Hands on with Morse code

Mike Bedford reveals something of the first-ever code for data transmission, and shows you how to try it out for yourself

Mike Bedford

Despite loving all things digital, Mike admits to being a bit of a Luddite, vinyl records and all.

On 24 May 1844, a message was sent from Washington DC to Baltimore. Normally it would have taken a day or more for a dispatch rider to cover the 66 km, but this message was received almost immediately. The words of the message – What hath God wrought – conveyed something of the enormity of the achievement and the awe of those observing it. The secret of this momentous event is revealed by the person sending the message – Samuel Morse. Indeed, the code that bears his name was the world's first code for data transmission. Here we delve into this historic code, and discover that it was a lot more sophisticated than most people think. We'll also see that, despite it being more than a little long in the tooth, you can still hear it in use today. And if you want to get some practical experience, we'll show you how to learn Morse code, and even try your hand at sending it.

Left
Morse code had already been in use for 72 years when this photo of a British Army signal exchange in the Battle of the Somme was taken, and it would be used on board ships for another 83 years

SPOTLIGHT ON THE CODE

Each letter, figure, or symbol in Morse code is represented by short and long elements – which could be sent by telegraph, radio, light, or even audibly – that are shown in written form as dots and dashes, as illustrated in **Figure 1**.

If you fancy learning Morse code, here's what we suggest. First, you need to memorise the code – it's not as difficult as you might expect. What's more, if you use Morse code to any significant degree, you'll probably never forget it. Previous generations might have learned it from a printed table, but there's a better way. Because you'll eventually want to receive Morse code by ear, it makes sense to learn it audibly, and there is no shortage of software and online utilities to help. You might be eager to try sending Morse code, but it's better to learn to receive it first. By listening to it, you'll learn the characters as rhythms, so it'll become second nature to reproduce the correct timings when you start to send it. So, you'll intuitively send dashes that are three times longer than dots, you'll send the spaces between dots and dashes in a character that are the same length as dots, and so on. It's also a good idea to learn to receive Morse code at a reasonably high speed – 15 words per minute or perhaps more – even if there are long gaps between the letters. Again, this helps you to learn the rhythms rather than trying to count dots

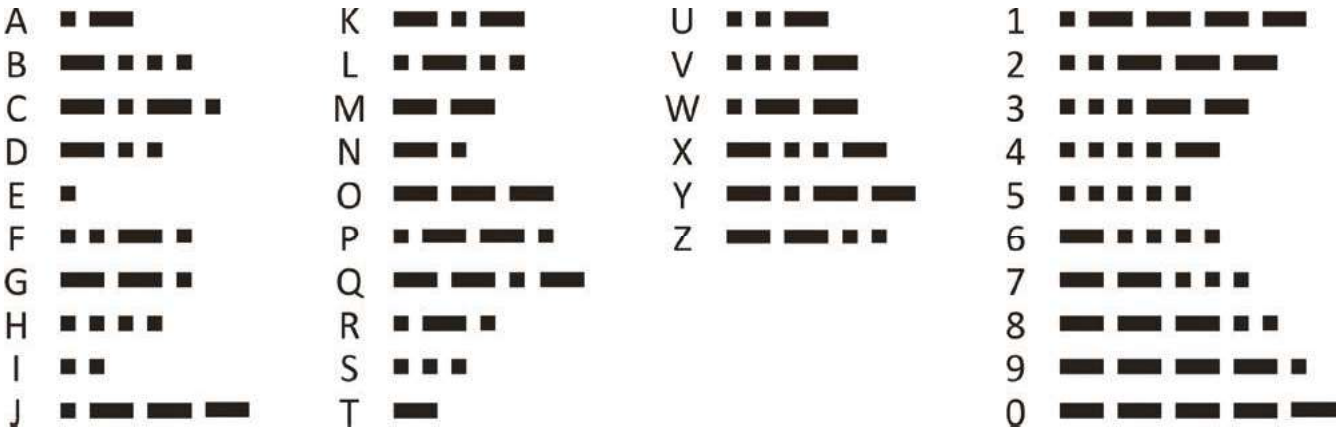


Figure 1 Characters in Morse code are represented by dots and dashes, but it's much more sophisticated than the apparently random codes might suggest

and dashes. A well-respected site for learning Morse code can be found at **lcwo.net**. It's free, although you do need to register.

When you progress to learning to send, you need a Morse key, and you can buy them new (from about £20) or second-hand ex-military (from less than £10 on eBay). Electrically, Morse keys are just like momentary action push-buttons, but they're finely balanced, and adjustable, so operators could use them all day long, even at high speed, without getting wrist injuries. Make sure you get one that works by pressing it down, not side to side (we'll look at those later). At a pinch, to start, you could even make your own key using a springy steel strip instead of a delicately engineered mechanism. You'll also need something to make a bleep, and the simplest solution is an active piezo sounder, which is the type that generates a bleep whenever a DC voltage is applied to it. All you need to do is wire the Morse key, sounder, and a battery in a series loop. If it's loud, and your family don't share your enthusiasm with incessant bleeping, you could put a piece of tape over the hole. A better solution would be to knock up a simple audio oscillator, ideally with a volume control, that can drive headphones. Alternatively, you could write some simple code on a Raspberry Pi or similar.

> "You might think of Morse code as being crude in the **extreme, but you'd be wrong; it's remarkably sophisticated**"

A SOPHISTICATED CODE

You might think of Morse code as being crude in the extreme, but you'd be wrong; it's remarkably sophisticated. If you look at the dot-dash combinations for the various letters, they might seem to be largely random, but they were chosen deliberately. The two most common letters in the English language are E and T, which are represented by a single dot and a single dash, respectively. On the other hand, J, Q, X, and Z are the least commonly used, and their codes are . - - - , - - . - , - . . -, and - - . ., which are some of the longest codes. Using a code in which letters are different lengths depending on how commonly they're used is very efficient compared to digital codes like ASCII, where all letters are the same length. Much the same method was reinvented in the 1950s as a method of data compression, and software that employs this Huffman code is still used today. Its principle differs from that of Morse code, only that instead of each character (or byte of data) always having the same code, the data is analysed and codes are assigned on the fly, according to how commonly they're found.

AUTOMATIC KEYS

Morse keys, like the ones we've already seen, limit the speed at which most people can easily send Morse code. Automatic keys allow Morse code to be sent more quickly, and they also force the user to send characters with accurate timing, for example, dashes always being exactly three times as long as dots. Automatic keys are operated by moving a lever – usually called a paddle – from side to side, instead of up and down.

The first automatic keys – or, strictly speaking, semi-automatic keys – were called bug keys and were purely mechanical. If you moved the paddle to the left it made an electrical connection, allowing you to send dashes, even though they had to be formed →

Above
You can practice sending Morse code using nothing more sophisticated than an old-fashioned Morse key, a battery, and an active piezoelectric buzzer

manually, just as they are with a manual key. However, when you moved the paddle to the right, a horizontal pendulum mechanism caused a string of dots to be sent. So, to send a letter B (- . . .), you'd move the paddle to the left for the correct length of time and then move it to the right until three dots had been sent. OK, so I chose a good example, but averaged over the alphabet, using a bug key certainly reduces the number of key movements required.

Going beyond the bug key, using mechanics is tricky, but with electronics, it's easy to produce a fully automatic key that sends strings of dashes when you move the paddle to the left and strings of dots when you move it the other way. It reduces key movements further, and it means that dashes will be perfectly formed as well as dots. If you'd like to try this way of sending Morse code – but, as it's generally suggested, not until you've mastered sending it manually – you'll either have to buy or make a side-to-side key that makes a different circuit depending on which way you move it, and build the simple circuit that we show in **Figure 2**, which is based on a design by Indian radio amateur Hari Sankar (call sign VU3NSH). It might seem strange that we've mixed logical families, using both a 4000 series and a 74HC series chip, but that's just because we had a 74HC32 at hand. It'll work just as well with a 4071 instead of a 74HC32, although note that the pinout is different. You can use pretty much any type of diode and any bipolar NPN transistor.

And it gets better. Some side-to-side keys have two paddles instead of one, so any electronics connected to the key can detect three active conditions: either of the paddles being closed by themselves, and both

paddles being closed at the same time because they're squeezed together. This type of key is usually used with more complicated electronics. It generates strings of dots or dashes, as the appropriate paddle is pressed by itself, or strings of alternating dots and dashes when the two paddles are squeezed together. So, if you squeeze the paddles together, with the dash paddle closing slightly before the dot paddle, the letter C (- . - .) could be sent. Circuits using discrete logic chips abound, but I'd be inclined to suggest you use something like a Raspberry Pi Zero. As well as implementing strings of dots and dashes, and alternating strings, you could also provide automatic spacing. Of course, short spaces the same length as →

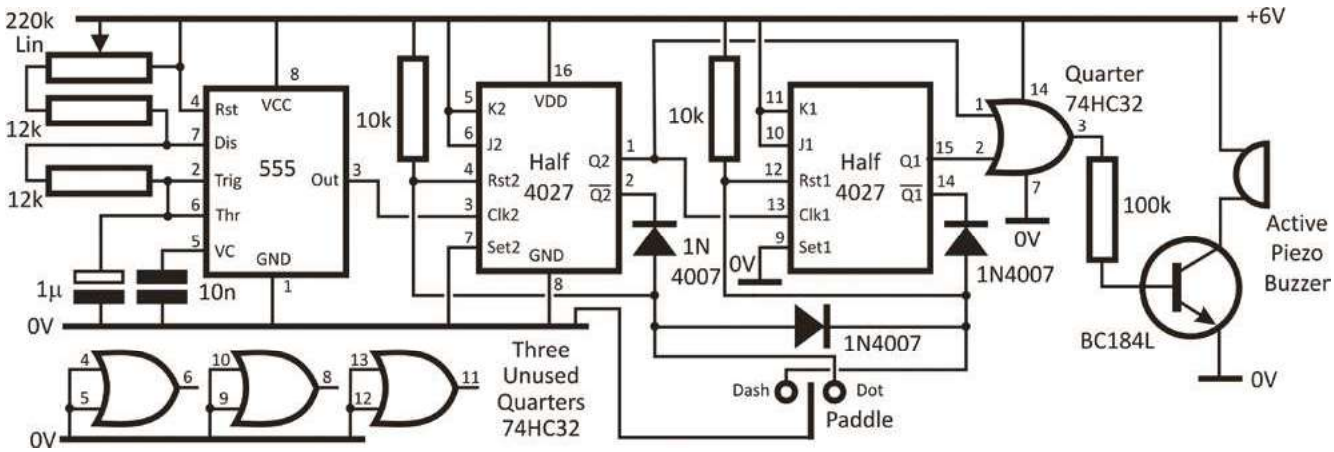
LISTEN **TO MORSE CODE**

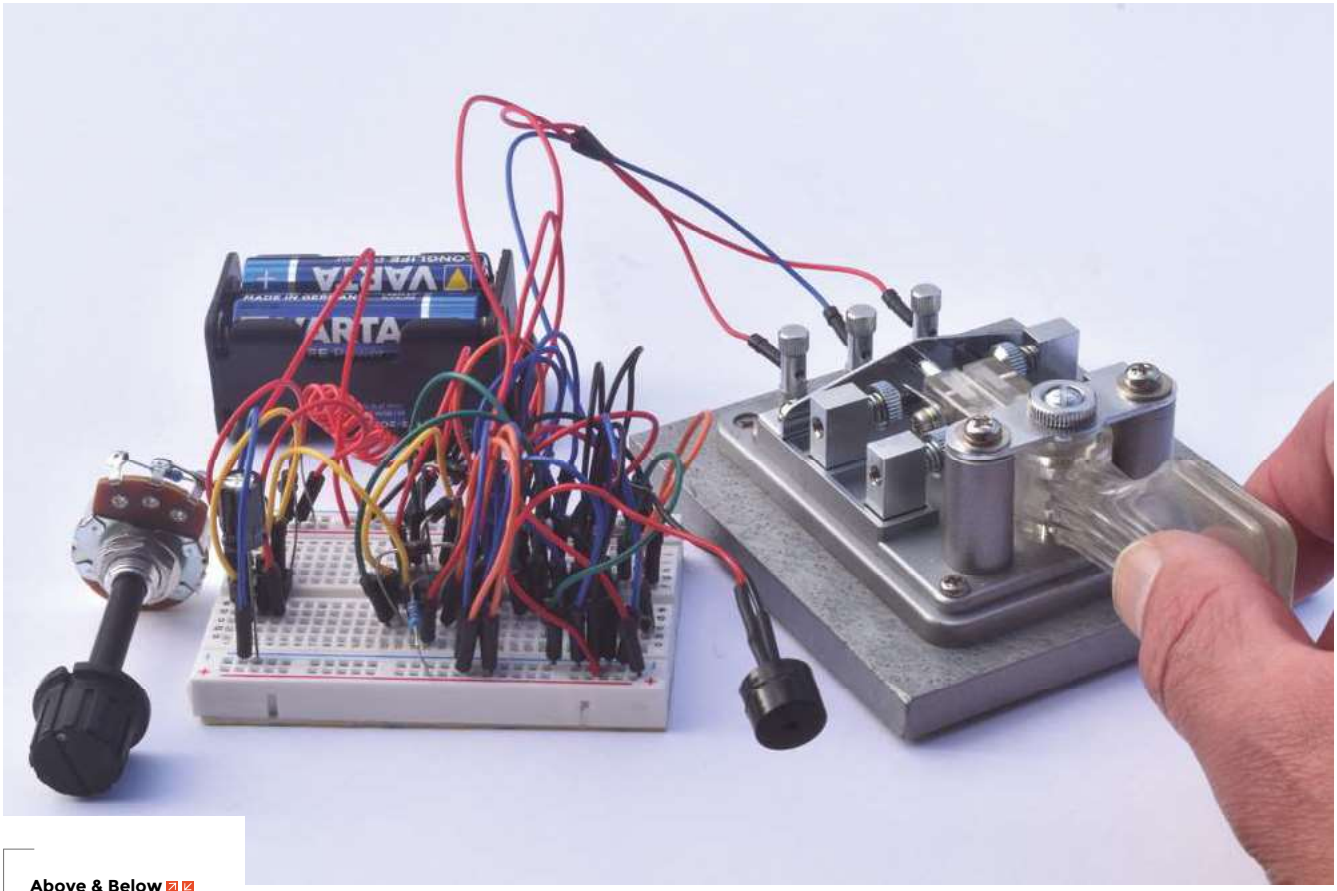
If you want to hear Morse code on the air waves, and if you have a shortwave receiver, you'll find it at the lower frequency end of each of the amateur radio bands, although which band you should choose will depend on various things, including the time of day. Not having a suitable radio is no handicap thanks to the network of WebSDRs – online software defined radios. Go to **websdr.org** and take your pick. We suggest choosing one that covers the 40 m amateur band (7.0 to 7.2 or 7.3MHz, depending on the country) and listen in from just above 7.0MHz to about 7.06MHz – at higher frequencies you'll find voice signals.

Left
Strings of dots can be sent with a semi-automatic 'bug' key, like this 100th-anniversary edition of the Vibroplex, the first Morse key ever to provide a degree of automation

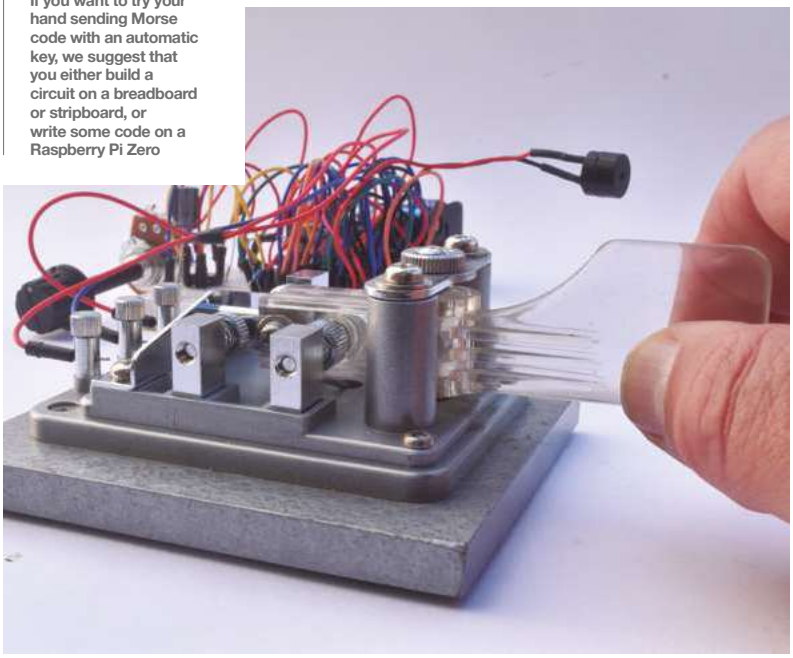
Image: CQDX.ru

Figure 2
This circuit can be used with a side-to-side key to implement an automatic key that takes the donkey-work out of sending strings of dots and dashes



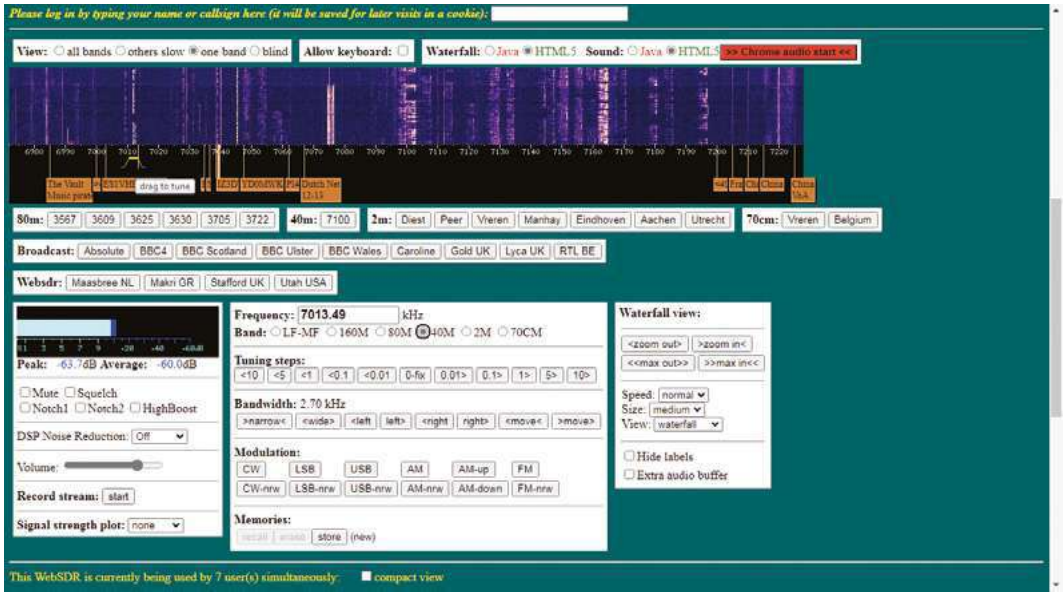


Above & Below
If you want to try your hand sending Morse code with an automatic key, we suggest that you either build a circuit on a breadboard or stripboard, or write some code on a Raspberry Pi Zero



dots will be an integral part of strings of dots and/or dashes, but if the paddles aren't pressed for, say, two dot lengths, you could prevent any more strings being sent until three dot lengths had elapsed, this being the length of a space between individual characters. So, if you try to send each character slightly early, the spacing will always be perfect. Similarly, you could enforce the space between words of exactly seven dot lengths.

Of course, if you're going to implement a squeeze key on an SBC, if you have a keyboard attached you could also create a fully automatic keyboard sender. These have been used but, surprisingly, you might think, they're not as popular with radio amateurs as side-to-side keys. Partially this is because many radio amateurs like to hone their skills in sending Morse code, skills that are barely needed with a keyboard sender. However, there's a more practical reason. Because software for reading Morse code often isn't as good as reading it by ear on noisy shortwave bands, it's usually read by ear. But, since most people can send Morse code more quickly than they can



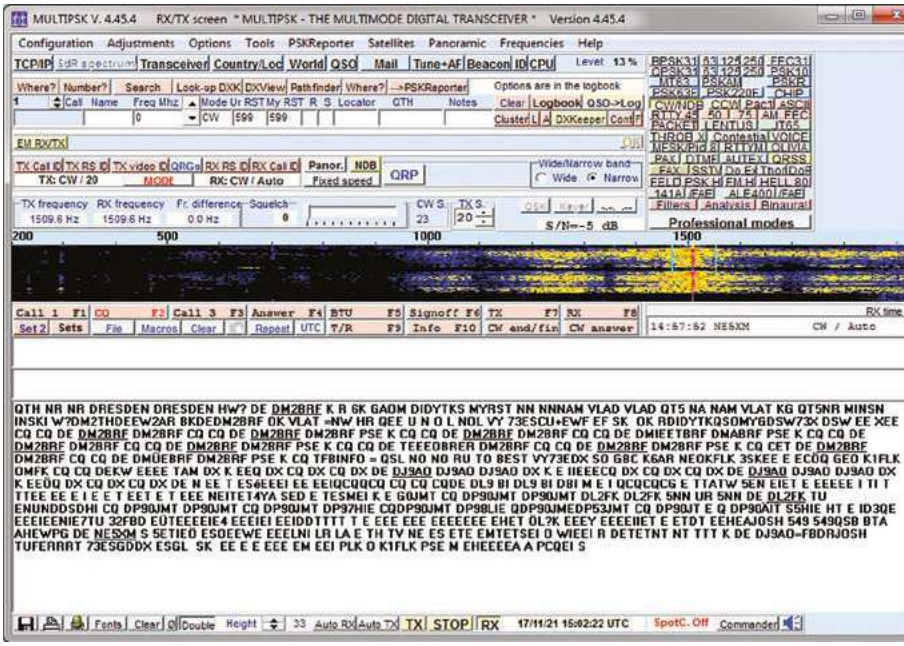
Left
Software like MultiPSK can decode signals on the amateur bands, but it will still take some time to understand the abbreviated nature of Morse code contacts

Below
Most of us don't own shortwave receivers, but we can receive Morse code on the amateur bands using the global network of WebSDRs

read it, even using a manual or side-to-side key, the benefit of sending very fast Morse code using a keyboard would be a retrograde step.

MORSE CODE TODAY

Morse code was phased out for maritime communication at the end of the 20th century, but that doesn't mean it's dead and gone. Today it's used by radio amateurs, and it spans longer distances than speech, all other things being equal. Using a simple, homemade Morse code transmitter, radio amateurs are able to send signals anywhere in the world. When most of us now carry a radio communication device that allows us to talk easily to anywhere in the world, this might not seem like a big deal, but we're not comparing like with like. Our mobile phones communicate with a base station usually no more than a few kilometres away, while simple transmitters and Morse code have covered almost 20,000 km. Not bad going for a method of communication that first saw the light of day almost 178 years ago.



DECODE MORSE AUTOMATICALLY

If you're new to Morse code, you're really going to struggle to interpret any code you might hear on the amateur bands. Software can lend a hand, though. It won't do a perfect job, because hand-sent Morse code isn't perfect, and the shortwave bands contain lots of interference, but if you pick a loud signal, you should have some success. Our recommended decoding package is MultiPSK, which is free for Windows from **f6cte.free.fr**, or fldigi (**w1hkj.com**) for Linux. You'll need to route the audio output from your browser to the decoding software while you're listening to the WebSDR. Possible utilities are the JACK Audio Connection Kit for Linux, or VB-Audio Cable for Windows.