

CW Memory Keyer

I recently purchased a new Paddle key, but since I have only used a straight key in the past I needed to get some practice using this strange new key. I started work making a keyer/practice oscillator and was happy (to start with) with how it worked, nothing fancy just speed control and that was it. Then I came across a keyer project on the web that offered many features, it was from Jan DK3LJ who made it open source, later modified by Jack AL4SV and then further modified by Don WD9DMP who made some changes that polished Jacks code more . The final product rivals many professional keyers costing 10 time the price.

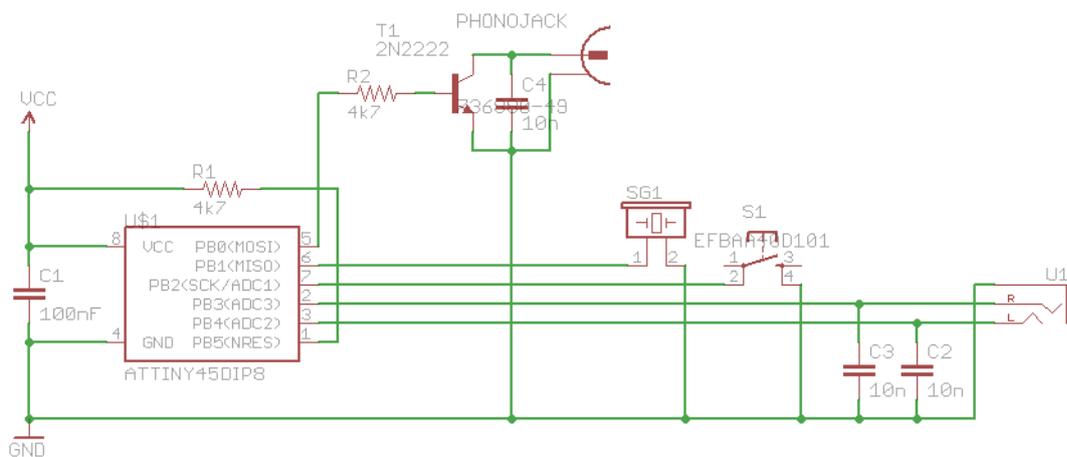
I have taken this open source code and just burnt it onto a Attiny85 chip, I have made a small strip board layout (with component mask) and put together a set of instruction on building it. I have included a short guide to using the keyer (By Don WD9DMP).

Here is a list of its features.

- Iambic A, B, Ultimatic modes
- DAH Priority Mode
- Paddle Swapping
- Sidetone Toggle
- Adjustable Sidetone Pitch
- ATU Tune Feature (20 Sec Key Down)
- TX Keying Toggle (Sidetone only for practice)
- TX Level Inverter mode
- WPM Speed feed back
- 4 x 100 char Memory's
- Automatic Beacon (for FOX hunts extra)
- Callsign Trainer (Sends random callsign, users send back)

For more details of each feature read the user guide at the end of this document.

Let me show you the circuit for the keyer.



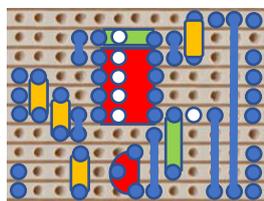
You can see its very simple.

Parts List

Programmed Attiny85 c/w socket
Strip Board & Mask
2 x 4K7 resistor ¼ watt
2 x 10nF capacitor (may be marked 103)
1 x 100nF capacitor (may be marked 104)
2N222 transistor
Piezo Sounder
Push Button

This can be built into any homebrew radio or boxed for use with a main station rig etc.

Like many of my projects I have made a strip board layout that can be used to make strip board more like a PCB.

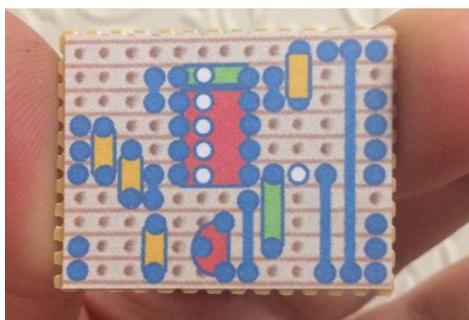


Print this out and use a glue stick to fix it to a small stripboard, the white dots are where the tracks need to be cut.

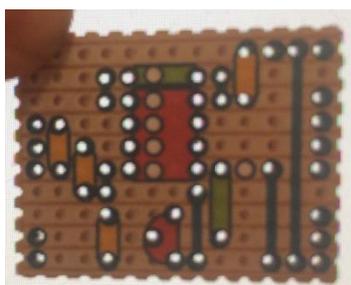
Building the Keyer

On a scale of 1 to 10 with 1 being very easy (no soldering) and 10 being complex and difficult this would rated as 3, suitable for beginners

Lets look at the board.



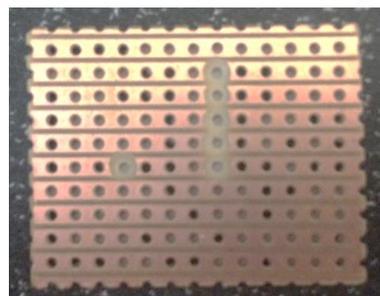
Use a glue stick to fix the mask to the board, do NOT use a wet glue as this can distort the size of the mask. Take time and make sure the holes line up.



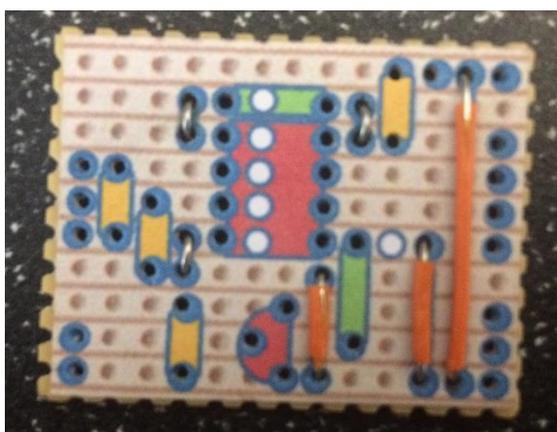
Next I pierced the holes in the mask, this is all the BLUE circles on the board. I check the holes are all done by holding the board against a white screen on my monitor.

Now make sure all the tracks indicated by a WHITE circle are cut, there are a total of 6 tracks to cut.

I use a 4 or 5 mm drill bit and just a couple of twists will produce a nice clean cut, if you need to put a pin through the positions on the mask where a track needs cutting, turn the board over and see where the pin is to make sure you cut the correct track.



Now we can start building the module, the order is not critical so feel free to do it your own way, below are details of how I made mine.

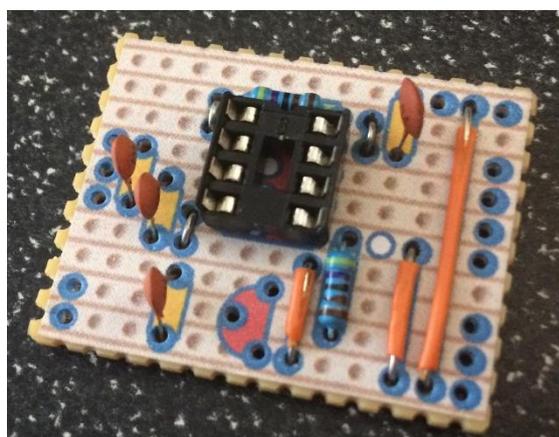
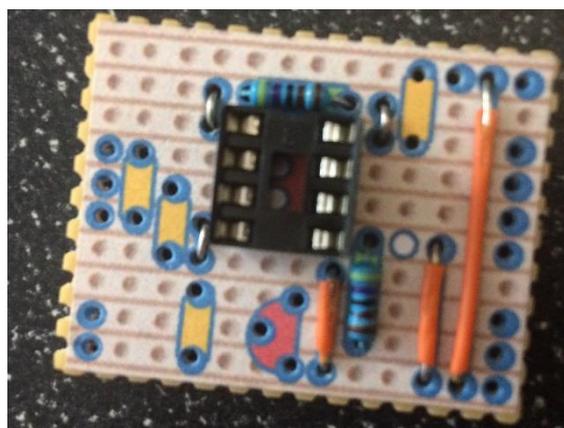


I fitted the links first, use insulated tinned copper wire if you can but for some links it would be near impossible to leave the insulation on so don't worry about it.

There should be 6 links to fit.

Next there are two resistors, both are the same value (4K7), they can be fitted either way around.

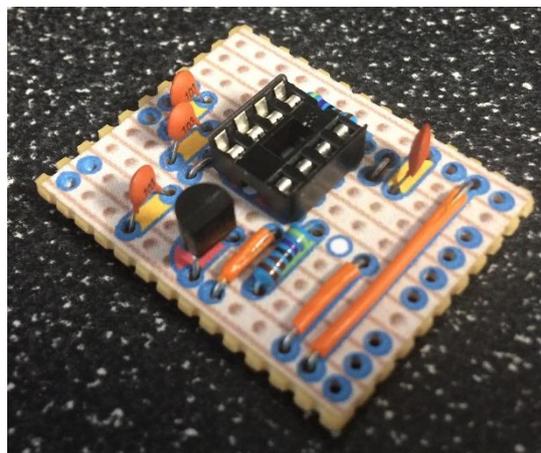
Also take the opportunity to fit the IC socket. Pin one is top left.



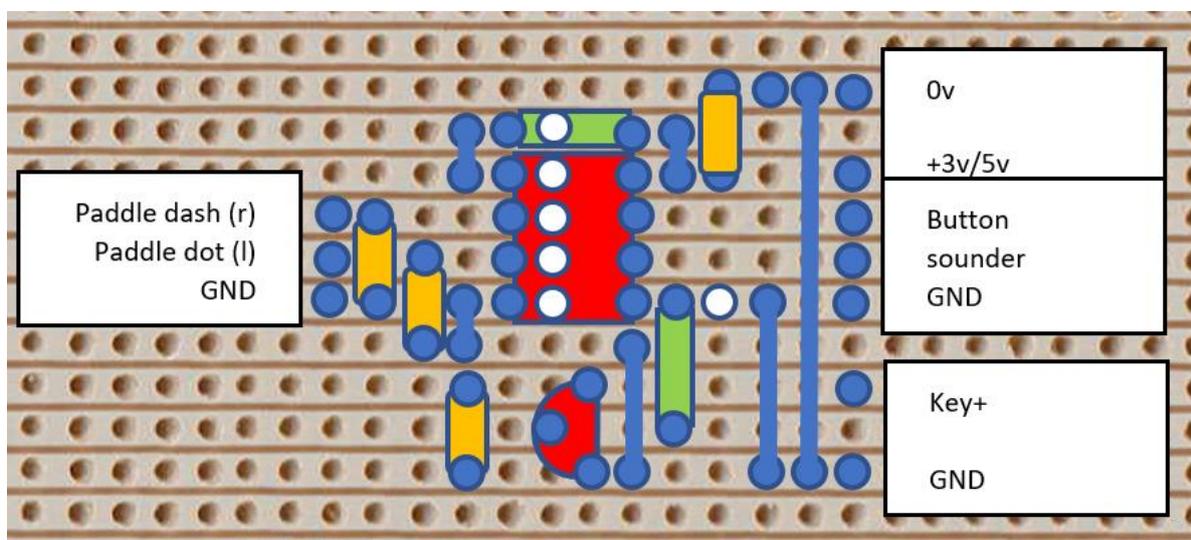
Next fit the 4 capacitors, 3 of them are 10nF (103) and 1 is 100nF (104).

Look at this picture, the capacitor in the top right section is the 100nF one, the other 3 on the bottom left section are all 10nF.

Finally fit the transistor, take care to fit this the right way, the layout on the mask shows which way to fit it.



Now you're done, use the connection details below to wire up your keyer.



My Keyer fitted to a Mint tin.



Look at my youtube video for a demo.

The Keyer draws about 1mA when active and after 60 secs of no use switches to sleep mode and the current falls into the microamp range, I have powered mine by two AAA batteries (total 3v) and haven't even bothered with a ON/OFF switch.

***** VERY IMPORTANT *** DON'T POWER THIS FROM 12V!!**

The ATTINY85 only needs 2.7 to 5.5 volts, more than this will destroy the chip, many people use a 3v button cell. If you need to power this from an external power source just use a 5 v regulator (or a 3v3) and everything will be fine.

So how well does it work?

Very well! When I first tested it I thought it didn't work right but that was me not understanding how it works and using it wrong. Let me briefly explain its use.

When powered up it sends '73', this is its start-up signal, the feedback signals are via the piezo sounder, NOT on the keyed output.

The keyer keyed my radio right away, a tap of the paddles and I found it worked backwards, the paddles seemed to turn the transmitter off when touched but then back on when left alone. I read the manual written by Don WD9DMP and found one feature was TX level invertor mode, the default is set for Positive keying , all my radios I use want the key shorted to ground for TX so that's the opposite to what is the default setting, changing this took about 2 seconds. Just push the command button and send the letter F, this toggles the TX level, as soon as I did this the keyer replied with 'R' for received and the problem was solved. I still thought I had problems since as soon as it cleared this problem I found as soon as I touched the paddles it started sending Morse to me via the sounder.

I was still in command mode and unknowingly had selected memory 1 TX.

After reading the instructions better I found all the problems are due to me. I didn't understand how the command mode worked. Push the command button and the keyer sends '?', it's now waiting for the user to make a selection, there are a number of different command's the user can send to the keyer to change many options, when you make a change the keyer sends 'R' to confirm its going to act upon the users selection, BUT it is still in command mode and is waiting for the next instruction, so sending another Morse character at this time will select another option, that's fine if you wish to do that but if making just one change you MUST either press the button a second time or wait 5 secs for the chip to exit Command mode itself. When it leaves Command mode it will send 'SK', now it's ready to be used. The most common change will be speed, just hold down the command button and either hold the Dash paddle to slow down a string of dash and dots being sent via the sounder (not keyed to the radio) or the other paddle to speed it up.

The fully instructions are below and I suggest sitting down and reading them fully before trying to use the keyer. If you don't like something it's probably just an option setting.

This is possibly the lowest cost memory key around, a PCB is available from to special order from the USA but it works out to about \$25 with the postage for just the blank board.

One feature I really like is the CW training mode, at the keyer speed it will generate a callsign (not a real callsign but follows the format) and sends it to the user, the user ends back the callsign, if right you get a 'R' and a new callsign is sent. If you get it wrong you get the error signal (.....) and it resends the same callsign again until you get it right. It a nice extra feature that makes practice more fun. BTW if you turn Farnsworth mode on it will use that in this mode which is a big help if learning the code.

Keyer Instructions

Version: 0.87 Usage (WD9DMP Fixes and Enhancements)

After reset in default mode, the keyer plays "73" to verify that the Code and EEPROM data are programmed correctly. The circuit operates as a regular IAMBIC keyer in IAMBIC B at 15 WPM (words per minute = 60 CPM), with 800 Hz side tone. By default, the transmitter keying signal is positive. The ATTINY85 will automatically power off after 60 seconds into ultra low-power Mode to conserve power. The unit draws only a few microamps in this state, making battery power from a single 3V lithium coin cell very practical.

Speed Change

Speed can be changed by pressing and holding the command key while operating the DIT and DAH paddles, in any mode.

DIT reduces speed while DAH increases speed. The keyer plays an alternating sequence of dit and dah while changing speed without keying the transmitter.

Command mode

Pressing the command button without changing speed will switch the keyer into command mode. This will be confirmed with the '?' character. Another press of the same button takes the keyer back into regular keyer mode and will be confirmed by the 'SK' prosign.

During Command Mode the transceiver is never keyed and sidetone is always activated. Further functions can be accessed by keying one-letter commands as listed below. In general, changes made in Command Mode will be written to EEPROM after a short delay following input and confirmed by an "R" acknowledgement, or by another short press of the Command button (for some commands). The unit will remain in Command Mode for subsequent commands, timing out to Normal Mode after a few more seconds with an "SK" indication. The keyer can also be returned to Normal Mode with an additional button press.

V -Version

The keyer responds with the current keyer software version number

P -Pitch

Allows modifying the sidetone pitch to a higher or lower level. A sequence of dits will be played and the pitch can be modified with the dit and dah paddles. If no paddle is touched for 5 seconds, the acknowledgment signal 'R' is sounded and the mode terminates, writing the new pitch to EEPROM and leaving the user in command mode.

R -Reset

All settings are returned to their default values except for the stored messages in the message EEPROM area. Restored settings include speed and pitch, paddle swap, TX level inversion, sidetone Farnsworth setting, and TX keyer settings.

U -Tune

The transceiver is keyed for a duration of 20 seconds for tuning purposes. Tuning mode is aborted once either DIT or DAH paddles are touched or the control key is pressed.

A -Iambic A

Keyer sets IAMBIC A as permanent keying mode. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request.

B -Iambic B

Keyer sets IAMBIC B as permanent keying mode. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request.

L -Ultimatic

Sets the keyer into ULTIMATIC mode as permanent keying mode. In Ultimatic mode always the last paddle to be touched is repeated indefinitely when paddles are squeezed. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request.

D -DAH priority mode

Sets the keyer into DAH priority mode as permanent keying mode. In squeezed state a sequence of DAHs is sent. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request. Some of the first generation keyers exhibited this behaviour so the chip can simulate that.

X -Paddle swapping

DIT and DAH paddles are swapped. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request.

S -Sidetone toggle

The built-in sidetone generator setting is toggled (ON -> OFF or OFF -> ON). NOTE: This setting is only of relevance for regular keying mode. Sidetone is always on in Command Mode. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request.

K -TX keying toggle

Toggles the setting of the TX keyer output. In default state the keyer switches the output line when it is in keyer mode. Toggling this setting enables or disables that function. NOTE: Keying is always off in Command mode. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request.

Z -Set Farnsworth pause

Allows setting of an extended inter-character pause, which makes fast keying easier to understand. A series of "Dit-Dah" sequences will be played. Pressing the "Dah" paddle will decrease the Farnsworth inter-character spacing. Pressing the "Dit" paddle will increase the Farnsworth Intercharacter spacing. If no paddle is touched for 5 seconds, the acknowledgment signal 'R' is sounded and the mode terminates, writing the new setting to EEPROM and leaving the user in command mode. Note that this only affects memory/program playback and transmission. If you desire Farnsworth mode in manual transmission, please manually pause between characters.

F (Flip) -TX level inverter toggle

This function toggles whether the "active" level on the keyer output is VCC or GND. The default is VCC. This setting is dependent upon the attached keying circuit. The setting is written to EEPROM and an 'R' is sounded to acknowledge the request.

W -Query current WPM speed

Keyer responds with current keying speed in WPM.

1, 2, 3, 4 -Record internal messages 1, 2, 3 or 4

The keyer immediately responds with "1" or "2" or "3" or "4" after which a message up to 100 characters can be keyed at current WPM speed. If no paddle is touched for 5 seconds, the acknowledgment signal 'R' is sounded and the mode terminates, writing the new setting to EEPROM and leaving the user in command mode. Choosing "1" or "2" or "3" or "4" but not keying a new message deletes the chosen message buffer content. A command key press during the recording function returns the keyer to command mode, leaving the memory unchanged.

E, I, T and M -Play back internal messages 1 or 2 or 3 or 4

The stored messages 1, 2, 3, or 4 are played back with keying enabled (if configured). A press of the command key aborts the message and immediately returns the keyer to Normal Mode for a QSO, with an "SK" confirmation. After the message plays to completion, the keyer remains in command mode with no "R" indication so another message playback sequence may be started. If no additional message is started, the keyer returns to Normal Modewith an "SK" confirmation.

N -Automatic beacon

The keyer responds with 'N' after which a number between 0 and 9999 can be keyed. After a 5 second timeout the keyer responds by repeating the number and 'R'. Once the keyer returns to keyer mode, the content of message buffer 4 is repeated in intervals of n seconds. The setting is preserved in EEPROM so the chip can be used as a fox hunt keyer. Returning to command mode and entering an interval of 0 (or none at all) stops beacon mode.

0 (zero) -Lock configuration

The 0 (zero) command locks or unlocks the main configuration items but not Version, Pitch, Tune, Callsign Training, WPM Query and Memory Playback functions.

C -Callsign trainer

The keyer plays a generated callsign (sidetone only) and the user must repeat it. If it was repeated correctly, 'R' is played and the next callsign is given. If a mistake was sensed, the error prosign (8 dits) is sounded and the current callsign is repeated again for the user to try once more. If nothing is keyed for 10 seconds, the keyer returns to command mode.

Speed Change = Command +Paddle
V -Version
P -Pitch
R -Reset
U -Tune
A -Iambic A
B -Iambic B
L -Ultimatic
D -DAH priority mode
X -Paddle swapping
S -Sidetone toggle
K -TX keying toggle
Z -Set Farnsworth pause
F (Flip) -TX level inverter toggle
W -Query current WPM speed
1, 2, 3, 4 -Record internal messages
E, I, T and M -Play back internal
messages
N -Automatic beacon
0 (zero) -Lock configuration

Handy Command Card to print off