What is Winkeyer ?

WinKeyer is a low-cost full featured external Morse keyer capable of being fully controlled via a USB interface. It enables software developers to create a fully integrated Morse keyer within their programs, which will operate with Windows XP, Win7, Win8, and now Win10. It ensures accurately timed CW letters and strings no matter what the host PC is doing.

A Bit of History

Few, if any, serious contesters or DXers would contemplate operating without using a computer-based logging program these days. Contest logging was one of the first serious Amateur Radio applications to appear on the PC and within a few years there were many excellent logging programs on the market. Virtually all of these programs had two things in common:

1. They were DOS-based
2. They offered some form of CW keying

The DOS environment lent itself well to the task of internal CW generation. Firstly, it was possible to capture the internal system clock and harness it to provide the timing for the code generator. Second, DOS inherently supported an interrupt system, which provided an easy way to implement CW keying as a background task. Finally, unfettered access to the parallel or serial port meant that the key output could be easily interfaced to the rig.

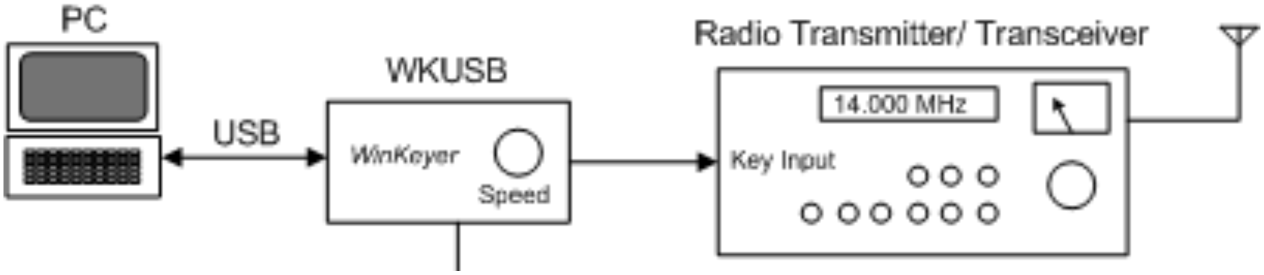
All this changed with the advent of Windows. Indeed it is fair to state that the huge step forward that Windows represented to the general PC user was at least equalled for Radio Amateurs by the step backwards in CW keying capability. Windows is a multi-tasking operating system which means that the CPU is shared between many different tasks. This makes it very difficult to accurately time CW due to constant task switching. This results in unevenly timed dits and dahs. For example, in the middle of a word you might have an R with a really long dah in the middle or an A that sounds more like an M.

WinKeyer is specifically designed to completely overcome this problem.

By off loading CW generation to a separate dedicated microcontroller, all the timing problems disappear. Applications running on the PC send ASCII letters to Winkeyer for conversion to Morse, allowing them to focus on more important things. Winkeyer provides several other crucial features. An external speed control allows the operator to tweak speed based on operating conditions. Having the ability to run fills with a paddle is essential and Winkeyer provides that also. Finally, to add a level of safety, Winkeyer isolates the PC from the the radio with optocouplers and provides two sets of KEY/PTT outputs for two radios. Winkeyer provides full control of CW timing and keying options all programmable from the PC. To add to Winkeyer's utility, a standalone mode is included which means it is a full featured CW keyer able to be operated without a PC connection.

So to summarize, Winkeyer gives the radio operator an Ideal Morse Keyer:

1. Which can be fully integrated with a PC based logging program via a standard USB com port.
2. Supports standard Iambic paddles
3. Provides a physical speed control knob
4. Capable of supporting functions such as callsign type-ahead, paddle interrupt, keyboard CW etc. matching the best practices of internal keyers in existing logging software
5. Simple character by character serial interface, capable of implementation on all platforms and operating systems
6. Fully featured specification, that matches or exceeds the best of the current standalone keyers
7. Capable of being operated standalone or in conjunction with a host
8. Low cost, simple to build kit
9. Compact, unit capable of being powered directly the USB port
10. Open architecture, with schematic (circuit) diagram and full interface specifications published, to enable anyone to integrate the keyer into their software project. A sample 32-bit Windows application with source is also available.
11. Rugged and reliable, including ESD protection and RFI filtering.



WinKeyer host requirements
WinKeyer is genuinely host and operating system independent and it is easy to implement in all the following environments:

- DOS – all versions
- Windows – all versions
- Linux – all versions
- Apple Mac – all versions

WinKeyer requires a single USB com port. Options are provided for dual radio support, and PTT.

The WinKeyer software interface
WinKeyer receives single byte commands, generally supported by one or occasionally two parameter bytes. WinKeyer responds to these commands with single byte responses. Under certain circumstances, WinKeyer will send unsolicited status information, so the host program must be able to accept inbound data at any time. WinKeyer implements a proprietary flow control mechanism for the sole purpose of preventing send buffer overflow when a large CW message is sent from host to WinKeyer. This flow control mechanism is easier to implement than XON/XOFF and simplifies the command structure. WinKeyer sends the following information to the Host:

- Speed pot setting. This allows the host to know what speed CW WinKeyer is sending. The speed byte is sent whenever the speed pot is turned.
- Character echo. This permits the implementation of callsign type-ahead and other similar features
- WinKeyer status. This includes busy/idle, flow control, paddle break-in, etc.

WinKeyer receives the following commands from the host (NB this list is not exhaustive):

- ASCII characters to send, including prosigns
- Set CW speed. The host can override the speed pot
- Get CW speed
- Set CW pot speed range
- Set weight
- Set mode (includes paddle swapping, Iambic A/B, bug mode, etc.)
- Set key compensation
- Set dit/dah trigger point (including the ability to disable stores)
- Get WinKeyer status
- Force key down
- Set PTT characteristics (lead-in and tail delays, 1st code bit extension)

It is possible to put together a simple WK interface in a couple of hours. Developers have interfaced to WK from MS Windows, Linux and the MAC using Visual Basic, C, C++, and Java. K1EL offers a free WK DLL and test platform written in C that gives a good foundation to work from. Note that the interface is published under a Creative Commons License that allows any host to interface to Winkeyer but limits design of a Winkeyer like device to personal or not for profit uses.