

Détecteur de CO2 v3

Présentation



Caractéristiques du détecteur de CO₂:

- compact (10,6 x 4,3 x 5,4 cm)
- autonome grâce à un accumulateur (testé à plus de 12h d'autonomie avec 750 mAh)
- mesure du CO₂ (avec calibration), de la température et du taux d'humidité
- microcontrôleur ESP32 (extensible avec des fonctionnalités Bluetooth ou WiFi par exemple)
- écran de type ePaper (consomme très peu d'énergie) avec historique
- LED multi-couleur permettant une palette variée de couleurs et Buzzer (désactivable)
- Seuils d'alerte réglables

Je tiens à remercier Henri BACHETTI: c'est grâce à la lecture de son blog [RitonDuino](#) que j'ai fini par me lancer dans l'électronique et l'Arduino.

Cette page peut être exportée au format PDF (menu en haut à droite).

Utilité de surveiller le taux de CO2

Les concepts énoncés ici ne sont pas approfondis, pour plus d'information on pourra consulter les sites *Projet CO2* (<http://projetco2.fr/>), *Nous aérons* (<https://nousaerons.fr/>). La science a maintenant montré que la transmission de la COVID-19 se fait majoritairement par le biais des aérosols. Les aérosols sont des «micro-goutelettes» potentiellement chargées en virus qui se diffusent et restent en suspension dans l'air (penser par exemple à de la fumée de cigarette dans un pièce). Une méthode

simple et efficace de limiter la transmission est d'aérer le plus possible les espaces intérieurs pour renouveler l'air. La mesure du taux de CO₂ dans l'air est un marqueur fiable de l'état de confinement d'un local et permet d'être alerté lorsqu'il devient nécessaire voire impérieux d'aérer.

Électronique

Liste du matériel

Capteur de CO2

Le [Senseair S8 LP](#) a été testé et recommandé par [Projet CO2](#) (capteur de type NDIR avec processus de calibration manuelle). On peut s'en procurer pour une trentaine d'euros chez Aliexpress par exemple [ici](#). Vérifiez bien la référence 004-0-0053 !



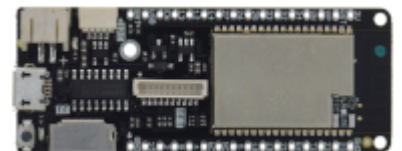
Capteur de température et d'humidité

Cette fonction est assurée par un circuit Sensirion SHT31D: composant classique i²c ayant fait ses preuves et pas trop gourmand en énergie (voir par exemple [Les capteurs de température, humidité, pression & luminosité](#)).



Microcontrôleur et écran ePaper

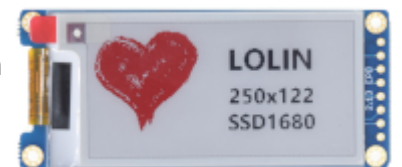
J'ai choisi la carte [D32 PRO v2](#) de chez LOLIN. Cette carte est équipée d'un ESP32-WROVER classique et possède de plus quelques avantages comme la présence de connecteurs dédiés pour les périphériques i²c et l'écran (ce qui réduit le nombre de soudures à effectuer). On la trouve chez Aliexpress sur la [boutique officielle](#).



Pour l'écran j'ai choisi le [ePaper 2.13](#) de chez LOLIN qui offre une taille et une résolution confortables et dispose du connecteur compatible avec le microcontrôleur. On peut également l'utiliser sans le connecteur avec des connexions classiques par soudure ou Dupont.



Malheureusement en voulant me réapprovisionner je me suis rendu compte que ce modèle n'était plus commercialisé (voir [ici](#)). Cependant un modèle plus récent existe: le [Tri-Color ePaper 2.13 \(SSD1680\)](#) qui, à priori, est identique au niveau des dimensions et de la résolution. Cependant le contrôleur est différent et nécessitera une petite adaptation du code source (*j'ai commandé un exemplaire et suis en attente de réception*).



Le câble pour la connexion TFT: soit chez LOLIN [ici](#) soit en prenant un câble de type JST SH1.0 SH à 10 fils droit (non croisé).

Le câble pour la connexion i²c: soit chez LOLIN [ici](#) soit en prenant un câble de type JST SH1.0 SH à 4

fils croisés.

Le support et circuit de charge pour l'accumulateur

Concernant l'alimentation nous avons la majorité du circuit alimenté en 3,3V excepté le capteur S8 qui nécessite du 5V. D'autre part, afin d'être autonome un accumulateur et donc un circuit de charge sont requis. Après recherche et plusieurs essais afin de créer un circuit dédié à cette tâche les résultats n'étaient pas convaincants par rapport à l'utilisation d'un *shield* complet gérant le circuit de charge et la sortie des deux tensions.

La v2 du détecteur utilisait un *shield* basé sur deux accumulateurs 18650 ce qui était surdimensionné: l'autonomie avoisinait les 24 heures. De plus le circuit comportait 5 LED allumées constamment ce qui n'était pas des plus économes (pas de possibilité de les éteindre hormis en les dessoudant). Finalement [ce modèle](#) basé sur un accumulateur 16340 est suffisant concernant l'autonomie. Les LED restent allumées uniquement si branché au secteur et il possède un interrupteur.



L'accumulateur

N'importe quel accumulateur de type 16340 à votre convenance fonctionnera. La boutique en ligne [NKON](#) est particulièrement bien fournie à des tarifs raisonnables. Je commande chez eux depuis plus de 10 ans. Plus près de chez nous le magasin [GO TRONIC](#) est également une valeur sûre.

La LED RGB

L'emploi d'une LED RGB permet de simplifier le circuit (par rapport à l'emploi de 3 ou 4 LED colorées) et de pouvoir faire évoluer les codes couleurs suivant les besoins. Attention à bien sélectionner une LED à anode commune (*long lead +*) de 5mm de diamètre. J'ai sélectionné [ce modèle](#) (*Diffused Lens Anode +*).

Le Buzzer

Le buzzer permet d'émettre des bips lorsque le seuil de CO₂ est trop élevé. Il faut veiller à prendre un buzzer passif de type piézo-électrique. [Ce modèle](#) convient. La version v2 du détecteur utilise [ce modèle](#) qui peut également convenir (il faudra juste souder des fils dessus et le caler dans le boîtier).



L'encodeur rotatif

L'encodeur rotatif permet de gérer les interactions de l'utilisateur: la rotation du bouton permet de faire défiler un menu ou des valeurs et l'appui permet la sélection. Ce modèle est assez compact, facile d'utilisation et fonctionne en 3,3V. *Attention à bien sélectionner le modèle doté d'un pas de vis afin de pouvoir le fixer à l'intérieur du boîtier grâce à l'écrou !* Il est disponible [ici](#) à l'unité ou [là](#) par lot de 10.



Connecteurs et en-tête Dupont

Toutes les connexions se font à l'aide de connecteurs Dupont, cela nécessite de posséder les connecteurs adéquats ainsi que la pince à sertir:

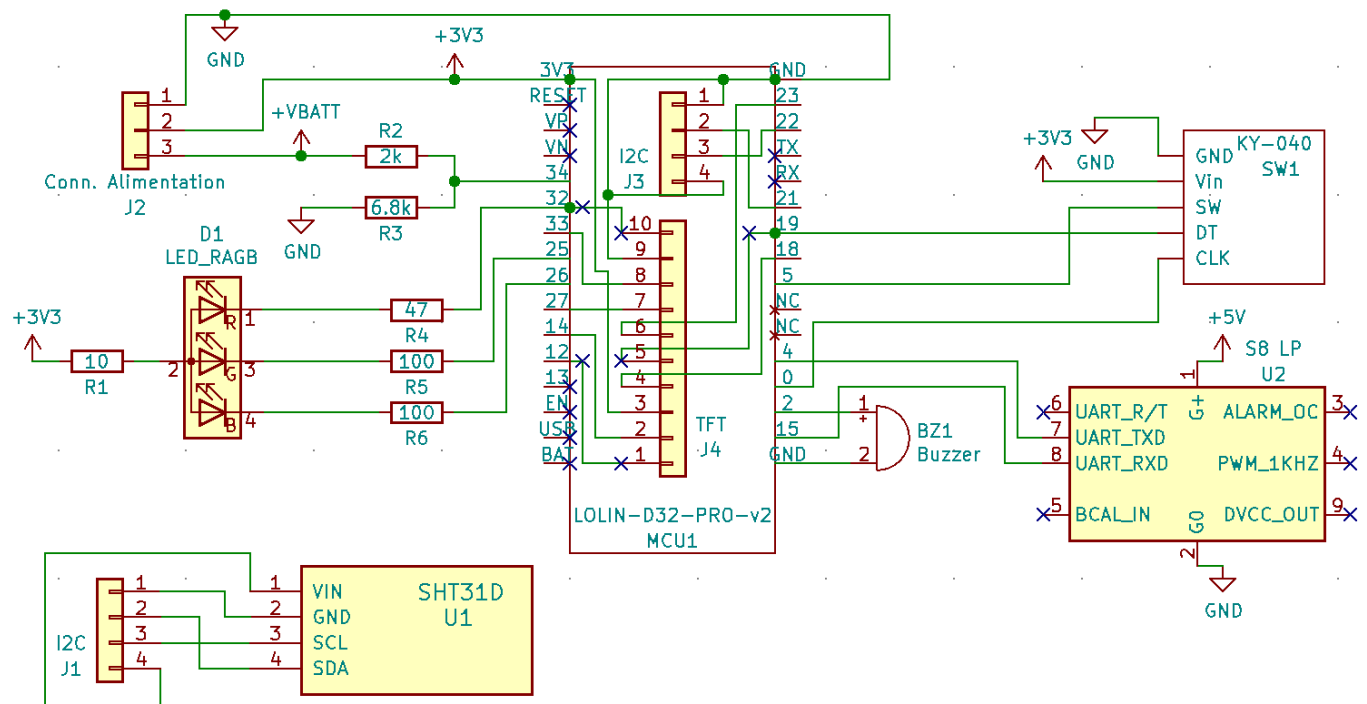
- 2 supports de 1×16 points pour la carte ESP
- 1 support 1×4 points pour le capteur S8
- des connecteurs coudés sécables (4 points pour la LED, 3 points pour l'alimentation et le KY-040 et 7 points pour le circuit de charge)

Il est également possible de ne pas utiliser de connexions Dupont et de souder directement tous les composants ensembles. Cela risque néanmoins d'être délicat concernant le détecteur S8 pour le caler dans le boîtier.

Autre

- les résistances suivantes: 10 Ω , 2 x 100 Ω , 47 Ω , 2 k Ω et 6,8 k Ω ;
- une plaque PCB à pastilles de 3 x 7 cm (24 x 10 trous);
- bien sûr un fer à souder et tout ce qui avec;
- du [câble électrique de couleur](#);
- de la gaine thermorétractable.

Schéma



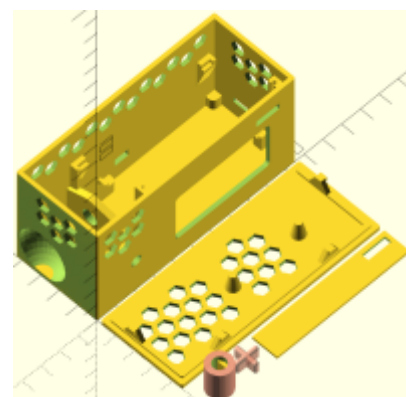
Le plus compliqué dans ce type de montage réside dans le choix des broches à répartir entre les différents éléments:

- Certaines broches sont imposées comme celles pour l'i2c (21 et 22) ou le connecteur TFT (32, 33, 27, 23, 19, 18, 14 et 12). Dans notre cas, avec l'écran choisi, les broches 32 (TFT_LED), 19 (SPI/MISO) et 12 (TS_CS) ne sont pas utilisées et restent donc disponibles pour d'autres usages.
- Afin de mesurer le niveau de charge de l'accumulateur il faut utiliser l'un des deux circuits ADC de l'ESP32. À savoir que l'ADC2 est utilisé par le module WiFi et entraîne donc un conflit. Pour l'instant le WiFi n'est pas utilisé mais en cas d'évolution il serait dommage d'être bloqué; c'est pour cela que la broche choisie pour mesurer la tension de la batterie est la 34, reliée à l'ADC1.

En rédigeant cette documentation je me rends compte d'un problème si, lors d'évolutions futures, la carte SD est requise: les broches nécessaires sont utilisées par le schéma actuel: la 19 (SPI/MISO) et la 4 (TF_CS). Il serait judicieux de libérer ces broches en déplaçant par exemple S8_TX sur la broche 12, S8_RX sur la 13 et KY040_DT sur la 15.

Boîtier

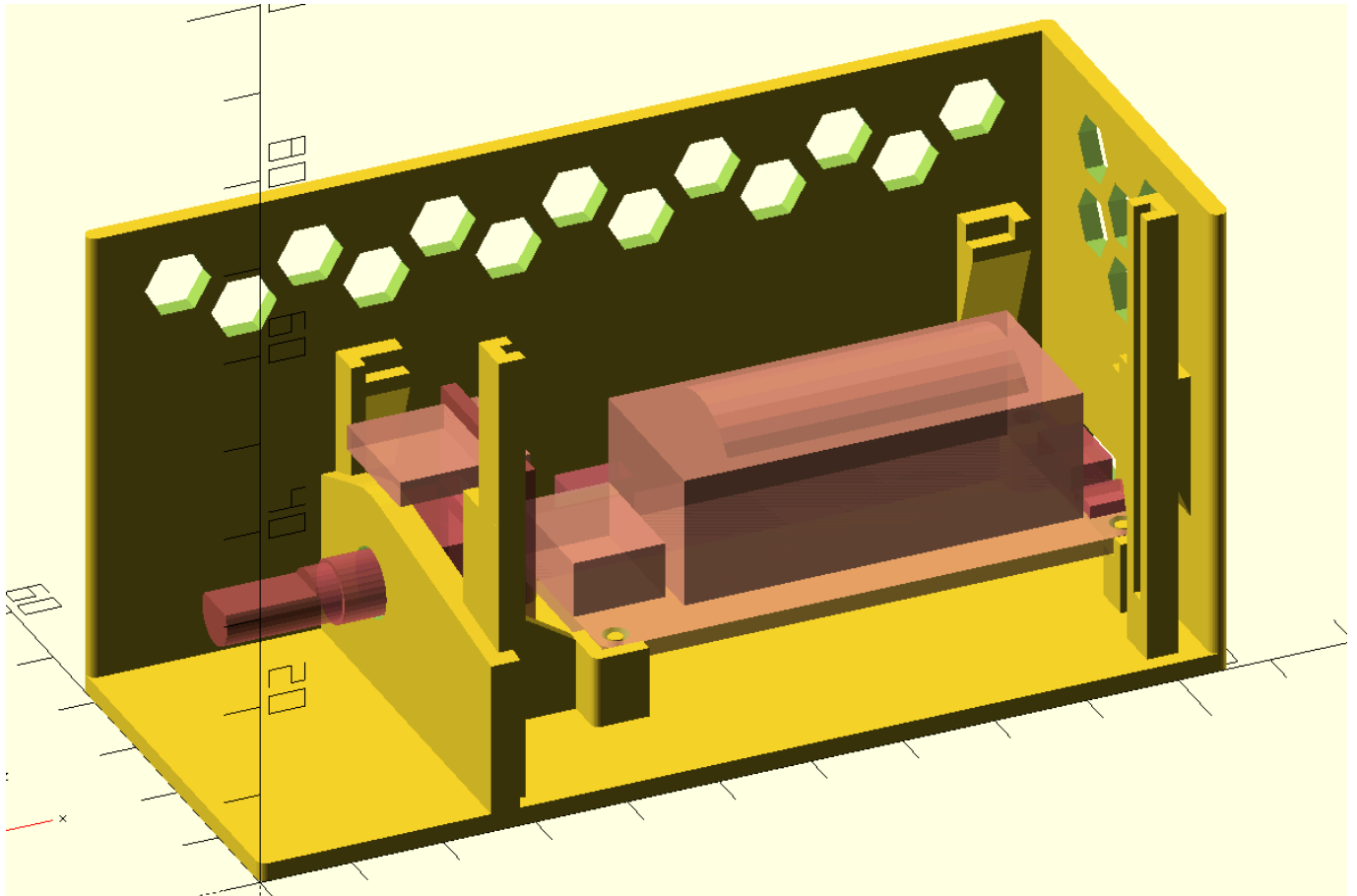
Le boîtier a été créé au besoin avec le logiciel OpenSCAD afin d'être le plus compact possible et aéré afin que le capteur CO₂ puisse effectuer ses mesures correctement. Le boîtier est composé d'une base, d'une cale pour l'écran, d'un couvercle et d'un bouton en deux parties (à coller) pour l'encodeur rotatif.

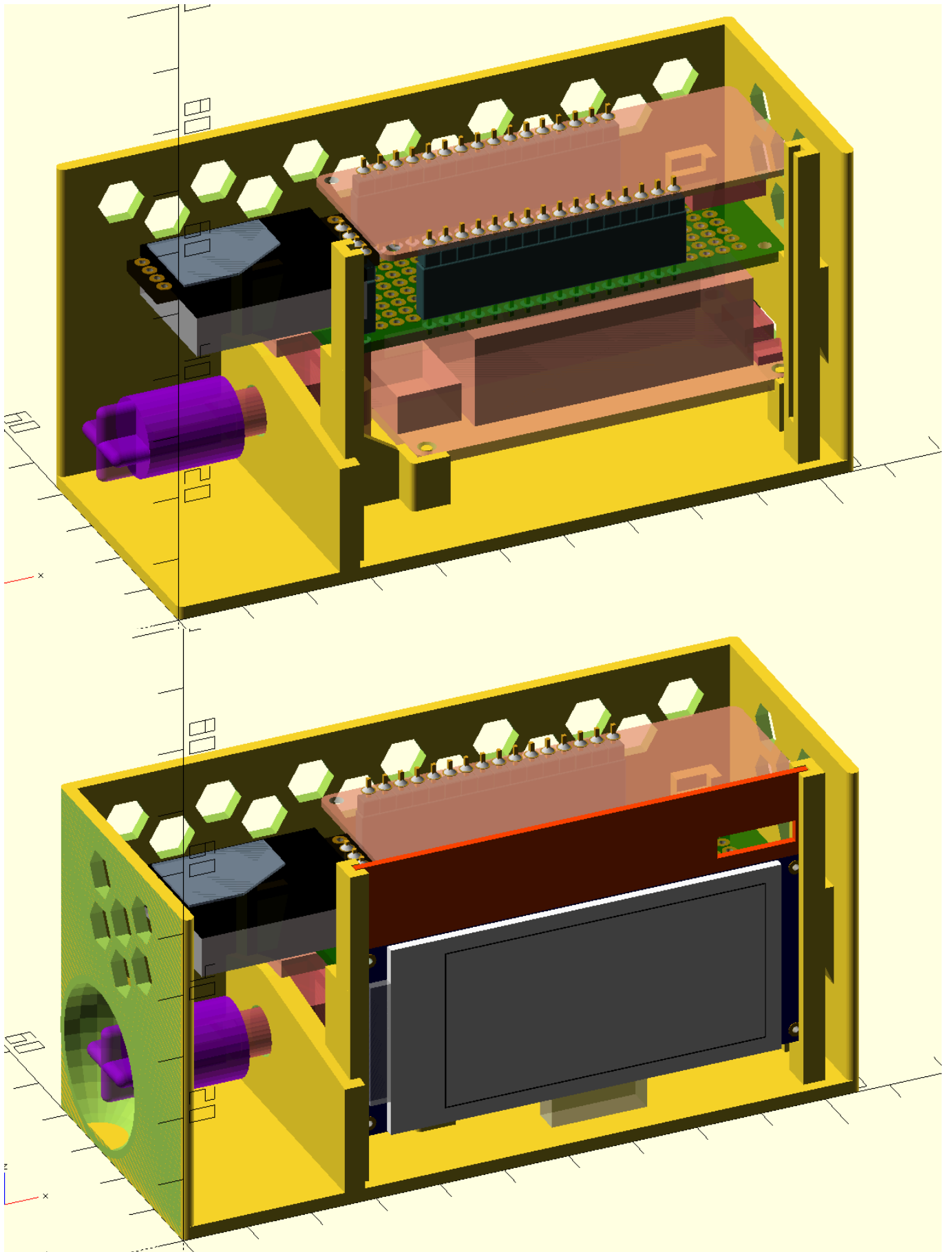


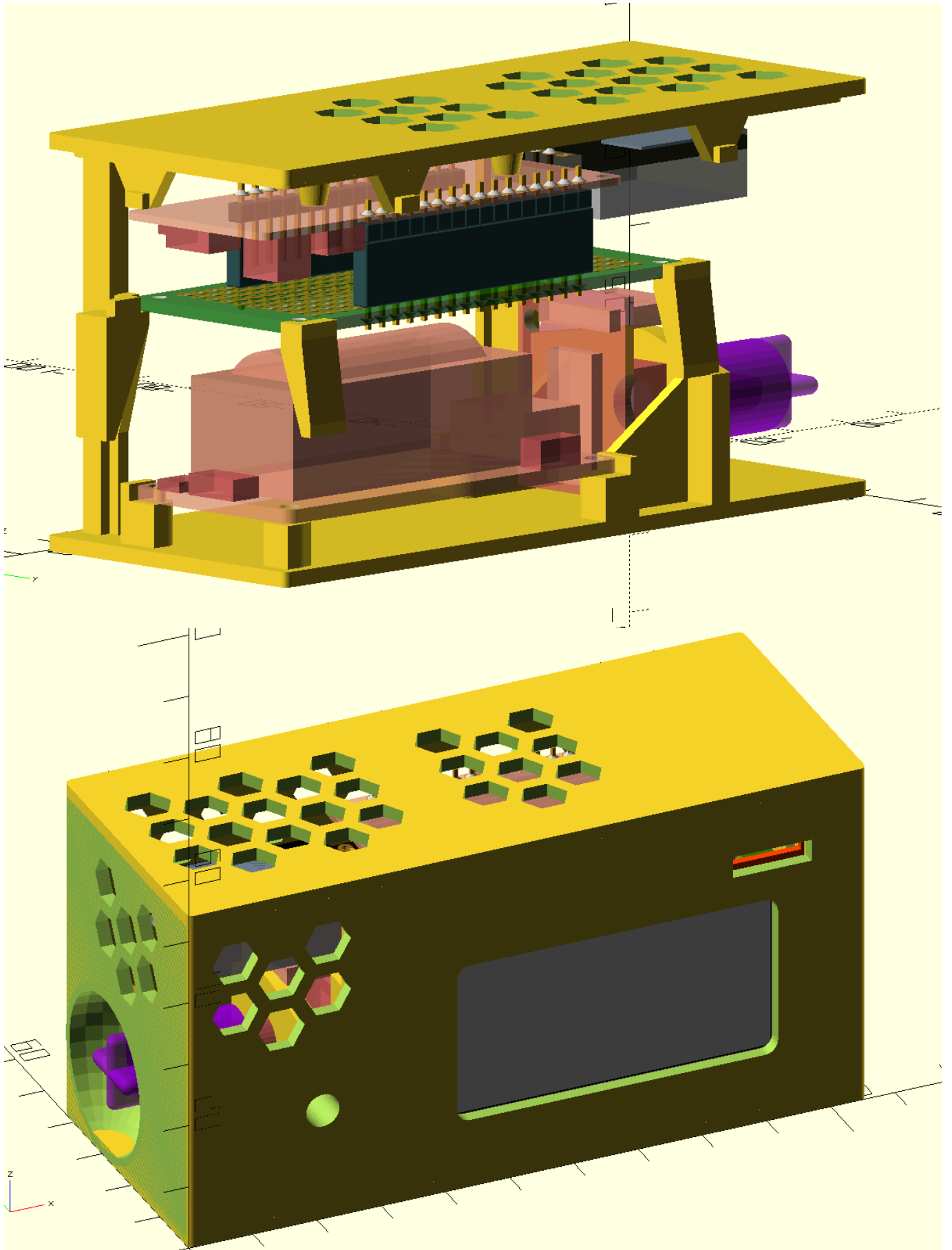
Le fichier zip fourni contient les fichiers STL prêt l'emploi pour l'impression ainsi que le fichier source

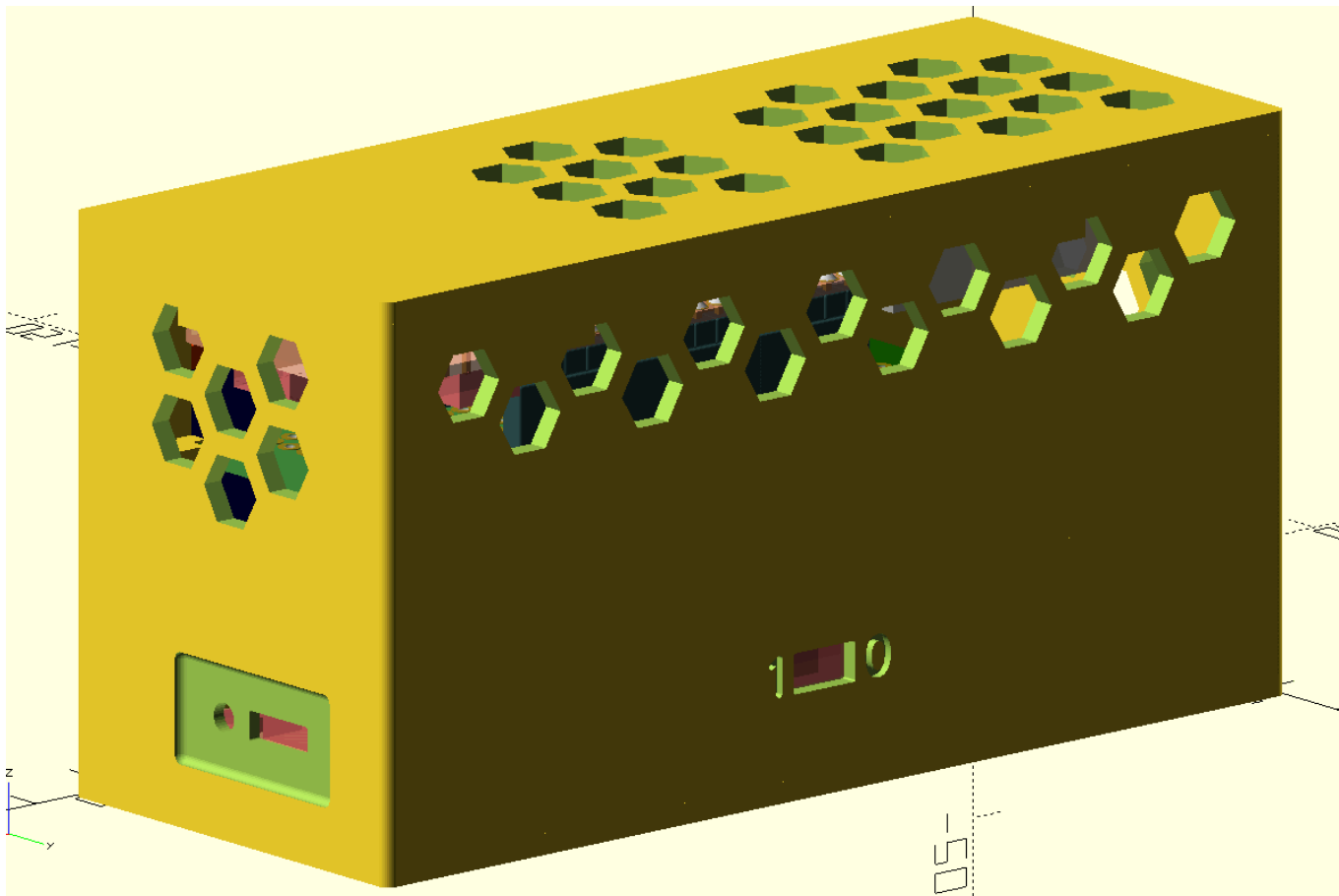
OpenSCAD et les fichiers de bibliothèques additionnelles.

J'ai effectué l'impression en PLA avec hauteur de couche classique de 0.2mm, remplissage à 50%, avec support pour les inévitables parties en porte à faux mais sans support d'adhérence. Comptez plus d'une douzaine d'heures pour l'impression de toutes les pièces. L'impression en PETG est bien sûr possible.

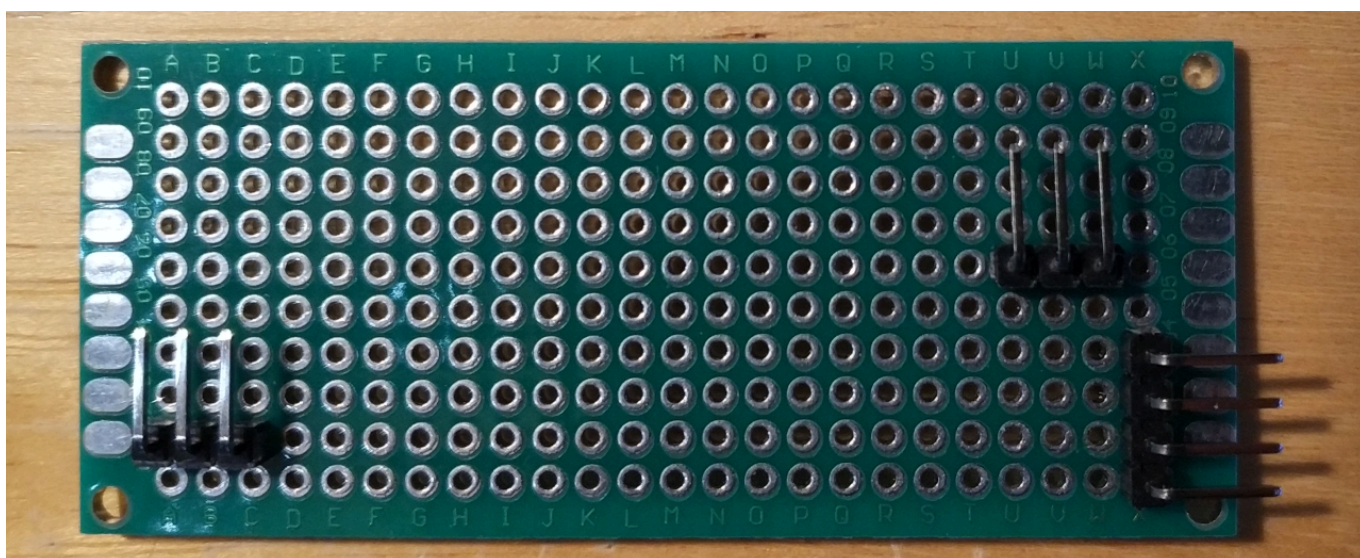




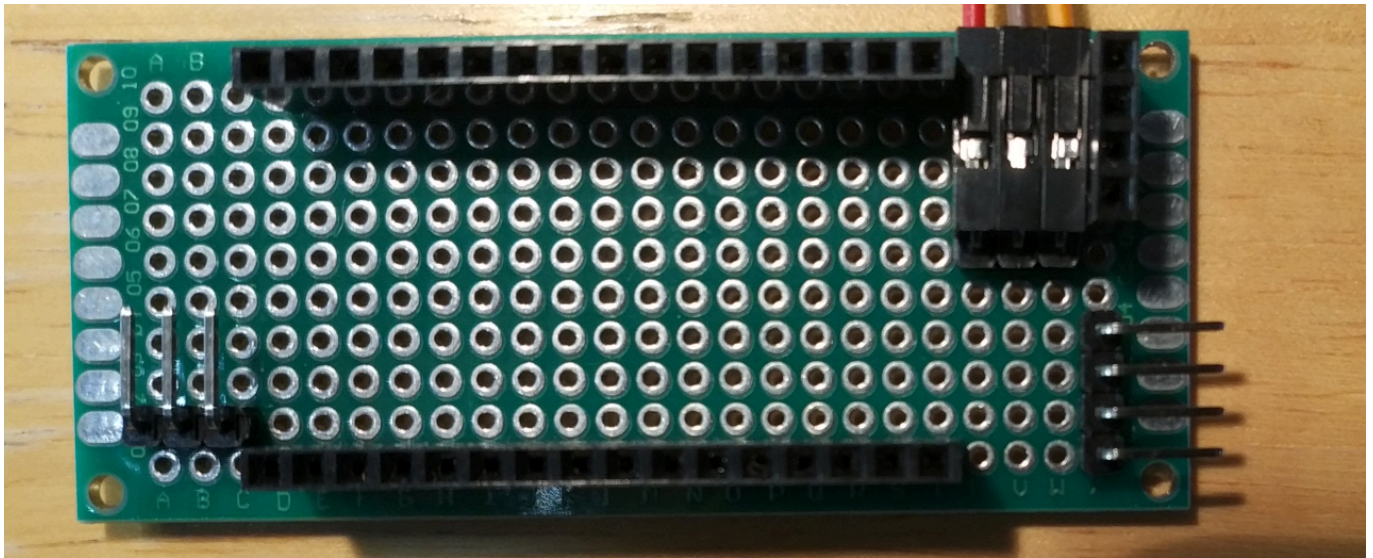




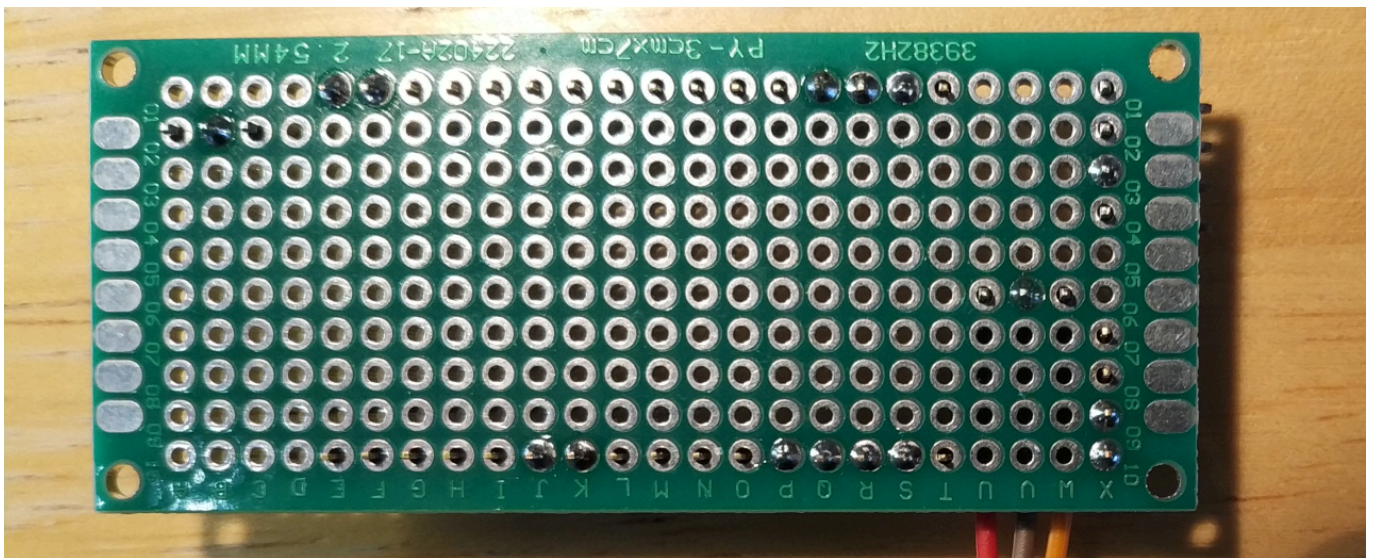
Montage



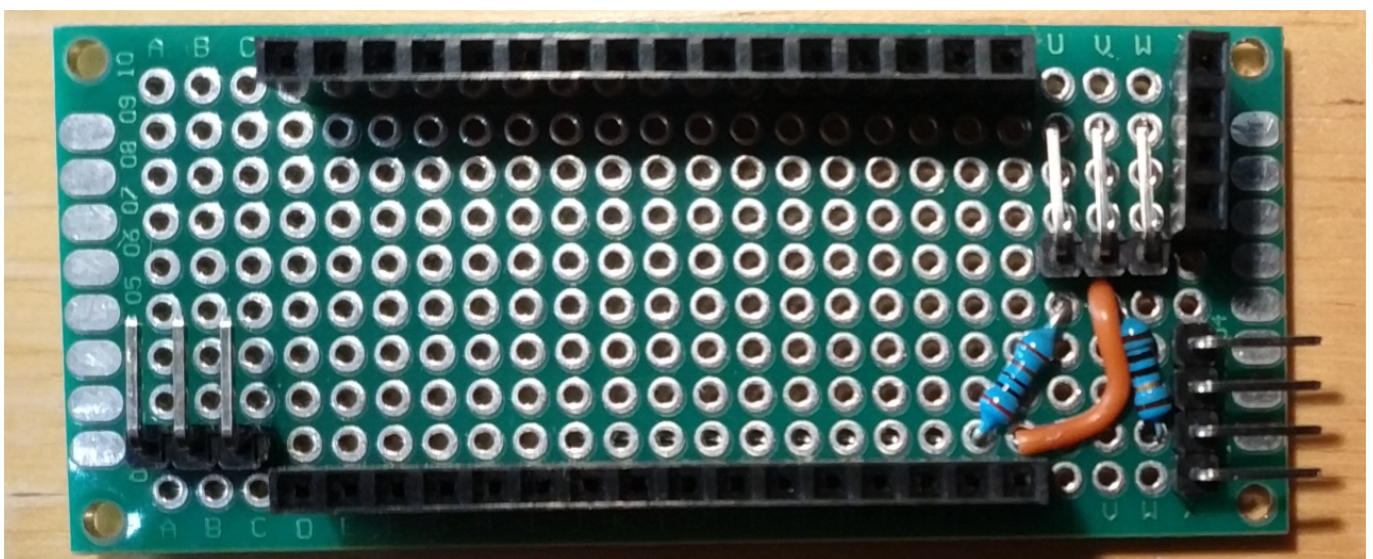
Positionner les 3 en-têtes coudées Dupont et souder une seule broche par connecteur (par exemple B2, V6, X3).



Positionner les 2 supports pour la carte ESP32 (souder les broches inutilisées) et le support pour le capteur S8 (souder les 2 broches inutilisées: X9 et X10).

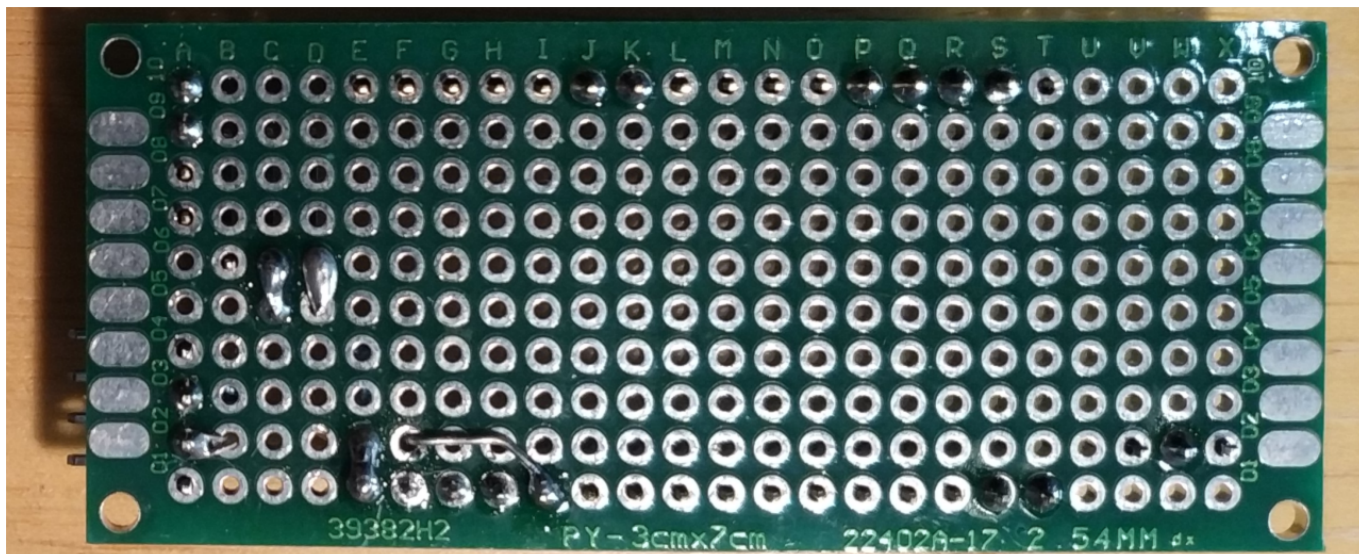


Vue côté soudures des connecteurs et en-têtes coudées Dupont.

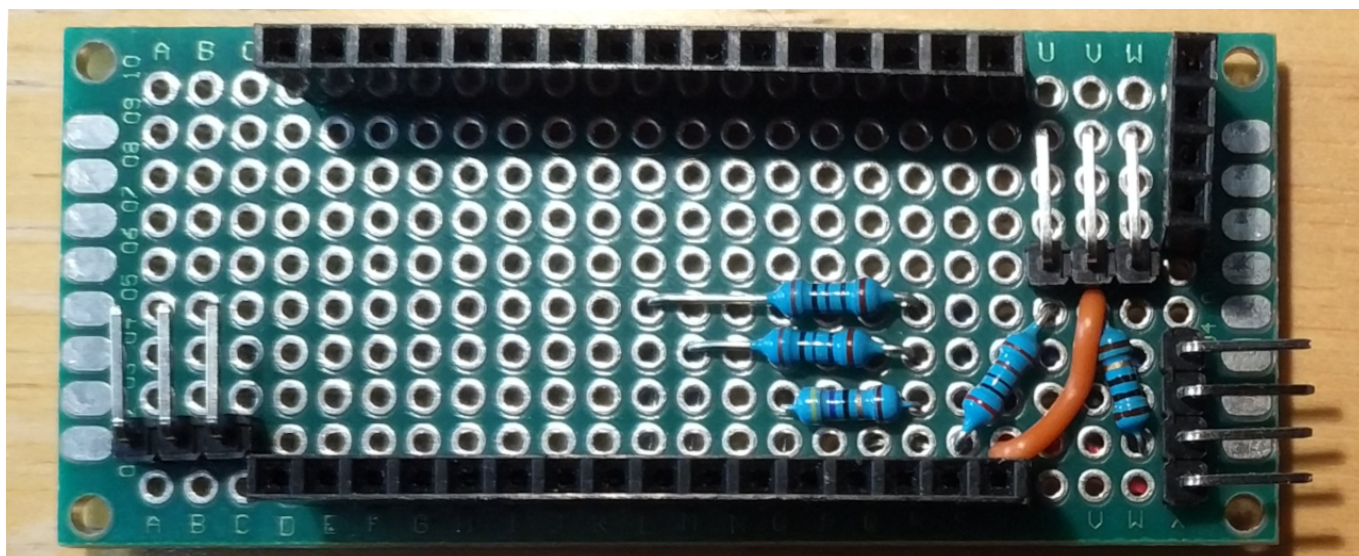


Positionner la résistance de 2 kΩ entre la broche 34 (S2) et le connecteur Vbatt (U6); un fil orange de

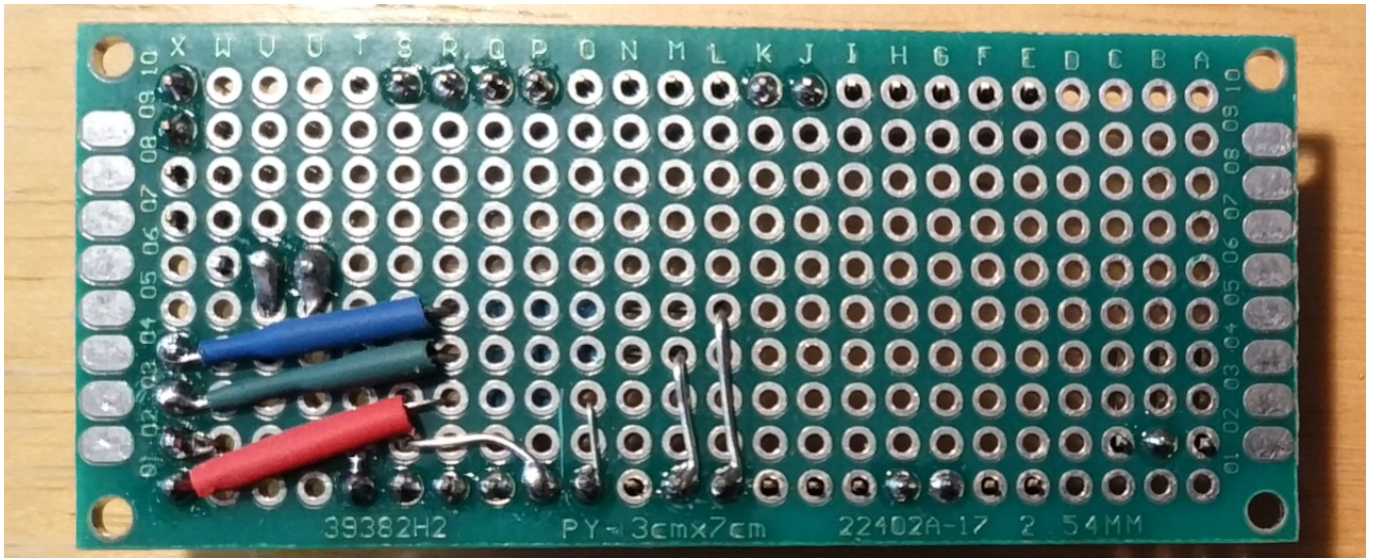
2cm entre le connecteur 3.3V (V6) et la broche 3V3 (T2) ainsi que la résistance de 10 Ω entre le connecteur 3.3V (V6) et l'anode du connecteur de la led RGB (W2).



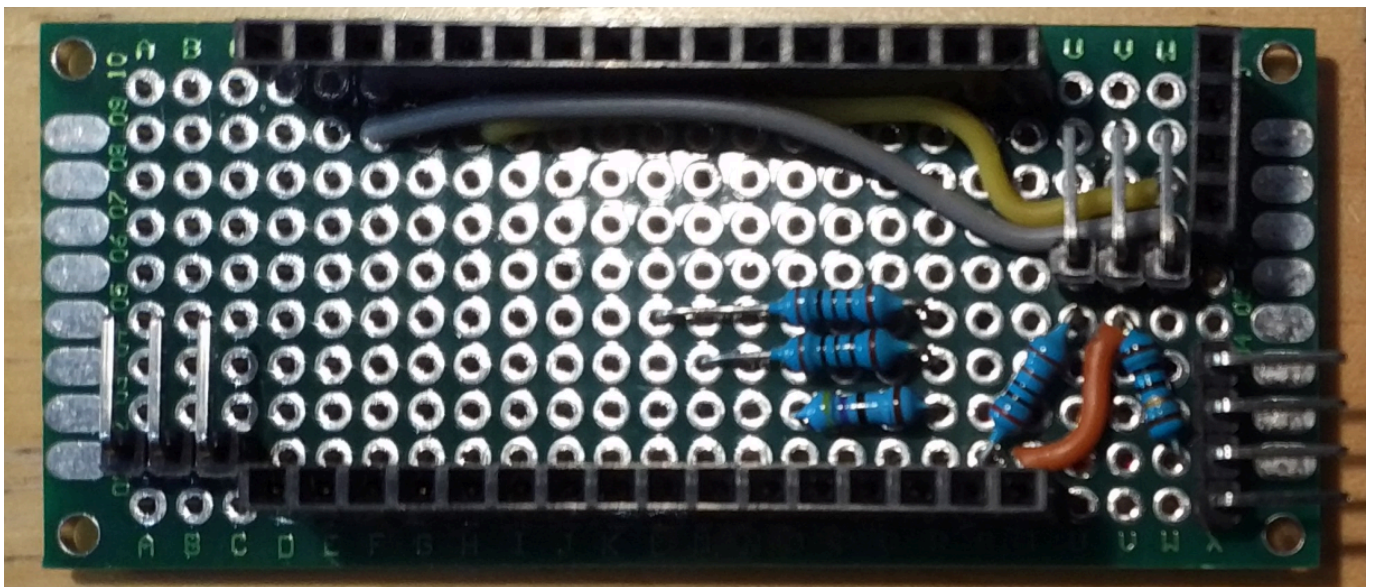
Souder la résistance au connecteur led RGB (A2-B2); le connecteur 3,3V avec la résistance et le câble orange (C5-C6); le connecteur VBatt avec la résistance (D5-D6). Ne pas souder la résistance de 2 kΩ avec la broche 34 (S2) (contrairement à la photo).



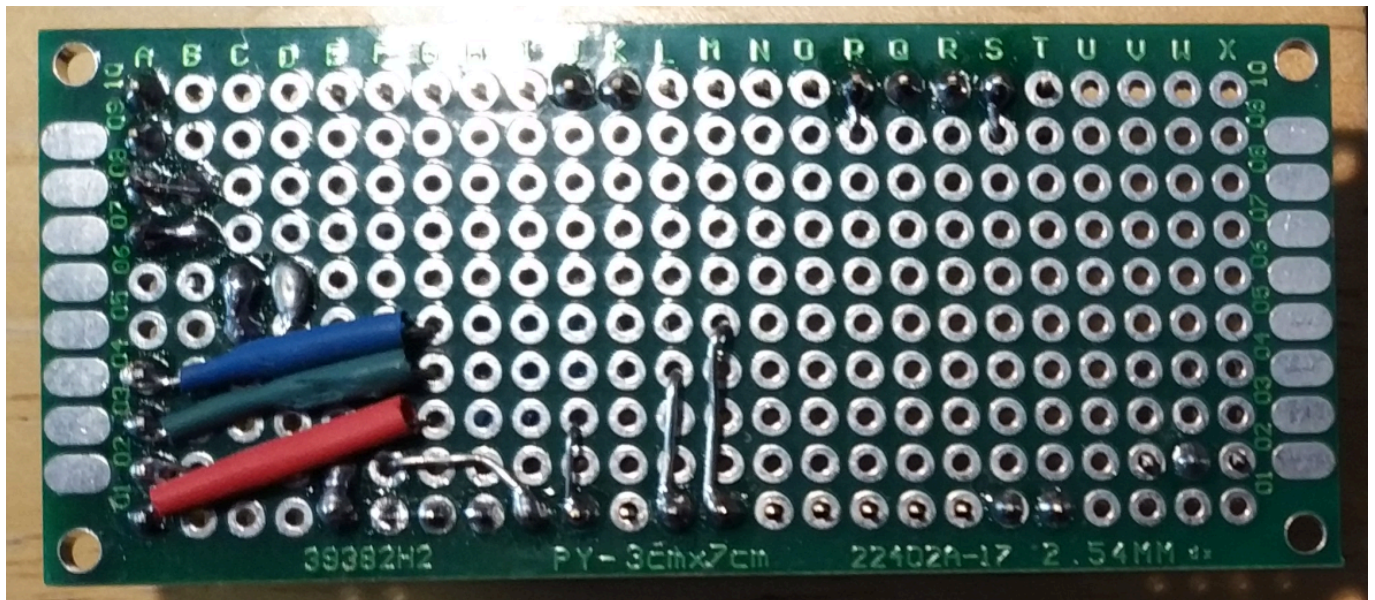
Positionner la résistance de 47 Ω en O3-R3; les résistances de 100 Ω en M4-R4 et L5-R5.



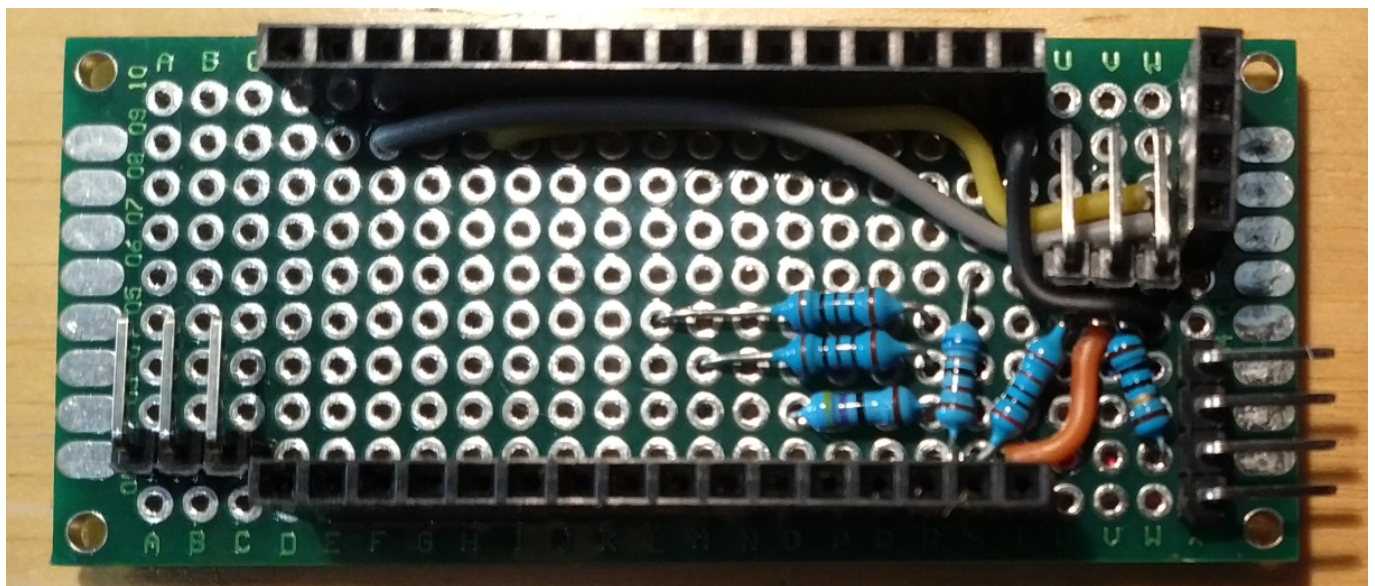
Utiliser 15 mm de gaine thermorétractable sur les pattes des résistances côté connecteurs RGB puis les souder. Côté ESP32 souder les résistances sur les points L1, M1 et O1.



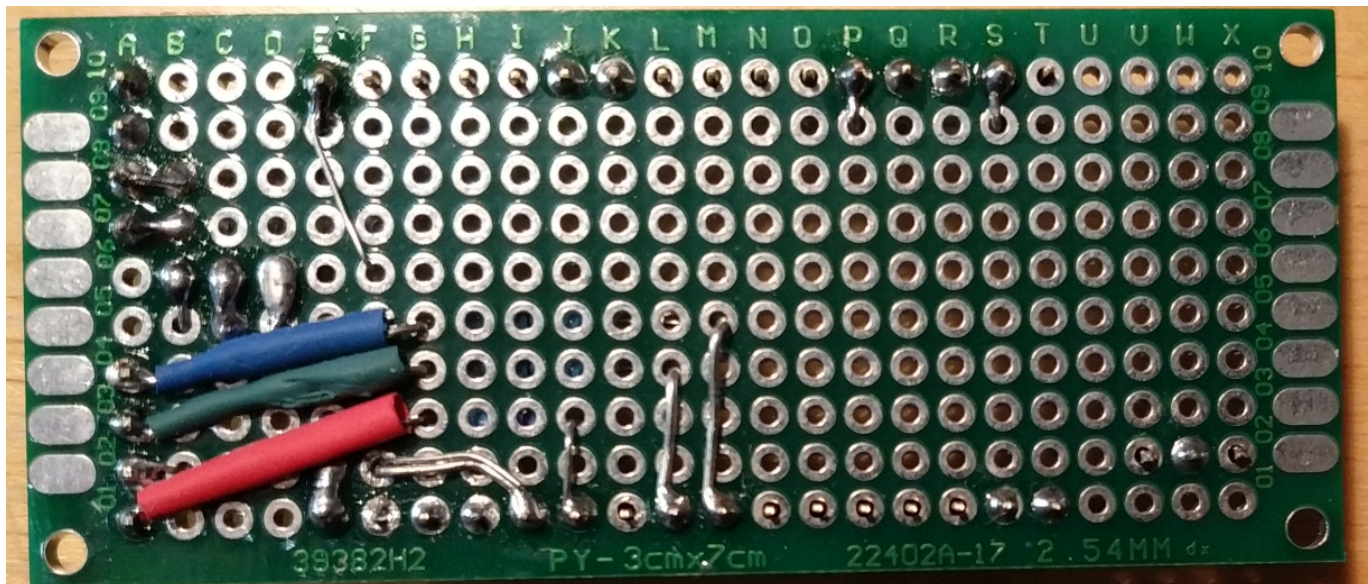
Pour la liaison avec le capteur S8 préparer un fil gris de 55 mm entre F9-W7 et un fil jaune de 50 mm entre I9-W8.



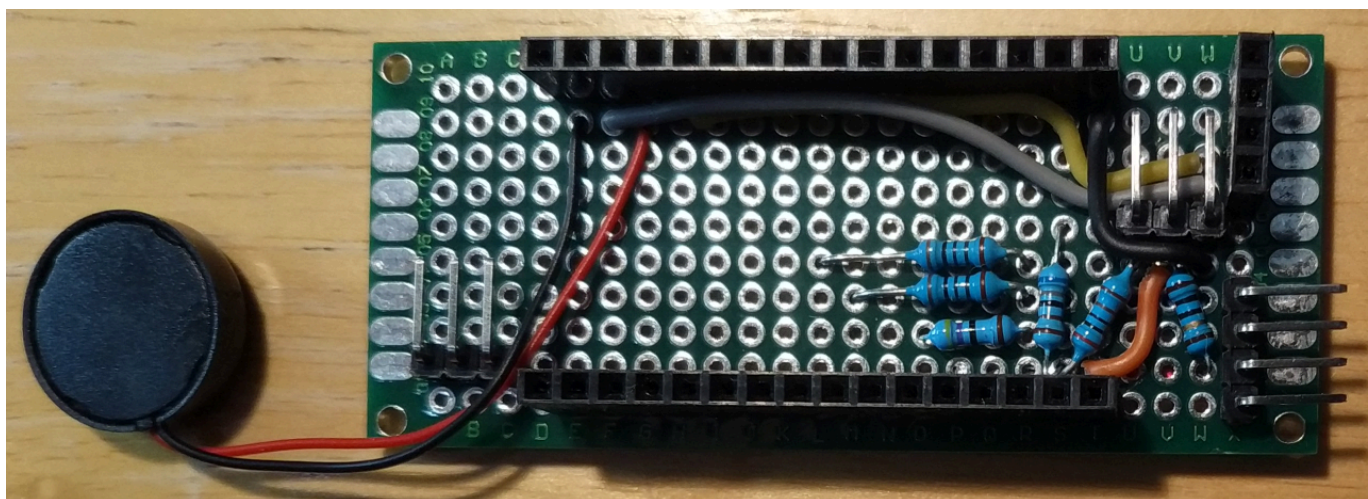
Souder les fils en F9-F10, I9-I10, W7-X7 et W8-X8.



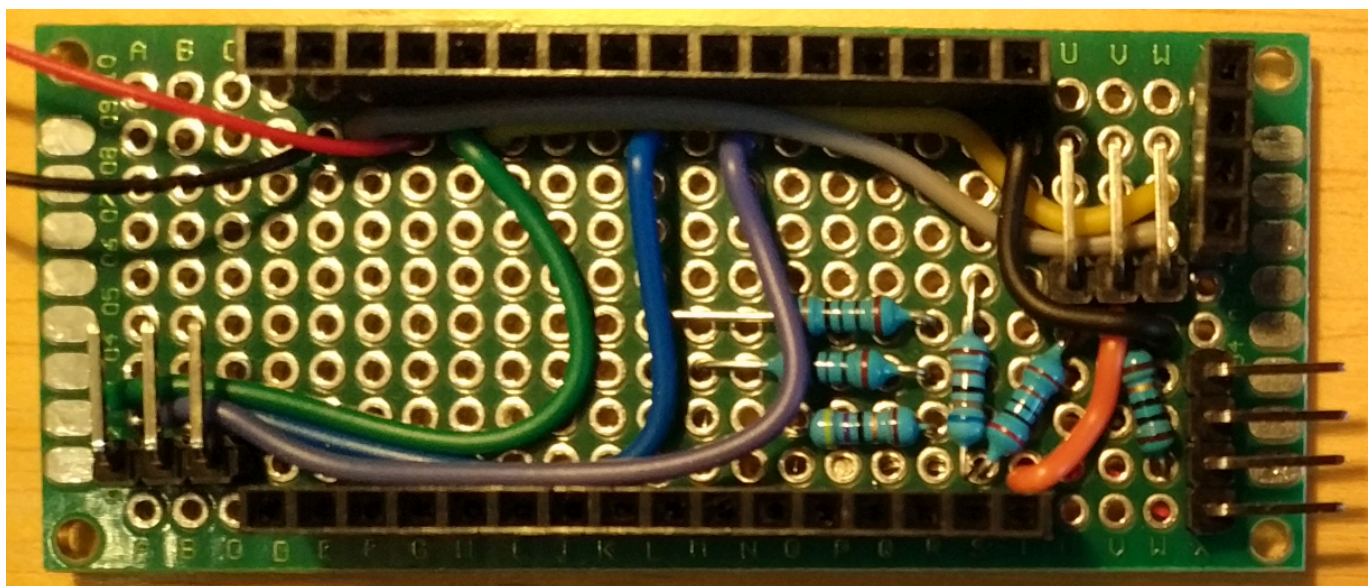
Préparer un fil noir de 28 mm et le positionner entre W5 et T9; installer la résistance de 6,8 kΩ entre S2 et S6.



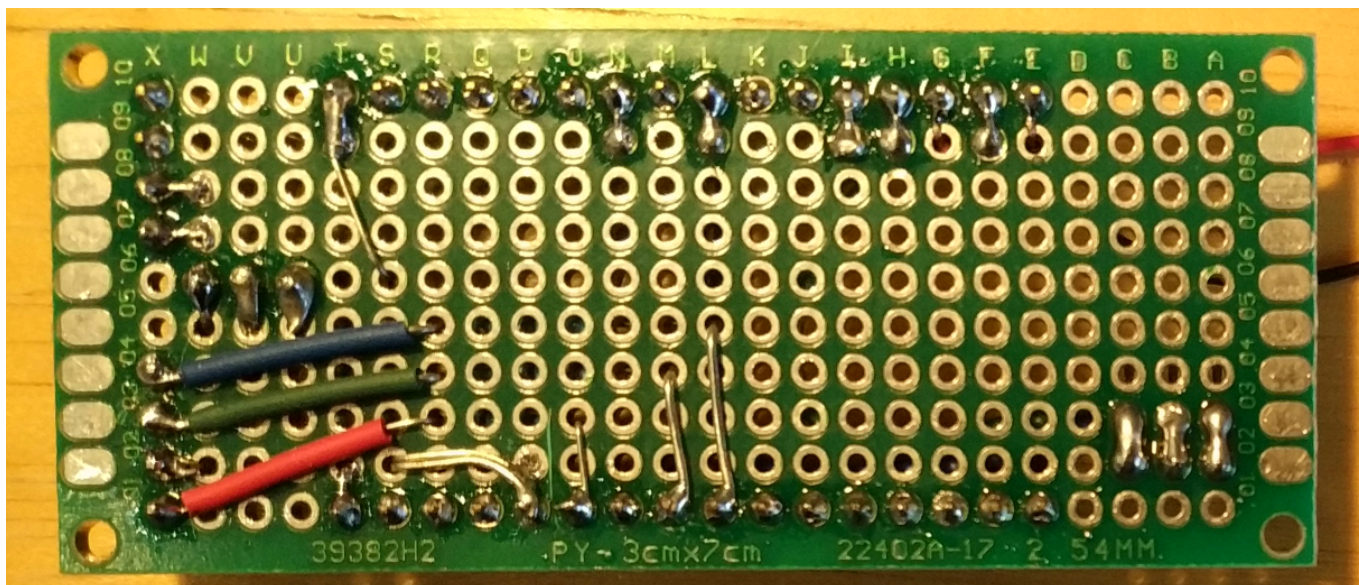
Souder la résistance et le fil noir entre T9-T10; les deux résistances en P01 et le fil noir sur le connecteur d'alimentation entre W5-W6.



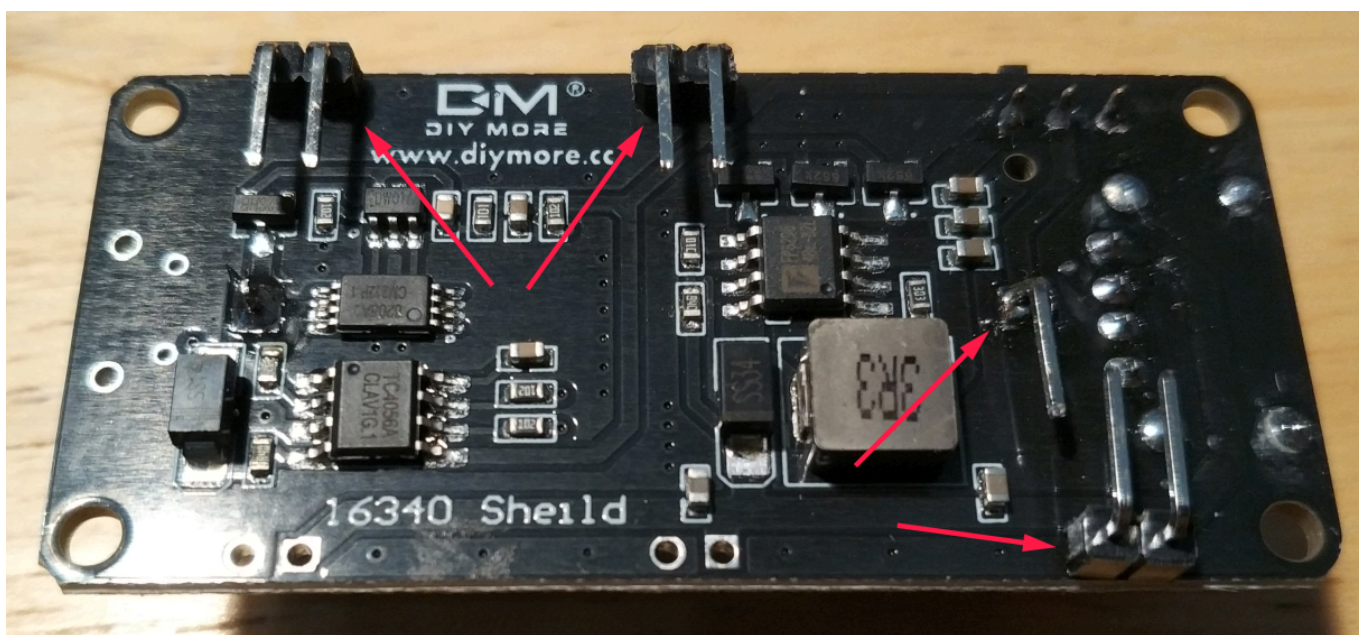
Positionner le fil noir (GND) du buzzer sur E9 et le fil rouge (+) sur G9; puis souder entre E9-E10 et G9-G10.



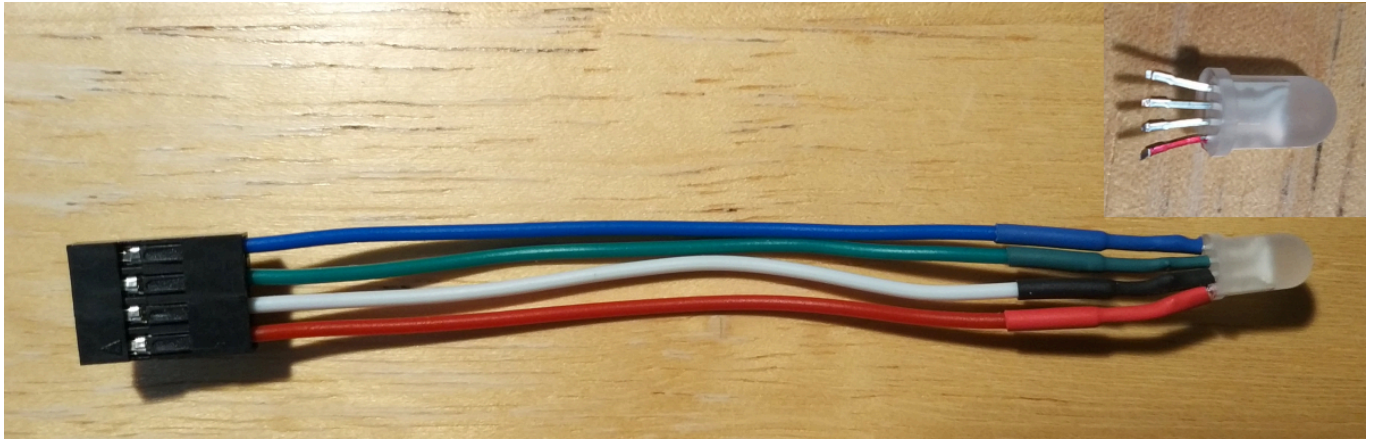
Positionner un fil bleu de 50 mm entre L9-C3, un fil violet de 55mm entre N9-B3 et un fil vert de 55 mm entre H9 et A3. La longueur choisie permet d'éloigner ces câbles de l'emplacement du connecteur de l'écran sur la carte ESP32 afin de libérer de l'espace pour le câble.



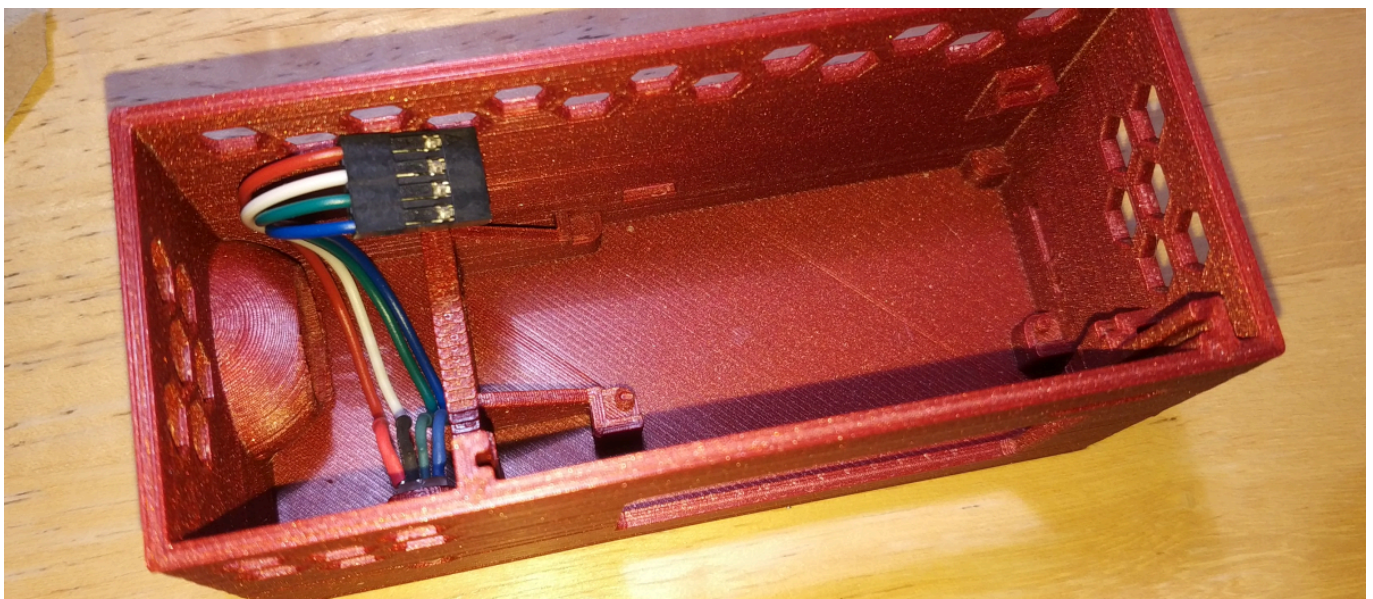
Souder les fils avec le connecteur entre A2-A3, B2-B3 et C2-C3 ainsi que tous les connecteurs non encore soudés. La carte est terminée.



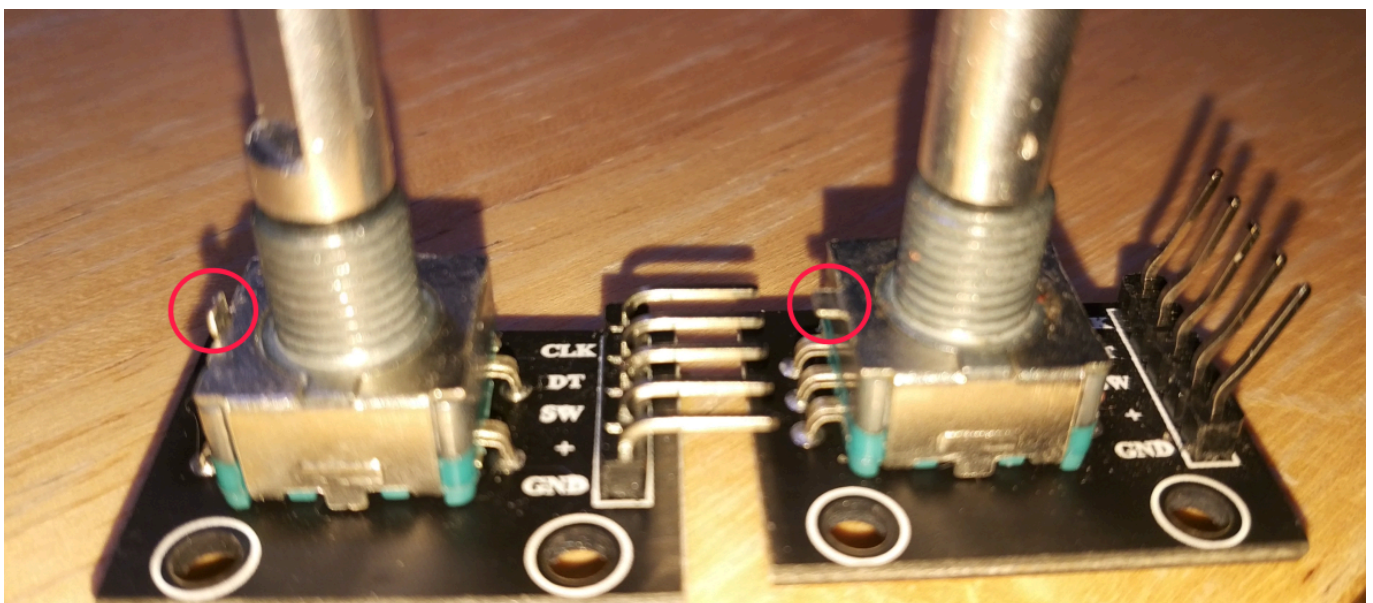
Sur le circuit de charge de l'accumulateur, souder des connecteurs 2 points coudés sur 1 sortie 5V et 2 sorties 3.3V comme sur la photo. Puis sur un connecteur coudé supprimer la partie plastique et couper une patte à mi-longueur. Il faut souder cette patte sur la borne positive de l'accumulateur afin de pouvoir mesurer sa tension.



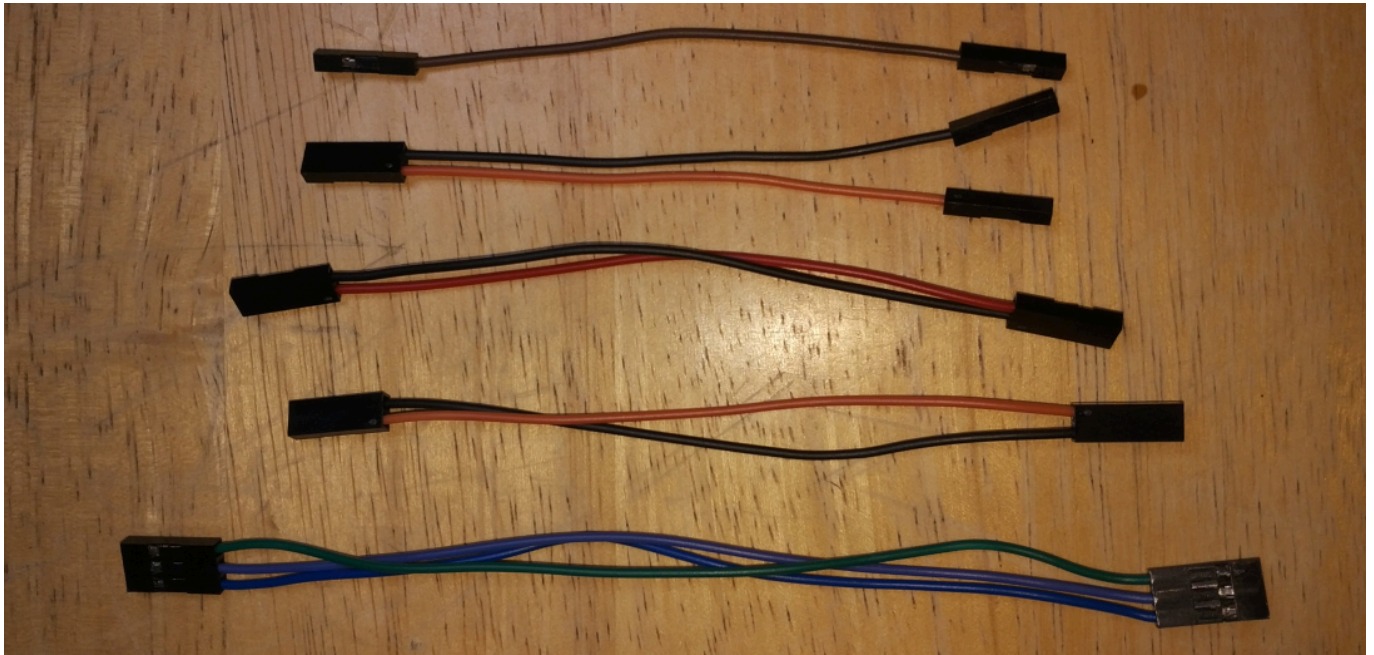
Couper les pattes de la LED (n'oubliez pas de marquer la patte rouge !) Souder ensuite 4 fils de 9 cm sur les pattes et isoler avec de la gaine thermorétractable. De l'autre côté sertir les câbles dans le même ordre sur un connecteur Dupont femelle à 4 plots.



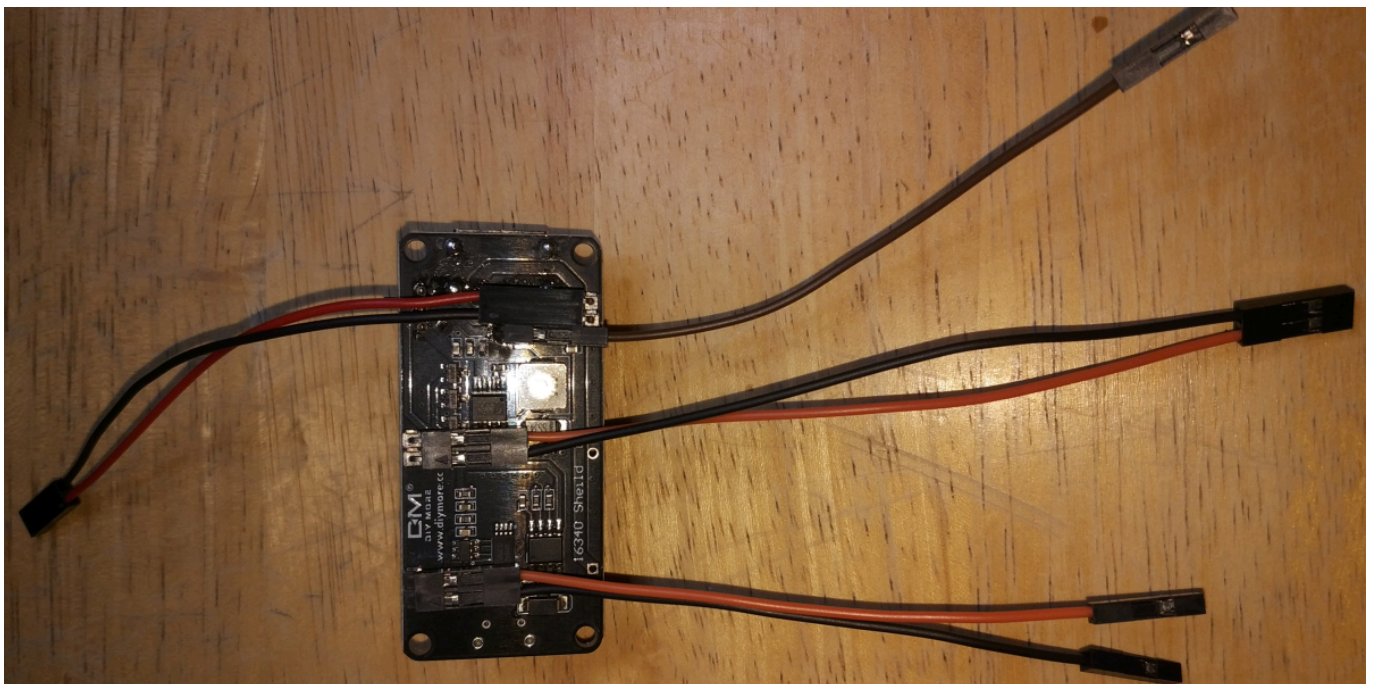
Mettre en place la LED après avoir donné forme à la tresse de câbles: celle-ci doit passer en dessous de l'axe de l'encodeur rotatif et remonter au fond du boîtier.



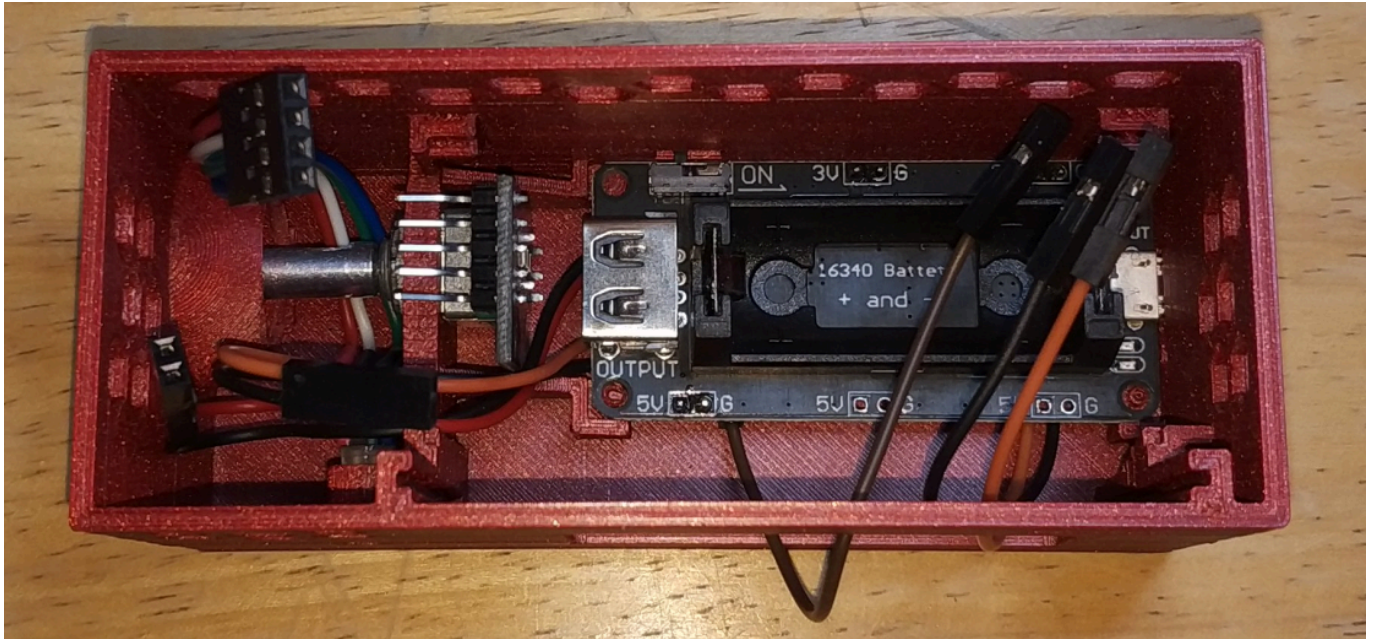
Préparer l'encodeur rotatif: tordre la petite patte afin qu'elle ne gêne pas et redresser les broches du connecteur Dupont.



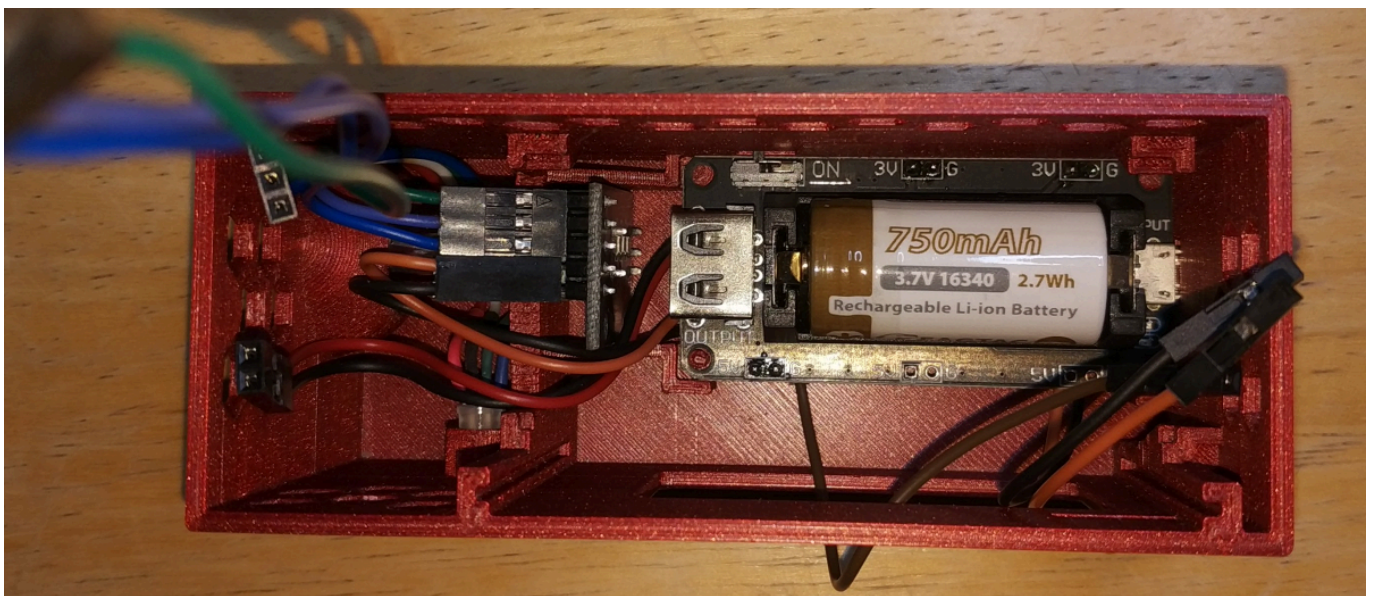
Il faut maintenant préparer tous les câbles de connexion. De haut en bas: marron, noir et orange de 9 cm pour le PCB; rouge et noir de 11 cm pour le S8; noir et orange de 11 cm pour l'alimentation du KY-040; vert, violet et bleu de 14 cm pour les données du KY-040.



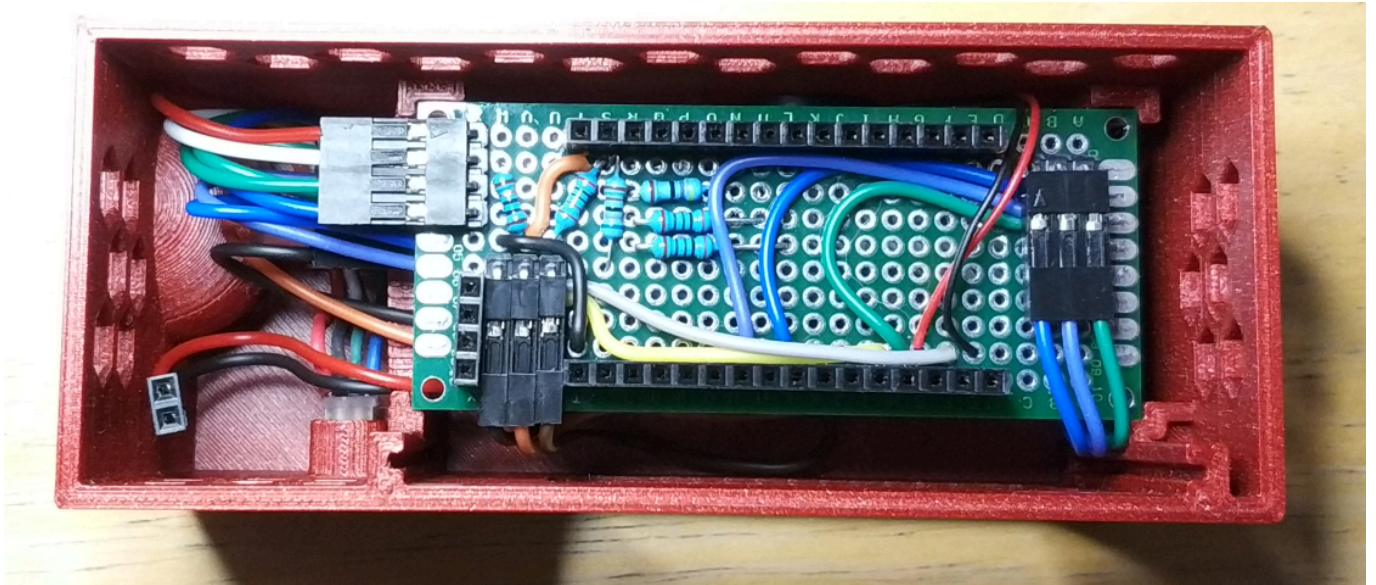
Brancher les câbles sur le circuit de charge en faisant bien attention à la polarité !



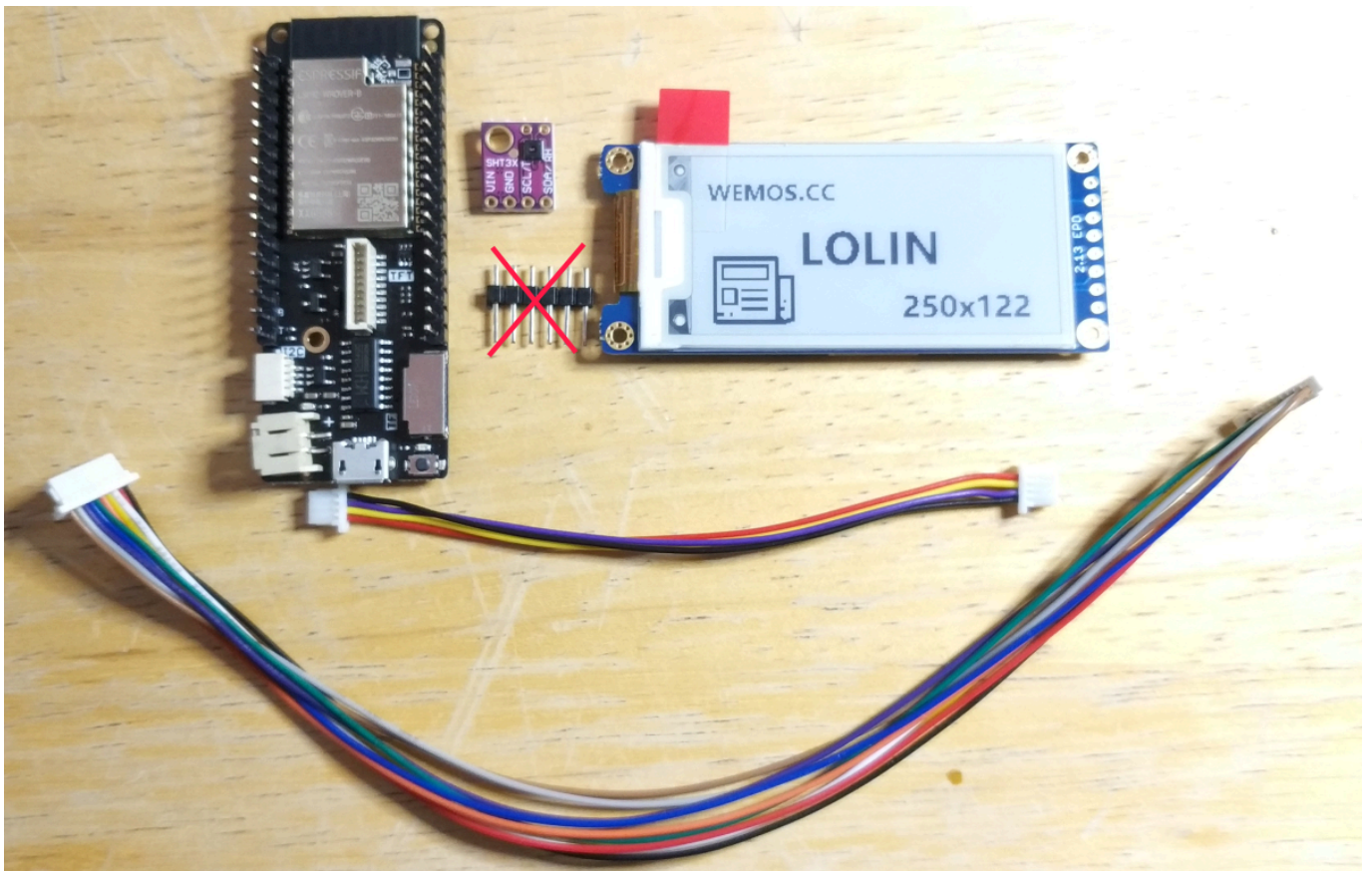
Installer le circuit de charge en l'insérant à 45° pour passer l'interrupteur au fond. Une fois en place il doit se clipser en forçant légèrement. Mettre en place les câbles en attente.



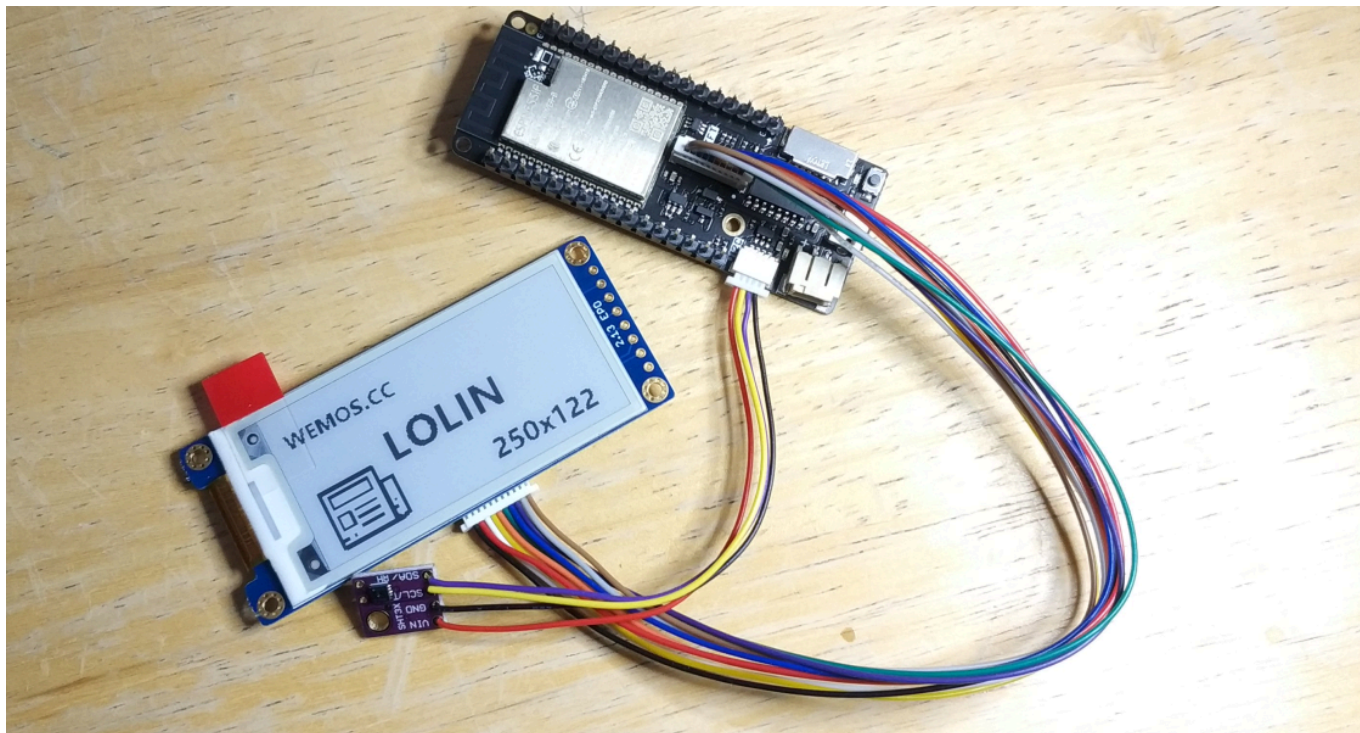
Brancher sur le KY-040 les câbles de données et d'alimentation.



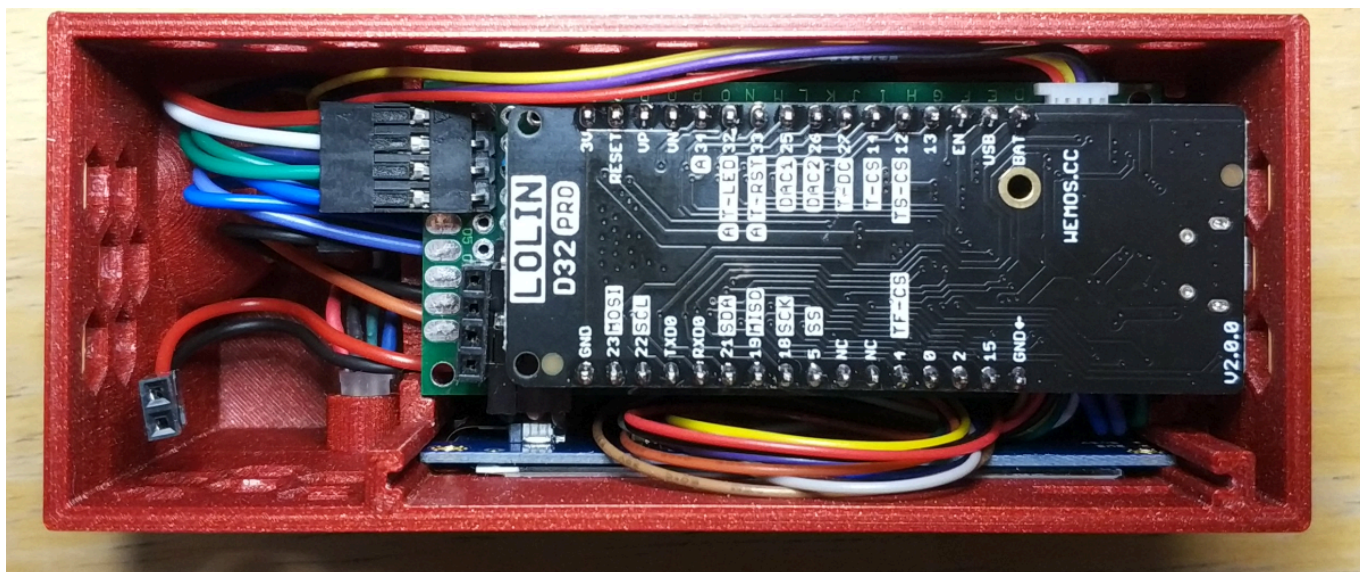
Installer le PCB dans les encoches prévues à cet effet en ayant préalablement passé les câbles de données du KY-040 en dessous afin de les brancher sur le PCB. Brancher également les câbles de la LED et les câbles d'alimentation.



Rassembler les éléments suivants: la carte ESP32, l'écran, le capteur SHT31D et les câbles. Souder les en-têtes Dupont sur la carte ESP32 «à l'envers» comme sur la photo.



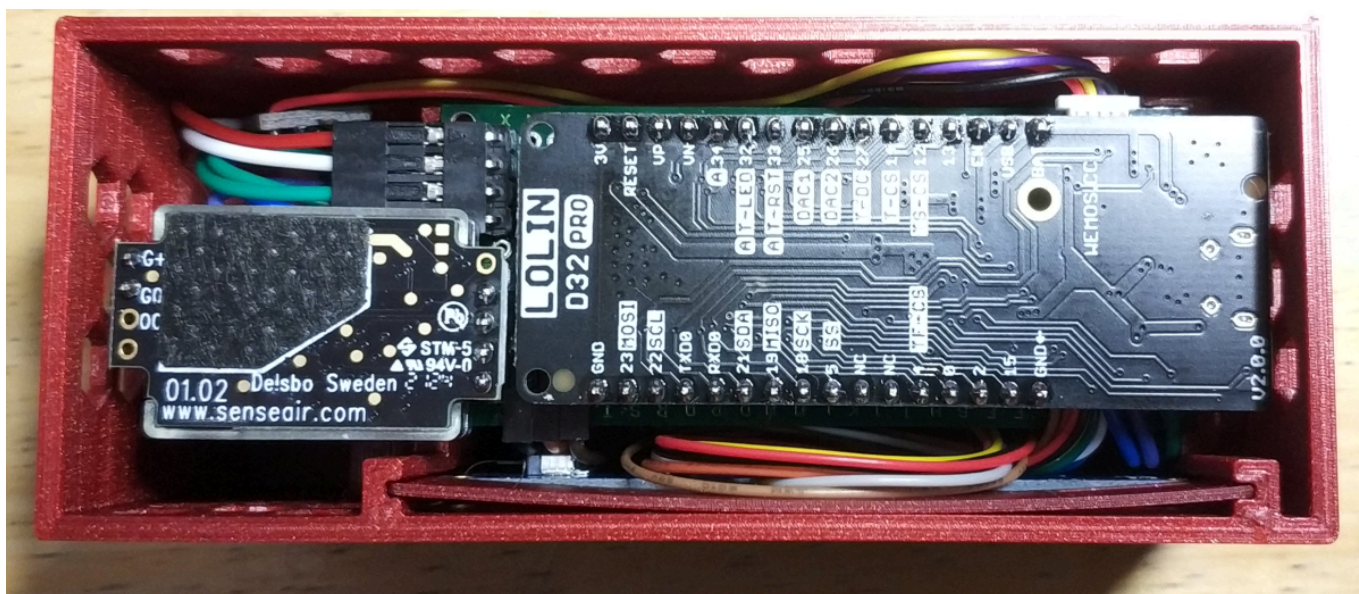
Sectionner un des connecteurs du câble i2c et souder les fils sur le capteur SHT31D comme indiqué. Brancher l'écran et le capteur sur l'ESP32.



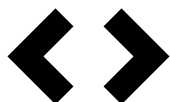
Installer la carte ESP32 à l'envers en faisant très attention au passage du câble de l'écran. Caler le capteur SHT31D et son câble au mieux.



ATTENTION à ne jamais toucher la partie capteur indiquée par la flèche rouge sur le S8 ! Souder les 2 en-têtes pour l'alimentation d'un côté et les 4 autres pour les données.



Brancher l'alimentation sur le S8 (attention à la polarité il n'y a pas de protection contre l'inversion !) puis insérer le S8 à son emplacement 4 broches. Glisser ensuite la cale pour l'écran dans le rail.



Reste à coller les deux éléments du bouton de l'encodeur rotatif: utiliser de la colle cyanoacrylate (attention aux doigts et aux yeux !). Une fois la colle sèche, insérer le bouton sur l'axe de l'encodeur rotatif. Le montage est maintenant terminé et vous pouvez passer à l'étape de la programmation. Pour cela, vous pouvez surélever la carte PCB pour atteindre le port μ USB de l'ESP32 mais avant lisez bien la remarque au début du chapitre *Programmation*.

Le couvercle s'installe de la même façon que le circuit de charge dans le boîtier: à 45° en calant les 2 plots dans les nids d'abeille à l'arrière (attention au câble i²c) puis en rabaisant l'avant. Il faut

légèrement arrondir le couvercle pour pouvoir passer les 2 plots latéraux à l'avant. Pour le démontage il suffit de presser les deux côtés du couvercle pour l'arrondir et déclipser les 2 plots latéraux à l'avant.

Programmation

Il ne faut jamais faire fonctionner le montage en alimentant la carte D32 simultanément par l'USB et le circuit de charge 16340 ! Sinon vous allez voir fumer votre carte au bout de quelques temps, je pense que le régulateur ne tient pas le coup sur la durée ! C'est pour cette raison que le port μ USB de la carte D32 n'est pas accessible sans ouvrir le boîtier.

Le plus simple pour programmer la carte est d'éteindre le circuit de charge 16340 lorsque le port μ USB de la carte D32 est branché. Dans ce cas le circuit n'est pas utilisable car ni le capteur S8 ni l'encodeur rotatif ne sont alimentés. Si vous voulez vraiment alimenter tout le circuit et en même temps programmer la carte alors vous **DEVEZ** débrancher le connecteur Dupont 3.3V sur la carte PCB (broche du milieu sur les 3) mais conserver la masse (GND) pour que le circuit fonctionne correctement.

La programmation s'effectue depuis l'environnement Arduino avec la bibliothèques Espressif pour l'ESP32. Vous aurez également besoin de bibliothèques supplémentaires comme indiqué dans l'en-tête du fichier:

- *Tone32*: v1.0.0 (<http://github.com/lbernstone/Tone32>)
- *GxEPD2*: v1.3.8 (<https://github.com/ZinggJM/GxEPD2>)
- *Adafruit SHT31*: v2.0.0 (https://github.com/adafruit/Adafruit_SHT31)
- *Adafruit GFX* v1.10.4 (<https://github.com/adafruit/Adafruit-GFX-Library>)

Téléchargement

L'archive ZIP suivante [co2-v3-2022-01-19.zip](#) contient tout le nécessaire pour fabriquer son propre détecteur et même l'améliorer:

- le fichier *co2-v3.ino* contient le code source pour programmer la carte ESP32
- le répertoire *3d* contient:
 - les fichiers *STL* prêts à l'emploi pour imprimer le boîtier
 - les fichiers *SCAD* pour modifier et adapter le boîtier à vos besoins ainsi que le fichier *Makefile* pour générer les fichiers *STL* plus facilement.

Le code fourni n'est pas vraiment propre et élégant à bien des endroits et mériterait un peu de nettoyage et réorganisation. J'ai cependant préféré le publier en l'état au cas où cela puisse être utile à certains. N'hésitez pas à me faire part de vos améliorations à l'adresse suivante: geek+co2@le-17.net !

Mises à jour

v3.3 (23 août 2023)

- Mise à jour de plusieurs bibliothèques et du Espressif arduino-esp32 core en v2.0.11.
- Suppression de *Tone32* puisqu'inclus dans la version 2.0 de Espressif arduino-esp32 core.
- Utilisation de *S8_UART* pour gérer le S8.
- Correction de bugs et amélioration du menu (bug d'affichage, ajout du scrolling, sélection immédiate pour les fonctions de sortie, calibration, etc.).
- Ajout d'une option de gestion de la LED: toujours éteinte, clignotante toutes les 8 secondes à la lecture du S8 ou tout le temps allumée (dans ce cas pas de sleep pendant les 8 secondes entre deux mesures!).
- Ajout d'une option de luminosité pour la LED (3 niveaux).

Fichier source (à renommer en *.ino*): [co2_v3.3.ino.txt](#)

Utilisation du détecteur

L'utilisation reste relativement simple. Pour l'installation du boîtier et les concepts de base on pourra se référer aux sites *Projet CO2* (<http://projetco2.fr/>) et *Nous aérons* (<https://nousaerons.fr/>).

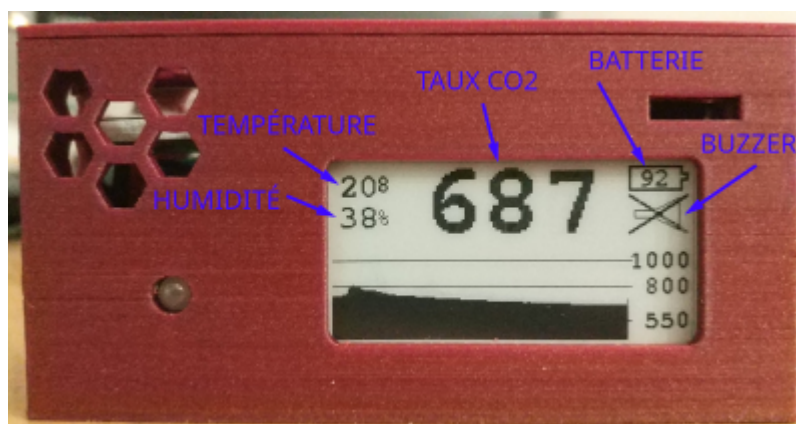
Mise en route du détecteur

Pour allumer le boîtier utiliser l'interrupteur coulissant à l'arrière. Au démarrage le boîtier s'initialise:

- LED blanche
- LED verte: initialisation de l'écran
- LED bleue: initialisation du capteur CO₂ (S8)
- LED orange: initialisation du capteur de température et d'humidité (SHT31D)
- LED rouge: initialisations internes

Affichage

Au bout de quelques secondes l'écran affiche les informations mesurées:



- en haut à gauche: la température (en °C)
- en dessous: le taux d'humidité (en %)
- au centre: le taux de CO₂ mesuré (en PPM)
- en haut à droite: la charge de la batterie (en %)

- l'état du Buzzer (désactivé lorsqu'il est barré)
- en partie basse: l'historique avec les différents seuils réglés

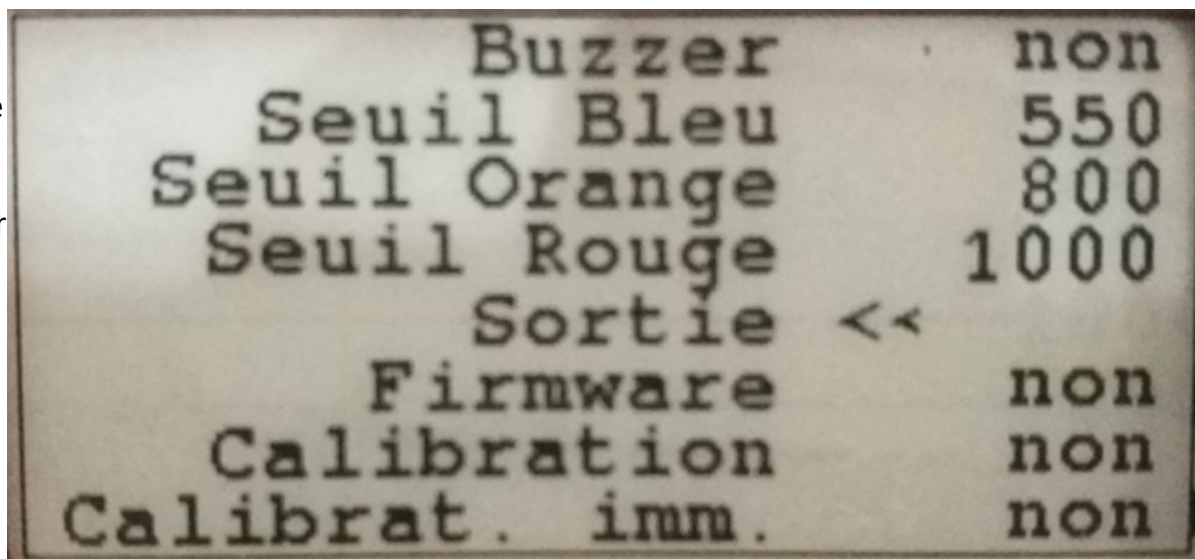
Le capteur CO₂ a une certaine inertie et la mesure des variations n'est pas immédiate. Afin d'augmenter l'autonomie le taux de CO₂ n'est mesuré que toutes les 8 secondes. À chaque période l'affichage est mis à jour, la LED clignote suivant le taux mesuré et le buzzer émet une «mélodie» si besoin.

Les seuils de taux de CO₂ sont réglés comme suit par défaut:

Taux de CO ₂	Couleur LED	Buzzer	Description
< 550	verte	non	Qualité de l'air excellente <i>Limite acceptable sans masque (par exemple cantine ou restaurant)</i>
550 à 799	bleue	non	Qualité de l'air correcte (avec masque)
800 à 999	orange	oui	Qualité de l'air mauvaise : il faut aérer immédiatement
> 1000	rouge	oui	Qualité très mauvaise : il faut aérer immédiatement en grand et sortir !

Réglages

À tout moment il est possible de désactiver ou réactiver le buzzer en appuyant sur la bouton de l'encodeur rotatif.



Pour accéder au menu de configuration il faut tourner l'encodeur rotatif pendant le clignotement de la LED ou l'affichage de l'écran. Le menu permet:

- d'activer ou désactiver le buzzer;
- choisir le mode de fonctionnement de la LED:
 - toujours éteinte
 - clignotante toutes les 8 secondes
 - toujours allumée (dans ce cas, le boîtier ne se mets pas en veille entre deux mesures toutes les 8 secondes et l'autonomie sur batterie est diminuée significativement)
- modifier la luminosité de la LED (3 niveaux)
- modifier les seuils de taux de CO₂
- afficher les informations avancées du capteur S8
- lancer le processus de calibration

La rotation dans un sens ou dans l'autre permet de choisir l'option à modifier. Un appui sur le bouton

sélectionne l'option. La rotation permet ensuite de choisir une valeur et un appui sur le bouton valide et retourne sur le choix des options.

Les réglages sont enregistrés dans l'EEPROM de l'ESP32 et sont donc préservés lors des cycles d'arrêt et d'allumage.

Calibration

La calibration n'est à faire qu'en cas de constat d'une dérive ou d'une anomalie des mesures. Le processus est simple: le capteur utilise la mesure de l'air extérieur pour se calibrer à la valeur de 400 ppm. Pour cela il est nécessaire de positionner le détecteur à l'extérieur et à l'abri de la pluie environ 20 minutes avant et après la calibration afin de stabiliser le processus.

Le mode *Calibration* est automatisé: une fois sélectionné, placer le boîtier à l'extérieur et attendre la fin du processus. Une barre de défilement indique l'avancée: environ 20 minutes d'attente, calibration et encore 20 minutes d'attente. Une fois le processus terminé le capteur indiquera *CALIB: OK* pendant encore une vingtaine de minutes avant de disparaître. Il n'est pas nécessaire d'attendre avant de rentrer le détecteur.

Le mode *Calibrat. imm.* exécute la calibration immédiatement sans temporisation. C'est à vous de placer le détecteur à l'extérieur le temps requis.

From:
<https://le-17.net/dokuwiki/> - **Le 17**

Permanent link:
<https://le-17.net/dokuwiki/geek/co2/v3>

Last update: **2024/03/23 17:41**

