Vinicio Coletti site

IK0WRB Keyer

Last released on June 30, 2001

What is a keyer?

A keyer is an electronic circuit connected to a two-paddle Morse key (a.k.a. bencher key or horizontal key or automatic key or iambic key) and to a radio transmitter. Left paddle transmits dots, while right paddle transmits dashes. Pushing the right paddle while keeping pressed the left one, transmits a dot, then a dash and another dot. Keeping pressed the two paddles, continues the dot-dash sequence, until one of the paddles is released. The same thing happen if you press the right dash and then the left, resulting in a dash-dot sequence.
Transmitting this way is far more simple and quick than using a standard vertical Morse key, thus many radio amateurs use iambic keys. Modern transmitters have the keyer circuit built in. An external keyer, like this one, is useful for old transmitters and also for new ones, because of the other features it delivers: the message memory, the beacon mode, the possibility to practice Morse.

How to build the IK0WRB Keyer

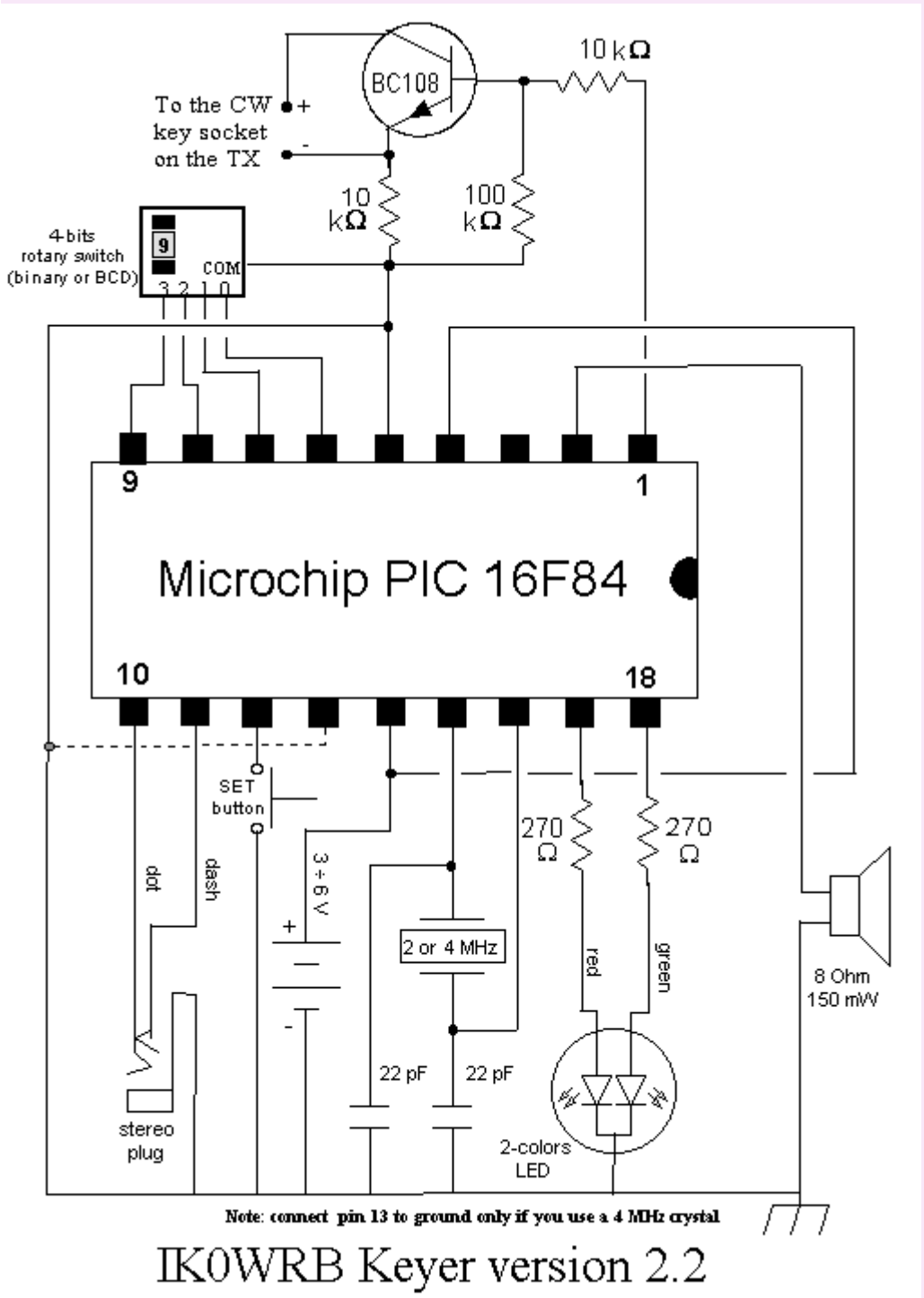
My keyer is based on a Microchip PIC 16F84 microcontroller and it uses a small number of components. You only need a crystal, a transistor and a few capacitors, resistors and connectors, plus a small speaker and a button. By the way you also need a small box and some batteries, just to give the 3.5 to 5 Volt DC the keyer needs.
The current is below 30 mA, with led glowing yellow and the speaker on. During pauses current drops to less than 20 mA and to only a few microamperes in sleep mode (for this reason there is no on/off switch).
A printed circuit board is not mandatory for this circuit, because of the small number of components.

How do I program the 16F84?

If you are lucky enough to live not too far from my home (which is unlikely if you are reading in English), I will be happy to give you one chip programmed with the **keyer software** I developed for this project.
Otherwise, keep reading and you will find the **hex module** you must put inside the 16F84 chip. If you don't have a chip programmer circuit, normally connected to the serial, parallel or USB port of your computer, you can buy it or look for a radio amateur owning it and kind enough to program the chip for you.

Version 2.2 circuit

What follows is the circuit of the IK0WRB Keyer version 2.2



If you need this drawing at a higher resolution, press here.

The components list

- a Microchip PIC 16F84, in the 18 pins package
- a 2 MHz or 4 MHz quarz crystal, parallel resonance
- 2 capacitors of 22 pF, 15 V
- 2 resistors of 270 Ohm, 0.25 W
- a resistor of 100 kOhm, 0.25 W
- 2 resistors of 10 kOhm, 0.25 W
- a 2-colors LED, common cathod, red and green
- a button, normally open
- 2 jack sockets, 3 poles (stereo)
- a loudspeaker, 4 - 8 Ohm 150 mW
- a BC108 transistor, or similar
- a 4-bits rotary switch (contraves) either binary (0-F) or BCD (0-9)

The program

You can download here the hex module you need to program the Microchip PIC 16F84 you need for the keyer. This is the software for **version 2.2** of the keyer. Be sure tu match the software version with the circuit version (below you will find some older versions).
The zip archive **wrbk22e.zip** contains the circuit drawing, this page and the hex module.

How to use the keyer

The use of the keyer is basically identical to the previous version 2.1, but I totally rewrote the manipulation (iambic) routine, after some enthusiasts of very high speed CW criticised a bit my keyer. I think now iambic is ok even for these strange chaps :-)
Due to difficulties to find 4-bits (thus 5 pins) 16-positions binary rotary switches, I mapped the positions A through E also on positions 5 to 9, that were unused before. So now, for example, to set audio on/off you can rotate the switch to 5 or to A, at your choice. This means that you can now also install a 4-bits (5 pins) 10-positions BCD switch, witch is more easy to find.
For the same market reason, I decided to support both 2 and 4 MHz quarz crystals, but for this you must tell the chip in some way (no, wispering doesn't work...) what crystal you are using. Since I got 2 unused pins, one of them is used now for this purpose. So, leaving pin 13 of the 16F84 free (logic 1) tells the chip that you are using a 2 MHz crystal (this is compatible with previous versions of the circuit), while connecting pin 13 to ground (logic 0) tells the chip that you are using a 4 MHz crystal.
All these changes were possibile because I changed the way to compute delays for each transmitting speed. Up to version 2.1 I used a pre-computed table stored in program memory, but now I execute a division routine and the space occupied by the table is now used for real istructions.
Thus I could add also a new feature to direct mode: pressing the button will trasmit a 5 seconds carrier, very useful for tuning.
I prolongued also the timeout for sleep mode, from the very short 33 seconds to a more normal 2'45". This doesn't affect the way you use the keyer, it's only a psycological trick: the keyer doesn't look dead while listening in a CW QSO.
Finally, I measured the actual trasmitted speed of the keyer and found it was about 9.8% slower than it should have been. Not a real problem, but I tuned delay constants anyway.
The following table shows how to operate version 2.2 of the keyer.

Commands of IK0WRB Keyer, version 2.2					
Rotary switch position	**Status**	**LED color**	**Left paddle**	**Right paddle**	**SET button**
0	Sleep	Off	No effect	No effect	No effect
1	Ready	Green while rx. Red while tx. Yellow if tx local.	Dot	Dash	Send single CQ message
2	Speed	Yellow	Decrease speed	Increase speed	Sound current speed (Snn)
3	Weighting	Yellow	Decrease weight	Increase weight	Sound current weight (Wn or Wn.5)
4	TX	Green if disabled. Yellow if enabled.	Enable transmitter	Disable transmitter	Sound status (TY or TN)
5 or A	Audio	Green if disabled. Yellow if enabled.	Enable audio monitor	Disable audio monitor	Sound status (AY or AN)
6 or B	Beacon	Green if message present. Off if no message.	Starts Beacon sequence. During carrier: immediate call.	During call: go to carrier. During carrier: more carrier.	Starts Beacon sequence. During tx or carrier: stops Beacon sequence.
			Autostart beacon sequence at power on		
7 or C	CQ	Green if message present. Off if no message.	Starts CQ sequence. During pause: immediate CQ.	During CQ: go to pause. During pause: more pause.	Starts CQ sequence. During CQ or pause: stops sequence
8 or D	Direct	Green	Carrier while pressed	Carrier while pressed	5 seconds carrier
9 or E	Enter	Green	Store a dot	Store a dash	Single pressure: insert inter-word space. 2 consecutive pressures:
			Store an inter-character space when		

Rotary switch position	Status	LED color	Left paddle	Right paddle	SET button
			pausing 0.25 sec.		delete last char.
F	Sleep	Off	No effect	No effect	No effect

Notes

- Speed can be changed with a 1 wpm step, from 1 to **98 wpm**.
- Weighting can be set from 2 to 5, with a step of 0.5
- CQ message stored in the memory can be 236 *symbols* long, where a symbol can be a dot, a dash or an inter-character space. Inter-word space is coded as two consecutive inter-character spaces. If the memory is full while storing, the keyer stops storing, retaining the message stored up to that point. The message is in the EEPROM, so it's never lost, unless you decide to store another one.
- CQ mode: you can send a single CQ by pressing the SET button in the Ready state or start a continuous CQ sequence by pressing left paddle or SET button in the CQ mode. RX period is fixed at 10 seconds, but you can momentarily reduce or prolongue it buy using the paddle keys. See table for details.
- Beacon mode works as CQ mode, but with a 20 seconds pause with carrier ON. You can modify the pause periode as in CQ mode.
- There is no on/off switch, because the keyer goes into sleep mode after about **2'45"** of inactivity. A pression of one of the paddles or the SET button, awakes the chip and the keyer continues its activity, as if it had been always on. In sleep mode the circuit drains less than 10 microamperes, comparable to the auto-discharge of batteries.
- Sleep mode locks all commands and is thus useful when carrying the keyer or to avoid unwanted transmissions while setting up the rig etc.
- Direct mode emulates a vertical key (and you can actually connect one of them to the keyer, either on dot or dash contact). Pressing the button with transmit a 5 seconds carrier, useful for tuning.
- Crystal: you can use 2 or 4 MHz crystals. For 2 MHz leave pin 13 free, for 4 MHz connect pin 13 to ground.
- Switch: you can use either a binary 0-F rotary switch or a BCD 0-9 switch. Both have 5 pins (or more), one common and 4 for output.

This project was done by IK0WRB, Vinicio Coletti
Rome, Italy, EU