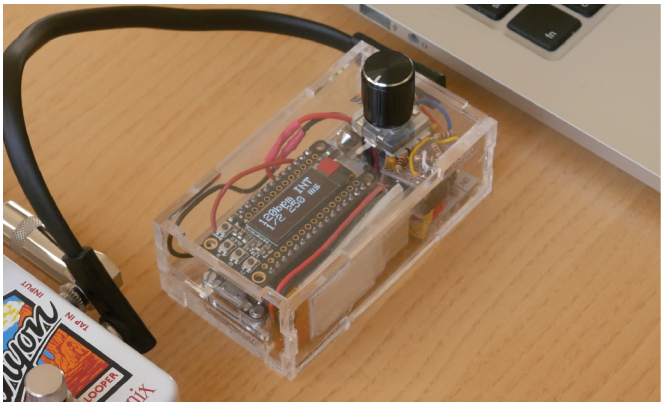
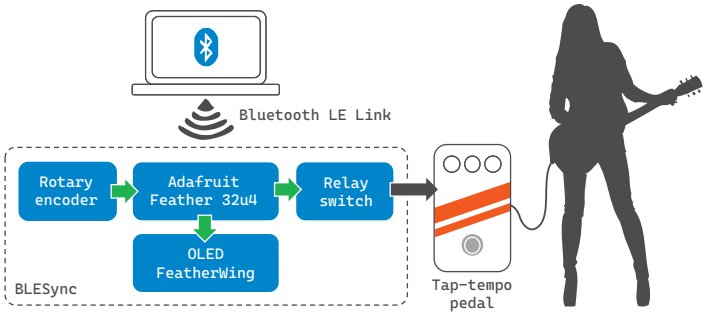
BLESync: Wireless Synchronization Between Computers and Tap Tempo Effect Pedals

JP Carrascal
Microsoft / Spacebarman
jp@spacebarman.com





In a performance incorporating computers and guitars, delay time and other time-based parameters are ideally synchronized with song tempo as determined by software. For a lot of effect pedals, there is no way to make this synchronization happen automatically, as tempo needs to be "tapped" by stepping on an external footswitch. BLESync is a MIDI Bluetooth LE (BLE) device designed to solve this problem. It simulates tempo tapping by means of a relay switch, whose tapping speed is defined by a precise tempo value as received wirelessly via BLE MIDI from a computer. This allows better physical on-stage independence between the computer and the guitar performer. Additionally, tempo can also be precisely entered using a rotary encoder. Handy beat divisions are available, including 1/1 (quarter note), 1/2 (eight note) and 3/4 (dotted-eight, also known as "The Edge delay").

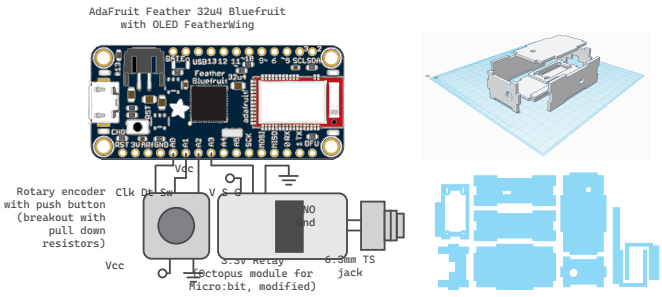
Hardware

Electronics

The device's brain is an Adafruit Feather 32u4 microcontroller with an embedded Bluetooth LE radio. It runs an Arduino sketch that manages the OLED FeatherWing display, Bluetooth connection, and generates the signal to drive the relay switch.

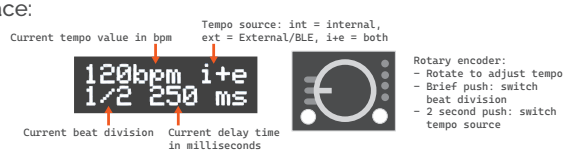
Enclosure

The box was designed using TinkerCAD and exported to SVG for laser cutting. It was cut from 3-mm thick methacrylate.



Interface

An OLED display and a rotary encoder provides a simple interface:



Software

Tempo encoding

Tempo should be encoded at the computer end as a MSB and LSB combination using CC 16 and 48 (general purpose CC, as per MIDI specs) to allow a wide range of tempos:

```
tempoMSB = bpm / 10
tempoLSB = bpm % 10
```

Where `tempoMSB` and `tempoLSB` are integers, and '%' is the modulo (division reminder) operator. Decoding is done by the Arduino sketch running on BLESync, as per this expression:

```
bpm = tempoMSB * 10 + tempoLSB
```

Demo

In this demo, I use Ableton Live running on MacOS as a master tempo source to control BLESync, but any DAW capable of generating CCs and running on a MIDI-BLE-capable OS should work. I use the MacOS Audio MIDI Setup utility to make the BLESync MIDI port active and available for MIDI software. A Max4Live patch obtains the current song tempo by means of the `[plugsync~]` object and applies the encoding described earlier. The patch is inserted in a MIDI channel, whose output is sent to the BLESync MIDI port.

Open-source

All the software used in this project, including the Arduino sketch, the Max4Live patch, and the laser-cutting box template is available for download from GitHub (link below).

Download software from:
github.com/jpcarrascal/BLESync

www.spacebarman.com
www.jpcarrascal.com