

# History and Philosophy of the MorseKOB Program

Les Kerr

24 May 2006

Revised, 15 November 2007

The story of the MorseKOB program parallels my own interest in landline telegraphy and American Morse. When I was a kid, I spent a lot of time hanging around the local railroad depot. I was fascinated with the clatter of the telegraph sounder, and amazed by the skill of the station agent as he copied train orders over the wire. Although I worked a lot of CW on the ham bands as a teenager, I never had the opportunity to learn American Morse.

My interest in landline telegraphy lay dormant for 45 years until I became a member of the Northwest Railway Museum in Snoqualmie, WA, where I worked as a volunteer train crew member, signal maintainer, and maintenance-of-way worker. One day in May 2002, I was chatting with James Sackey in the museum bookstore and we began talking about Morse code. James told me about a program called The Mill, which I immediately downloaded and installed on my computer. Soon after that, I bought a KOB<sup>1</sup> and built an interface circuit to drive the sounder from the serial port. I was hooked!

## The Telegrapher's Toolkit

After I had been working with The Mill for a few months, I got the urge to write my own Morse code program. I wanted to create a collection of modules, each of which would perform a specific task, so that the whole collection could be assembled together in a very flexible way. The basic concept was to provide all the components necessary to construct a virtual telegraph office. For example, if you didn't have a sounder, there'd be a simulated sounder module you could plug in.

Although I had many years of experience with computers, I'd never written a Windows program before. Thus began the long and ongoing challenge of learning about Windows programming in more and more detail. My son, David, gave me a copy of Visual Basic, and the Telegrapher's Toolkit project was underway.

One module at a time, I built up the following set of capabilities:

- Choice of simulated sounder or CW tone
- Simulated key, using the keyboard or mouse buttons
- Serial port I/O, for connecting an external key or sounder
- Simulated bug, using a paddle connected to the serial port
- Recorder for capturing and playing back sent or received code
- Histogram for analyzing the timing of sent or received code
- Code reader and sender for American or International Morse
- Ability to send Morse over a phone line or over the internet<sup>2</sup>

---

<sup>1</sup> KOB stands for "key on board", which is a key and sounder mounted on a wooden base.

<sup>2</sup> Due to a lack of awareness on my part, the mechanisms I came up with for sending Morse over a phone line or over the internet were not compatible with the dialup hub or CWCom.

The Telegrapher's Toolkit was designed for flexibility, rather than ease of use. As such, it served me very well as a testbed for experimenting with new concepts, although it would have been a bit complicated for anyone else to use.

## Dial-up Morse

In June 2003, I joined the *slowspeedwire* and *brasspounders* internet groups and I became aware of the Morse activity on the dialup hub. I was anxious to get on the hub so I could hear what real American Morse sounded like, but to do that I needed a 300 baud modem. My first thought was, "My computer is just sitting here with a modem already built in. Why don't I just use *it* to connect to the hub?"

I quickly discovered it's not so simple. Even though many modems today can still connect at 300 baud, they're designed to transmit ASCII characters, not arbitrary marks and spaces. I wrote to modem manufacturers for advice, but their responses were not encouraging.

I scoured the internet for any clues on how I could trick a normal modem into doing dialup Morse. After five months of frustrating false starts and dead ends, I hit upon the idea of using my modem's voice commands. I devised an algorithm for synthesizing and demodulating the necessary tones to simulate a 300 baud modem. At last I was able to take part in the "slow speed wire" sessions on the hub and use American Morse to communicate with other operators.

## MorseKOB

I had a great time using my software, checking into the hub as often as two or three times a week. I also used the program to build up my code speed by downloading newspaper articles from the internet and reading them in Morse.

Despite having some reservations about supporting my software on other computers, I decided I was having way too much fun with my program to keep it all to myself. I repackaged the software with two goals in mind:<sup>3</sup>

1. Keep the program as simple and easy-to-use as possible. In fact, I wanted the program to reflect the simplicity of the electric telegraph itself.
2. Handle the timing of Morse code so accurately that it would be indistinguishable from an actual telegraph wire.

Recognizing there were already plenty of good CW programs, I dropped support for International Morse and focused entirely on American Morse. I also eliminated many of the bells and whistles from the Telegrapher's Toolkit to reduce the program's complexity from the user's point of view. I designed the user interface for dialup Morse to match the controls and indicators on the Radio Shack DCM-6 modem.

I gave the name of the program a fair amount of thought. I chose "KOB" because (a) the program could simulate a key and sounder, (b) it was particularly suitable as a training tool, and (c) the name KOB had a nice "low tech" feel to it. I added "Morse" to the name

---

<sup>3</sup> These remain the two main goals for the KOB program. They haven't been fully achieved yet, but I'm still working at it.

(a) to strengthen its association with American Morse, and (b) to make the program easier to find by internet search engines.

MorseKOB made its first public appearance in February 2005, and was successfully used by Dave Freeman on the dialup hub the following month.

## **Internet Morse**

The KOB program didn't really come into its own as a generally useful tool until the addition of Internet Morse. I had been aware of CWCom since mid-2002, but from the beginning I ran into problems with inaccuracies in its timing of Morse code.

In March 2004, I created a software testbed to analyze CWCom's communications protocol, in an effort to isolate the cause of the problem. I wrote some prototype software that mimicked the CWCom program, but without its timing flaws. I also devised a scheme where the CWCom protocol could be extended to support closed-circuit telegraphy with realistic "break" behavior.

This effort was not much more than an academic exercise until Keith LeBaron put me in touch with Ted Wagner. Ted had a copy of the CWCom server software running on his computer, and he agreed to let us use his server for testing a new version of the KOB program that would support Internet Morse. With Sid Vaughan spearheading the concept of Internet Morse, and additional help from Maurie Challinor in New Zealand, MorseKOB finally became the program that we know today. After an intense period of testing and debugging, it was released for general use on July 27, 2005.

## **Acknowledgements**

The MorseKOB program literally would not have been possible without the help of many supporters and collaborators, including:

- **James Sackey**, for rekindling my interest in American Morse
- **Jim Farrior**, for his outstanding program, The Mill
- **David Kerr**, for getting me started with Visual Basic
- **Don Reser** and **Mike Taylor**, for their help in testing early versions of the Telegrapher's Toolkit
- **Derek Cohn** and **John "Ace" Holman**, for their help and encouragement in getting on the dialup wire
- **Jim Wades**, for ensuring the continued availability of the dialup hub
- **Dave Freeman**, for beta testing the dialup Morse module
- **John Samin** and **Andrew Spurrier**, for developing CWCom
- **Sid Vaughan**, for pioneering the use of American Morse on the internet
- **Ted Wagner** and **Tom Curtola**, for hosting the KOB server
- **Keith LeBaron** and **Maurie Challinor**, for their help in testing early versions of the Internet Morse module
- **John Beasley**, **Adrian Taylor**, and **Gil Verret**, for beta testing new versions of the program
- **Pat O'Brien**, for promoting the KOB program and recruiting new users