

# Brancher des périphériques USB sur un ESP8266

Saviez vous que vous pouvez brancher des périphériques USB sur un Arduino ou un ESP ? Que ce soit un clavier, une souris, un manette de jeu ou un contrôleur MIDI, c'est possible !

 Difficulté Moyen

 Durée 2 heure(s)

 Catégories Électronique, Musique & Sons, Jeux & Loisirs

 Coût 6 EUR (€)

## Sommaire

Introduction

Video d'introduction

Étape 1 - Choisir l'USB Host

Étape 2 - Régler le problème d'alimentation

Étape 3 - Brancher l'USB Host

Étape 4 - Régler le problème avec la bibliothèque USB Host

Étape 5 - Downgrader ESP8266

Étape 6 - Test l'USB Host

Étape 7 - Exemple : Manette XBOX360

Étape 8 - Exemple : Souris

Étape 9 - Exemple : Contrôleur Midi

Étape 10 - Conclusion

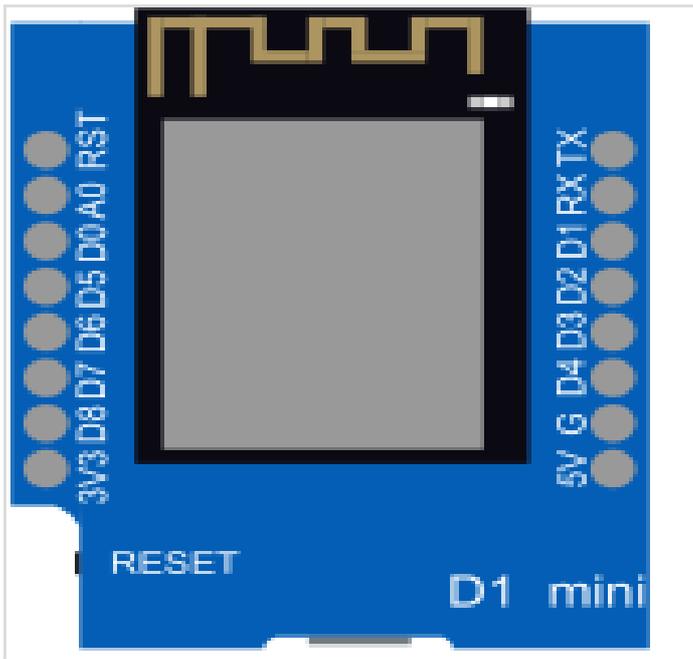
Étape 11 - Adaptation ESP32

Commentaires

## Introduction

Je jouais avec **Mozzi**, une **bibliothèque de synthèse audio** (<https://sensorium.github.io/Mozzi/>), quand je me suis rendu compte que si je voulais vraiment pouvoir exploiter tout le potentiel de cette bibliothèque, j'allais devoir faire un contrôleur.

Si seulement, je pouvais brancher un **clavier** ou un **contrôleur midi**, ça serait carrément plus simple !

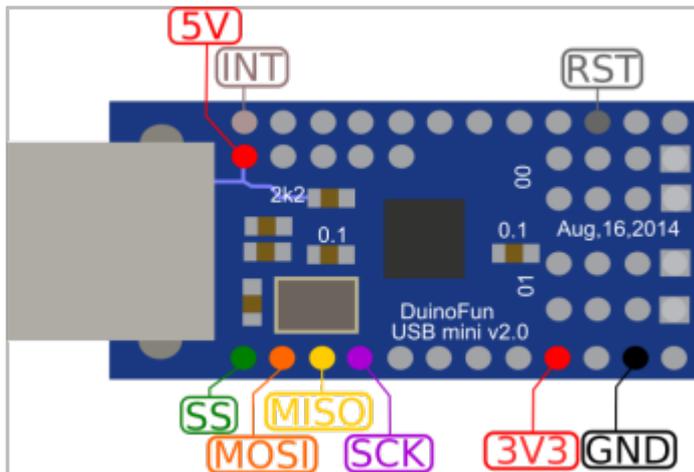


## Matériaux

- Un microcontrôleur 3.3v (par ex: Wemos Mini)
- Un USB Host Mini

## Outils

Un petit tournevis plat / Un cutter fin



## Étape 1 - Choisir l'USB Host

Merci à CircuitatHome pour la création de l'USB Host <https://www.circuitsathome.com/usb-host-shield-hardware-manual/>

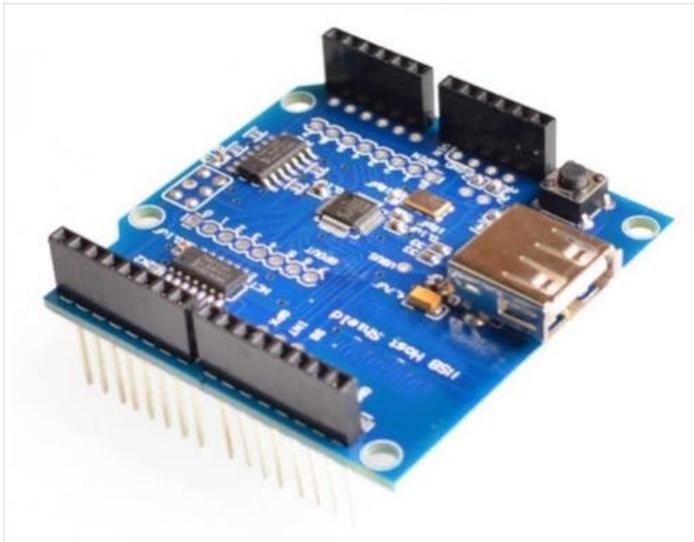
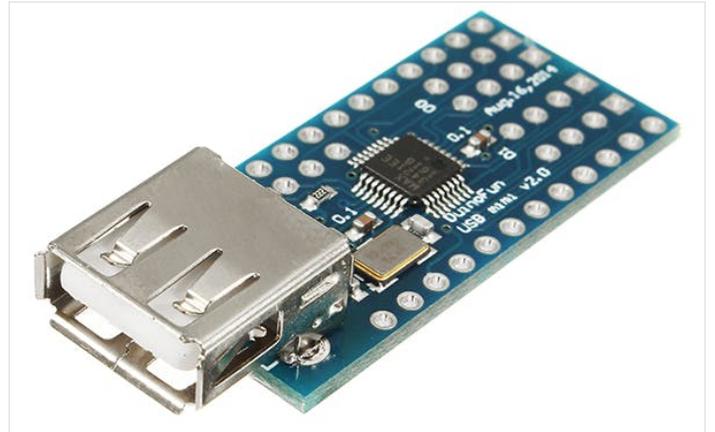
Dans ce tutoriel nous allons utiliser un clone de l'USB Host Mini.

La puce sur celle-ci est un MAX3421E qui ne fonctionne quand 3.3v



Si vous voulez utiliser un microcontrôleur en 5v (comme l'Arduino Uno / Nano) utiliser plutôt le Shield USB Host.

Malheureusement avant de pouvoir brancher notre USB Host, il va nous falloir régler plusieurs problèmes sinon cela ne marchera pas.



## Étape 2 - Régler le problème d'alimentation

Si vous avez acheté un clone de l'USB Host Mini, il n'y a pas de jumper pour le VBUS.

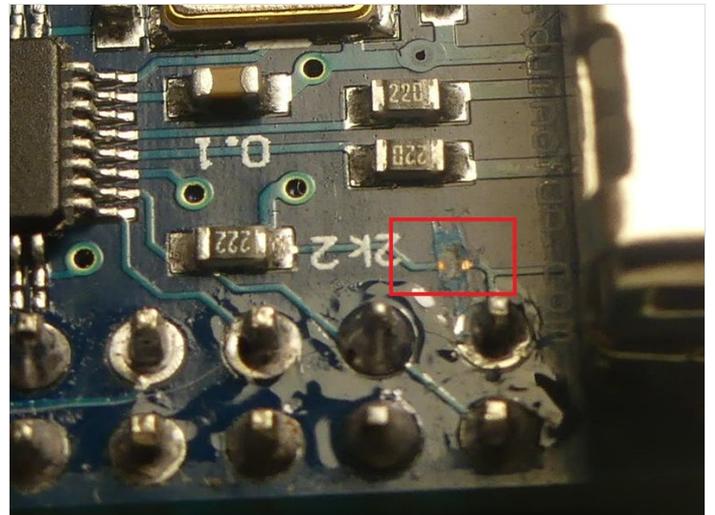
Cela veut dire que les périphériques USB seront alimentés en 3.3v. Malheureusement, aucun de mes périphériques ne marche, car ils nécessitent tous du 5v.

Vous pouvez en apprendre plus sur ce lien (en japonais)

[https://ht-deko.com/arduino/shield\\_usbhost\\_mini.html](https://ht-deko.com/arduino/shield_usbhost_mini.html)

Il va falloir couper une trace sur le circuit pour régler ce problème.

J'ai utilisé un **petit tournevis plat**, mais si vous avez un **cutter fin** c'est sûrement mieux.



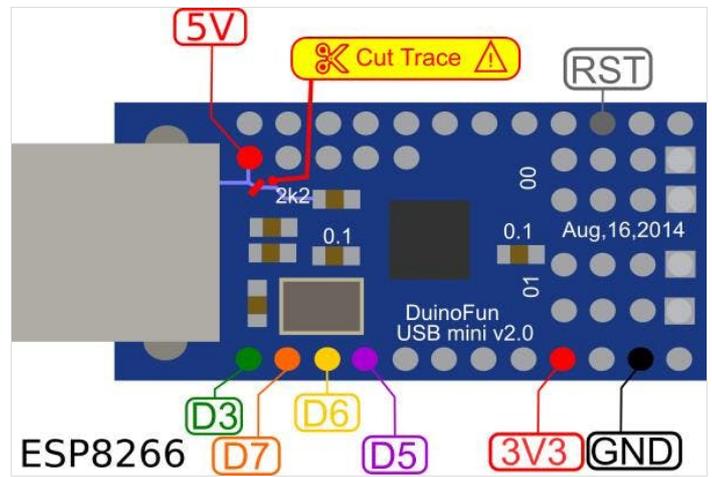
## Étape 3 - Brancher l'USB Host

OK, j'espère que vous n'avez pas cassé votre carte! Maintenant il est temps de la brancher !

J'ai utilisé un Wemos Mini D1 avec mon USB Host Mini, voici comment le brancher.

**i** Je n'ai pas relié la broche INT, car celle-ci ne semble pas nécessaire au fonctionnement

- D3 -- SS
- D7 -- MOSI
- D6 -- MISO
- D5 -- SCK
- RST -- RST



## Étape 4 - Régler le problème avec la bibliothèque USB Host

Vous avez peut-être remarqué que pour la broche SS (Chip Select / CS) j'ai utilisé la broche D3 au lieu de la broche par défaut D8.

J'ai eu un problème avec l'USB Host qui mettait mon Wemos Mini en mode programmation et bloqué l'exécution du sketch.

[https://github.com/felis/USB\\_Host\\_Shield\\_2.0/issues/457](https://github.com/felis/USB_Host_Shield_2.0/issues/457)

Afin de changer la broche SS, il va falloir changer une ligne dans la bibliothèque.

Télécharger USB Host Shield Library 2.0

Ouvrez le fichier `UsbCore.h` dans

`libraries/USB_Host_Shield_Library_2.0` et changez cette ligne

```
#elif defined(ESP8266)typedef MAX3421e<P15, P5> MAX3421E; //  
ESP8266 boards
```

Avec cette ligne

```
#elif defined(ESP8266)typedef MAX3421e<P0, P5> MAX3421E; //  
ESP8266 boards
```

**i** Pourquoi P0 au lieu de D3 ? c'est parce que le GPIO0 est le D3 sur le Wemos Mini.

<https://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/>

## Étape 5 - Downgrader ESP8266

On y est presque, si vous essayez de compiler un exemple dans la bibliothèque, **il ne marchera pas!**

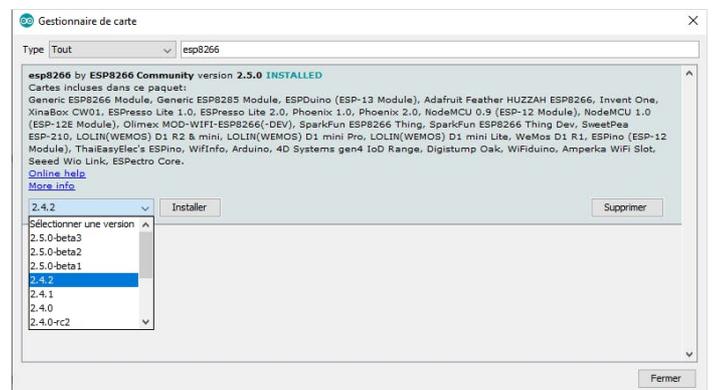
**i** Ce problème a été réglé récemment, cette étape n'est nécessaire que si vous utilisez la version 1.3.2 de cette bibliothèque

Il y a une incompatibilité entre la bibliothèque et la version 2.5 d'ESP8266 sur le logiciel Arduino.

Il va falloir downgrader celle-ci à 2.4.2

[https://github.com/felis/USB\\_Host\\_Shield\\_2.0/issues/449](https://github.com/felis/USB_Host_Shield_2.0/issues/449)

Aller dans Outils / Type de carte / Gestionnaire de carte et changer la version à 2.4.2





# Étape 11 - Adaptation ESP32

Le branchement sur ESP32 (testé sur DOIT ESP32 DEVKIT V1):

SS : GPIO5

INT : GPIO17

SCK : GPIO18

MISO: GPIO19

MOSI: GPIO23

RST : 3.3V

---