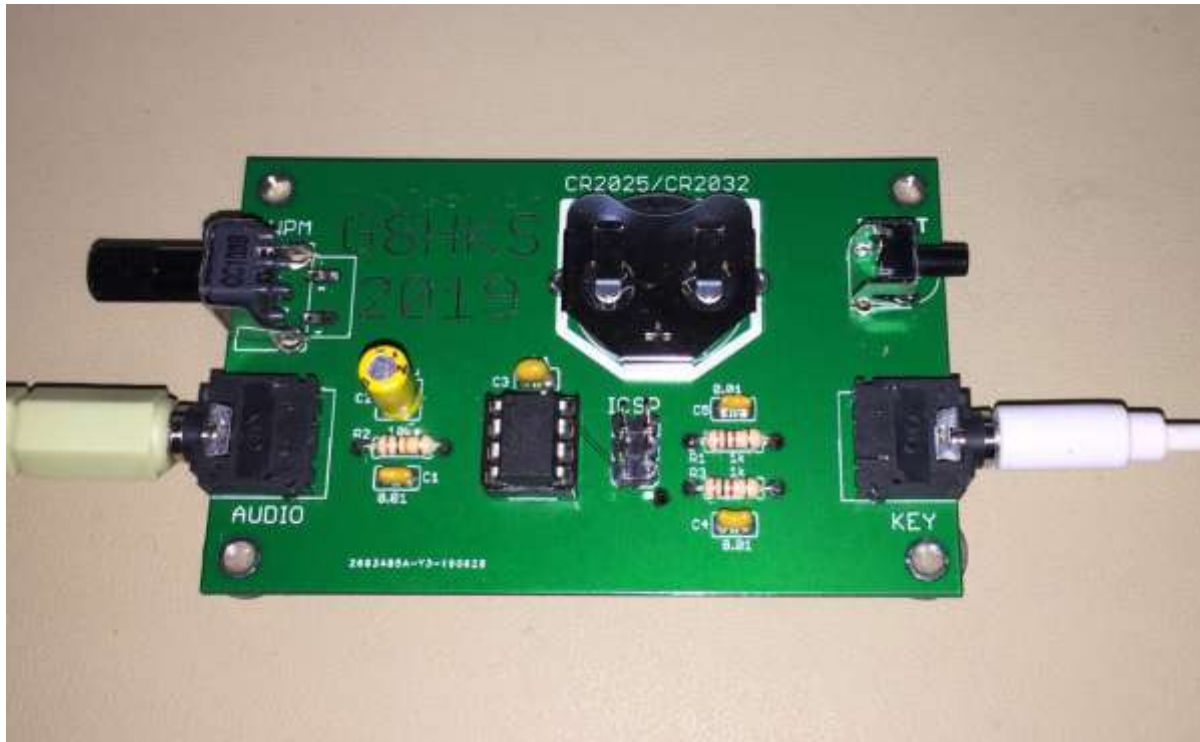


## TinyTutor Morse Trainer

The TinyTutor is based on an Atmel ATTiny 85 microcontroller which has been programmed to generate a sine wave tone in response to an external Morse key or under programme control to play random words.



The aim is to build a subliminal recognition of the 100 most common words in the English language plus many Ham words and abbreviations by repetition, whilst also giving the user a method of practicing their keying skills.

The board will play back what you **actually** sent.....

### Parts:

3 x 10 nF	1 x 10 k $\Omega$	1 x switch
1 x 100 nF	2 x 3.5mm socket	1 x battery holder
1 x 10 $\mu$ F	1 x 10 k $\Omega$ pot	1 x CR2032 battery
2 x 1 k $\Omega$	1 x IC socket	1 x ATTiny85 microcontroller
4 x self-adhesive feet		

## **Key Features**

You can use a straight key, a paddle, or both.

The default is straight key, but as soon as a Dah is sent on a paddle, the mode is changed.

To switch back, hold the straight key until the Dits change to a continuous tone.

What you send with the key will be replayed when you pause or after 50 characters are entered.

The keying for a paddle speed is set by the pot from 10 – 25 WPM

The keying speed for a straight key is automatically adjusted to the speed that you key

If you do nothing, random words will be played every few seconds. The board can be switched to Farnsworth timing (and back) by sending <UI>

If you still do nothing, the board will go to sleep after approx 10 mins

Or you can send it to sleep by keying <SK>

Or a continuous string of Dah's.

A 'Dit' on the key will wake it up, as will the reset button.

In sleep mode the board consumes << 1µA. Whilst active it takes roughly 3mA so the coin cell battery should last for about 60 hours of full activity..... or 20 years in standby!

The board recognises a Morse backspace to correct errors

All user settings, such as volume and tone, are captured in EEPROM.

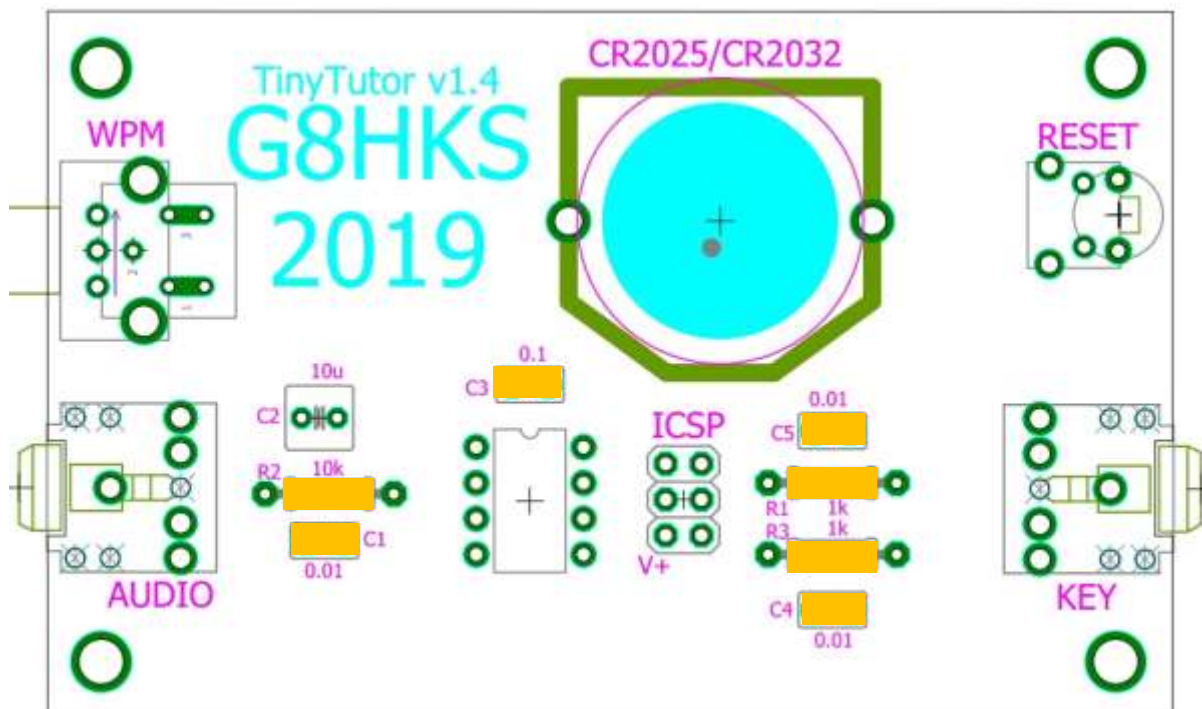
The microcontroller holds a list of the 100 most used words in the English language plus many commonly used ham words.

It defaults to Iambic B paddle, where an early Dit will be detected during a Dah.

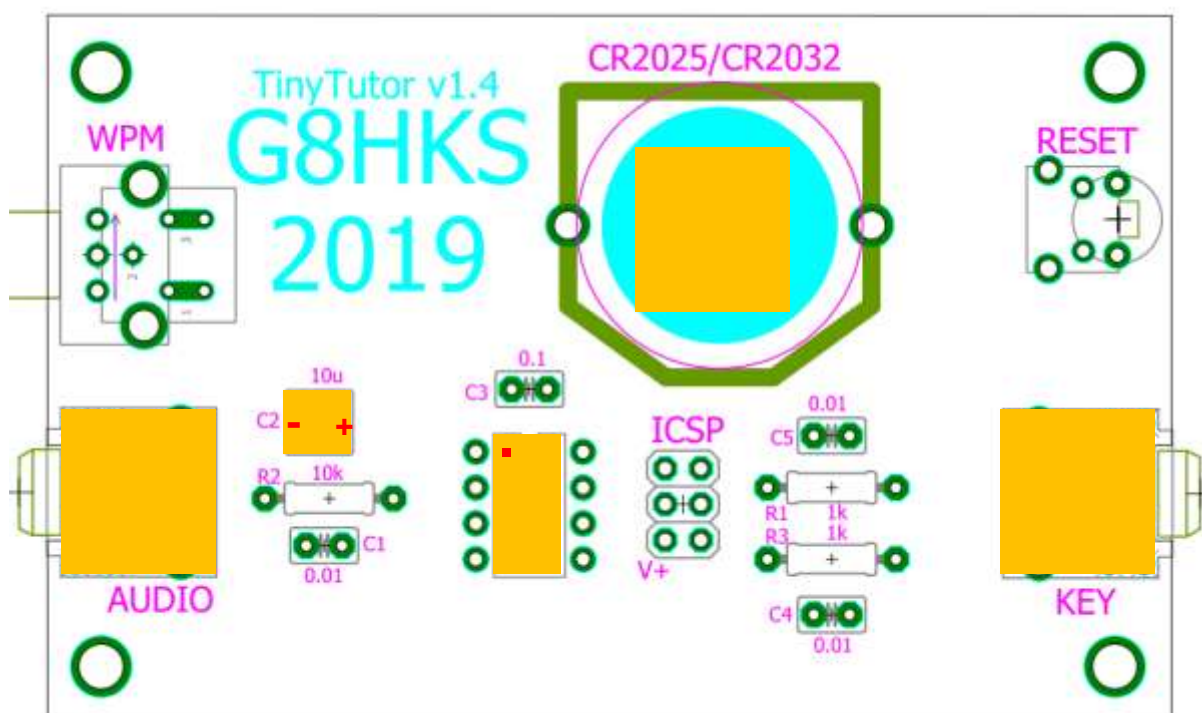
The board is smart enough to recognise correct keyed entry of the random word last played and gives appropriate feedback: 'R R'.

The random word will be repeated if you enter a '?'.

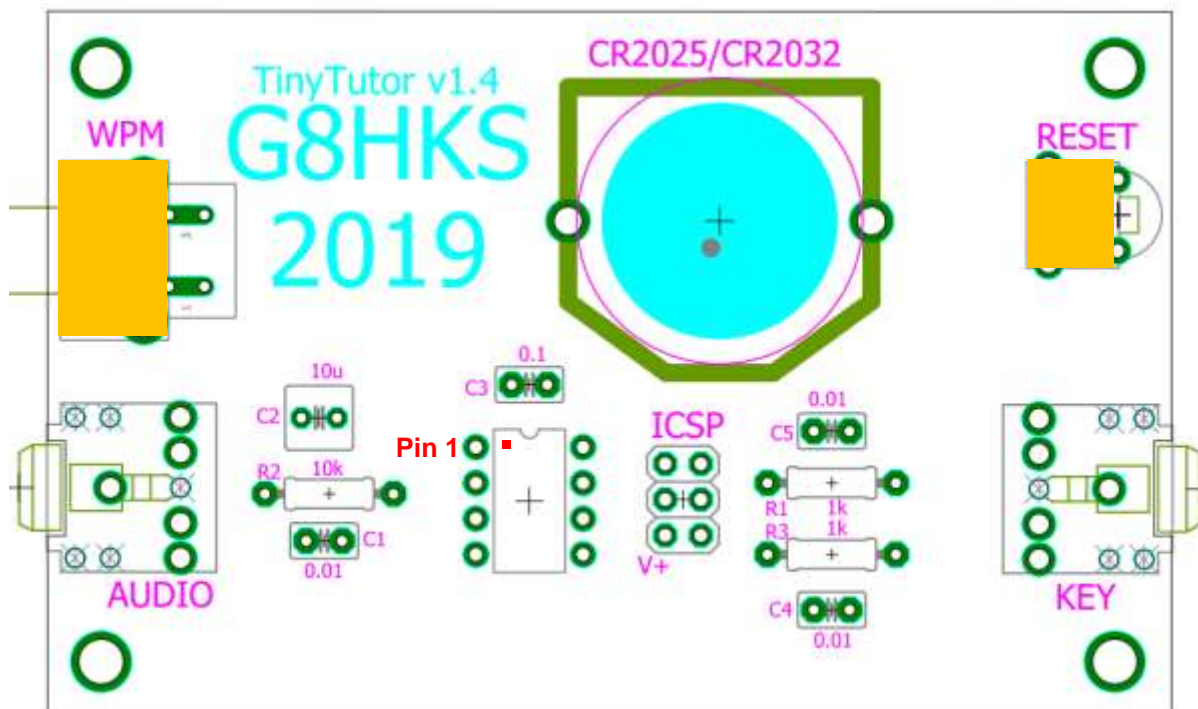
Install components in the same order as shown on the following pages to make it easy: **lowest height parts first.**



1) Install R1 + R3 (1 kΩ), R2 (10 kΩ), C1, C4, C5 (10 nF), C3 (100 nF)



2) Install: IC socket, J1, J2 (3.5mm jacks), Battery housing and C2 (10 μF)  
**Check orientation of the IC socket** – dimple at top matching the silk screen



- 3) Install WPM Pot and Reset switch
- 4) Double check that you have soldered and trimmed all wires.
- 5) Carefully insert the IC into the socket: be careful to **make sure** that the pin 1 dimple on the chip is at the same end as the dimple on the socket
- 6) Finally, slide the battery into the housing with the **+** uppermost

### **Set Volume**

Plug in a speaker or headphones and a key. Hold the key while clicking the reset button and adjust volume with the pot.

Release the key and you will hear the welcome message at 15 WPM.

### **Set tone**

Whilst the welcome message is playing, hold the key again and at the end of the message you will get a continuous tone. Use the pot to set the pitch to suit your ears then release the key.

Random words will start after a few seconds: use the pot to set the WPM to a comfortable speed.

### **Farnsworth**

At low WPM Farnsworth fixes the minimum character speed at 18 WPM, but increases the inter-character and space timing to give more time to recognise the characters. Above 18 WPM, all timing will be as set by the speed pot.

Key <UI> to toggle Farnsworth mode on & off. Get "OK" in response.

### **Random words or 5 character groups**

The board can be switched between playing random words and playing 5 character groups by keying <AS>. Response is "random words" or "random characters".

It is now up to you....