Morse Keying Practice for Novice Telegraphers by W5AWS

1 Motivation

Practice makes perfect! Actually, perfect practice makes perfect, especially when sending Morse code. The first time I tried to send Morse by hand, my sending was very erratic, and the muscles in my arm ached from the repetitive exertion. Finally, I decided to relax into a comfortably slow rhythm. What one needs is a strict task-master to force correct sending. What better task-master than a Morse decoder? Hence, this paper.

2 Background

I started learning Morse several years ago using the book by David Finley, N1IRZ, *Morse Code: Breaking the Barrier*, based on the Koch Method, in conjunction with an MFJ-418 Pocket Morse Code Tutor. Later, and more recently, I finished learning the code by copying eight-character groups at 25 WPM using the Android app by IZ2UUF (https://www.iz2uuf.net/cw/).

Then I found that I couldn't understand slow code, prevalent among beginning telegraphers on air. Not only that, there is a big difference between reading machine-generated Morse and reading it on-air. So, it was back to the woodshed for me, mainly listening to call signs and QSOs generated by the IZ2UUF app, and sending code to a computer program using a straight key. Straight keys the Long Island CW Club highly recommends using by beginning telegraphers, on the premise that the rhythm of sending helps to reinforce the code.

Despite joining the Long Island CW Club, I decided that I really needed to get on the air. It was the push provided by the Tulsa Amateur Radio Club's CW Nervous Novice Net, started by Ky, KY5VAR, and Mike, KI5EGH, that got me going. But, I still need the continuous training of sending text and pangrams, which brings me to the MFJ-461 *Pocket CW Reader*, a device that I've owned for several years and not used much.

3 Logging Code Keying Practice

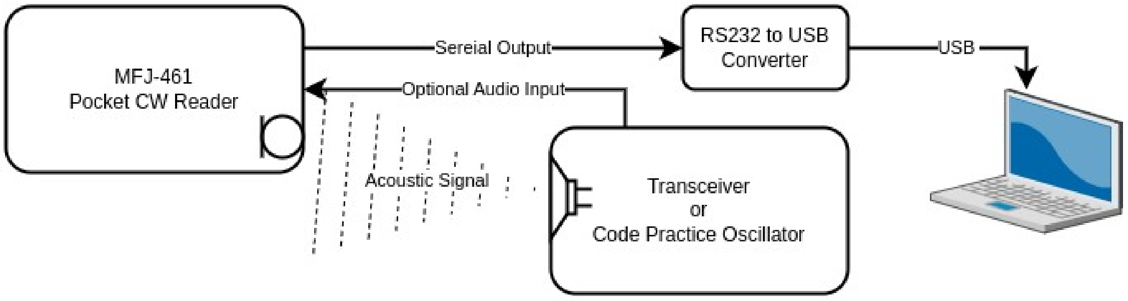
To work properly, CW readers are very picky about receiving the correct dit-dah spacing, which forces the telegrapher to send correctly. If not done correctly, the reader ignores or misinterprets the code.

I have had a lot of practice with *xcwcp*, an application on my computer that allows me to send it Morse code from my straight key via an interface that I built. However, this only applies to users of the GNU Linux operating system; in my case Linux Mint. Users of Mac-OS or Microsoft Windows may or may not have access to a similar software application, so I thought the MFJ-461 would be a device with cross-platform compatibility on any laptop or desktop computer.

4 System Configuration

Figure 1 is a block diagram of the morse keying practice system.

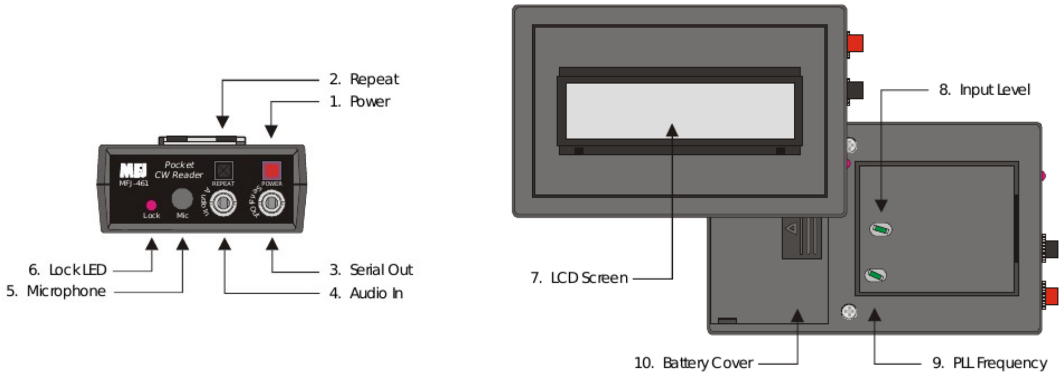
Figure 1



4.1 MFJ-461 Pocket CW Reader

Nice features of the MFJ-461 are the built-in microphone that can receive Morse acoustically, the audio-in jack, and the serial output jack that when connected to an RS232 port can transmit the Morse it receives to a terminal emulator on a computer that can then log the keying practice, which then can show improvement as time goes by.

Figure 2



4.1.1 Serial Data Connection

A cable with a 3.5 mm TRS plug connects to the Serial Out jack of the MFJ-461 at one end; at the other end, a DB9 Female D-Sub 9-pin connector mates with the Male D-Sub 9-pin connector of the RS232-USB convertor. The connecting cable is flexible three-core, wired as shown in Table 1.

Table 1

TRS 3.5 mm Plug	DB9 Female D-Sub
Tip	Pin-2 TXD
Ring	Pin-3 RXD
Sleeve	Pin-5 Ground

Building a cable like this is well within the capabilities of all amateur radio operators, probably from parts already to hand somewhere in the shack. The trick will be finding them.

4.1.1.1 Data Transmission Characteristics

Serial output data transmission characteristics are: 19200 baud, 8 data bits, no parity, 1 stop bit, and no flow control.

4.1.2 Acoustic Signal

On your computer, go to the ARRL web-site at this URL by clicking the link or scanning the QR-Code:

https://www.arrl.org/10-wpm-code-archive

Scroll down the page and start one of the *.mp3* files playing and increase the audio volume of computer. Switch on the MFJ-461 and place it near the speaker of the computer. Twiddle the Input Level and PLL Frequency controls until the Lock LED flashes brightly in time with the playback of the ARRL Code Practice Archive recording.

You should see characters scrolling across the LCD screen of the MFJ-461 along with an assessment of the words per minute of the Morse speed.

4.1.3 Optional Audio Input

If you have an auxiliary audio output on your transceiver or code practice oscillator, you can connect directly to the MFJ-461 through the Audio In jack by plugging in a mono 3.5 mm plug on the end of the connecting cable; the other end is whatever fits your device. Connecting the Audio In jack disconnects the internal microphone.

4.2 RS232 to USB Converter

A potential problem is that most, if not all, modern computers do not support RS232 serial communication. With Linux, a simple solution is the Digi Edgeport 1 port RS-232 DB9 to USB Converter with captive 2 meter USB cable, which just plugs in and goes:

https://www.digi.com/products/models/301-1001-15

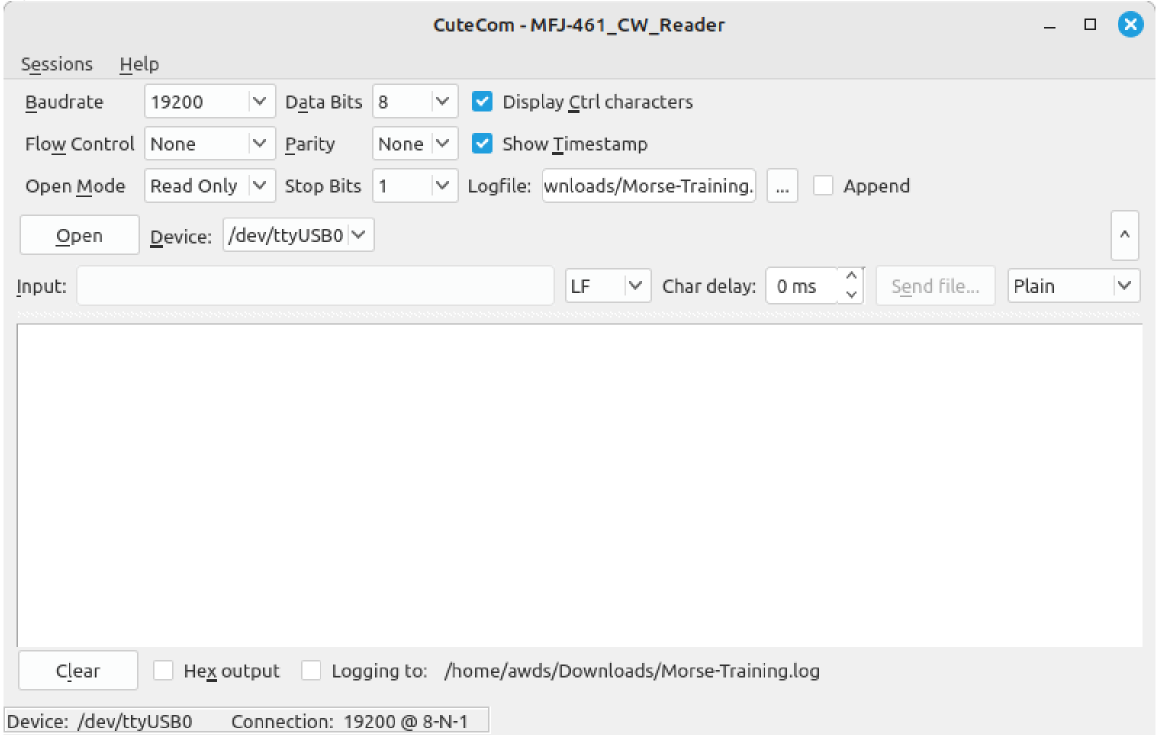
Other computer systems may require download and installation of driver software, and there will likely be other RS232 to USB converters options available to suit your particular operating system and computer hardware.

5 Computer

On your computer, you need a terminal emulator that connects to the serial port. On Linux, there are all kinds of free and easy options available. Linux Mint happens to have *CuteCom*, an application with a graphical user interface, available for download and installation from the software repositories. Once you have the emulator installed, adjust the communication settings to match those listed in section *4.1.1.1 Data Transmission Characteristics*.

Figure 3 is a screenshot of the *CuteCom* settings on my computer, with the settings saved to a configuration file named *MFJ-461_CW_Reader*.

Figure 3



CuteCom - MFJ-461_CW_Reader

Baudrate: 19200 | Data Bits: 8 | ☑ Display Ctrl characters

Flow Control: None | Parity: None | ☑ Show Timestamp

Open Mode: Read Only | Stop Bits: 1 | Logfile: wnloads/Morse-Training. | □ Append

Device: /dev/ttyUSB0

Input: | LF | Char delay: 0 ms | Plain

□ Hex output | □ Logging to: /home/awds/Downloads/Morse-Training.log

Device: /dev/ttyUSB0 Connection: 19200 @ 8-N-1

CuteCom can log traffic to a text file named *Morse-Training.log*, by checking the *Logging to:* option.

6 Conclusion

In Figure 4 you can see that I need more practice sending clean code when I tried to send these two pangrams:

- HEAVY BOXES PERFORM QUICK WALTZES AND JIGS.
- AMAZINGLY FEW DISCOTHEQUES PROVIDE JUKEBOXES.

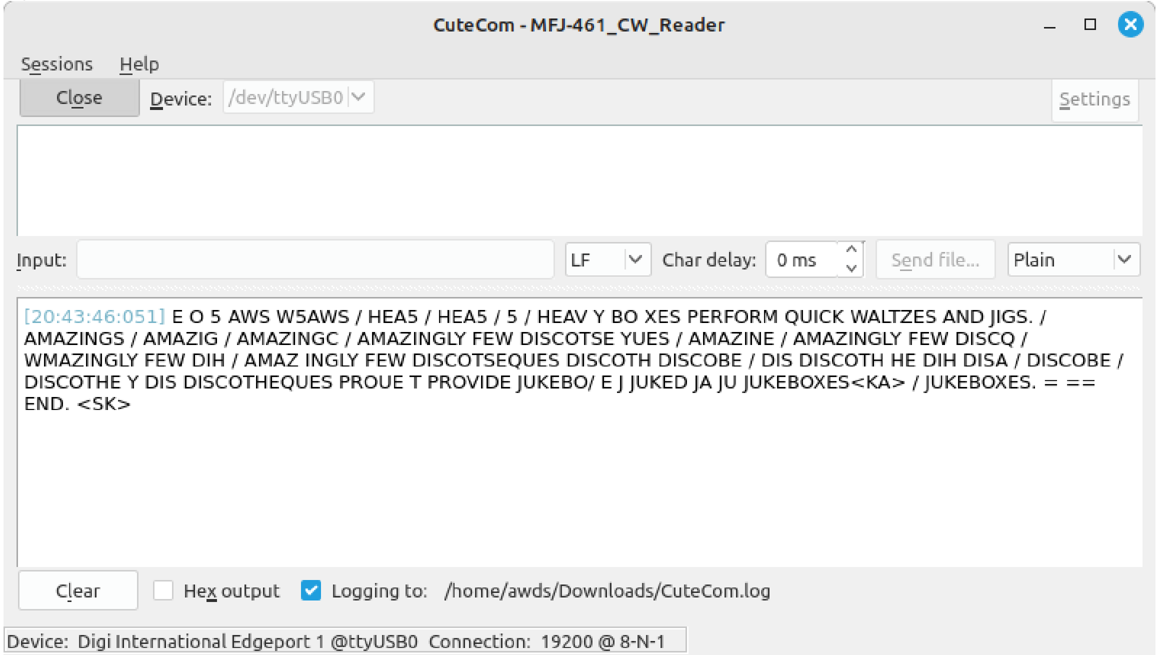
At first, I sent my call sign a couple of times to get the CW Reader tuned into the speed of my fist, which takes a moment or two, then I sent the pangrams.

Even though I was sending at about 12 or 13 WPM, some character combinations my mind blocked, preferring something else entirely, so I sent a slash then tried again.

The *LICW Club* tutors recommend short bursts of practice of not more than fifteen or twenty minutes duration, two or three times a day.

By my lights, this off-line practice rig is a good way to hone one's fist without adding to the on-air QRM, especially when band conditions prevent on-air operations anyway. I can turn down the RF power of my Penntek TR-45 Skinny and practice sending with the MFJ-461 listening to the sidetone, either acoustically or with direct audio input via the phone output of the radio, adjusting the audio level until the MFJ-461 is receiving correctly.

Figure 4



Device: /dev/ttyUSB0

Input: LF Char delay: 0 ms Plain

```
[20:43:46:051] E O 5 AWS W5AWS / HEA5 / HEA5 / 5 / HEAV Y BO XES PERFORM QUICK WALTZES AND JIGS. /
AMAZINGS / AMAZIG / AMAZINGC / AMAZINGLY FEW DISCOTSE YUES / AMAZINE / AMAZINGLY FEW DISCQ /
WMAZINGLY FEW DIH / AMAZ INGLY FEW DISCOTSEQUES DISCOTH DISCOBE / DIS DISCOTH HE DIH DISA / DISCOBE /
DISCOTHE Y DIS DISCOTHEQUES PROUE T PROVIDE JUKEBO/ E J JUKED JA JU JUKEBOXES<KA> / JUKEBOXES. = ==
END. <SK>
```

□ Hex output ☑ Logging to: /home/awds/Downloads/CuteCom.log

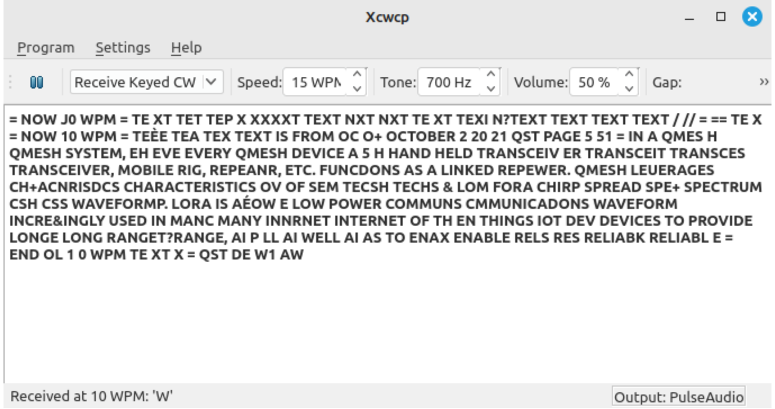
Device: Digi International Edgeport 1 @ttyUSB0 Connection: 19200 @ 8-N-1

LICW Club tutors also recommend practice sending free text, either sentences from a book, magazine, newspaper, or conjured from one's mind.

We need numerical sending practice too, so it helps to send the numbers 0 through 9, and the numerical date and time.

One practice I follow is to copy with pencil on paper the 10 WPM ARRL code practice files then open the *.txt* file and send it back to the computer with my straight key. Figure 5 is an example sent to *xcwcp*. Again, my mind blocks certain combinations of letters, forcing me to try again, which is no problem off-air in the safety of my shack with no need to announce errors with *dit dit dit*, prior to sending corrections.

Figure 5



Receive Keyed CW Speed: 15 WPM Tone: 700 Hz Volume: 50 %

```
= NOW J0 WPM = TE XT TET TEP X XXXXT TEXT NXT NXT TE XT TEXI N?TEXT TEXT TEXT TEXT / // = == TE X
= NOW 10 WPM = TEÈE TEA TEX TEXT IS FROM OC O+ OCTOBER 2 20 21 QST PAGE 5 51 = IN A QMES H
QMESH SYSTEM, EH EVE EVERY QMESH DEVICE A 5 H HAND HELD TRANSCEIV ER TRANSCEIT TRANSCES
TRANSCEIVER, MOBILE RIG, REPEANR, ETC. FUNCDONS AS A LINKED REPEWER. QMESH LEUERAGES
CH+ACNRISDCS CHARACTERISTICS OV OF SEM TECSH TECHS & LOM FORA CHIRP SPREAD SPE+ SPECTRUM
CSH CSS WAVEFORMP. LORA IS AÉOW E LOW POWER COMMUNS CMMUNICADONS WAVEFORM
INCRE&INGLY USED IN MANC MANY INNRNET INTERNET OF TH EN THINGS IOT DEV DEVICES TO PROVIDE
LONGE LONG RANGET?RANGE, AI P LL AI WELL AI AS TO ENAX ENABLE RELS RES RELIABK RELIABL E =
END OL 1 0 WPM TE XT X = QST DE W1 AW
```

Received at 10 WPM: 'W'

Output: PulseAudio

I notice that the MFJ-461 reports the ARRL 10 WPM code practice files being sent at 15 WPM, but with Farnsworth spacing to 10 WPM.

7 Glossary

app	Application
ARRL	Amateur Radio Relay League
CW	Continuous Wave, synonymous with Morse code
DB9	D-subminiature 9-pin connector, synonymous with DE-9
GNU	Gnu Not Unix
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LICW	Long Island CW
Mac	Macintosh
MFJ	Martin F. Jue
OS	Operating System
PLL	Phase-Locked Loop
QRM	Interference
QSO	Contact
RS232	Recommended Standard 232
RXD	Receive Data
TRS	Tip, Ring, Sleeve
TXD	Transmit Data
URL	Universal Resource Locator
USB	Universal Serial Bus
WPM	Words Per Minute
xcwcp	X-windows CW Code Practice