

**NAME**

**morse** - reformat input as morse code

**SYNOPSIS**

**morse** [-**elrps**] [-**d** *device*] [-**w** *speed*] [-**c** *speed*] [-**f** *frequency*] [*string ...*]

**DESCRIPTION**

The **morse** command reads the given input and reformats it in the form of morse code. Acceptable input are command line arguments or the standard input.

Available options:

- l**     The **-l** option produces output suitable for led(4) devices.
- s**     The **-s** option produces dots and dashes rather than words.
- p**     Send morse the real way. This only works if your system has speaker(4) support.
- w** *speed*  
      Set the sending speed in words per minute. If not specified, the default speed of 20 WPM is used.
- c** *speed*  
      Farnsworth support. Set the spacing between characters in words per minute. This is independent of the speed that the individual characters are sent. If not specified, defaults to the effective value of the **-w** option.
- f** *frequency*  
      Set the sidetone frequency to something other than the default 600 Hz.
- d** *device*  
      Similar to **-p**, but use the RTS line of *device* (which must be a TTY device) in order to emit the morse code.
- e**     Echo each character before it is sent, used together with either **-p** or **-d**.
- r**     Decode morse output consisting of dots and dashes (as generated by using the **-s** option).

The **-w**, **-c** and **-f** flags only work in conjunction with either the **-p** or the **-d** flag.

Not all prosigns have corresponding characters. Use '#' for *AS*, '&' for *SK*, '\*' for *VE* and '%' for *BK*. The more common prosigns are '=' for *BT*, '(' for *KN* and '+' for *AR*.

Using the **-d** flag, it is possible to key an external device, like a sidetone generator with a headset for training purposes, or even your ham radio transceiver. For the latter, simply connect an NPN transistor to the serial port *device*, emitter connected to ground, base connected through a resistor (few kilohms) to RTS, collector to the key line of your transceiver (assuming the transceiver has a positive key supply voltage and is keyed by grounding the key input line). A capacitor (some nanofarads) between base and ground is advisable to keep stray RF away, and to suppress the minor glitch that is generated during program startup.

## ENVIRONMENT

Your LC\_CTYPE locale codeset determines how characters with the high-order bit set are interpreted.

ISO8859-1

ISO8859-15 Interpret characters with the high-order bit set as Western European characters.

KOI8-R Interpret characters with the high-order bit set as Cyrillic characters.

ISO8859-7 Interpret characters with the high-order bit set as Greek characters.

## FILES

*/dev/speaker* speaker(4) device file

## SEE ALSO

speaker(4)

*ITU-T Recommendation F.1*, Operational provisions for the international public telegram service, Division B, I. Morse code.

*ITU-R M.1677-1*, International Morse code, <https://www.itu.int/rec/R-REC-M.1677-1-200910-I/>, 2009.

## HISTORY

Sound support for **morse** added by Lyndon Nerenberg (VE6BBM) <[lyndon@orthanc.ca](mailto:lyndon@orthanc.ca)>.

Ability to key an external device added by Jörg Wunsch (DL8DTL).

Farnsworth support for **morse** added by Stephen Cravey (N5UUU).

## BUGS

Only understands a few European characters (German and French), no Asian characters, and no continental landline code.

Sends a bit slower than it should due to system overhead. Some people would call this a feature.