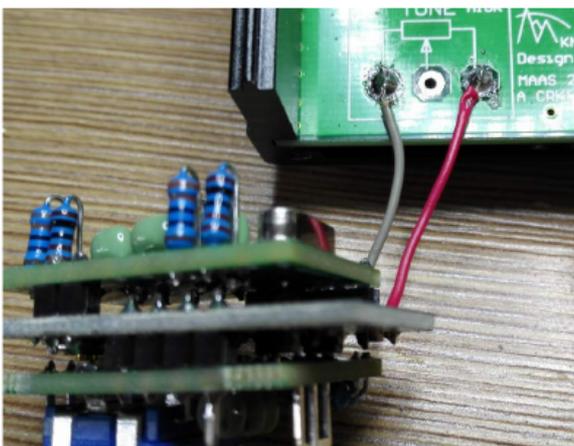


Enchufe la placa de Oscilador en la cara inferior de la placa de Arduino.
 Compruebe que no hay cortocircuitos entre las placas.
 Asegúrese de que están bien asentadas y suéldelas.
 Corte las patillas sobrantes a ras.

Modifique la placa principal del KN-Q7A para extraer los componentes en los recuadros marcados con un "#". En la zona del **VXO**, a la izquierda, quite **los dos condensadores de 47 pF o 27 pF**, **los dos cristales Xb**, el resistor de 100 k Ω , la bobina DIY7-7, el diodo BB910, un condensador de 100 nF (104), otro resistor de 100 k Ω , un resistor de 10 k Ω y **el potenciómetro de sintonía A10k**. En la sección del **OFB**, quite **los dos condensadores de 47 pF**, **el cristal Xa'**, la inductancia de 22 μ H o el condensador de 15 pF y el condensador ajustable VC. Como mínimo debe retirar los componentes que están dentro de los recuadros amarillos (en negrita en el texto) y el potenciómetro de sintonía..



Suelde dos hilos (de unos 5 cm de largo) a la placa de Arduino: Tensión positiva a la isleta RAW y negativo a GND.



Suelde los otros extremos a las isletas dejadas por el potenciómetro de sintonía A10K: masa, a la izquierda, y positivo a la derecha, para así alimenta la placa con 8 V.



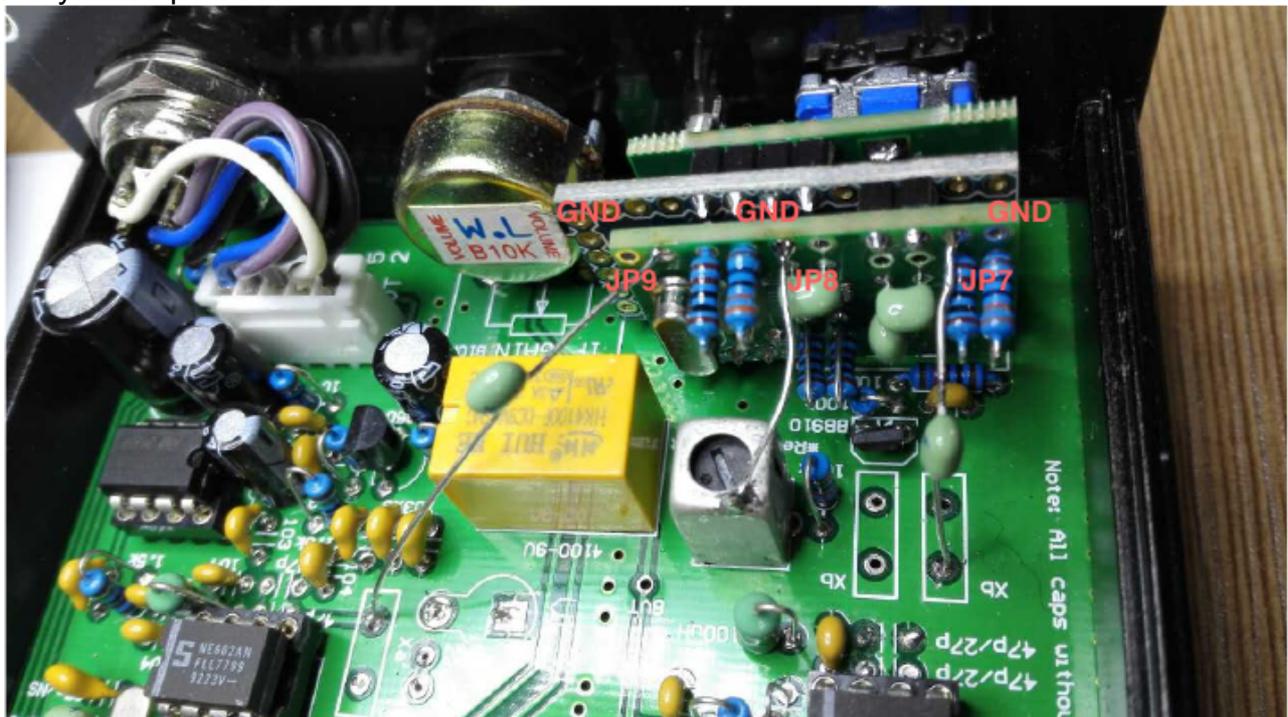
Use un condensador de 100 nF (104) para conectar JP9, isleta de la derecha, con el OFB (una de las isletas de Xa', tal como muestra la fotografía. La isleta derecha de JP9 (en la esquina) es masa (GND).

Conecte con otro condensador de 100 nF (104) la salida derecha de JP7 a la isleta de Xb que corresponde al OFV, como muestra la fotografía. La isleta izquierda de JP7 es masa (GND).

Si lo desea, puede mejorar la puesta a masa de Sandwich usando un cable de unión entre la isleta izquierda de JP8 y la inductancia metálica DIY7-7 (o la isleta de unión a masa, si la ha retirado).

El condensador de 15 pF que se ve en la foto se puede instalar en la posición indicada para reducir las espurias, pero es opcional. En algunas versiones previas del kit se

suministró un diodo 1N4148. Era para insertarlo en la línea de alimentación y bloquear la fuga de corriente cuando se hiciera la programación del Arduino una vez instalado. En la mayor parte de los casos no es necesario porque el regulador LDO de 3,3 V del Arduino lo lleva ya incorporado.



Necesitará perforar en el panel frontal un agujero de 3 mm de diámetro el led indicador. Debe hacerse en medio de los dos agujeros ya hechos en la placa para los potenciómetros.

Puede reutilizar el botón de Sintonía, pero uno de menor tamaño, como el usado para IF GAIN puede servir igualmente.

Descripción de las funciones

El programa actual (26 de febrero de 2017) soporta dos modos: Normal y Calibración. Este último se activa pulsando y manteniendo pulsado el pulsador del codificador rotativo mientras se da tensión al equipo.

Cuando se enciende el equipo es muy fácil saber en qué modo está: el led rojo indica modo Normal mientras que el naranja indica modo de calibración del OFB.

En modo normal, al pulsar el encoder se cambia los pasos de sintonía entre 1 kHz y 100 Hz, más que suficiente para su uso en SSB. En los pasos de 1 kHz el led se pone verde cuando se está en frecuencias impares (terminadas en 1, 3, 5, 7, o 9 kHz) mientras que está apagado para frecuencias pares (2, 4, 6, 8 kHz). El led se pone naranja cuando se pasa por las decenas de kHz (10, 20, 30, 40, 50, 60, 70, 80 y 90 kHz) y en rojo con las centenas y cuando se sale de banda: 0, 100 y 200 kHz. Cuando se está en pasos de 100 Hz el led se pone verde cada frecuencia par (2, 4, 6, 8 kHz). El siguiente gráfico le ayudara a entender esto de manera sencilla: la primera línea de colores es en paso de 1 kHz, la segunda es para paso de 100 Hz.

70	70	70	70	70	70	70	70	70	70	71	71	71	71	71	71	71	71	71	71	
90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10
Orange	Green	Red	Green	Orange																
Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green

En el modo de calibración, pulsando el encoder se alterna entre calibración del OFB y calibración del cristal. Los datos, según se va cambiando, se almacenan en EEPROM. En el modo de calibración del OFB el led estará naranja a no ser que se salga del rango predeterminado (se pondrá rojo), siendo los pasos de 100 Hz. En el modo de calibración del cristal, el led estará verde y los pasos serán de 20 Hz.

Calibrado

Mantenga pulsado el encoder mientras da tensión al equipo: el led se pondrá naranja, indicando que está en modo de calibrado del OFB.

Pulse de nuevo para cambiar a modo de calibrado del cristal. Conecte con cuidado un frecuencímetro a la isleta derecha de JP8 y gire el mando en el sentido contrario a las agujas del reloj unos 70 pasos. La frecuencia irá disminuyendo y se acercará a 10.00000 MHz.

Cuando llegue a la frecuencia exacta pulse de nuevo el botón para memorizar el calibrado del cristal. Desconecte el equipo.

Para calibrar el OFB, puede hacerlo de igual forma como hizo cuando ajustó su KN-Q7A.

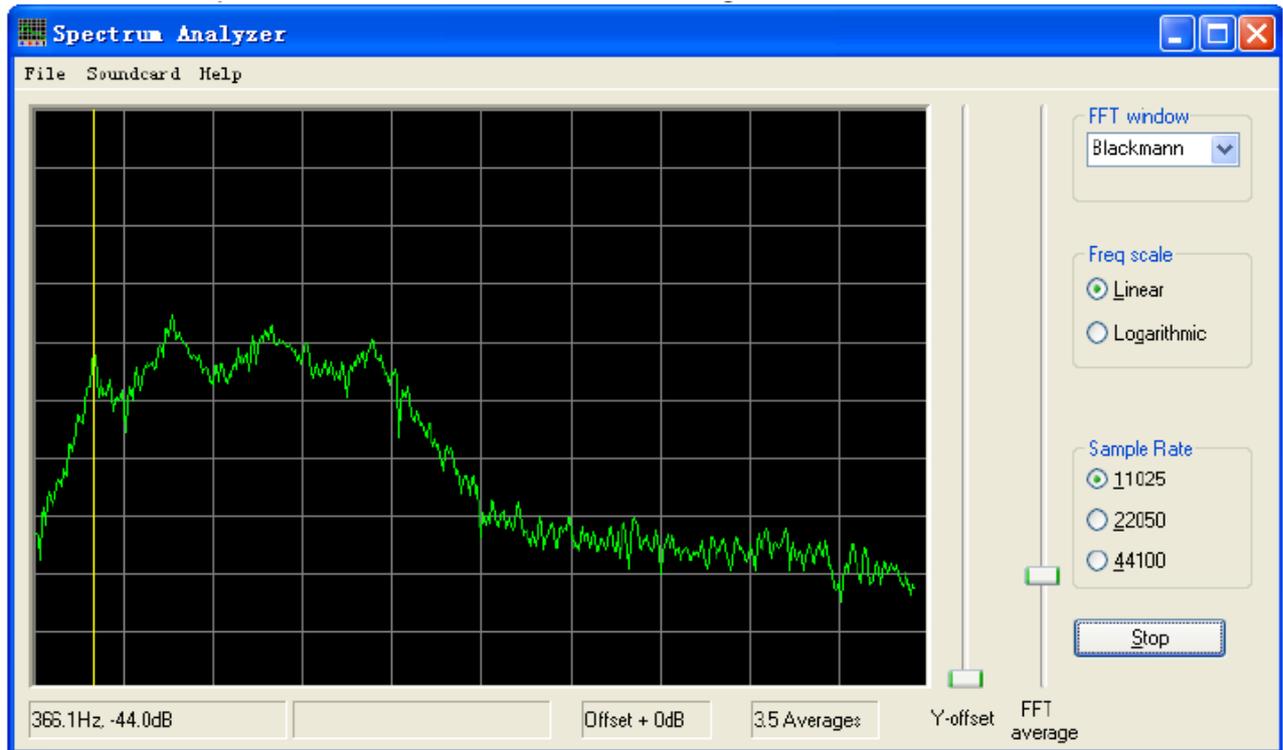
Necesitará descargar el programa Audio Spectrum Analyzer desde:

<http://www.qsl.net/zl1an/Software/Spectrum3.zip>

Abra el zip y haga doble clic en "spectrum2.exe" para ejecutarlo. Conecte el altavoz del equipo a la entrada de audio de la tarjeta de sonido de su PC. Necesitará también conectar a la antena del equipo un generador de ruido de RF o, simplemente, una antena.

Vuelva a entrar en el modo de calibrado manteniendo pulsado el encoder mientras vuelve a dar tensión al equipo. Use los mismos ajustes de la imagen a continuación y pulse "Start" para observar el espectro de audio recibido. Si no observa algo parecido necesitará ajustar el mando de IF GAIN del transceptor. Si usa una antena busque una frecuencia libre de transmisiones para poder escuchar el ruido de fondo. Después gire el encoder para desplazar la banda pasante hacia la izquierda o la derecha. Puede usar el cursor para leer la frecuencia y el nivel de la señal (en la esquina inferior izquierda), identificando dónde quedan los extremos de la banda pasante: Estos deben ser de, aproximadamente, 350-2200 Hz,

aunque puede ser algo más ancha o más estrecha. Asegúrese de que siempre el extremo inferior queda a 350 Hz. Como ahora el rango del OFB es mucho más amplio, es posible ajustar el equipo para usarlo en LSB o USB. Si Vd. puede ver un rango espectral de audio correcto pero no entiende las voces recibidas necesitará ajustar de nuevo el OFB porque está en la banda lateral incorrecta. En la mayoría de los casos los datos por defecto suelen estar muy próximos a los valores necesarios para el modo requerido, necesitando unos pocos pasos para un ajuste perfecto. Una vez que el ajuste sea correcto pulse el botón para memorizarlo en la EEPROM.



Posibles modificaciones

El Sandwich es un pequeño kit que permite disponer de un OFV digital. Si Vd. conoce bien el Arduino y cómo programarlo puede que quiera añadirle otros detalles.

En la placa del oscilador se pueden encontrar algunas patillas del Arduino sin utilizar, como la patilla analógica A3 o la patilla digital D6, que puede configurarse como una salida PWM. Posiblemente Vd. necesitará detectar el estado del equipo, Rx o Tx, para poder iluminar ledes de colores o para generar un desplazamiento para telegrafía CW. Posiblemente necesitará también detectar la potencia directa, la ROE y otros parámetros de operación. Otra idea interesante es generar una señal de audio en Morse como indicador de la frecuencia de operación, en lugar de contar pasos.

Lo divertido de este kit no es solo el montarlo, sino el modificarlo y compartir los resultados con otros, por ejemplo a través de http://groups.yahoo.com/group/CHINA_QRP/

Adaptación a otros equipos SSB

Tomemos como ejemplo el BITX-40. Tiene un OFB alrededor de 12 MHz mientras que el rango de frecuencias del OFV queda alrededor de 4,7 ~ 5,0 MHz para la región 2 de ITU y alrededor de 4,8 ~ 5,0 MHz para las regiones ITU 1 y 3. A diferencia del KN_Q7A, donde se obtiene la frecuencia final por diferencia entre OFV – OFB, en el BITX-40 la frecuencia final se obtiene por resta de OFB – OFV. Estos serían los cambios que habría que hacer en el “sketch” de Arduino:

```
if (digitalRead(JP)) //Para el BITX-40; JP se usa para seleccionar la región ITU
{
    op_freq = 7100000; //ajusta la frecuencia de operación por defecto
    band_high_limit = 7200000; //ajusta el extremo superior de la banda a 7,2 MHz
    bfo_freq = 11998000; //para una FI = 12000 kHz, por defecto en modo LSB
    bfo_low_limit = 11992000;
    bfo_high_limit = 12002000;
}
else
{
    op_freq = 7100000; //ajusta la frecuencia de operación por defecto
    band_high_limit = 7300000; //ajusta el extremo superior de la banda a 7,3 MHz
    bfo_freq = 11998000; //para una FI = 12000 kHz, por defecto en modo LSB
    bfo_low_limit = 11992000;
    bfo_high_limit = 12002000;
}
```

También hay que cambiar el código que ajusta la frecuencia de CLK0 en varias partes del “sketch” sustituyendo:

```
si5351.set_freq(uint64_t((bfo_freq + op_freq)*100), SI5351_PLL_FIXED, SI5351_CLK0);
```

por:

```
si5351.set_freq(uint64_t((bfo_freq - op_freq)*100), SI5351_PLL_FIXED, SI5351_CLK0);
```

Además de este cambio en el firmware, hay que ajustar también la tensión de salida del oscilador. Como el BITX40 usa un mezclador por diodos que requiere mayores tensiones tanto para el OFV como para el OFB, se recomienda eliminar los atenuadores resistivos compuestos por R5 a R8 de 1 kΩ y 330 Ω, usando un puente en lugar de los resistores de 1 kΩ. Tenga en cuenta que los resistores de 1 kΩ usados como “pull-up” para las líneas I2C (R3 y R4) no deben ser eliminados.