

Specifications and features:仕様と機能

- ・寸法 : 17.8x33.2x4mm (+ 3.2mm、マイクロ USB ポートおよびマイクロ SD カード付き)
- ・ 100%オープンソース、好きな機能を追加できます (GPLv3)
- ・電源 : 2.6~4.5 ボルト、LED 出力 1~6 あたり最大 10A。単一のリチウムイオン 3.6-3.7V (低 2.6V、フル 4.2V) バッテリーを推奨
- ・スピーカー : 4 オームまたは 8 オーム、2-5W (推奨)
- ・無制限の量のサウンドバンク/フォント、通常の (Plecter、NEC) および「Smoothswing」サウンドフォントをサポート
- ・ Sound FX (WAV サウンドファイル) : ブート、ブラスター偏向、ロックアップ、ハム、スイング、クラッシュ、ドラッグ、フォント、フォーカス、イグニッション、リトラクションなど
- ・ Light FX : ブレードのちらつき、脈動、衝突時のフラッシュ、ドラッグ、突き刺し、ブラスターの偏向、ロックアップなど
- ・アイドルモードでの音楽トラック (WAV サウンドファイル) の再生とセイバーサウンドエフェクトのバックグラウンド
- ・マイクロ SD カード : SanDisk ブランドの 4-32Gb クラス 4-10 を推奨
- ・Bluetooth を介したリモートコントロールのサポート (外部 Bluetooth モジュールアドオンを使用)
- ・サウンドフィルター、合成、mp3 再生などの高度な機能を備えた高速 32 ビットプロセッサ
- ・3 ワット 5V サウンドアンプ、16 ビットデジタル出力
- ・ネオピクセル LED / ストリップ制御用の 4 つのデータ信号出力
- ・サンプルレートは 44kHz (デフォルト) で、22kHz と 11kHz がサポートされ、自動的に 44kHz にアップサンプリングされます。
- ・ギャップレス再生。1 つのサンプルを中断して別のサンプルに移動すると、2.5ms のクロスフェードが発生します。
- ・ポリフォニック再生、現在最大 5 つの同時サンプル用に構成されています
- ・スムーズスイング モーションからサウンドへのアルゴリズムのサポート
- ・ PL9823 (RGB) 、WS2812B (RGB) 、SK6812 (RGB、WWA、RGBW) ネオピクセルのサポート
- ・ 1/2/3/4 色 LED スター (Tri-Cree および Quad (RGBW) LED モジュール)
- ・セグメント化された (6 セグメント+フラッシュストリング) クラシックストリングブレードのサポート
- ・デュアルおよびクロスガードセットアップのマルチブレードサポート
- ・抵抗のさまざまな値によるブレード LED タイプ、プリセット、およびブレードスタイルの選択 (ブレード ID 機能)
- ・クリスタルチャンバーサポート
- ・ネオピクセルブレードを備えた電力レベルインジケータ
- ・ OLED PLI および FONT、アニメーション表示
- ・ブレード検出ピン
- ・リアルタイムの色変更、IR 受信サポート (ProffieOS 3.9 ファームウェアアップデート以降)
- ・ Bluetooth モジュールや OLED ディスプレイなどの衛星デバイスに電力を供給するための 3.3V (約 250mA が利用可能) パッド
- ・サウンドファイルを PC から直接 USB ケーブル経由で SD カードにアップロード (大容量記憶装置のサポート)
- ・ POV (視覚の持続性) モードのサポート
- ・アクセント LED のサポート (追加の「ブレード」としても実装)
- ・音声エラーとバッテリー低下メッセージ
- ・ユーザーによる簡単で無料のファームウェアアップデート

Software installation and setup

① 最新の [ArduinoIDE ソフトウェア](#) をインストールします (ベータ版は使用しないでください)。

Windows アプリとしてインストールすることもお勧めしません。

特定の保護されたフォルダにインストールされるためです。

追加のソフトウェア/プラグインをインストールすることはできません。

ProffieBoard が COM ポートに表示されない場合は、以前の ArduinoIDE バージョン。→

Download the Arduino IDE

ARDUINO 1.8.11
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Release Notes
Source Code
Checksums (sha512)

Previous Releases

Download the [previous version of the current release](#) the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

② [ProffieboardArduino プラグインと Zadig プログラム](#) をインストールします。このページのインストール手順に従ってください。

インストール

① Arduino IDE (少なくともバージョン v1.6.8) をダウンロードしてインストールします

② ArduinoIDE を起動します

③ 環境設定に移動

④

https://profezzorn.github.io/arduino-proffieboard/package_proffieboard_index.json 「追加のボードマネージャー URL」として追加

[ツール] → [ボード] メニューからボードマネージャーを開き、[Proffieboard プラグイン] をインストールします

Proffieboard Tools → Board メニューを選択します

File Name	Description	Last Modified
variants	get rid of IAP, save 1KB	a day ago
README.md	stm32f4 -> proffieboard	4 months ago
boards.txt	enable -ffast-math	4 days ago
platform.txt	fix urls and stuff	4 months ago
programmers.txt	Add NUCLEO-L476RG support	2 years ago

Arduino Plugin for Proffieboard

Installing

1. Download and install the [Arduino IDE](#) (at least version v1.6.8)
2. Start the Arduino IDE
3. Go into Preferences
4. Add https://profezzorn.github.io/arduino-proffieboard/package_proffieboard_index.json as an "Additional Board Manager URL"
5. Open the Boards Manager from the Tools -> Board menu and install "Proffieboard Plugin"
6. Select Proffieboard Tools -> Board menu

OS Specific Setup

```
ProffieOS | Arduino 1.8.10
File Edit Sketch Tools Help
ProffieOS
1 /*
2 Proffie
3 #EEPROM
4 Copyright
5 Additio
6
7 This pr
8 ic hard
9 the Fre
10 ter yo
11
12 This pr
13 pur WE
14 MERCHAN
15 GNU Ge
16
17 You sho
18 along w
19 */
20
21 // You can have multiple configuration files, and specify which one
22 // to use here.
23
24 #define CONFIG_FILE "config/Graflex_config.h"
25 // #define CONFIG_FILE "config/my_testproffie_v2_config.h"
26 // #define CONFIG_FILE "config/crossguard_config.h"
27 // #define CONFIG_FILE "config/graflex_v1_config.h"
28
29 #ifdef CONFIG_FILE_TEST
30 #undef CONFIG_FILE
31 #define CONFIG_FILE CONFIG_FILE_TEST
32 #endif
33
```

③ [ツール] → [ボード] で [ProffieboardV2] を選択します。

USB タイプ - シリアル + 大容量記憶装置 (または + WebUSB)

DOSFS - SDCARD (SPI)

CPU 速度 - 80 MHz

最適化 - 最小コード (または高速/高速/最速)

ポート - COM (PC が割り当てた番号) (ProffieBoard)

ProffieBoard を USB ケーブルで PC に接続して、ポートを選択できるようにします。

Uploading firmware

① ProffieBoard ファームウェアと SD カードのコンテンツをダウンロードします。

ProffieOS-v2.2.zip を Documents ディレクトリまたはに解凍します。

デスクトップですが、Arduino プログラムフォルダやその他の場所にはありません。

プログラムディレクトリ！ ProffieOS-v2.2 フォルダと その中の ProffieOS フォルダー：... ¥ ProffieOS-v2.2 ¥ ProffieOS。

これらのフォルダやファイル内の名前を変更したり、移動したりしないでください。

ProffieOS フォルダ外の他の場所に！ 解凍する。

ProffieOS_SD_Card.zip を保存するディレクトリに移動します。ProffieOS-v2.2 フォルダー。ProffieOS_SD_Card からすべてのファイルを SD カードへコピーします。

② ファイルエクスプローラーの設定でファイル拡張子を再表示して、...h の末尾、拡張子を確認します。

設定ファイル名に「.h」を追加しないでください。config フォルダーに移動し、独自の config.h ファイルを作成します。（やり方については次のページを参照してください）ProffieOS.ino ファイルをダブルクリックします。

③ 図のように config.h ファイルの名前を追加し、これを ProffieOS.ino ファイルに保存します。他の設定ファイルがコメントアウトされました。CONFIG_FILE は 1 つだけである必要があります。

「//」はナシ。複数の設定ファイルを持つことができます。

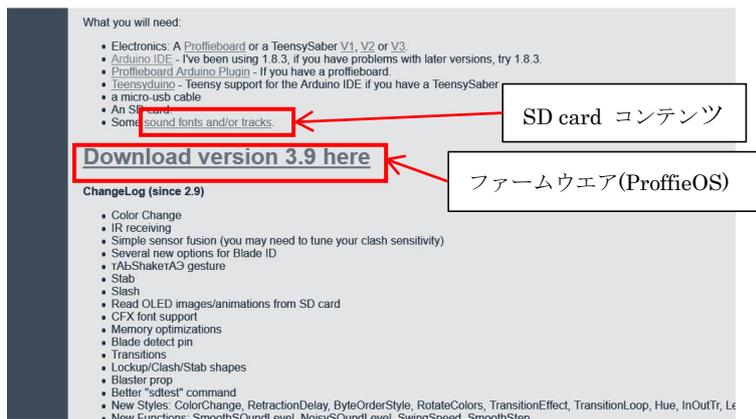
ProffieOS ¥ config フォルダーで、必要なものを定義するだけです。

ProffieOS.ino ファイルに保存し、Proffieboard に再度アップロードします。

④ データ転送マイクロ USB ケーブルを使用して Proffieboard を PC に接続します。矢印ボタンを押すとコンパイルされてボードへのファームウェアアップロードされます。赤いテキストのプログレスバーが 100% で停止するのを待ちます。

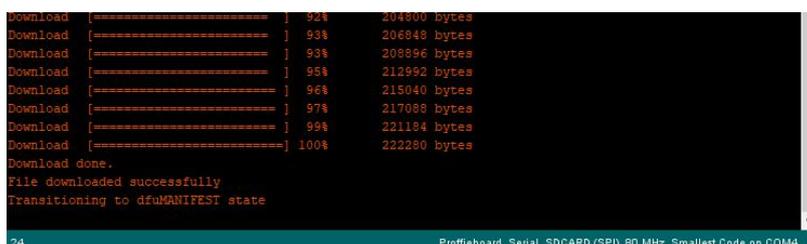
スピーカーが接続されている場合、ProffieBoard は起動音を再生します。「シリアル+大容量記憶装置」を使用している場合は、SD カードを正しく取り出してください。これで、USB ケーブルを抜くことができます。これで完了！

エラーが発生する場合、これは config.h ファイルに問題、#define CONFIG_FILE 名に誤りがある、config.h ファイル config フォルダーの外にあり、PC ユーザー名がラテン語ではありません...



名前	更新日時	種類	サイズ
.github	2021/02/13 19:37	ファイル フォルダー	
blades	2021/02/13 19:37	ファイル フォルダー	
buttons	2021/02/13 19:37	ファイル フォルダー	
common	2021/02/13 19:37	ファイル フォルダー	
config	2021/02/15 1:44	ファイル フォルダー	
display	2021/02/13 19:37	ファイル フォルダー	
fontconvert	2021/02/13 19:37	ファイル フォルダー	
functions	2021/02/13 19:37	ファイル フォルダー	
ir	2021/02/13 19:37	ファイル フォルダー	
motion	2021/02/13 19:37	ファイル フォルダー	
mtp	2021/02/13 19:37	ファイル フォルダー	
pov_tools	2021/02/13 19:37	ファイル フォルダー	
props	2021/02/13 19:37	ファイル フォルダー	
scripts	2021/02/13 19:37	ファイル フォルダー	
sound	2021/02/13 19:37	ファイル フォルダー	
styles	2021/02/13 19:37	ファイル フォルダー	
transitions	2021/02/13 19:37	ファイル フォルダー	
videotobc	2021/02/13 19:37	ファイル フォルダー	
cvsignore	2021/02/13 19:37	CVSIGNORE ファイル	1 KB
.gitattributes	2021/02/13 19:37	GITATTRIBUTES ファ...	1 KB
Arduino.mk	2021/02/13 19:37	MK ファイル	68 KB
Common.mk	2021/02/13 19:37	MK ファイル	4 KB
LICENCE.txt	2021/02/13 19:37	テキストドキュメント	35 KB
Makefile	2021/02/13 19:37	ファイル	5 KB
Proffieboard.mk	2021/02/13 19:37	MK ファイル	16 KB
ProffieOS.ino	2021/02/13 19:37	Arduino file	49 KB
README.md	2021/02/13 19:37	MD ファイル	1 KB
Teensy.mk	2021/02/13 19:37	MK ファイル	5 KB

```
lightsaber | A
File Edit Sketch Tools Help
lightsaber
15 //
16 //
17 // You can have multiple configuration files, and specify which one
18 // to use here.
19 //
20 #define CONFIG_FILE "config/my_saber_config.h"
21 // #define CONFIG_FILE "config/default_proffieboard_config.h"
22 // #define CONFIG_FILE "config/default_v3_config.h"
23 // #define CONFIG_FILE "config/crossguard_config.h"
24 // #define CONFIG_FILE "config/grafx_v1_config.h"
25 // #define CONFIG_FILE "config/prop_shield_fastled_v1_config.h"
26 // #define CONFIG_FILE "config/owk_v2_config.h"
27 // #define CONFIG_FILE "config/test_bench_config.h"
28 // #define CONFIG_FILE "config/toy_saber_config.h"
29 // #define CONFIG_FILE "config/proffieboard_v1_test_bench_config.h"
30
31 #ifdef CONFIG_FILE_TEST
32 #undef CONFIG_FILE
33 #define CONFIG_FILE CONFIG_FILE_TEST
34 #endif
35
36 #define CONFIG_TOP
37 #include CONFIG_FILE
38 #undef CONFIG_TOP
39
40
41
42
```



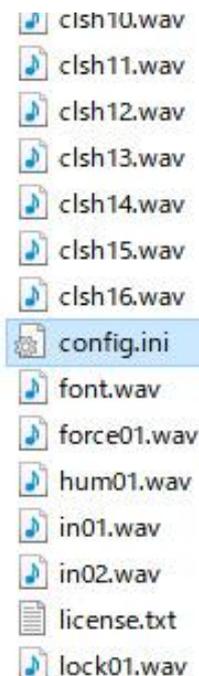
config.h file structure, editing

すべてのサウンドファイル（サウンドフォント、音楽トラック）はマイクロSDカードに保存されます。必要なサウンドフォントフォルダを追加します。（Plecter、NEC、Smoothswing フォントがサポートされています。デフォルトで行われているように、WAV ファイル名を変更する必要はありません。コピーして貼り付けるだけです）。

ProffieOS_SD_Card コンテンツフォルダと音楽トラックを tracks フォルダに追加します。

すべての音楽トラックとサウンドフォントフォルダには、ラテン文字を使用し、特別な文字（?、。| ¥）{[/-など）を使用しないで名前を付けてください。config.ini ファイルと smoothsw.ini ファイルがない場合は、各サウンドフォントフォルダにあることを確認してください。デフォルトの Proffieboard から1つコピーしてください。

サウンドフォントを作成し、新しく追加したサウンドフォントフォルダに貼り付けます。



すべてのブレード効果、LED 構成、音量レベル、衝突感度などは、ProffieOS ¥ config フォルダーにある config.h ファイルで変更されます。

これを行うには、メモ帳などのテキストエディタで..._ config.h ファイルを開き、すべてのテキストを削除し、ここに配線図の構成コードを作成し、コピーして

..._ config.h ファイルに貼り付け、新しい名前で保存します。27-28 ページの手順に従って、ボードにアップロードします。詳細>>>[こちら](#)<<<。※次のページに記載

```
#ifndef CONFIG_TOP
#include "proffieboard_v2_config.h"
#define NUM_BLADES 1
#define NUM_BUTTONS 2
#define VOLUME 1000
const unsigned int maxLedsPerStrip = 144;
#define CLASH_THRESHOLD_G 1.0
#define ENABLE_AUDIO
#define ENABLE_MOTION
#define ENABLE_WS2811
#define ENABLE_SD
#define SAVE_STATE
#endif

#ifndef CONFIG_PRESETS
Preset presets[] = {
  {"TeensySF", "tracks/venus.wav",
   StyleNormalPtr<CYAN, WHITE, 300, 800>(), "cyan"},
  {"SmthJedi", "tracks/mars.wav",
   StylePtr<InOutSparkTip<EASYBLADE(BLUE, WHITE), 300, 800>>(), "blue"},
  {"SmthGrey", "tracks/mercury.wav",
   StyleFirePtr<RED, YELLOW>(), "fire"},
  {"SmthFuzz", "tracks/uranus.wav",
   StyleNormalPtr<RED, WHITE, 300, 800>(), "red"},
  {"RgueCmdr", "tracks/venus.wav",
   StyleFirePtr<BLUE, CYAN>(), "blue fire"},
  {"TthCrstl", "tracks/mars.wav",
   StylePtr<InOutHelper<EASYBLADE(OnSpark<GREEN>, WHITE), 300, 800>>(), "green"},
  {"TeensySF", "tracks/mercury.wav",
   StyleNormalPtr<WHITE, RED, 300, 800>(), "white"},
};

BladeConfig blades[] = {
  { 0, WS2811XBladePtr<144, bladePin, Color8::GRB, PowerPINS<bladePowerPin2, bladePowerPin3>>(), CONFIGARRAY(presets) },
};
#endif

#ifndef CONFIG_BUTTONS
Button PowerButton(BUTTON_POWER, powerButtonPin, "pow");
Button AuxButton(BUTTON_AUX, auxPin, "aux");
#endif
```

Proffieboard v2 config
number of "blades" used
number of buttons used (1-3)
volume level (0-3000)
Clash sensitivity (lower = more sensitive, higher = less sensitive, try with 0.5 step)
Makes the board save last selected Volume, Preset and blade Color
sound font folder name
track name
Blade style code
Preset 1
Preset name
blade configuration

The config file : 設定ファイル

NoSloppy は 2020 年 2 月 3 日にこのページを編集しました・9 つの改訂

ProffieOS のすべてを変更することは可能ですが、ほとんどの場合、変更する必要があるのは構成ファイルのみです。構成ファイルは、実際には残りのコードの動作を指定する C ++コードにすぎませんが、多くの ProffieOS ユーザーは C ++の専門家ではないため、このページでは ProffieOS 構成ファイルについて詳しく説明します。

まず、典型的な設定ファイルを見てみましょう。 ※以下をすべてコピーして config.h ファイルに貼り付けます。

```
#ifdef CONFIG_TOP
#include "proffieboard_v1_config.h"
#define NUM_BLADES 1
#define NUM_BUTTONS 2
#define VOLUME 1000
const unsigned int maxLedsPerStrip = 144;
#define CLASH_THRESHOLD_G 1.0
#define ENABLE_AUDIO
#define ENABLE_MOTION
#define ENABLE_WS2811
#define ENABLE_SD
#endif

#ifdef CONFIG_PRESETS
Preset presets[] = {
    { "TeensySF", "tracks/venus.wav",
      StyleNormalPtr<CYAN, WHITE, 300, 800>(), "cyan"},
    { "SmthJedi", "tracks/mars.wav",
      StylePtr<InOutSparkTip<EASYBLADE (BLUE, WHITE), 300, 800> >(), "blue"},
    { "TthCrstl", "tracks/mars.wav",
      StylePtr<InOutHelper<EASYBLADE (OnSpark<GREEN>, WHITE), 300, 800> >(), "green"},
    { "TthCrstl", "tracks/uranus.wav",
      StyleStrobePtr<WHITE, Rainbow, 15, 300, 800>(), "strobe"},
    { "TeensySF", "tracks/venus.wav",
      &style_pov, "POV"},
    { "SmthJedi", "tracks/mars.wav",
      &style_charging, "Battery¥nLevel"}
};

BladeConfig blades[] = {
    { 0, WS2811BladePtr<144, WS2811_ACTUALLY_800kHz | WS2811_GRB>(), CONFIGARRAY(presets) },
};
#endif

#ifdef CONFIG_BUTTONS
Button PowerButton(BUTTON_POWER, powerButtonPin, "pow");
Button AuxButton(BUTTON_AUX, auxPin, "aux");
#endif
```

ご覧のとおり、このファイルには、TOP、PRESETS、および BUTTONS の 3 つのセクションがあります。各セクションは次のようになります。

```
#ifdef CONFIG_NAME
Some stuff goes here
#endif
```

ここで、「NAME」はセクションの名前です。このファイルには3つのセクションがありますが、4つのセクションがあるものもあり、将来的にはさらに多くのセクションが可能になる可能性があります。各セクションには、技術的には有効なC++コードを含めることができますが、意図したとおりにセクションを使用することをお勧めします。各セクションは、独自のwikiページで説明されています。詳細については、以下のリンクをたどってください。

https://github.com/profezzorn/ProffieOS/wiki/The-CONFIG_TOP-section
https://github.com/profezzorn/ProffieOS/wiki/The-CONFIG_PRESETS-section
https://github.com/profezzorn/ProffieOS/wiki/The-CONFIG_BUTTONS-section
https://github.com/profezzorn/ProffieOS/wiki/The-CONFIG_PROP-section

Blade Styles : ブレードスタイル

Proffieboardは、メインのサーベルブレードにブレードスタイルを使用し、その他のアクセントLEDを使用して、すべての光の効果（色の変更、フラッシュ、ちらつき、遅延、点火/収縮のタイミングなど）。

[ブレードスタイルエディタ](#)を使用して、ブレードスタイルを作成および調整します。Megtooth Sith Sabresは、すばらしい[ビデオチュートリアル](#)を行い、彼が見せて説明しました。

ブレードスタイルエディタの使用法。また、既製のブレードスタイルを入手したり、TRAフォーラムやFett263ライブラリから共有したりすることもできます。

開始時にスパーク、クラッシュ、ブラスター、ロックアップとドラッグ、点火/収縮効果を備えた単純なちらつきグリーンブレードのブレードスタイルの例：

```
StylePtr<InOutHelper<SimpleClash<Lockup<Blast<OnSpark<AudioFlicker<Rgb<0, 255, 0>, Rgb<50, 100, 0>>, Rgb<255, 255, 0>, 150>, Rgb<255, 50, 0>>, AudioFlicker<Rgb<100, 255, 0>, Rgb<255, 0, 150>>>, Rgb<255, 100, 150>, 40>, 200, 300, Black>>
```

-これは、ブレードスタイルコードがconfig.hファイルプリセットに貼り付けられたように見える方法です（StylePtr <...>コンテナ内にあります）

```
InOutHelper<SimpleClash<Lockup<Blast<OnSpark<AudioFlicker<Rgb<0, 255, 0>, Rgb<50, 100, 0>>, Rgb<255, 255, 0>, 150>, Rgb<255, 50, 0>>, AudioFlicker<Rgb<100, 255, 0>, Rgb<255, 0, 150>>>, Rgb<255, 100, 150>, 40>, 200, 300, Black>
```

-これは、ブレードスタイルエディター内で編集したときのブレードスタイルコードの外観です。各ブレードスタイルはさまざまなエフェクトで構成されており、追加された各エフェクトは、前のエフェクトのベースカラーの代わりに使用されます。

```
InOutHelper<base color, 200, 300, Black>
```

-ベースカラーは、単語（WHITE、RED、GREEN、PURPLEなど）またはRgb <0-255, 0-255, 0-255>の値で定義できます。カスタムシェード; 200はミリ秒単位の拡張長です。300はミリ秒単位の収縮長です。黒は引込まれたときの色です（他の色でもかまいません）

```
SimpleClash<base color, clash color, 40> -衝突効果; 40はミリ秒単位の衝突時間です
```

```
Lockup<base color, lockup color> -ロックアップ効果
```

Blast<base color, blast color> - ブラスター効果

OnSpark<base color, spark color, 150> - 点火効果の火花; 150 はミリ秒単位のスパーク持続時間です。

AudioFlicker<" A" color, " B" color> - ちらつき効果 (ブレードが実際のセイバーハム音にちらつきます)。 「A」と「B」の色の違いが多いほど、突然がちらつく

Rgb<255, 50, 0> - RGB 形式の実際の色 (0 は光なし、255 は赤、緑、または青のチャンネルの最大輝度値)

Quick troubleshooting tips : 最も一般的な問題を解決する方法

- ①「フォントディレクトリが見つかりません。 SD カードが見つかりません」音声エラー...
 - SD カードのサウンドフォントフォルダ名が、config.h ファイルの各プリセットのフォント名と完全に一致するかどうかを確認します。 FAT32 で SD カードを再フォーマットします。そしてさらに試みる。
- ②Proffieboard は、常に「STM32 BOOTLOADER」としてのみコンピューターによって認識されます (Arduino IDE のポート選択では何も認識されません)。
 - Zadig ドライバーが正しくインストールされていても、Proffieboard が PC によって常に「Proffieboard」ではなく「STM32BOOTLOADER」として認識される場合BOOT ボタンを押しながら RESET ボタンを押した後でのみ- Arduino IDE を開き、最新の ProffieOS と Proffieboard を使用していることを確認してください。プラグインバージョン。[ツール] タブの[ポート]を選択せずに、[コードの確認] ボタンをクリックし、完了したら[アップロード] ボタンをクリックします。ファームウェアは Proffieboard で更新する必要があるため、次に USB ポートに接続したときに正しく認識されます。
- ③Arduino IDE でのスケッチ (コード) コンパイルエラー...
 - 開いた ProffieOS.ino ファイルの#define CONFIG_FILE " config /..._ config.h" 行が、config / で正しく記述されているかどうかを確認します。 かどうかを確認しますProffieOS.ino スケッチファイルで定義した..._ config.h ファイルは ProffieOS と同じ名前です-「ファームウェアバージョン」 / ProffieOS / config フォルダ。このフォルダにあります。
- ④Arduino IDE でのスケッチ (コード) コンパイルエラー:... Sketch ¥ config / proffiev2config.h : 42 : 1 : エラー : 「StyleFactory *」を「constchar *」に変換できません。初期化...
 - いずれかのプリセットにブレードスタイルがありません : NUM_BLADES が 2 以上に設定されていますが、プリセットのブレードスタイルは少なくなっています。
- ⑤Arduino IDE でのスケッチ (コード) コンパイルエラー:... Sketch ¥ config / proffiev2config.h : 42 : 1 : エラー : 「プリセット」の初期化子が多すぎます...
 - #define NUM_BLADES 値が各プリセットのブレードスタイルの数と一致するかどうかを確認します。一部のプリセットには、より多くのブレードスタイルがあります。必要以上に。
- ⑥音が出ない...
 - SD カードを取り外して再度挿入し、スピーカーの配線を確認します。 SD カード上のすべてのサウンドファイルに正しい名前が付けられていることを確認してください。 SD カードを再フォーマットする。FAT32、サウンドファイルをロードして再試行し、別の SD カードを試してください。

⑦ボードは「低電力」と言います...

- バッテリーを充電。

⑧シリアルモニターにはボードから送信された情報が表示されますが、コマンドが機能しません...

-シリアルモニターウィンドウの右下隅で、[行末]ドロップダウンが[改行]に設定されていることを確認します。

⑨音が変で歪んでいる...

- SD カードの速度を確認します (31 ページを参照)。スピーカーの配線を確認し、新しい高品質のスピーカーを試してください。バッテリーを 4.1~4.2V まで充電し、推奨バッテリーを使用していることを確認してください。

さらにヘルプが必要な場合は、次のリンクを確認してください。

[ProffieOS/ProffieBoard/TeensySaber wiki on GitHub](#)

[Ask your question on The Rebel Armory forums](#)

[Ask your question on FX-sabers forums](#)

[Ask your question in facebook group](#)