RADIO MAGiC
Robert C. Mazur, VA3ROM | www.va3rom.com | va3rom@gmail.com

First published in the July-August 2023 issue of the Canadian Amateur

Snap Circuits Part 2: “Mr. Morse”

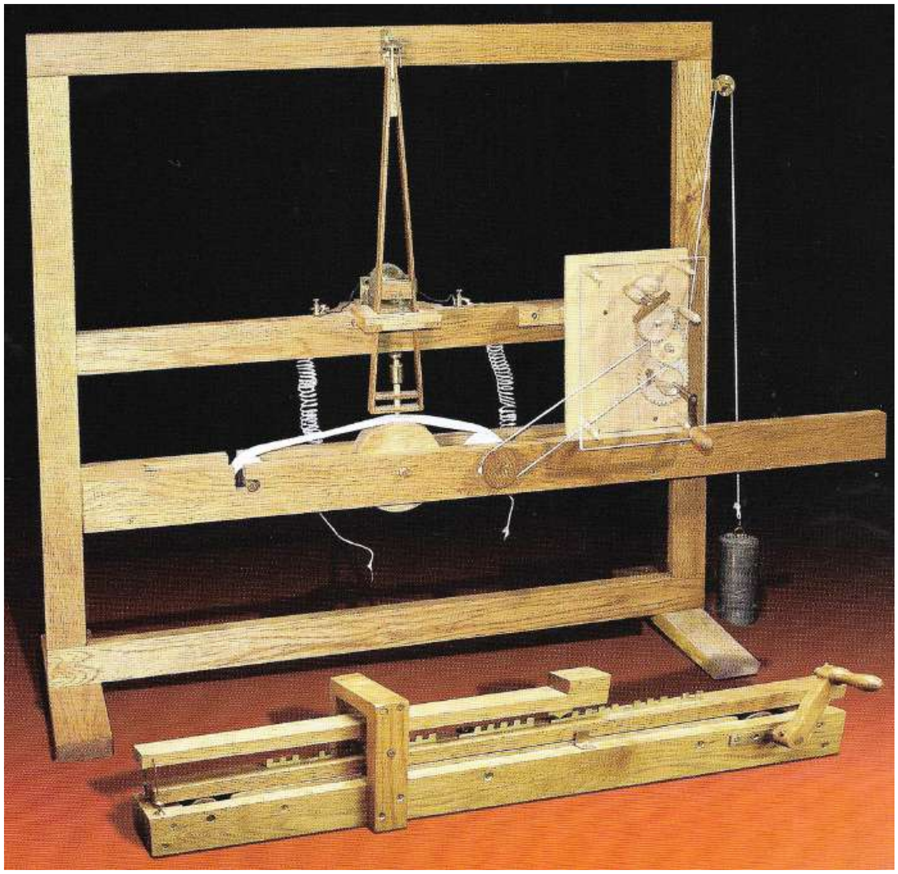
Introduction

This isn’t the full history and story of the “Electro-Magnetic Telegraph” as Samuel Morse and Alfred Vail called it or about the complex love/hate personal/business relationships between their two families. But it’s rather bittersweet because Vail died broke and largely forgotten while Morse became rich and famous. After ten years of avoiding it, I’m finally writing about a mode that we Amateurs either love or hate with a passion. The inspiration for this article came about from Elenco Electronics’ amazing Snap Circuits ARCADE microcontroller kit, which resulted in a gadget (to borrow the Arduino term) that I call “Mr. Morse”’.

Genesis

The original Morse numbers code and electro-magnetic telegraph system used movable type (custom cast lead “slugs”) and required a words/phrases to/from numbers encode/decode dictionary paired with a printing (pen, pencil or steel stylus) ticker-tape register (receiver) for visual decoding of short messages at each end of a single wire (Figures 1A and 1B, next page). Morse was a professional artist (painter)—not an engineer—so his numbers code and mechanical methodology wasn’t commercially viable, but it was proof of concept that information could be transmitted by electrical pulses sent over a single wire plus a record of messages could be made and saved. The original paper tape, embossed using a steel stylus, of the first public Morse code message (24 May 1844) was hand annotated by Samuel Morse for posterity and is available for viewing here http://tiny.cc/9qx5wz.

Figure 1A: Morse Telegraph System (1837)—Pre-Vail
A working replica of Morse's original printing register receiver (background) and port rule transmitter (foreground). The telegrapher used a dictionary to encode words/phrases into numbers then loaded custom cast moveable type lead slugs sequentially into the port rule. A hand crank moved the slugs under a rocker arm dipping a wire staple in/out mercury filled capsules to open/close a battery external circuit (not shown). Credit: "Were Discovery Sparks Imagination", John D. Jenkins, 2009, page 37.



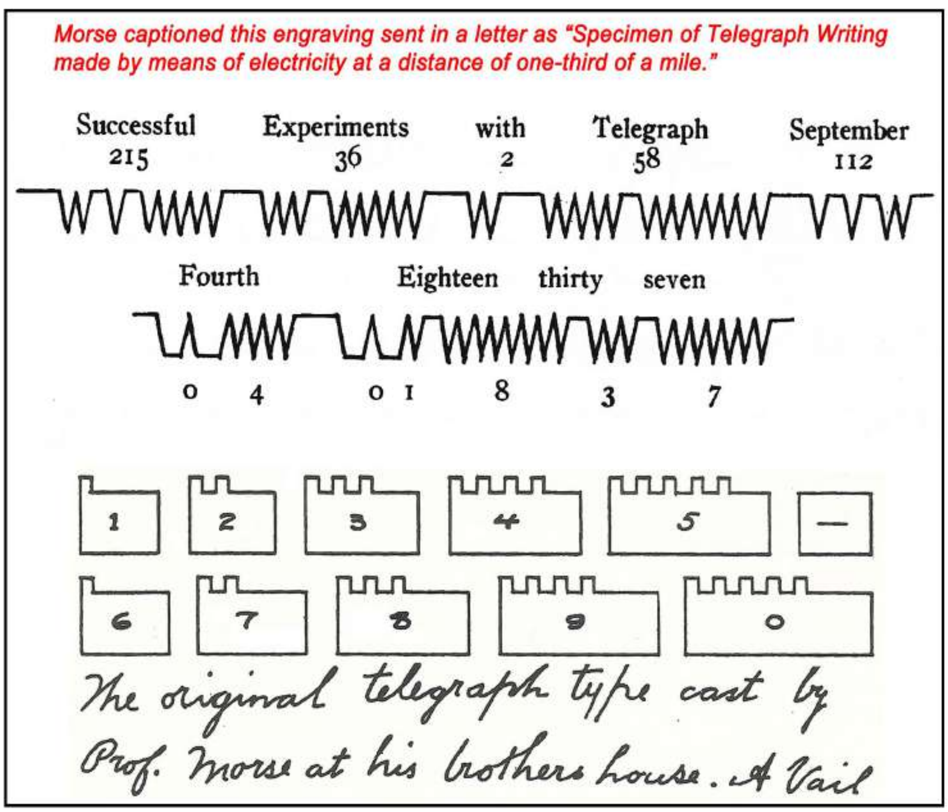


Figure 1B: Morse Printing Telegraph Example (1837)
The telegrapher at the other end of the wire used the same dictionary to decode the received numbers back to words/phrases then wrote the message in longhand on a telegram form for delivery to the customer. Credit: "Early History of the Electro-Magnetic Telegraph from the Letters and Journals of Alfred Vail. Credit: University of Michigan library collection.

In 1837, after a chance meeting with Samuel Morse, who was an art instructor at the University of the City of New York (now NYU), young graduate student and engineering genius Alfred Vail became a business partner (so he thought) or was only a work-for-hire assistant to Morse (so he thought). Either way, Vail contractual agreed to improve the mechanical side of Morse's telegraph system with the financial assistance of Alfred's wealthy businessman father Stephen Vail. I've read the original contact wherein the Vail's agree to finance the entire cost of redesigning Morse's electro-magnetic telegraph system with the intent that it would be sold outright to the U.S. government in exchange for a percentage of the sale, and that Samuel Morse would retain all patent rights (and all legal liabilities) of said system as being the sole inventor. And Vail also had a "do-or-die" deadline of only a few months, but he did it only a week late with his redesign of the register making it more compact and more reliable by using a windup clock-driven control mechanism instead of the drop weight (Figure 1C) along with his invention of the "finger" key then the "straight" (level correspondence) hand key. The hand key simplified transmission, increased speed and accuracy of messaging, eliminating the port rule and moveable type. Basically, Vail improved on Morse's original "hardware" while Morse improved his "software".

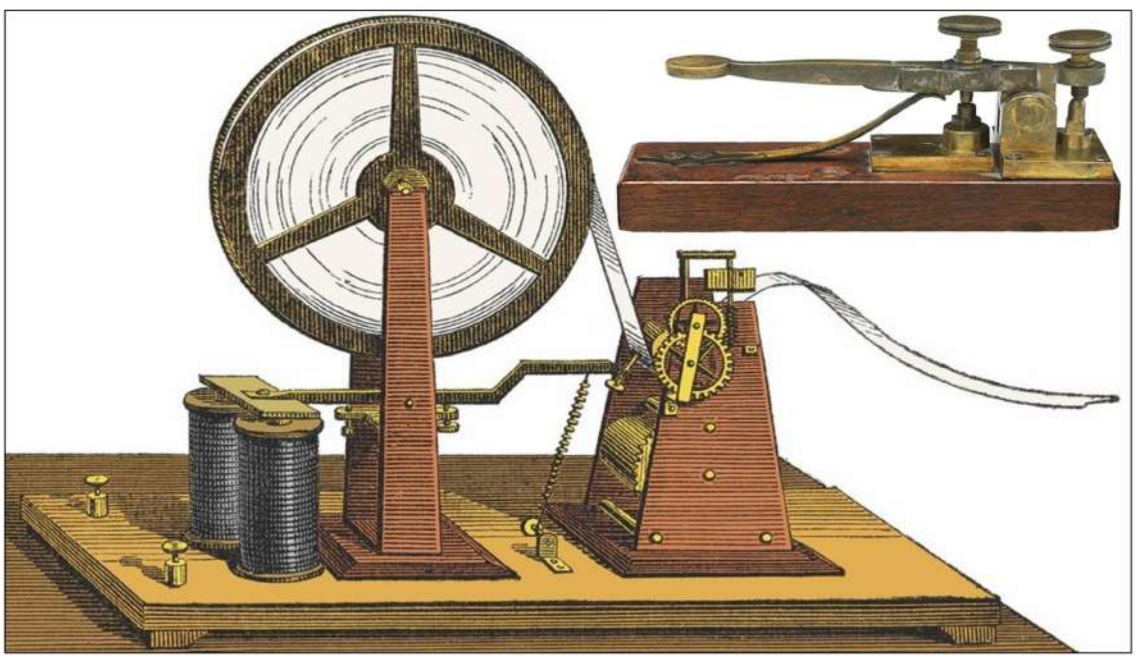


Figure 1C: Alfred Vail early register and hand key (insert)

Vail continued refining the register and bequeath his personal handmade favourite to Morse in his Will. If they had any acrimony in this life, Vail didn't want it to carry over to the next.
Credit: Drawing from the Smithsonian Institution.

A new dot/dash code was created to include letters of the alphabet (eliminating the dictionary). It was based on the binary system, which was created by the famous 17th century mathematician Gottfried Wilhelm Leibniz ("Explication de l'Arithmétique Binaire"). Morse visited local print shops and counted the number of lead slugs in each letter's storage bin then he used this information to assign the most frequently used letters the shortest time units. In English, the letter E is the most used so it was defined as the short mark (dot) of one time unit while T (the second most used) was defined as the long mark (dash) of two time units (not three as today). This became known as the American Morse (also railway/landline) code to distinguish it from the optical and electric telegraph systems used in Great Britain and continental Europe at the time. Alfred Vail explained this in his 1845 book "The American Electro Magnetic Telegraph".

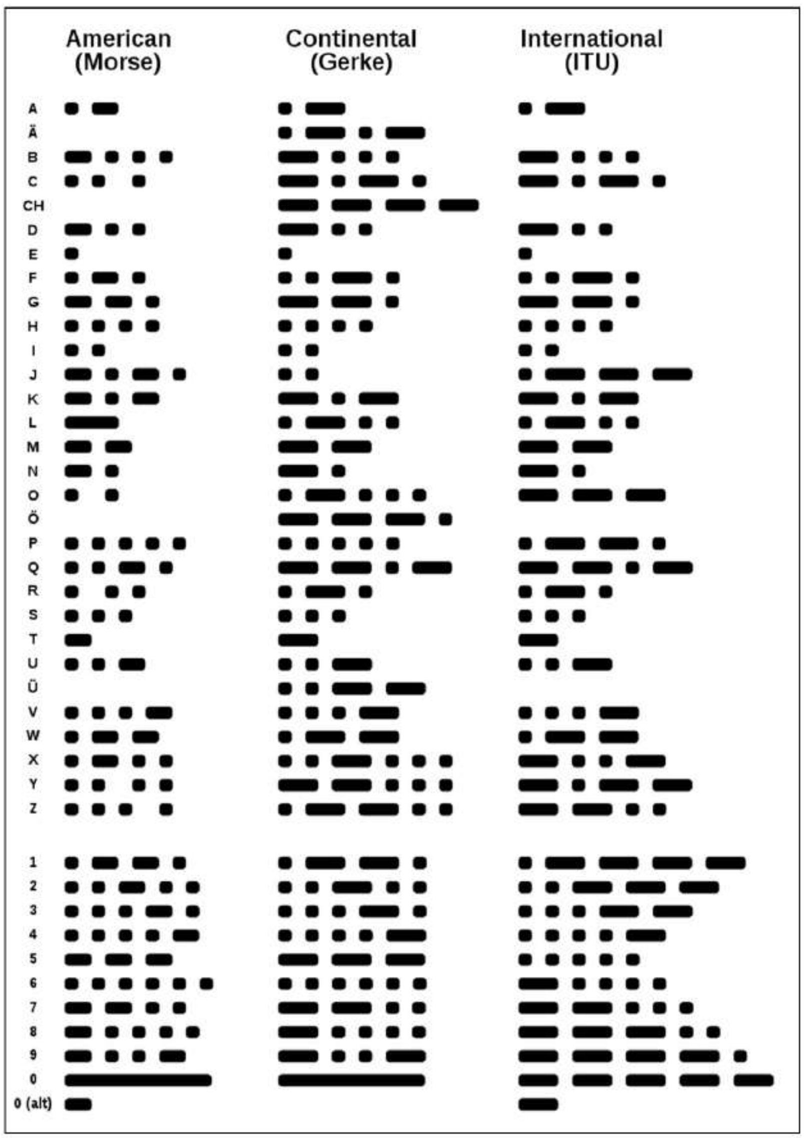
"By examining the telegraphic alphabet, the characters will be found to be made up of dots: short and long lines—and short and long spaces. A single touch of the key, answers to a single dot; which represents the letter, E. One touch of the key prolonged, that is, the contact at the key continued for about the time required to make two dots, produces a short line, and represents T. A single touch for about the time required to make four dots, is a long line, and represents L. A single touch for about the time required to make six dots, is a still longer line and represents the 0 of the numerals. If the use of the key be suspended for about the time required to make three dots, it is a short space, used between letters. If suspended for the time required to make six dots, it is a long space, used between words, and a longer space is that used between sentences. These are the elements which enter into the construction of the telegraphic characters, as used in transmitting intelligence. The alphabet is represented by the following combination of these elements."

Note 1: The "long space" was written as a period in messages delivered to customers. Other punctuation wasn't defined until the American Telegraph Conference of 1854.

Note 2: Inept/inexperienced "fists" often sent sloppy "hog-Morse" or "ham-fisted" code caused by letters with the same number of dots only differentiated by their inter-element spacing: I and O (two dots); C, R and S (three dots); H, Y and Z (four dots). Figure 2 refers. To be called a "hog" or a "ham" on the circuit wasn't a telegrapher's term of endearment!

Morphosis

In 1847, Friedrich Gerke witnessed demonstrations of Morse's code and telegraph system in Hamburg, Germany. He immediately realized that it was far superior to the optical telegraph system used within the German Confederation (central Europe). But he also realized that while it worked well for English text, most European languages had accented letters and digraphs with alphabets having more than 26 letters. But only a year later, the very determined Gerke had not only translated Vail's "The American Electro Magnetic Telegraph" into German, he had also redesigned large portions of the American Morse code!



Gerke gave the dot problematic letters their own dot/dash combinations. The two different length dashes (short and long) were discarded leaving just the dot (still one time unit) and the modern longer dash (now three time units). Added were the German umlaut letters Ä, Ö and Ü plus the digraph CH. This became known as the Hamburg or Continental (German) code, adopted in 1851 by the German-Austrian Telegraph Association. *Note: Gerke's revisions of numbers plus letters O, P, X, Y and Z had some really strange dot/dash combinations plus the letters I and J used the same code. Figure 2 refers.*

Figure 2: The Three "Morse" Codes

This comparison chart shows the almost Darwinian-like evolution.

Credit: "Spinningspark"; Wikipedia: CC BY-SA 3.0.

In May 1865, the Continental (German) code's problematic numbers and letters were again redefined. The umlaut letters and digraph were removed because they were considered too "Prussian" for French Emperor Napoleon III, who was also the host of the first international telegraphy conference held in Paris, France. The result was the International Morse code, which was adopted by the newly created International Telegraph (now Telecommunication) Union or ITU. The original mandate of the ITU was to establish international rules and regulations for telegraph networks among the first 20 signatory treaty nations (now nearly 200). It created a framework standardizing telegraphy equipment, established uniform operating procedures, set tariff rates and accounting guidelines. But the three "Morse" codes weren't mutually comprehensible; American landline/railway telegraphers stubbornly stuck with their version for domestic use (allowed by the ITU treaty) while the Germans naturally preferred theirs. However, the International Morse code quickly became the de facto "lingua Franca" by the rest of the world because its use was mandatory when transmitting messages via oceanic submarine communication cables that rapidly encircled and interconnected the world by the end of the 19^{th} century, creating the very first "internet".

Today

Morse code (meaning the international version from now on) has been adapted to control devices and assist persons with various disabilities by using puff-and-suck, head-tilt or eye-blink, foot or finger-tap, etc. Your smartphone can switch between the standard QWERTY to a Morse code virtual keyboard. Languages based on the Cyrillic, Arabic and Japanese alphabets use character matching of their alphabets to the same or nearly the same Latin alphabet letters with modifications and/or additions for letters that don't exist in the Latin alphabet. Of course, Morse is very much alive and thriving in the Amateur Radio world and it's still the only digital mode that we humans can encode/decode in our heads.

As an aside, most Amateurs refer to Morse code as "CW", but CW means "continuous wave"—a transmission method not a mode. In 1902, radio pioneer Reginald Fessenden invented the mechanical (alternator driven) continuous wave transmitter and "CW" was used to differentiate between it and spark gap (damped wave) Morse code. This is no longer required because spark gap use is prohibited on the Amateur bands and there are umpteen "CW" modes in use today.

Meet "Mr. Morse"

Elenco's Snap Circuits ARCADE kit was introduced in the part 1 article (see *TCA* May-June 2023) wherein it was described how to build and code a simple PICAXE microcontroller Morse code practise oscillator (CPO). Over the ensuing months, it has "evolved" far beyond my expectations (Figure 3, below and Figures 4A and 4B, next page). Snap Circuit kit builds may be for "children" and are large, colourful and cute or artsy looking, but Elenco also packs in a lot of learning and fun for "Ages 8 to 108", as their box cover art proudly states. And once I reversed engineered the undocumented ARCADE source code, it was fairly easy to develop my own variant or "fork".

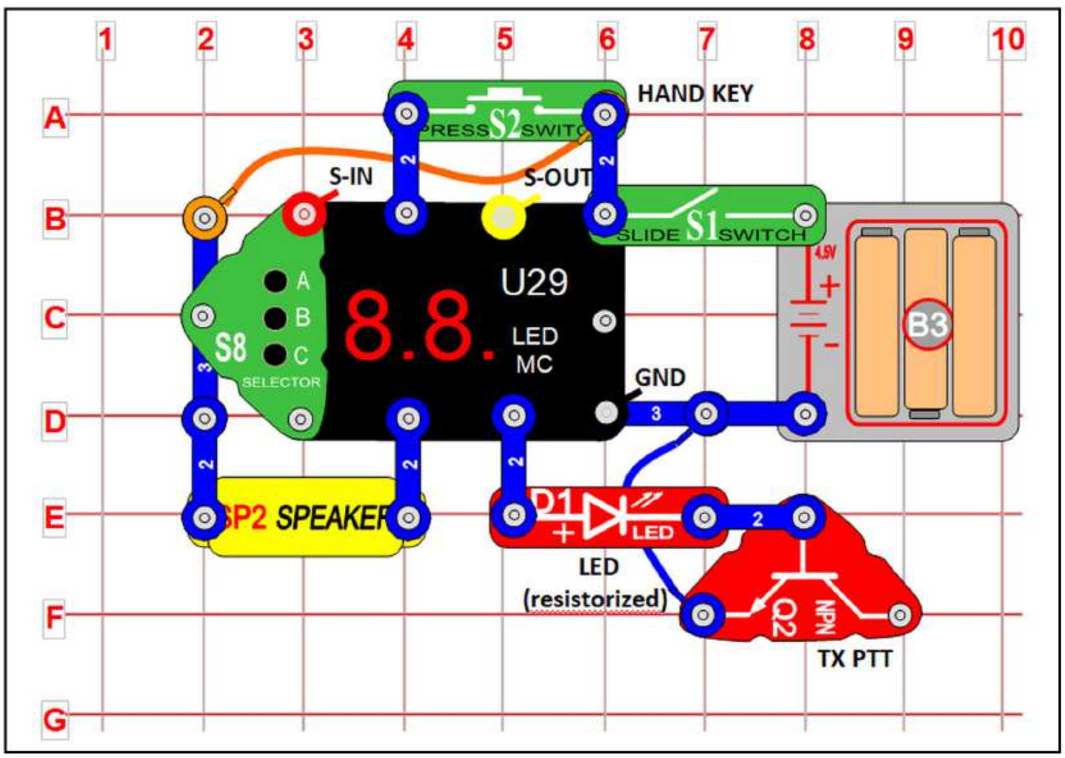
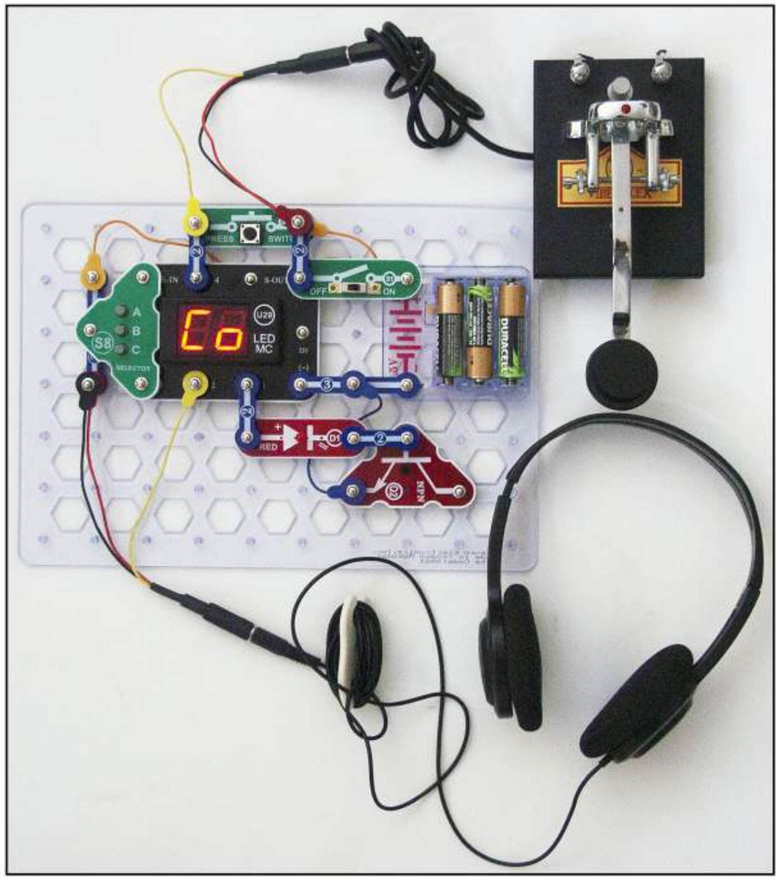


Figure 3: "Mr. Morse" Pictorial Elenco's Circuit Designer (MS Word, only) makes it easy to swap virtual Snap Circuits electric or electronic components in/out and/or position them as required before you build any real-world snap circuit.

Revolution Education's (Rev. Ed.) free PICAXE Editor 6.0 was used to develop the "Mr. Morse" control code. As an old-school programmer, I still prefer to use free form line coding instead of PICAXE Blockly and object orientated programming (OOP). The part 1 article refers. My source code is fully commented so I won't go into the nitty-gritty of how it works because you can download the supplement zip file from my Radio Magic web page and study the code. I've also included several interesting articles including Samuel Morse's bound diaries (Volume 2) that cover the telegraph's genesis, its development, his relationship with the Vail family, etc.

Figure 4A: "Mr. Morse" CPO Wiring
Audio output is fairly loud because there's no volume control so you may want to use a headset with one built-in. The PICAXE series of microcontrollers produce a rather raspy sounding square wave tone, which is probably close to what rotary spark-gap Morse code sounded like 120 years ago. Note: My TRS jacks are wired as per the PICAXE programming wiring pattern so the tip and ground (sleeve) must be reversed to work like a "regular" TRS stereo jack. See Figure 5, next page, for the PICAXE TRS wiring pattern.



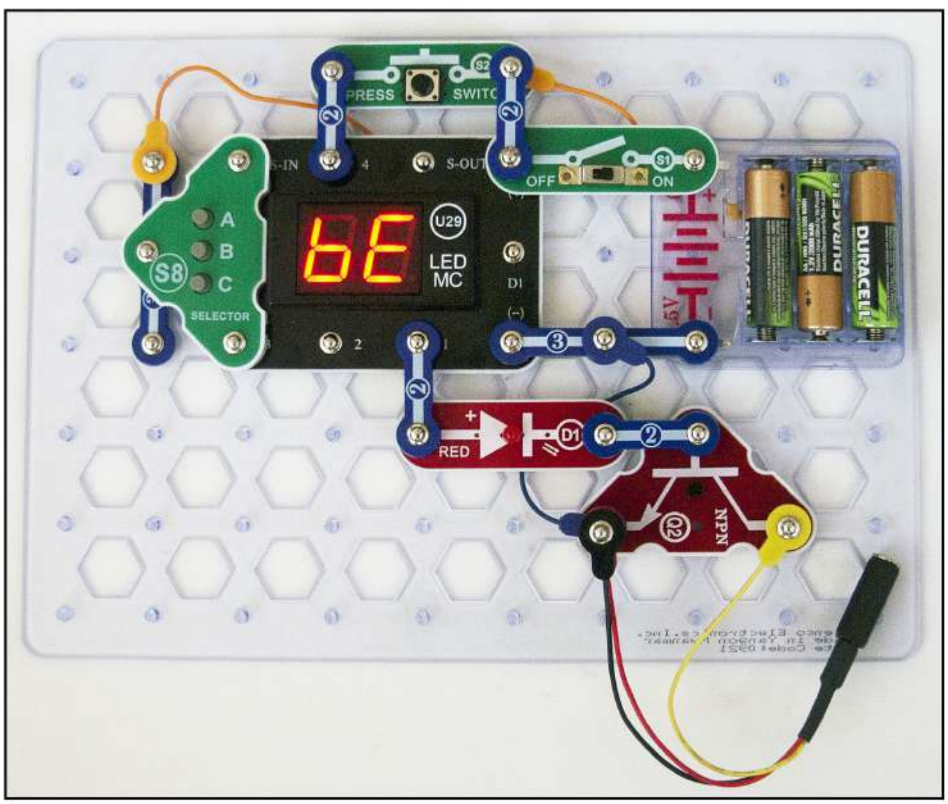


Figure 4B: "Mr. Morse" Beacon/Keying Wiring
I've added NPN transistor Q2 from the ARCADE kit to connect Mr. Morse to an external solid state transmitter for either manual transmission using a hand key or automatic standalone beacon (bE) mode.

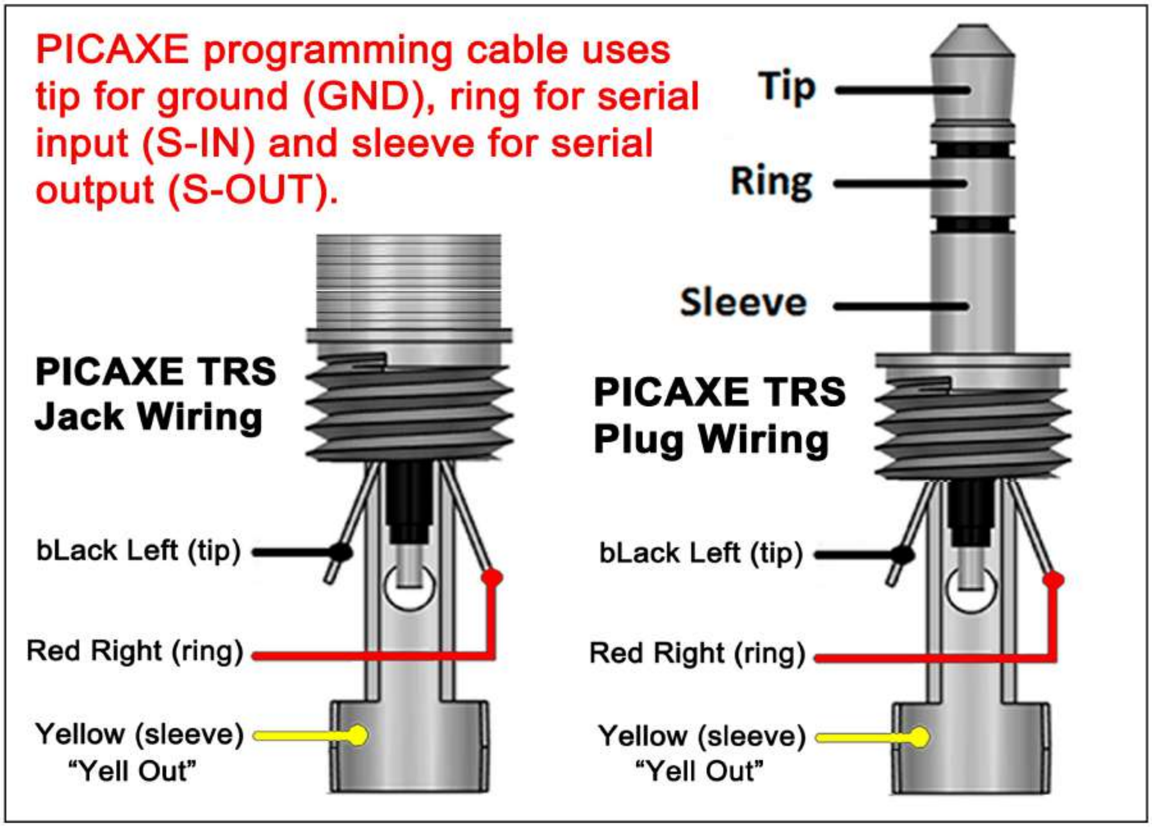


Figure 5: PICAXE TRS Wiring Pattern
Credit: Components101.com

“Mr. Morse” Features and Functions:

A. Code practise oscillator (CPO). Use the press switch or hand key (preferred).
B. Battery voltage monitor (useful with battery packs).
C. Set number of five-character code groups to send.
D. Set character sending speed.
E. Set user programmable beacon transmission repeat interval.

The following functions also print characters to the PICAXE Editor serial terminal if connected via the PICAXE programming cable to the host computer.

F. Transmit user programmable beacon (optionally keys external transmitter).
G. Send random five-character alphanumeric groups plus common punctuation.
H. Send random five-character number groups only.
I. Send random five-character alphabet group only.

Note: The values for items C, D and E are automatically stored in electrically erasable programmable read-only memory (EEPROM) and reloaded into random access memory (RAM) on power up.

A built-in dual 7-segment light emitting diode (LED) displays the two letter abbreviations of menu items plus parrots them back in Morse code via speaker (SP2) or headphones, as well sending a Morse code "R" for "Roger" to confirm user selections when pushbutton (S8) button C is pressed. Buttons A and B scroll up/down through menu items and also increment/decrement any numbers. One caveat; the 7-segment LED display serial data control red LED (D1) and NPN transistor (Q2) share the same PICAXE pin/port so an external transmitter shouldn't be connected until after activating the beacon mode. I've included a 10-second delay for this purpose, which should be enough time for you to connect the keying line (Q2) to your transmitter. Some transceivers also have a menu "BK" key on/off function (Yaesu FT-857D) or other method and this may be more convenient so the 10-second delay can be removed.

Speaking of cables, if you're not a PICAXE user, you'll need to purchase the custom programming cable, a half dozen coloured (red, black and yellow) snap-connector wires (RobotShop), and a couple of 3.5 mm stereo or tip, ring, sleeve (TRS) jacks (any electronic parts supplier) to make two snap-jack TRS adapters. Rev. Ed. reversed the tip and ground (sleeve) connections of their programming cable creating a teeny-tiny problem with the de facto TRS standard (Figure 5 refers). You can either solder the TRS jack adapters as per normal then remember to switch the tip and ground around when connecting to the PICAXE LED MC to use the programming cable, or you can follow Rev. Ed.'s TRS wiring pattern (my preference) then do vice versa to use as a regular audio/hand key jack adapter. Figures 5A and 5B refer.

Computerized Morse Encoding/Decoding Process

Using 8-bit binary arithmetic, we can encode up to 256 Morse characters (0 to 255) or most anything else for that matter. There're several methods you can use but here's the simplest, fastest and therefore the best, IMHO.

> **Example:** Ä (umlaut A) is · — · — (dot dash dot dash) in German Morse code. What're its encoded binary/integer values?

Solution: Let all dashes equal binary 1; let all dots equal binary 0; let a stop bit equal binary 1 (marks the end of a character's elements). Map all dots and dashes including the stop bit to an 8-bit binary number using left to right bit encoding (my preference). All unused bits are binary 0. Add all non-zero bit values to determine the encoded binary/integer value.

Method 1: I'm a leftie plus we send character elements from left to right.

1	**2**	**4**	**8**	**16**	**32**	**64**	**128**	
0	1	0	1	1	0	0	0	= 2 + 8 + 16 = 26
·	—	·	—	stop bit				

Method 2: If you prefer right to left encoding. (Note the bit reversal!)

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**	
0	0	0	1	1	0	1	0	= 16 + 8 + 2 = 26
			stop bit	—	·	—	·	

Therefore umlaut A is encoded 8-bit binary 00011010 and the equivalent encoded integer value is 26. What we've done is "plant" a "seed" for a binary search tree (BST) that we can "grow" by adding more and more encoded Morse code characters (nodes) interconnected by "branches" that "sprout" left (for a dot) or right (for a dash). To explain how "Mr. Morse" searches the BST for any encoded binary value that we need to decode, I'll use plain text pseudo-code. Figure 6, next page, refers.

Start BST Routine

Is encoded binary value odd or even? (Modulo function used)

If odd, take the right branch and send a dash (three time units)

If even, take the left branch and send a dot (one time unit)

Divide encoded binary value by 2 (cuts the branches to search in half)

Is encoded binary value now equal to 1? (stop bit)

If Yes, exit BST routine (all done with this character)

If No, pause for inter-element space (one time unit)

Repeat BST Routine

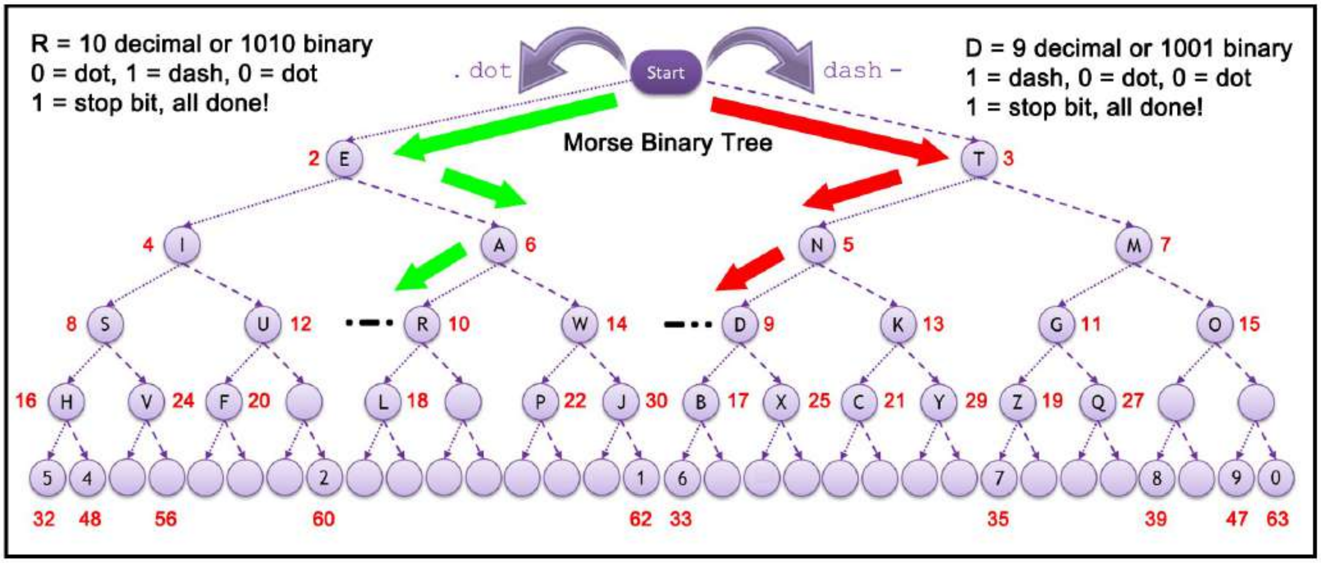


Figure 6: Morse Binary Tree

Accented letters, digraphs and punctuation were removed for easier visualization. Credit: "Aris00" at English Wikipedia; transferred by "Ddxc" to the Wiki Commons.

As you can see, the Morse code BST has a very small number of nodes and branches, but even a BST with a billion branches would only need 29 divisions by 2 (binary right bit shifting) to get to a bottom node. This takes perhaps a fraction of a microsecond but it gives the illusion of artificial "intelligence" (AI). But one typo, logic or syntax error ("bug") in any computer program usually results in GIGO, so having plain language comments within your code will really help hunt for any bug(s).

Epilogue

Today, many revisionists feel that Alfred Vail was cheated of his rightful due because there's a contentious issue as to how much of the revised code and telegraph system was Morse's and how much was Vail's. It's a rather moot point considering Vail signed a work-for-hire contract. When the U.S. government opted not to buy or even invest in the project in 1838, Alfred lost all interest in the project and abandoned Morse, as was his right by the terms of the contract. But Morse refused to let his dream die, and he struggled on alone for years always fearing that someone, somewhere in the world would "steal his thunder". Vail only returned once the political wind had changed and Morse had secured the needed federal funding to build a test telegraph line between Washington and Baltimore in 1844. And the rest is history...

My Final

Feel free to make your own modifications or even translate the PICAXE code for use with other microcontrollers like the Arduino Uno. And because I finished most of this article on May 4th or “Star Wars” day, along with “May the fourth be with you!” I’ll also add “May the Morse be with you!”—73

References and Resources

American Morse https://tinyurl.com/ycx6x58h

Binary Search Tree https://tinyurl.com/49f2mxnj

Elenco Designer https://tinyurl.com/bddsh8m5

iCW https://tinyurl.com/mrzwr3pn

ITU https://tinyurl.com/53kh9rz8

Morse code Assistive Technology
https://tinyurl.com/6ewzfkbk and https://tinyurl.com/52vr7dcs

Morse code for non-Latin alphabets https://tinyurl.com/5f3smww6

PICAXE (Revolution Education) https://rev-ed.co.uk

RobotShop https://ca.robotshop.com

Snap Circuits https://tinyurl.com/2p84je9r

"The American Electro-Magnetic Telegraph" https://tinyurl.com/2p8a3tb4

The 1865 International Telegraph Conference https://tinyurl.com/yc5tkjpn