

# Systeme Ventilation Fablab

- [Main](#)

# Main

Il faut update ce doc et corriger les erreurs et bien le formater, je fais cela rapidement ici car je n'ai pas accès à l'ancien wiki

Code:

<https://github.com/alexcrerra/fablabVentilateur>

Il faut tout télécharger et l'ouvrir avec Arduino comme un fichier (le fichier et segmenter en plusieurs parties pour éviter d'avoir un grand bloc difficile à lire.

Le code est près. Il faut passer maintenant à la partie de conception de la PCB + boîtier et connecter les ventilateurs.

1 capteur contrôle 1 ventilateur - plus pratique si présence de gradient de température.

Pour la suite:

- estimer conso. relay (amp) pour dimensionner l'alimentation
- utiliser ecosystème groove au lieu de cables et vérifier comment alimenter en externe

04/12/2024

On a mis à jour le code et reconnecté tout.

**Ne pas débrancher svp.**

-Écran remplacé par un écran plus petit, bouton remplacé par celui de 2 encoders-

Pour un capteur de température --> problème pour ceux qui utilisent i2c car même adresse. Faut donc soit changer de capteur, soit changer les adresses avec un fer à souder.

Une fois le capteur remplacé (2/3 sont bons), tout est ok pour passer en phase de test.

Par la suite, il faut faire un boîtier

Pour plus tard il faut ajouter une carte SD.

## **06/12/2024**

Aujourd'hui, nous avons commencé la modélisation du boîtier.

Après avoir jeté un coup d'oeil à la grow box, nous avons conclu que la boîte ne doit pas faire plus de 7cm de profondeur.

Suite à cela, nous avons mesuré tous les composants électroniques qui constituent le système de ventilation, plus précisément la distance des trous de fixation afin de les prendre en compte lors de la modélisation. La boîte n'est pas totalement finis mais voici une première version sans le couvercle en pièce-jointe.

Par ailleurs, il faut aussi ajouter le troisième capteur. Nous n'avons pas eu l'occasion de le faire mais il suffit juste de brancher le capteur à l'emplacement D3 du shield. Les fils sont déjà branché sur le shield, il suffit de le brancher de l'autre côté les fils noir, rouge et jaune avec le capteur (utilisez le capteur noir, il marche à merveille). Connectez les fils de même couleurs entre eux.

Vous remarquerez que les fils sont un peu abîmés au bout, et qu'ils ne tiennent pas très bien. Pour assurer la connection, on pourra souder les fils du capteurs avec ceux brancher sur le shield (et éventuellement, ajouter un ou deux résistance pour protéger le circuit).

## **11/12/2024**

Essai avec carte SD, nouveaux bugs qui ont apparus d'un coup après module SD. À vérifier et debugger.

La version d'avant sur le github marche bien.

## **13/12/2024 : Miro von der Borch**

J'ai réussi à debugger le bug graphique qui était dû à une mauvaise attribution des pins.

Je n'ai toujours pas réussi à comprendre ce qui ne va pas avec la carte SD, peut être que l'arduino est sous alimenté ?

Il faudrait modifier la librairie Adafruit\_SHT31 car on utilise un SHT35 et peut être nettoyer un peu les variables car il y en a un certain nombre qui ne servent à rien.

Je n'ai pas réussi à upload mon code sur le github, mais la seule modification que j'ai faite est de vérifier tout les pins.

## **18/12/2024 : AlexC**

Ajout sauvegarde paramètre car problème avec le buffer créer par fichier ouvert dans la carte sd - > 100k cycles dure plusieurs années voir décades voir siècles donc ok. en effet il faut clean le code. pins vérifiés car bug à cause du switch qui se toggle tout seul. la partie sauvegarde de données est codée et presque prête... faut juste debug les fonctions dans sdutilities (à renommer) car erreurs lors de écriture dans eeprom (c'est rapide faut juste check le code). la lecture est bonne c'est juste qu'au démarrage on lit des paramètres du 1er test et pas les paramètres updates).

## **20/12/2024 : Mikhail KOGAN**

Continué à faire le boîtier, qui est quasiment terminé avec le couvercle (fichiers en pièce jointe), néanmoins on pourrait peut être trouver des charnières et de modifier le modèle pour les y intégrer.

## **22/01/2025 : Miro VDB**

Pour le boîtier, au vu du nombre de capteurs à mettre, on pensait passer à un arduino méga, donc il faudra peut-être revoir le design de la boîte en conséquence. De plus, serait-il possible d'avoir un dessin de l'agencement des pièces dans la boîte ainsi que leurs dimensions (il est difficile de deviner ce qui va où simplement avec les trous de vis) ?

Update : j'ai terminé la partie stockage de données sur l'EEPROM, ça fonctionne à merveille. Il ne reste plus qu'à nettoyer le code. je vais commencer et faire ce que je peux.

Update bis : Il faudrait qu'on réimplémente la fonction de logbook qui normalement ne posait pas de problème.

Update ter : j'ai nettoyé une partie du code et ajouté la fonction de logbook. J'ai identifié d'où pourrait venir le problème : les librairies des capteurs de température qui semblent être en conflit avec celles de la carte SD. Je propose de s'en faire des custom pour pallier à ça (j'en ai déjà fait une pour le SHT35 qu'il suffit d'implémenter dans le code). D'autre part, je n'ai pas pu ajouter le

code sur le repo, alors je l'upload ici. il est fonctionnelle, il suffit de mettre la fonction "handleLogBook" en commentaire.

## 22/01/2025 : AlexC

Le switch du potentiomètre n'était pas connecté.

Fonction pour bypass la carte SD et logbook ajoutée pour pouvoir debugger le système sans besoin de tout connecter (suffit de faire BYPASSSD = false; dans la première tab.

Ensuite je vois un problème avec la gestion du overflow de la fonction de millis() qui en 50 jours se remet à 0. J'ai pensé a faire un fabs(Delta temps) et cela devrit fonctionner, puisque lorsque millis va à 0 la condition dans le if est vérifiée. Ceci dit, pour le logbook cela pose problème pour la gestion du temps. On peut ajouter une condition diff. pour ce cas:

```
if(millis() - timeLogBook){
```

```
faire truc normal
```

```
}
```

```
if(millis() -- timeLogBook <= 0){
```

```
// overflow et dcp il faut modif comment on gère le temps.::
```

Car oui la gestion du temps n'est pas présente dans le fameux logbook donc soit on ajoute un module RTC (assez simple), soit on doit convertir nous-mêmes le millis en min, heures et jours...

Ensuite oui comme discuté avec Milo il faudrait passer sur une mega vu le nombre de I/O...

Nous avons également vu en quoi le module sd pourrait être, par sa connection SPI en conflit avec un des capteurs. Je vais directement basculer sur un arduino mega.

Il y a un monde ou la mega rentre sans devoir modifier la boîte

D'ailleurs je pense que 6 relays c'est un peu overkill mais bon, au pire enlever 1 paire serait sensé.

Sauf si vous comptez utiliser 2 ventilateurs par capteur, car sinon il faudra trouver 3 autres capteurs (en plus je pense que même avec 2 on peut parfaitement s'en sortir...);

Bon, 15 min de debugging pour rien... quelqu'un a modifié la fonction du switch du potentiometre et a remplacé la fonction qui bouge le curseur de configuration par le curseur d'affichage température... problème réglé.

Bien joué sinon Milo pour l'EEPROM, cela marche très bien. Je pense qu'il ne reste qu'à remplacer les capteurs de température (il y en a 3 différents) et voir souder les cables et ensuite régler le problème avec la gestion du temps dans le logbook. D'ailleurs il faut ajouter une option de créer un nouveau logbook a chaque fois que le système est réinitialisé (éviter de réécrire si restart). Une solution est de prendre un espace dans l'EEPROM comme n° de fichier et faire log\_99.csv où 99 est le nombre stocké +1. je le ferais si j'ai le temps mais c'est très facile est je l'ai déjà utilisé et je sais que c'est très pratique!

Finalement, il faut ajouter un module SD différent car la carte SD doit être récupérable sans besoin de tout démonter. en plus je ne pense pas qu'avoir le shield vert qui écrase les câbles soit une bonne idée. J'ai trouvé un module externe dans une boîte:

