

interrupteur sans contact DIY



Mathias Claußen (Elektor)

La pandémie m'a donné l'idée de réaliser un interrupteur qui interprète des gestes de la main. L'absence de contact physique réduit en effet la transmission de virus et bactéries d'une personne à l'autre. C'est un atout pour les espaces de travail très fréquentés. Cet article décrit le principe de cette idée et sa mise en œuvre sur un ESP32 (avec carte d'extension) qui évalue les gestes de la main.

Début 2020, lors de la 1^{ère} vague de COVID-19, pour contenir la propagation, des mesures furent prises dans les bureaux. Nettoyage et désinfection des surfaces devinrent le rituel quotidien. S'il est facile d'essuyer machines à café et bouilloires avant et après utilisation, s'assurer que d'autres surfaces, par ex. les interrupteurs

des bureaux paysagers et des couloirs, ne puissent devenir des sites de prolifération du virus est plus difficile.

En y réfléchissant, j'ai réalisé que le problème serait résolu à l'aide d'interrupteurs ne nécessitant aucun contact physique. Dans la

Avertissement

Tous les circuits présentés dans cet article fonctionnent sur le secteur. Ni les circuits ni les circuits imprimés présentés ici n'ont été testés du point de vue fonctionnalité et sécurité. Il est déconseillé de fabriquer ou utiliser les CI sans les faire vérifier au préalable. Pour réaliser un projet à partir de cet article, il est vital d'être conscient des dangers. Vous travaillerez entièrement à vos risques et périls. Seuls des ingénieurs ou des concepteurs qualifiés peuvent le faire !

boutique d'Elektor [1], j'ai trouvé le HAT 3D Gesture & Tracking de Seeed Studio pour Raspberry Pi (fig. 1).

Le MGC3130

Début 2019, James Rowley et Mark Omo ont présenté un interrupteur d'éclairage sans contact *The Open Smart Switch* [2] sur la plateforme *hackster.io*. Ce projet utilise le MGC3130 de Microchip Technology. Le HAT 3D Gesture & Tracking pour Raspberry Pi envisagé utilise la même puce. La chaîne *ElektorTV* [3] a abordé en détail les caractéristiques de cette puce.



Le MGC3130 capte le champ électrique. La puce envoie un signal basé sur la mesure du champ électrique produit à la surface d'une tablette tactile. Cinq électrodes disposées autour de la tablette servent à détecter le champ et tout changement subtil de ce dernier quand un objet comme une main s'approche de la tablette et le perturbe. La figure 2 montre le principe de fonctionnement du capteur et le champ produit. Le doigt peut être reconnu et sa position calculée. En outre des gestes 3D, par ex. « roue à air » ou « pichenette », peuvent être évalués dans les quatre directions. Une pichenette consiste à passer devant le capteur avec un doigt en l'air. La puce peut être calibrée et configurée à l'aide des outils fournis par Microchip. La page [4] fournit toutes les informations nécessaires pour utiliser la puce dans une réalisation personnelle.

Elle communique avec le monde extérieur par une interface I²C et nécessite les deux lignes usuelles (données + horloge) plus deux autres (réinitialisation et interruption bidirectionnelle). Le document *DS40001718E* [5] de Microchip fournit les détails de l'interface. Pour cette puce, on peut concevoir un pad personnalisé, mais il faut noter que la carte devra comporter au moins quatre couches. Et dans ce cas, il sera nécessaire de paramétrer et configurer la puce pour correspondre au nouveau dessin du pad.

Quel microcontrôleur ?

En ce qui concerne le microcontrôleur, nous avons vraiment l'embarras du choix de nos jours. Pour n'en citer que quelques-uns, cela va du Raspberry Pi Pico à l'ESP8266 ou à l'ESP32, en passant par le STM32 Blue Pill et l'éprouvé ATmega328. L'application ne nécessite pas de calculs compliqués et ne produit qu'un seul signal pour commuter le relais.

Il nous faut aussi réfléchir au mode de fonctionnement de l'interrupteur : doit-il être autonome pour allumer une seule lampe ou intégré à une installation domotique existante ? La 2^e option,



Figure 1. Le 3D Gesture & Tracking Shield (carte d'extension pour suivi et gestuelle 3D).

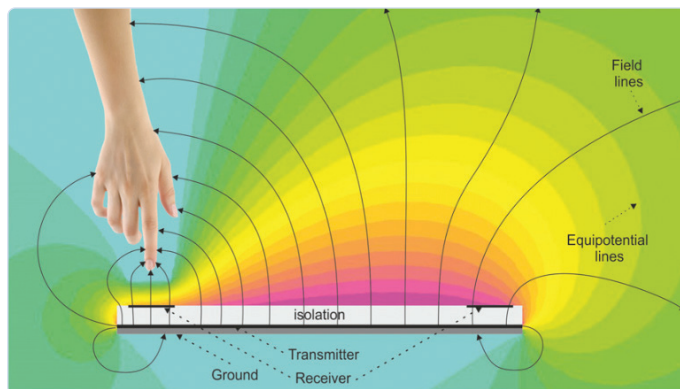


Figure 2. Le champ du capteur MGC3130. (Source : Microchip)

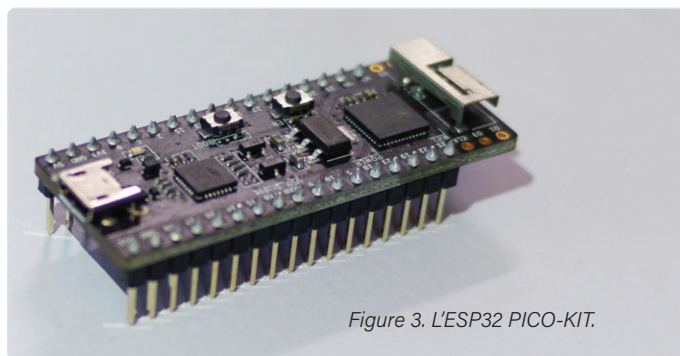


Figure 3. L'ESP32 PICO-KIT.

l'intégration dans un système de commande tel qu'ESPHome, aurait l'avantage de centraliser la commande de la lampe ou de la charge. Il convient également de déterminer comment effectuer les mises à jour du micrologiciel sur le système. Avec l'ESP8266 ou l'ESP32, une liaison Wi-Fi permet ces mises à jour sans fil (en anglais OTA = Over The Air).

Nous avons opté pour une variante de l'ESP32 qui dispose du Wi-Fi, du Bluetooth, d'une mémoire Flash embarquée suffisante et des mises à jour via Wi-Fi : l'ESP32 PICO-KIT (photo fig. 3).



Figure 4. Boîtier à encastrer.
(Source : Würth [9])

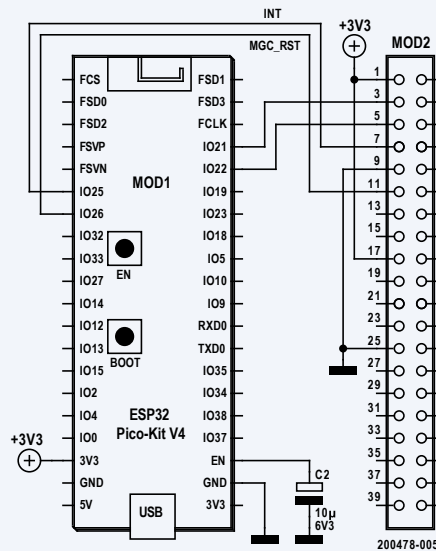


Figure 5.
Schéma du prototype.

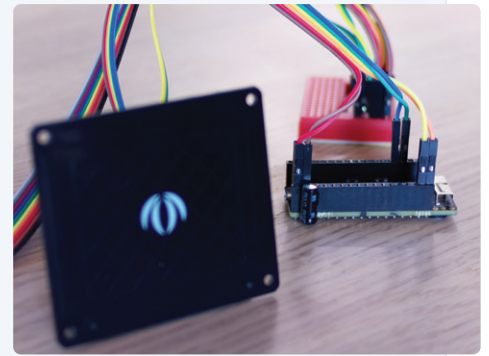


Figure 6. Interconnexion du tout.

Concept de base

Comme pour tout projet, il faut définir le concept de base et vérifier tous les composants potentiellement disponibles pour accomplir la tâche. Ici, nous possédons déjà un capteur, le 3D Gesture & Tracking Shield pour le Raspberry Pi et un ESP32, en version PICO-KIT. Il nous faut aussi un relais pour allumer et éteindre la lumière et enfin une alimentation électrique. Il suffit d'un micrologiciel et le tour sera joué...

Mais, tout n'est pas si simple. Souvent, en cours de développement, nous nous heurtons à des problèmes imprévus qui doivent être résolus. Le processus de développement peut être divisé en trois domaines qui concourent à élaborer le produit fini. Il y a d'abord le logiciel qui évalue les données des capteurs et déclenche les actions adéquates. Ensuite le matériel électronique et la carte sur laquelle les composants sont montés. Un schéma de connexion des composants est produit. Idéalement, le développeur du logiciel a déjà demandé un retour d'information sur l'affectation des broches et les exigences particulières. Il arrive souvent que des erreurs

de disposition du matériel puissent être corrigées par de petites modifications du logiciel, mais cela peut être une source d'ennui pour le développeur du logiciel.

Enfin, il y a la conception mécanique de l'ensemble complet et du boîtier qui doit tout accueillir. Souvent, la taille du boîtier n'est pas critique et nous pouvons en choisir (ou fabriquer) un de dimension adéquate. Cependant, dans certains cas, nous aurons besoin de tout faire entrer dans un boîtier standard, par ex. une boîte d'encastrement électrique ordinaire, pour l'intégrer dans l'installation électrique domestique. Pour ce projet, notre plan initial était de faire tenir l'ensemble dans un volume de 65 mm × 55 mm × 25 mm. Toutefois, cela ne correspond pas à certaines boîtes d'encastrement courantes (**fig. 4**) utilisées dans certains pays européens. Pour conférer un caractère plus universel à cette réalisation, il serait souhaitable que l'ensemble soit assez petit pour être monté dans ce type de boîtier également.

Mise en œuvre du concept initial

Le schéma de la **figure 5** montre la connexion initiale du 3D Gesture & Tracking Shield pour Raspberry Pi à l'ESP32. La carte est installée sur une plaque d'essai (**fig. 6**) pour tester sa fonctionnalité. Plusieurs bibliothèques sont disponibles pour le MGC3130. La bibliothèque utilisée dans ce projet provient de Seeed Studio [6], elle est prévue pour un Raspberry Pi.

Pour l'ESP32, l'interface I²C et l'affectation des broches d'E/S doivent être adaptées. Une fois ces modifications effectuées, les tests peuvent démarrer avec cette bibliothèque. La **figure 7** montre la suite des coordonnées positionnelles du mouvement pour un « geste de pichenette » reconnu.

Alimentation et relais

Lors de la conception et du test du prototype, nous ne nous sommes pas préoccupés de l'alimentation électrique ni du relais à inclure dans la réalisation finale. Le contrôleur final fonctionnera à partir de la tension du secteur. À cet effet, nous devons inclure une alimentation de 5 V ou 3,3 V suffisamment puissante pour l'ESP32, le 3D Gesture & Tracking Shield et le relais.

Les HLK-PM03 et HLK-PM01 de Hi-Link (**fig. 8**) sont parmi les modules d'alimentation les plus compacts. Ils fournissent une

```

26  mgx3130_rst_pin = resetPin;
27  xTaskCreate( Touchinput_Task, "MGC3130Drv", 4096, NULL, tsxIDLE_PRIORITY, &x
28  configASSERT( xHandle );
29  }

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
x: 48540, y: 65534, z: 34456
x: 47465, y: 65534, z: 34252
x: 46156, y: 65534, z: 33950
x: 44656, y: 65534, z: 33498
x: 43026, y: 65534, z: 32867
x: 41337, y: 65534, z: 32204
x: 39667, y: 65534, z: 31760
x: 38103, y: 65534, z: 31560
x: 36732, y: 65534, z: 31502
x: 35621, y: 65534, z: 31525
x: 34798, y: 65534, z: 31599
x: 34243, y: 65534, z: 31788
x: 33899, y: 65534, z: 31841
x: 33696, y: 65534, z: 31986
x: 33569, y: 65534, z: 32119
x: 33467, y: 65534, z: 32215
x: 33360, y: 65534, z: 32265
x: 33237, y: 65534, z: 32278
x: 33103, y: 65534, z: 32277
x: 32972, y: 65534, z: 32275
x: 32857, y: 65534, z: 32284
x: 32764, y: 65534, z: 32304
Gesture: FLICK_SOUTH_TO_NORTH, class: FLICK_GESTURE, edge flick: no, in progress: no
x: 32687, y: 65534, z: 32331
x: 32611, y: 65534, z: 32358

```

Figure 7. Sortie de données du capteur de gestes.

puissance de 3 W, soit 600 mA en 5 V ou environ 900 mA en 3,3 V. L'ESP32 nécessite environ 500 mA (2,5 W sous 5 V, 1,65 W sous 3,3 V). Nous avons choisi un relais Panasonic de type ADW12. Il nécessite une impulsion d'environ 67 mA en 3,3 V ou 40 mA pour la version 5 V (220 mW). Le 3D Gesture & Tracking Shield lui-même nécessite encore 20 mA en 3,3 V (66 mW). Tout compris, nous arrivons à un minimum de 2,8 W si l'ESP32 PICO-KIT est alimenté en 5 V. Cela nous rapproche de la puissance de sortie maximale de ce module 5 V.

L'ESP32 PICO-KIT consomme moins s'il est alimenté en 3,3 V, car le régulateur LDO (Low DropOut) intégré n'est plus nécessaire et ne sera pas alimenté. C'est pourquoi nous utiliserons finalement un module 3,3 V tel que le HLK-PM03. Il peut fournir 900 mA maximum. Le relais choisi est un Panasonic ADW1203HLW. Les contacts de ce relais unipolaire bistable acceptent 16 A sous 277 VAC max.. Il est assez compact : 24 mm × 10 mm × 16 mm. Il suffit d'une courte impulsion pour l'activer ou le désactiver et la bobine ne consomme aucun courant permanent.

Vers une version finale du schéma

Le schéma préliminaire du circuit est présenté à la **figure 9**. Nous n'avons pas encore inclus de protection autour du module d'alimentation HLK-PM03. Habituellement on installe, côté entrée CA, un fusible thermique et une varistance et, côté sortie, un fusible et des condensateurs de lissage de la tension. Sans ces mesures de protection du côté de l'entrée, une défaillance interne du module peut conduire à un incendie. En sortie, une régulation et un filtrage insuffisants de la tension peuvent réduire considérablement

Figure 8. Convertisseur CA/CC Hi-Link.



la portée Wi-Fi ou déclencher un crash logiciel. L'alimentation finale comprendra ces modifications.

Un condensateur de 10 µF placé entre les broches GND et EN de l'ESP32 évite tout défaut de synchronisation qui empêcherait de passer en mode de programmation à la réinitialisation. C'est également utile si, à l'occasion, l'ESP32 refuse d'être programmé depuis le port micro-USB.

Comme déjà indiqué ci-dessus, la carte ESP32 PICO-KIT comprend un régulateur AM1117-3.3 censé fournir 3,3 V à l'ESP32 à partir de 5 V. Ce régulateur supporte 3,3 V sur sa broche de sortie s'il n'est pas alimenté. Cependant, dans notre schéma du circuit, une diode Schottky (D1) est connectée entre l'alimentation 3,3 V et l'alimentation 5 V afin qu'un régulateur, monté le cas échéant sur un autre type de carte, ne puisse pas être endommagé. En l'incluant, il circule un courant de l'alimentation 3,3 V à travers le régulateur. Le courant minimal de commande du relais 3,3 V est de 67 mA. C'est beaucoup plus que ce que les broches de l'ESP32 peuvent fournir. Un étage de commande à transistors (T1 et T2) est donc utilisé. Les diodes de roue libre D2 et D3 sont essentielles pour éviter de détruire les transistors par les pics de tension inverse à la coupure de la bobine du relais.

Le 3D Gesture & Tracking Shield est connecté à l'ESP32 via le

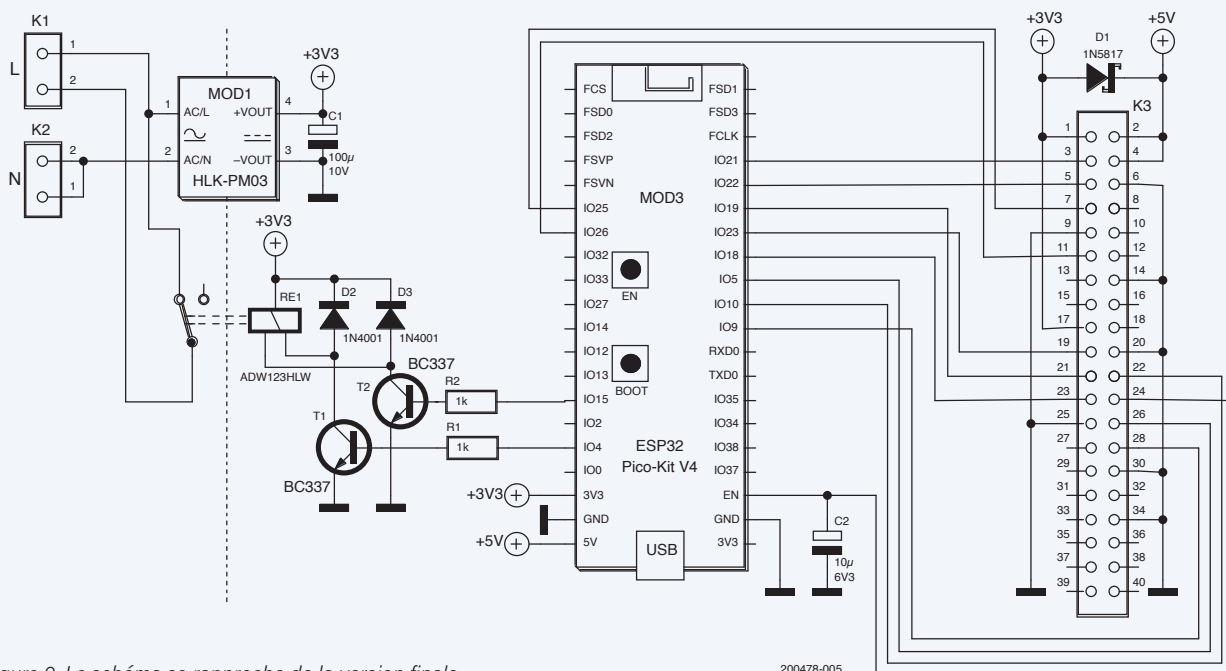


Figure 9. Le schéma se rapproche de la version finale.

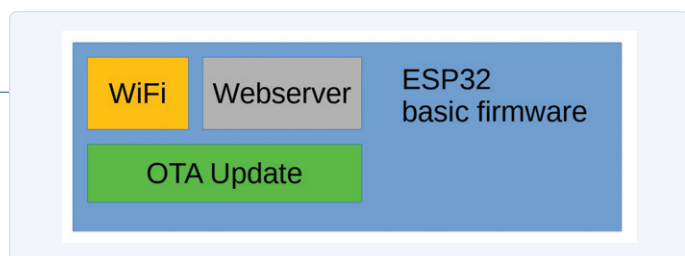


Figure 10. Blocs de base du microprogramme.

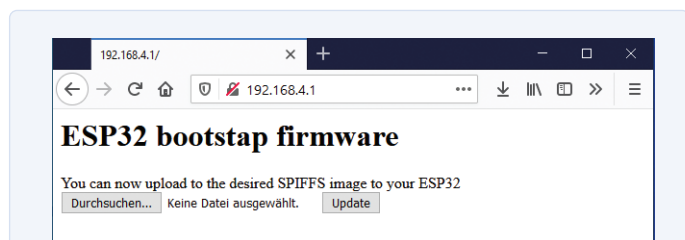


Figure 11. Téléchargement de l'image SPIFF via un navigateur web.

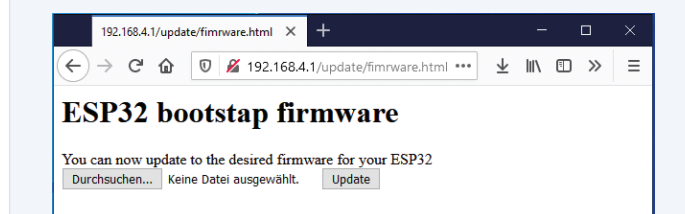


Figure 12. Mise à jour du micrologiciel OTA via un navigateur web.

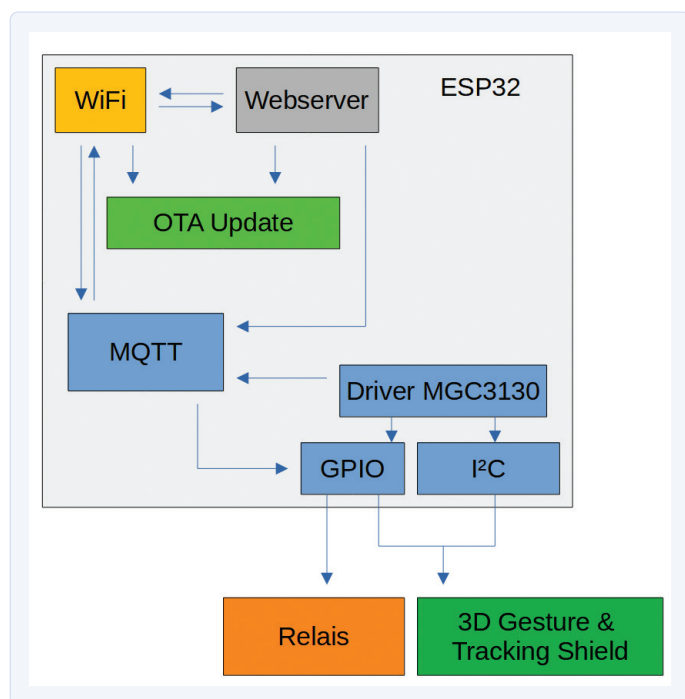


Figure 13. Liaisons du module de micrologiciel.

connecteur à deux rangées de broches K3. Les 3,3 V, les signaux I²C et les deux broches d'E/S sont les seules connexions requises pour le HAT. Il comprend déjà les résistances de rappel au positif du bus I²C. La carte a été conçue pour fournir des signaux SPI et autres broches d'E/S nécessaires pour l'ajout d'un moniteur TFT de 3,5 pouces.

La flexibilité de la matrice d'E/S de l'ESP32 facilite la connexion des périphériques. Pour le 3D Gesture & Tracking Shield, les broches 21 et 22 sont affectées au bus I²C, la broche 25 à l'interruption et la broche 26 à la réinitialisation du MGC3130. S'il n'est pas nécessaire d'intégrer un écran TFT, nous pourrions utiliser une variante ESP32 nettement plus petite ou même un module ESP32-C3.

Micrologiciel

Après avoir affecté les broches de l'ESP32, il faut compiler le micrologiciel pour l'utiliser. Les routines de la bibliothèque permettent de créer un serveur web qui produit deux pages web de base et d'effectuer des mises à jour sans fil du micrologiciel. La **figure 10** donne les modules du microprogramme de base. Cette approche évite de réécrire et de déboguer certaines des fonctions de base utilisées dans de nombreuses applications similaires.

Le microprogramme produit deux pages web de base. L'une sert à écrire une nouvelle image SPIFFS (Serial Peripheral Interface Flash File System) dans l'ESP32 (**fig. 11**) et l'autre à fournir une mise à jour du micrologiciel de l'ESP32 (**fig. 12**).

Aujourd'hui, le micrologiciel n'exécute que les fonctions de commutation de base. Un autre composant permet d'évaluer les informations du MGC3130. Une page web minimale permet de configurer la messagerie MQTT (Message Queuing Telemetry Transport) et une autre d'activer/désactiver la charge via un navigateur web. La **figure 13** illustre l'interaction des modules. La commande de l'interrupteur d'éclairage est donc logicielle ou manuelle.

Casse-tête...

Le micrologiciel et le schéma du circuit étant prêts, il faut s'assurer que tout le matériel tient dans l'espace disponible. En matière d'outils de conception de circuit imprimé (CI), chacun a ses préférences et utilise celui qui lui est familier. De mon côté, je préfère KiCad qui crée des schémas et des mises en page éditables librement sans grand investissement financier. Elektor a d'ailleurs publié le livre *KiCad Like a Pro* pour KiCad [7].

La **figure 14** illustre le CI réalisé et, en effet, il n'est pas très joli. Des découpes dans la carte assurent l'isolement nécessaire entre les zones sous tension du secteur et l'alimentation 3,3 V. Tout cuivre superflu à l'interconnexion des composants est retiré. La **figure 15** montre le routage obtenu. Les découpes et les fentes sur le pourtour de la carte sont réalisées à la fraiseuse, on ne peut donc pas s'attendre à des angles vifs à 90°. Si vous souhaitez faire fabriquer la carte, vérifiez les caractéristiques du fraisage pour avoir des angles internes de rayon minimal.

Des modèles 3D existent pour presque tous les composants. Ils serviront à étudier l'implantation des composants et à utiliser

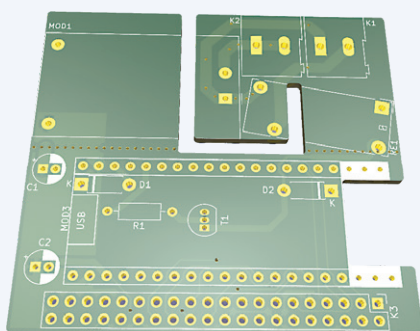


Figure 14. Circuit imprimé terminé.

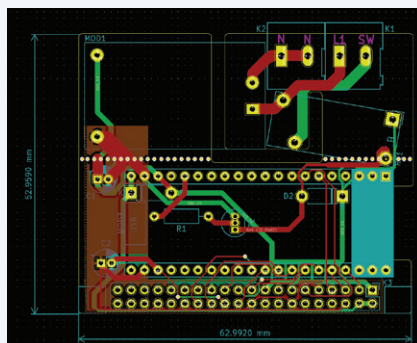


Figure 15. Schéma d'implantation terminé.

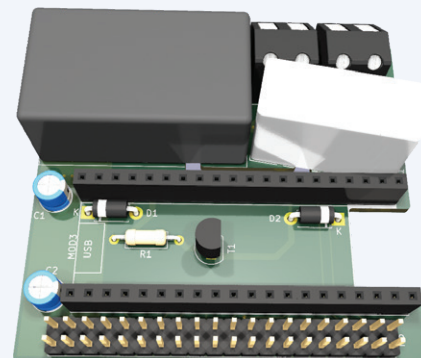


Figure 16. Disposition des composants sur le circuit imprimé.

au mieux l'espace disponible. Pour tout faire tenir, vous serez peut-être amené à utiliser des cartes filles, perpendiculaires à la carte principale. La **figure 16** montre une vue 3D de notre CI.

Chausse-pieds SVP !

KiCad a l'avantage de pouvoir exporter un fichier STEP utilisable dans un programme de conception 3D. Dans FreeCAD, nous chargeons et plaçons correctement le modèle de notre CI et son boîtier (impression 3D de la **figure 17**). Sur la **figure 18**, on voit que le CI ne tient pas ; son contour déborde de l'espace disponible. Un conflit apparaît entre les composants et les surfaces internes du boîtier et le bord de la carte (en vert) dépasse à l'avant. Est-ce une erreur sur les modèles 3D, un problème de mise à l'échelle, ou bien une simple erreur de mesure ? La dernière hypothèse est la bonne.

Après avoir revu les dimensions de la carte, tout semble un peu à l'étroit sur le CI. La vue de côté des composants de la **figure 19**

montre qu'il n'y a pas assez de place pour tous ceux-ci. Il faut garder une distance sûre entre pistes, il n'est donc pas possible de rapprocher les composants. La **figure 20** montre qu'une fois tout assemblé sous FreeCAD, le contour du module d'alimentation déborde encore sur un pilier d'angle. Il nous faut revenir en arrière et revoir la conception avec d'autres composants.

Et maintenant ?

Du point de vue matériel et micrologiciel, la conception de base de l'interrupteur sans contact fonctionne comme prévu. Pour que l'unité puisse être utilisée avec une autre gamme de normes électriques, la carte doit aussi pouvoir tenir dans un boîtier circulaire. La figure 4 montre une épure de boîtier pour interrupteurs et prises à encastrer dans des cloisons de plaques de plâtre ; ce modèle est courant dans certains pays européens. Pour résoudre cet écueil, il faudra étudier la compacité du module d'alimentation et la taille des modules ESP32 du marché pour

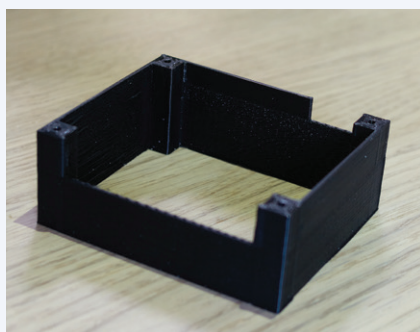


Figure 17. Le boîtier imprimé en 3D.

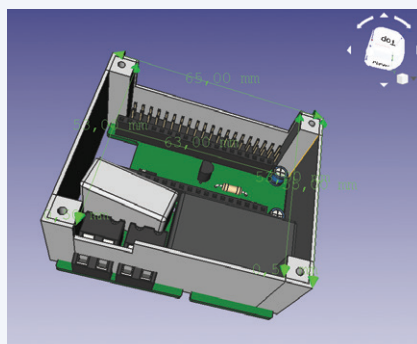


Figure 18. Le circuit imprimé ne rentre pas.

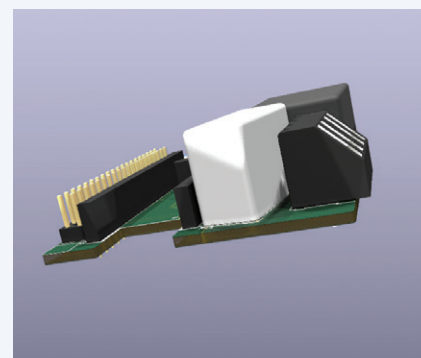


Figure 19. Vue de côté montrant les composants placés les uns contre les autres.

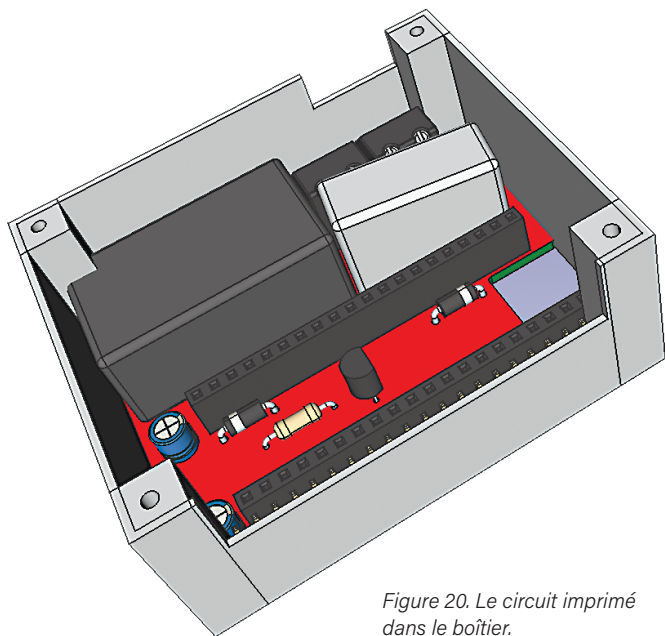


Figure 20. Le circuit imprimé dans le boîtier.

prendre moins de place que le PICO-KIT ESP32.

Avec un peu de chance, des modules ESP32-C3 devraient arriver sous peu sur la paillasse du laboratoire. Plus compacts, ces modules sont moins chers, moins énergivores et équipés de Wi-Fi & Bluetooth. Vu leur faible consommation, un module d'alimentation plus petit devrait suffire et faire gagner un peu plus de place.

À ce stade de développement, nous avons décidé de mettre ce projet en veille. L'expertise de nos lecteurs est la bienvenue ainsi que toute suggestion sur des composants alternatifs plus compacts ou l'optimisation de la disposition et du routage pour remplir l'espace disponible. Ce projet n'est pas achevé et nous pensons que les lecteurs trouveront bien des erreurs dans le circuit actuel. Des conseils et des astuces sur l'interopérabilité KiCad/FreeCAD seraient également les bienvenus. Si vous souhaitez jeter un coup d'œil au code source, au schéma, au circuit imprimé ou aux données du boîtier, tout est téléchargeable depuis le dépôt GitHub d'Elektor [8].

200478-04

Contributeurs

Texte et images : Mathias Claussen

Rédaction : Jens Nickel, C. J. Abate

Traduction : Yves Georges

Mise en page : Giel Dols

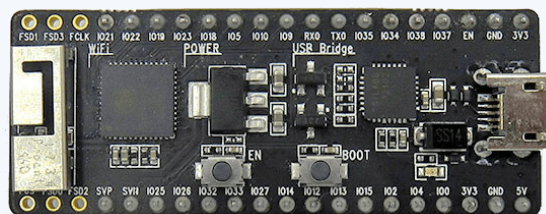
Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mathias.claussen@elektor.com) ou contactez Elektor (redaction@elektor.fr).



PRODUITS

> **ESP32-PICO-Kit V4**
www.elektor.fr/18423



> **Modèle 3445 de PeakTech ,**
multimètre numérique, mesure de la valeur efficace
vraie, avec Bluetooth (6000 points)
www.elektor.fr/18774

> **WT 1013, station de soudage numérique (95 W)**
de Weller
www.elektor.fr/19338

LIENS

- [1] Boutique d'Elektor : www.elektor.fr
- [2] Interrupteur intelligent « open source » : www.hackster.io/133854/open-smart-switch-44fa54#toc-future-developments-8
- [3] Vidéo Youtube de démonstration du Gesture Pad : www.youtube.com/watch?v=WVSVhEeMi_4
- [4] Page produit du MGC3130 : www.microchip.com/wwwproducts/en/MGC3130
- [5] Fiche technique du MGC3130 : <https://ww1.microchip.com/downloads/en/DeviceDoc/40001718E.pdf>
- [6] Bibliothèque Seeed Studio MGC3130 : https://github.com/Seeed-Studio/Seeed_Linux_mgc3x30
- [7] Peter Dalmaris, « KiCAD like a Pro », Elektor : www.elektor.fr/kicad-like-a-pro
- [8] Dépôt Github : <https://github.com/ElektorLabs/200478-Touchless-Lightswitch>
- [9] Boîte d'encastrement pour mur creux de Würth :
<https://eshop.wurth.fr/Categories-produits/Boite-d-encastrement-pour-mur-creux/310755020304.ciid/3107.cgid/fr/FR/EUR/>