

Andrew Whaley's Blog

Security and other geeky stuff.

Morse Code (CW) via USB

Posted by Andrew on 04/28/2017

Tags: CW , MORSE



Since I started studying for my Amateur Radio exam, I wanted to find ways to practice Morse Code. There are a couple of really good websites :-

[Morsecode.me](#) – interesting and lively online chatroom where all communication is in morse code

[MorseBlitz](#) – fun game for improving your speed by keying letters fast enough to defuse bombs.

Both of these sites use the mouse button or a keyboard key to key the Morse and so is accessible to all. But what if you want to use a real telegraph key as shown above ?

There are a couple of methods :-

1. **Modify a mouse to take key input** – this is pretty easy and very effective. Just find an old or cheap USB mouse and solder the key input across the left mouse button switch. Every time you key down, it's like the mouse button is pressed. This is easy but means carrying a bulky mouse and keying can only be used as mouse input.
2. **Build a dedicated circuit to emulate a mouse or keyboard** – this is what we're going

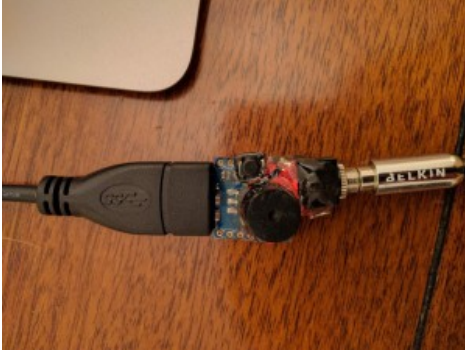
RANDOM POSTS

[Morse Code \(CW\) via USB](#)

[RBS / Natwest Touch ID Security](#)

to discuss in the rest of this article and enables keying to be used either as mouse input or as a full keyboard via the built in decoder. This enables CW input to be used on any app that can accept text input. It will work on any computer including Android phones.

Here's a close-up of the adapter :-



It's USB-A at one end and a 3.5mm jack socket at the other. The USB plugs into the computer (here via a USB-C adapter for the MacBook). The 3.5mm mono jack is the connector for the key.

The board has 4 modes :-

1. Mouse mode (silent) – emulating a mouse left button press,
2. Mouse mode with buzzer – to produce a sidetone whilst the key is down,
3. Keyboard mode (silent) – decodes your keying into a character and sends the key code like a keyboard would,
4. Keyboard mode with buzzer.

The push switch cycles through each mode.

The components required are :-

1. Beetle USB Atmega32u4 dev board ([like this one](#)),
2. Subminiature Piezo buzzer 5v ([like this one](#)),
3. Surface mount momentary push switch,
4. 3.5mm mono jack socket.
5. Epoxy resin to set it (clear epoxy glue works well)

Entire cost should be less than £10.

The Beetle USB board is a miniaturised Arduino Leonardo which supports USB HID peripheral mode, i.e. can act as a keyboard or a mouse. These boards are quite popular in the security world because they can be programmed to perform all manner of keyboard injection attacks on various computers and so are widely and cheaply available.

To assemble :-

1. Connect the +ve buzzer lead to D1 on the board,
2. Connect the other buzzer lead to GND,
3. Connect the button to D14 on one side and GND on the other,
4. Connect the 3.5mm socket tip to GND,

5. Connect the socket collar to D0,
6. [Optional] if the socket has an activation connector, connect this D2
7. Upload the sketch from the Arduino IDE
8. If it all works, set the components in place with the glue and leave 24 hours to dry.
Be careful not to allow the glue to get inside the buzzer or the 3.5mm socket or to cover the button.

If your particular board doesn't have a D14 don't worry just use one of the other digital pins and update the BUTTON pin in the sketch.

You'll need some proficiency with a small soldering iron as there's not much room and the solder tabs are small. You'll also need some short lengths of insulated wire to make some of the connections.

Here's the [sketch code](#).

You can have some great fun playing old text adventure games like [Zork](#) using the keyboard mode as text input – remember to use the CR pro-sign .-. for enter.

Enjoy !

Post a Comment

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment