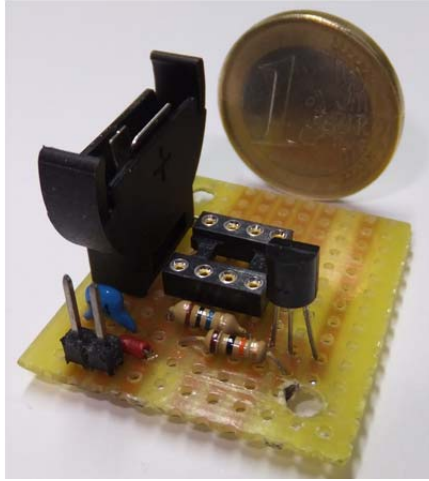


## KEYER EB5ABT (jaimorse)



Como aparte de la radioafición otras de mis aficiones es la electrónica y más concretamente los microcontroladores, desde hace ya mucho tiempo tenía ganas de embarcarme en el proyecto de un Keyer electrónico.

El pasado verano por fin me puse manos a la obra y empecé la construcción de este Keyer que he bautizado con el nombre de "jaimorse" ;).

Como en esto del CW me considero un dominguero, le pedí consejo al colega Pepe EA5AIO (que es un experto en el mundo de la telegrafía y ha construido y utilizado gran cantidad de manipuladores), y desde el primer momento empezó a darme sugerencias y a recomendarme distintos Keyers como ejemplo a seguir.

Por todo ello, antes de empezar a describir el circuito quiero agradecer a Pepe todo el tiempo que me ha dedicado y más sabiendo lo ocupado que está actualmente.

### CARACTERÍSTICAS KEYER

Las principales características de este keyer son:

- ✓ Bajo consumo.
- ✓ Circuito de pequeño tamaño y fácil de construir (tiene muy pocos componentes).
- ✓ 2 Memorias con un total de 118 caracteres.
- ✓ Seguridad ante bloqueo Paddle y pulsador transmisión memoria.
- ✓ Posibilidad de pasar de modo trabajo Keyer a modo baliza.

### ESQUEMA ELÉCTRICO

En la figura 1, podemos ver el esquema del Keyer.

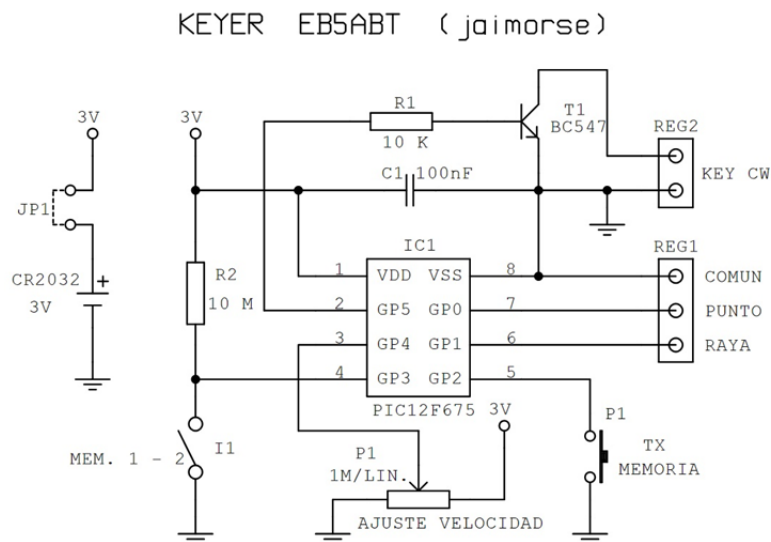


Figura 1

El circuito está basado en el microcontrolador Pic 12F675 de la casa Microchip.

Dada la sencillez del circuito, se puede realizar su montaje muy fácilmente en una placa perforada universal.

Como podemos observar, no se ha puesto un interruptor para desconectar la alimentación cuando no utilicemos el Keyer, ya que el consumo del circuito en reposo es de unos pocos microamperios.

Originalmente el circuito estaba alimentado con una pila de 9V y tenía un consumo en reposo de 3,5 mA, pero Pepe me hizo ver que con ese consumo la pila no duraría nada y me recomendó ver el Keyer de DL4YHF, que se alimentaba directamente con una pila de 3V, modelo CR2032.

Probé a alimentar el Pic directamente a esa tensión y como lo único que variaba era la frecuencia de trabajo del Pic, realizando unos pequeños cambios en el programa, finalmente pude alimentarlo de la misma manera.

El puente JP1 (jumper tipo placa ordenador) que está en serie con la pila, sirve para que podamos desconectar fácilmente la alimentación del circuito, ya que como veremos más adelante habrá situaciones en que el programa se quedará bloqueado y mediante este puente podremos resetear el Pic.

Recomiendo utilizar un zócalo para insertar el Pic (y no soldar directamente al circuito impreso), ya que al ser un dispositivo programable nos será más sencillo reprogramarlo si queremos cambiar los textos de las memorias, el indicativo, las futuras versiones de software (firmware), etc.

Si solamente necesitamos una memoria podríamos simplificar aún más el circuito, eliminando la resistencia R2, el interruptor de selección de memorias I1 y puenteando el pin 4 del Pic (GP3) al pin 1 (VDD).

Para activar la transmisión del equipo de CW donde se conectará el Keyer, se ha utilizado el transistor BC547, ya que es un elemento muy utilizado y fácil de localizar, pero si fuera necesario un transistor que soporte más intensidad, podríamos utilizar por ejemplo un MC140 y si procede, bajar el valor de la resistencia R1 a 4K7 ohmios.

Los altos valores de resistencia de R2 y del potenciómetro P1 son necesarios para obtener un bajo consumo.

Si para alimentar el circuito en lugar de la pila CR2032 vamos a utilizar pilas AA o AAA (incluso lo podríamos alimentar con un Power Bank para teléfono móvil [1] ) y el consumo no nos preocupa, podríamos cambiar los valores de R2 y P1 a 10K ohmios por ejemplo.

Lo que sí que es mandatorio, es que el potenciómetro utilizado sea lineal, para evitar saltos bruscos al regular la velocidad.

Otro punto importante que me recomendó Pepe, es el uso de condensadores de 1nF en caso de tener problemas con radiofrecuencia y el Pic se comportara de forma extraña al transmitir.

En tal caso podríamos colocar estos condensadores entre los pines de entrada del pic y masa (pines 4, 5, 6 y 7).

LISTADO COMPONENTES	
R1	10K
R2	10M
P1	1M / LINEAL
C1	100nF Poliester
T1	BC547
IC1	PIC 12F675
I1	Interruptor 1 circuito / 2 posiciones
P1	Pulsador N.A.
JP1	Jumper circuito impreso
Porta Pilas CR2032	
Pila CR2032	
C. audio hembra estereo 3,5 ó 6,3 mm	
C. audio macho mono 3,5 ó 6,3 mm	
Conector Altavoz	
Zocalo 8 Pins	

Figura 2

## MODIFICACIÓN TEXTO MEMORIAS TRANSMISIÓN

En primer lugar nos bajaremos el fichero que contiene el programa ejecutable del Pic (“KEYER\_EB5ABT\_\_V1\_0.hex”), de la siguiente dirección: <https://goo.gl/r9sWAO>.

Las dos memorias de texto tienen que ser editadas antes de programar el Pic, ya que posteriormente no es posible su modificación.

En futuras versiones del programa ejecutable, se podría incluir la función de edición de las memorias.

De todas formas pienso que por el precio que tiene el Pic, si necesitáramos más de dos memorias (para utilizar el Keyer en distintos concursos), nos sería más cómodo tener varios chips e ir intercambiándolos mediante el uso de zócalos de pin torneado.

A continuación, con la ayuda del programa IC-Prog [2], editaremos los mensajes de texto de las dos memorias que se encuentran situados como texto plano, en la memoria de datos del Pic.

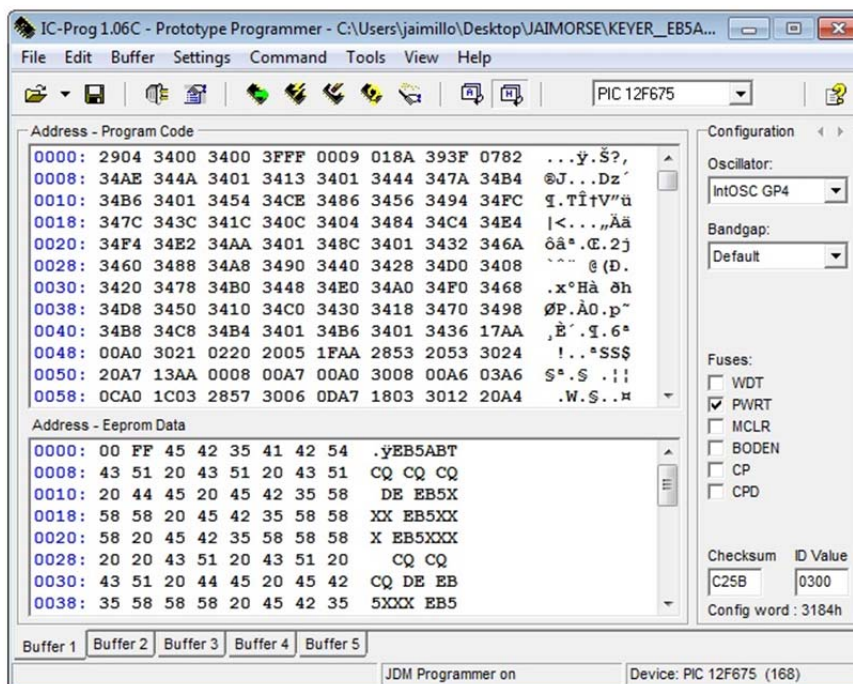


Figura 3

La dirección de comienzo de la primera memoria de texto es la 08.

Las 8 primeras direcciones (desde la 00 hasta la 07), están reservadas para el propio uso del programa y futuras modificaciones.

Las letras de las memorias tienen que estar en mayúsculas, y también se pueden utilizar números y demás símbolos utilizados en telegrafía.

La posición de fin de memoria (de ambas memorias) se establece mediante el valor 00 tras el último carácter a transmitir.

La dirección del comienzo de la segunda memoria (sin importar donde se encuentre) se sitúa tras el valor 00 de finalización de la primera memoria.

Simplificando, el almacenamiento de las memorias de texto utilizará el siguiente esquema:

08 --> “MEMORIA 1” 00 “MEMORIA 2” 00

En la figura 4, podemos ver un ejemplo en el que se ha señalado el comienzo y final de cada memoria de texto a transmitir.

00	00	FF	45	42	35	41	42	54	.	.	E	B	S	A	B	T
08	43	51	20	43	51	20	43	51	C	Q	C	Q	C	Q		
10	20	44	45	20	45	42	35	58	D	E	E	B	S	X		
18	58	58	20	45	42	35	58	58	X	X	E	B	S	X	X	
20	58	20	45	42	35	58	58	58	X	E	B	S	X	X	X	
28	20	20	43	51	20	43	51	20		C	Q	C	Q			
30	43	51	20	44	45	20	45	42	C	Q	D	E	E	B		
38	35	58	58	58	20	45	42	35	S	X	X	X	E	B	S	
40	58	58	58	20	45	42	35	58	X	X	X	E	B	S	X	
48	58	58	20	41	52	00	43	51	X	X	A	R	.	C	Q	
50	20	54	45	53	54	20	44	45	T	E	S	T	D	E		
58	20	45	42	35	58	58	58	20	E	B	S	X	X	X		
60	54	45	53	54	00	FF	FF	FF	T	E	S	T	.	.	.	.
68	FF	FF	FF	FF	FF	FF	FF	FF	.	.	.	.	.	.	.	.
70	FF	FF	FF	FF	FF	FF	FF	FF	.	.	.	.	.	.	.	.
78	FF	FF	FF	FF	FF	FF	FF	FF	.	.	.	.	.	.	.	.

Figura 4

Cuando hayamos modificado las memorias, sería una buena idea hacer una copia del programa del Pic, con los cambios realizados.

Para ello, pulsaremos en "File" "Save File As..." y le daremos un nombre al fichero.

## PROGRAMACIÓN DEL PIC

Una vez editadas las memorias de transmisión, procederemos a programar el microcontrolador Pic.

Aunque algunos programadores de microcontroladores Pic permiten recuperar el valor de calibración del oscilador interno que incorpora cada chip, recomiendo que antes de programar nuestro Pic, hagamos una copia del contenido de la memoria flash del mismo, para salvar el mencionado valor de calibración.

Si nuestro programador es compatible con el programa IC-Prog, directamente podremos programar el Pic.

No voy a entrar en más detalles, ya que sobre este tema hay mucha información en Internet.

## UTILIZACIÓN KEYER

Para conectar el Keyer a la emisora, normalmente utilizaremos un cable con conector de audio mono de 3,5 ó 6,3 mm.

Con el potenciómetro P1, regularemos la velocidad de transmisión (la duración de los puntos y las rayas) y si accionamos el pulsador P1, se transmitirá la memoria de texto seleccionada por I1.

Mientras se está transmitiendo una memoria, podemos regular la velocidad de transmisión, y si durante la transmisión accionamos el Paddle, la transmisión se detendrá.

Si se quedara cortocircuitado el pulsador de transmisión de memorias P1, solamente se transmitirá el texto una vez, ya que hay una seguridad que impide que las memorias se emitan continuamente por este fallo.

También hay una seguridad por si se cortocircuita el Paddle, y en este caso, si se emiten más de 100 puntos ó 100 rayas seguidas, el Keyer entrará en modo bloqueo.

Para poder poner de nuevo en servicio el Keyer, tendremos que desconectar la alimentación mediante la desconexión del Jumper JP1.

## MODOS TRABAJO

Aunque el uso habitual de este circuito va a ser como Keyer, también se ha incluido la función de baliza.

Si queremos trabajar en modo baliza, tenemos que indicar el nº de periodos de pausas que habrá entre cada transmisión.

El valor está comprendido entre 1 y 63.

Nótese que este valor no coincide con minutos, ya que dependiendo del valor de la tensión de alimentación del Pic, el oscilador interno varía su frecuencia y esto afecta directamente a la duración de las pausas.

Si queremos cambiar el modo de trabajo del circuito, necesitamos utilizar la emisora como monitor (también podríamos utilizar un oscilador de audio), para escuchar las indicaciones que nos manda el Keyer, mientras nos encontramos en "Modo Programación".

Si finalmente utilizamos la propia emisora, lo primero que tenemos que hacer, es bloquear la transmisión de la misma para evitar salir al aire.

Para entrar en modo programación, el circuito tiene que estar sin alimentación (desconectar jumper JP1).

Activamos el PUNTO en el Paddle y sin soltar pulsamos P1 (transmisión de memoria).

Ahora alimentamos el circuito conectando el puente JP1.

Si hemos realizado este procedimiento correctamente, el Keyer procederá a emitir continuamente las letras MP.

Al soltar el pulsador P1 y el Paddle, nos indicará el número de pausas que tiene programado.

Si estábamos trabajando en modo Keyer, este valor será 00.

Ahora con el Paddle subiremos (PUNTO) o bajaremos (RAYA) el valor de pausas que queremos programar, de forma que:

00 --> Keyer.

01 .. 63 --> Baliza + pausas intervalo.

Al pulsar P1, nos notificará el valor actual y lo memorizará.

Para salir del modo programación, volveremos a desconectar la alimentación con JP1.

En modo Baliza, se transmitirá la memoria seleccionada con I1, con un intervalo que depende del valor seleccionado.

## CONCLUSIONES

Dado la sencillez del circuito y el poco consumo que tiene, pienso que este Keyer se podría incluir sin mucha dificultad en algunos equipos de QRP de CW.

Al diseñar el circuito, decidí poner un conmutador de selección de memorias (pensando en hacer llamada para QSO o en hacer concursos), pero al ver el Keyer de DL4YHF (que dispone de un pulsador para cada memoria), veo más práctico este último.

También veo muy interesante el que el propio Keyer pueda ponerse de manera sencilla en modo baliza, para automatizar las llamadas en concursos, QSOs, etc.

Comentar también, que recientemente (estando cacharreando con otro tema), se me ha ocurrido una idea que si finalmente funciona, permitiría disponer al Keyer de 3 ó 4 memorias.

Para implementar todas estas funciones, necesitaría hacer algunas modificaciones importantes en la estructura del programa, y es un trabajo que añadido a la enorme lista de tareas que tengo pendiente, y que espero poder realizar en el futuro.

Si alguien necesita alguna aclaración, que le genere el fichero ejecutable del Pic con su indicativo, etc., que no dude en ponerse en contacto conmigo, mandándome un correo.

Tengo intención de subir algún video a YouTube [3], para mostrar el Keyer en funcionamiento.

Advertir que se trata de un circuito experimental, del que se podrían producir efectos no evaluados, por lo que no me puedo hacer responsable de los daños que pudiera ocasionar.

[1] <https://goo.gl/BSWeOe>

[2] <http://www.ic-prog.com/>

[3] <https://goo.gl/H4efjU>