# Goals

Make a respin of the STM32 Tesla Coil interrupter. The new interrupter will have the following features and/or improvements from the previous generation:

* Battery powered
* USB-C power and communication
  + If USB is connected, use USB. Assume battery will be charged.
* 20x4 LCD screen
* Speaker to test songs out loud
* Fixed SD card orientation
* Up to 3 Fiber optic outputs
  + Some way to route outputs
* Higher STM32 core speed
* Better structured code
  + Implement more advanced MIDI features like pitch bend
  + New USB application to load/store files on the SD card
* Live MIDI mode
* Fits in the same interrupter case is a plus but not a must

Required interfaces:

* SDMMC: 4 bit wide interface, FATFS
* USB device: virtual com port
* I2C1 🡪 LCD
* USART1: MIDI
* USART2: Coil A rx
* USART6: Coil B rx
* ADC1: 4 channel input. 1 potentiometer for each coil
* Components

V1 board is 91 x 64mm

* STM32F401RE
  + Power consumption: 21.8 mA @ 84 MHz, 15.8mA @ 60MHz
  + STM32 compiler path: C:\ST\STM32CubeIDE\_1.13.2\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.externaltools.cubeprogrammer.win32\_2.1.0.202305091550\tools\bin
  + 16 MHz external clock
* Red (1206) LEDs: V=BAT, CHARGE
* Green (0603) LEDs: 3v3GOOD, 5VGOOD
* Blue (0805) LEDs: Heartbeat and Fiber optics

# Directories

Eagle project is in C:\Users\bucht\OneDrive\Documents\EAGLE\projects\interrupter\_v2

Main repo is in repo which is in OneDrive under Projects/Interrupter.

# TODO

* Make I2C LCD faster by removing HAL\_Delay()s

# Assembly revision / Board changes / Case uses

* Add jumper between buck output and rails
* Add DC barrel jack connector
* Only 1 ADC will be needed to control volume. Use the other two ADCs to GPIO
* Reused GPIOs:
  + ADCIN12
  + ADCIN13
  + All OLED pins (6 GPIO)
* Fixed silk screen on SPI header

# Changing code structure to be OO

* NoteManager singleton which will handle fiber optic transmitters. Exposes the API:
  + int note\_on(note number, velocity, requested\_coil)
  + int note\_off(note number, requested coil)
* SD singleton which manages SD card transactions