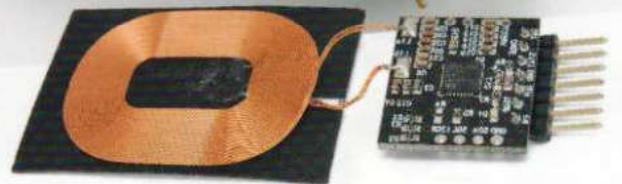


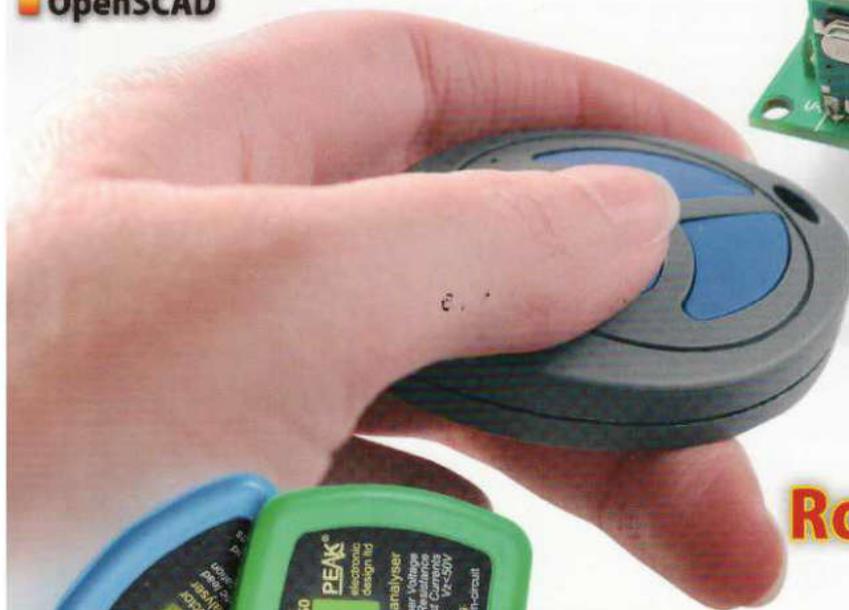
n° 146 PRINTEMPS 2019

- Analyseur de semi-conducteur
- Breakout Board AOP
- Chargeur sans fil
- OpenSCAD

Décodeur universel de télécommandes



Roulette électronique



- CAO kiCad EDA - 6
- Cours MPLAB X - 6
- VoIP sur Raspberry - 2
- 3DRAG+ : Imprimez en grand ! - 2



N° 146 Mars 2019

L 13270 - 146 - F: 8,30 € - RD



Sommaire

ARTICLES

Numéro 146
Printemps 2019
100 pages



04 ALIMENTATION

DÉCOUVRIR ET CONCEVOIR UN CHARGEUR SANS FIL

La charge sans fil est la dernière frontière technologique des appareils mobiles, elle joue un rôle fondamental dans l'industrie. Grâce à cette technologie, nous vous proposons de concevoir un système d'alimentation sans fil pour vos projets. La possibilité de transférer de l'énergie électrique au moyen d'ondes électromagnétiques a été démontrée pour la première fois en 1890 par Nikola Tesla.



14 LOISIRS

ROULETTE ÉLECTRONIQUE À LED

Nous mettons au goût du jour une version électronique de l'un des jeux de casino les plus célèbres, sans trucage et sans déception. Le mot « roulette » est évidemment français, mais beaucoup disent que le jeu a été inventé en Italie sous le nom de « girilla ». Quelle que soit son origine, nous pouvons affirmer en toute certitude qu'il s'agit de l'un des jeux de casino les plus célèbres et des plus pratiqués dans le monde.



24 IMPRIMANTE 3D

LA 3DRAG+ IMPRIMEZ EN GRAND ! DEUXIÈME PARTIE

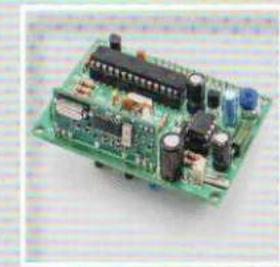
Après avoir abordé, dans le précédent numéro, le montage de la 3DRAG+ qui est capable d'imprimer des objets de dimensions 40 cm x 40 cm x 40 cm, nous allons compléter notre imprimante d'un plateau chauffant et d'une carte de contrôle autonome, puis nous décrirons la configuration logicielle de l'imprimante. Seconde et dernière partie.



42 TÉLÉCOMMANDE

DECODEUR UNIVERSEL POUR TÉLÉCOMMANDES RADIO

Ce montage permet de vérifier l'intégrité et la programmation correcte des télécommandes couramment utilisées pour les ouvre-portes, les portails, etc. Notre projet s'adresse à ceux qui utilisent souvent des télécommandes radio codées. Sa fonction est d'analyser un signal et de le décoder, en affichant la séquence relative des bits sur l'afficheur, jusqu'à un maximum de 32.



50 LABORATOIRE

ANALYSEUR DE SEMI-CONDUCTEUR

Nous vous présentons dans cet article un ensemble d'instruments de laboratoire vous permettant d'identifier, de vérifier les caractéristiques et de tester les transistors BJT, les transistors FET, les thyristors, les diodes, etc., à un coût abordable et simple d'utilisation. Grâce aux progrès technologiques remarquables, il est possible de trouver sur le marché des appareils ayant des hautes performances avec un faible coût.



Les typons des circuits imprimés et les programmes lorsqu'ils sont libres de droits sont téléchargeables

L'EDITO PRINTEMPS 2019

Chères lectrices et lecteurs,

En ce début d'année 2019, la Rédaction a répondu aux attentes d'un grand nombre de nos lecteurs qui désiraient réaliser une imprimante 3D capable d'imprimer des objets de grandes dimensions. Nous avons abordé succinctement la construction mécanique dans le précédent numéro, mais dans cette seconde partie nous développons l'aspect configuration logicielle qui pourra servir pour d'autres types d'imprimantes 3D.

Pour compléter le monde de la 3D, nous vous proposons un tutoriel en deux parties sur OpenSCAD, qui est logiciel gratuit permettant de créer des modèles 3D.

Enfin certains de nos lecteurs nous ont écrit en nous exposant leurs problèmes de télécommandes. En effet, certains en trouvent au fond de leurs tiroirs, d'autres perdent leurs télécommandes de portail ou d'alarme avec parfois l'unique solution de changer l'ensemble du système. C'est pour cela que nous avons développé un décodeur universel de télécommandes qui rendra bien des services.

Un autre montage intéressant est le chargeur sans fil dont l'article aborde l'aspect théorique de la charge sans fil avec à la clé une mise en œuvre pratique.

Enfin, nous continuons aussi nos cours détaillés concernant la programmation en C des PIC32 et la réalisation de circuits imprimés à l'aide du logiciel Open Source KiCad.

La Rédaction



54 LABORATOIRE

CARTES DE PROTOTYPAGE POUR AMPLIFICATEUR OPÉRATIONNEL

Nous vous proposons, dans cet article, de réaliser deux cartes de prototypage ou « breakout board » basées sur deux amplificateurs opérationnels, afin de tester des dizaines de configurations possibles. Le premier est un amplificateur de signaux analogiques unidirectionnels et peut être configuré en inverseur ou non inverseur, tandis que le second fonctionne comme un comparateur et peut piloter des charges allant jusqu'à 2 A.



59 COURS

COURS MPLAB X IDE

Dans les dernières leçons de ce cours, nous avons concentré notre attention sur les piles (stack) mises à disposition par Microchip. Nous continuons notre cours sur MPLAB-X, le nouvel environnement de développement intégré produit et distribué par Microchip et qui remplace l'ancien MPLAB IDE. Dans cet article, nous allons créer des applications, telles qu'un serveur Web en utilisant la pile TCP/IP de Microchip. Sixième leçon.



71 DIDACTIQUE

TUTORIEL OPENS CAD

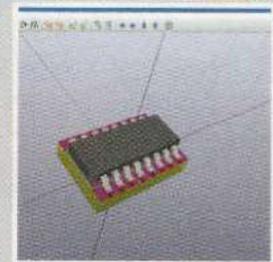
Ceux d'entre vous qui imprimez des objets en 3D, à l'aide par exemple de la 3DRAG+, avez besoin de logiciels de modélisation et d'édition en trois dimensions pour créer et éditer des fichiers 3D. Le logiciel OpenSCAD en est un bon exemple. Il permet entre autres de gérer la plupart des fichiers disponibles sur le site Thingiverse. OpenSCAD est un logiciel gratuit permettant de créer des modèles 3D basés sur une « géométrie de construction de solides ».



82 CAO

APPRENEZ À MAÎTRISER KICAD EDA

Nous poursuivons l'étude de Pcbnew, l'éditeur utilisé pour la mise en œuvre des circuits imprimés dans KiCad, en illustrant l'utilisation des bibliothèques et de l'éditeur d'empreintes. Nous continuons ainsi le développement de notre projet pratique et nous terminerons la réalisation de notre projet pratique. Enfin, nous aborderons la création des fichiers gerber nécessaires à la réalisation de nos circuits imprimés. Sixième et dernière partie.



92 INFORMATIQUE

UNITÉ VOIP SUR RASPBERRYPI

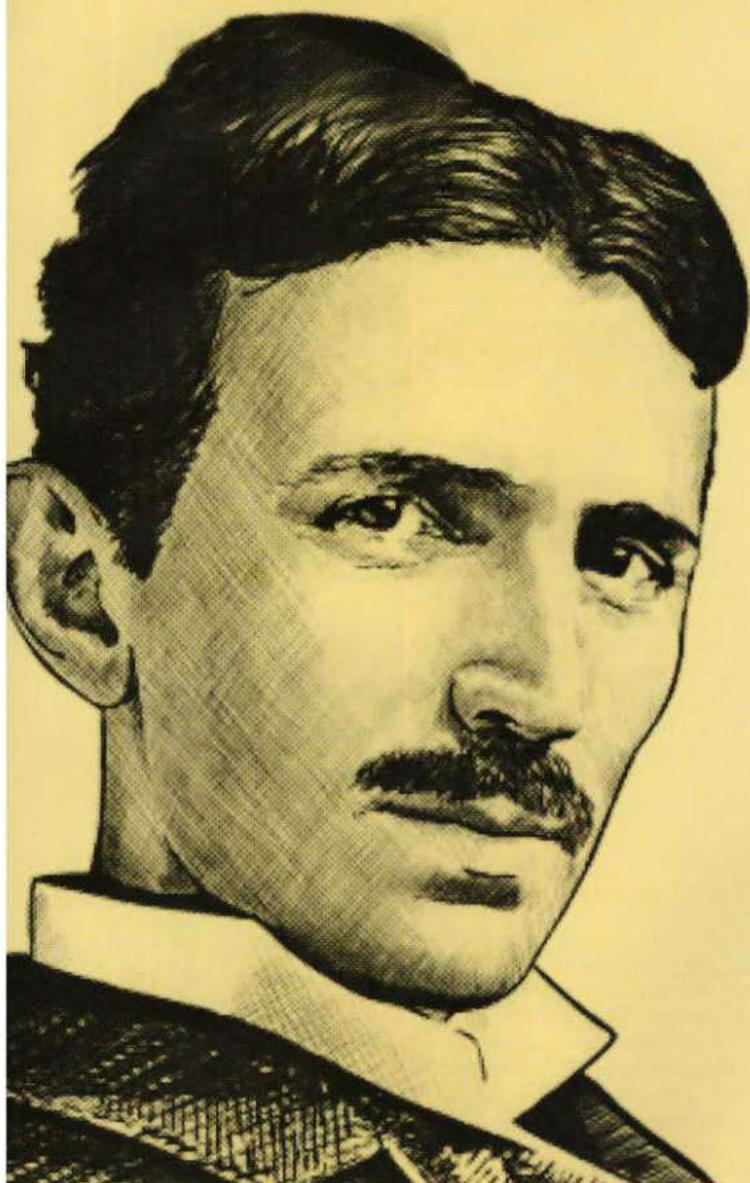
Nous mettons en pratique les notions de téléphonie sur IP que nous avons abordées dans le précédent numéro 145, en créant notre propre standard téléphonique complet doté de toutes les fonctionnalités les plus avancées. Dans cette seconde et dernière partie, nous allons étudier la configuration étendue de manière à obtenir des fonctions avancées avec lesquelles, par exemple, nous allons implémenter un service de sélection des appels entrants.



à l'adresse www.electroniquemagazine.com dans le sommaire de la revue 146 et à l'onglet « Télécharger ».

Découvrir et concevoir un chargeur sans fil

..... de Boris Landoni



La possibilité de **transférer de l'énergie électrique** au moyen d'**ondes électromagnétiques** a été démontrée pour la première fois en **1890** par **Nikola Tesla**, qui n'a peut-être pas imaginé à quel point il aurait été apprécié de nos jours.

Le standard « **Qi** » (prononcer « tchi ») et la **charge sans fil** sont des termes qui sont entrés fortement dans le secteur de la consommation et en particulier dans le secteur des smartphones et des tablettes. Ces appareils sont équipés de plus en plus de chargeurs sans fil pour recharger leur batterie interne.

Cette technologie a également été testée dans le domaine des véhicules électriques, où l'on trouve des expériences d'autobus urbains rechargés en s'arrêtant à une station où se trouve sous l'asphalte une grosse bobine produisant un énorme champ magnétique.

Considérant que le thème est important, nous avons décidé de réaliser une application de chargeur sans fil. Pour vous aider dans cette tâche, nous avons préparé une plate-forme de développement sur laquelle vous allez travailler et qui est composée de deux unités - un émetteur et un récepteur - que nous décrivons dans cet article afin de vous montrer le potentiel de cette technologie.

L'émetteur est l'unité qui irradie le champ électromagnétique dont l'énergie sera transférée au dispositif récepteur placé dans le champ de la bobine selon des modalités et des performances qui dépendent de facteurs que nous allons examiner maintenant.

La technologie

Transférer de l'énergie par des ondes électromagnétiques est facile, mais rendre cette technologie plus efficace et moins encombrante, en particulier pour des puissances élevées constitue un véritable défi, qui devient encore plus difficile si vous devez réaliser des solutions économiques et miniaturisées pour les plus petits appareils.

De nombreuses entreprises ont développé des dispositifs adaptés à cette technologie ces dernières années. Parmi les pionniers, nous pouvons citer **IDT** (Integrated Device Technology) qui a mis à disposition sur le marché des produits.

Le système de recharge sans fil décrit ici ne constitue pas une application réelle commerciale, mais permet de vous familiariser avec cette technologie.

La société Integrated Device Technology fournit un kit de démonstration Open Source pour lequel toute la documentation est disponible, y compris le schéma de câblage et les fichiers source nécessaires à la réalisation des circuits imprimés. Il comprend un émetteur et un récepteur, capables de transférer une puissance de 5 W sous une tension de 5 V avec un courant de 1 A (selon la norme WPC 1.1.2 de la spécification Qi).

Le projet a été conçu pour fournir une plate-forme de prototypage rapide et utile au développement de vos applications. Analysons maintenant en détail les deux unités, l'émetteur P9038-R et le récepteur P9025AC.

L'émetteur

Ce dernier est basé sur le circuit intégré P9838-R (voir la figure 1) qui génère un signal alternatif afin d'être amplifié pour pouvoir alimenter une inductance. Le rôle de celle-ci est d'**irradier l'espace environnant d'un champ électromagnétique**.

L'étage amplificateur fonctionne en pont, de manière à permettre à la bobine d'être pilotée avec une composante alternative, ce qui est impossible autrement lorsque l'on travaille avec une alimentation mono tension, comme dans notre cas.

Le circuit intégré est **capable de détecter la présence d'une charge**, c'est-à-dire d'une bobine à proximité à partir de laquelle l'énergie va être prélevée (pour simplifier, nous pouvons dire que l'émetteur « détecte » la présence du récepteur).

Regardons le schéma électrique de l'émetteur et examinons son fonctionnement. Il peut être interfacé à un ordinateur grâce au connecteur micro USB qui est utilisé pour fournir l'alimentation.

Il est aussi possible d'alimenter le circuit à partir des contacts « VIN » et « GND » afin de ne pas utiliser l'USB. Dans ce cas, la tension d'alimentation aux points « VIN » et « GND » doit être comprise entre 4,5 VDC et 6,9 VDC.

Le système peut être mis en veille (standby), tout en restant alimenté, afin de réduire la consommation pendant les phases où il n'est pas nécessaire de transférer de l'énergie sans fil. Le mode veille est géré par la broche « EN » (Enable) qui est active à un niveau bas.

S'il n'est pas nécessaire de mettre le circuit en veille, cette broche peut être connectée constamment à la masse (sur le circuit imprimé, elle est connectée à une résistance de tirage de 47 k Ω vers la masse). Dans le mode standby, l'émetteur ne consomme que 600 μ A.

Comme nous l'avons mentionné, **il est possible de choisir deux sources d'alimentation**, le circuit U1 les gérant de manière transparente pour l'utilisateur. Si le montage est alimenté à partir de l'USB, la broche 15 (VBUS) reçoit une tension de 5 VDC et commande la broche 47 (GATE) afin de polariser la grille du MOSFET Q1. Ce dernier devient conducteur et fait passer le courant vers la ligne « VIN », laquelle alimente la broche « REGIN » connectée au régulateur interne de U1.



La charge sans fil est la dernière frontière technologique des appareils mobiles, elle joue un rôle fondamental dans l'industrie. Grâce à cette technologie, nous vous proposons de concevoir un système d'alimentation sans fil pour vos projets.

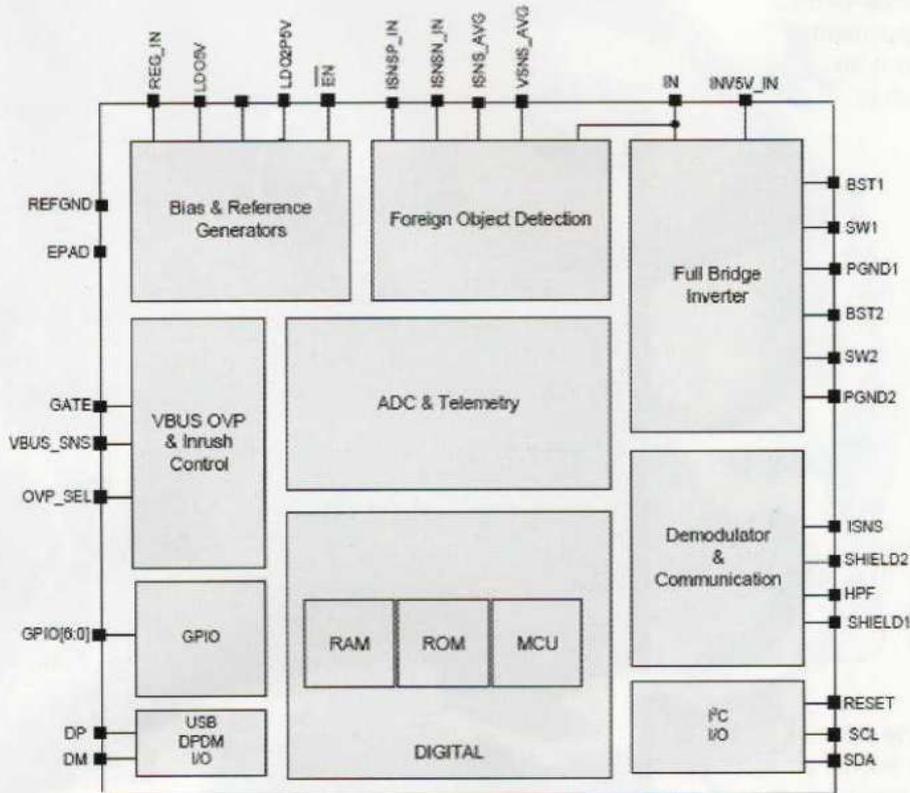


Figure 1 : schéma de principe du circuit intégré IDT P9038-R.

Ainsi le régulateur interne fournit une tension typique de 5 VDC à l'ensemble de U1. La ligne « VIN » est filtrée à l'aide des condensateurs C22 et C23, qui sont nécessaires quelle que soit la source d'alimentation du montage.

Notez que le condensateur **C1** permet un démarrage progressif du régulateur interne lorsque le circuit est alimenté par l'USB. En fait, il polarise progressivement la grille du MOSFET.

Quant à la résistance R1, elle permet une décharge du condensateur C1 lorsque la broche « GATE » revient à un niveau bas, sinon C1 resterait chargé et au prochain allumage le MOSFET conduirait instantanément (risque de surtensions).

Le circuit intégré **U1** comprend également un régulateur de tension linéaire interne **LDO** (Low-dropout ou faible tension de déchet) dont l'entrée est connectée à la broche « REGIN » et la sortie à la broche « LDO5V » (19), filtrée par un condensateur de 1 µF. Il existe également dans le circuit intégré U1

un autre régulateur de tension linéaire LDO fournissant 2,5 VDC sur la broche LDO2P5V (20).

La broche « RST » permet de réinitialiser le système et est maintenue à un niveau logique bas par une résistance de tirage vers la masse (pull-down) de 47 kΩ.

Comme dans le cas de la fonction « ENABLE », un report externe est également disponible pour la fonction RESET, ce qui permet de réinitialiser l'émetteur à l'aide d'un bouton poussoir externe par exemple.

Le seul autre circuit intégré utilisé dans le module émetteur est une **mémoire EEPROM** de type « 24AA64T », à l'intérieur de laquelle se trouve un firmware « standard » configuré pour fonctionner de manière optimale avec le circuit imprimé présenté et la bobine adéquate.

Le firmware est automatiquement chargé à chaque démarrage du système et la communication s'effectue via le bus I²C.

La broche « WP » (Write Pulse) de l'EEPROM est accessible depuis l'extérieur via la broche portant le même nom.

En ce qui concerne l'étage générateur de signaux qui pilote la bobine, la **fréquence générée est définie de manière interne** par le firmware.

Les sorties en pont sont situées sur les broches SW1_A, SW1_B, SW1_C et SW1_D (elles sont toutes connectées en parallèle et reliées à la borne L1a) pour un côté et sur les broches SW2_A, SW2_B, SW2_C et SW2_D (connectées en parallèle et à la borne L1b) pour l'autre côté.

Les condensateurs C10 et C21 permettent une contre réaction du signal de « bootstrap » afin de démarrer l'oscillateur.

La **bobine irradiante correspond à L1** et est découplée en continu par 4 condensateurs en parallèle qui déterminent la fréquence de résonance. La tension alternative présente entre la broche L1b et la masse est redressée par la diode D2 et traverse le diviseur de tension constitué par les résistances R17 et R11. En sortie du pont diviseur se trouve le condensateur C16 qui filtre la tension afin d'obtenir une tension pratiquement continue.

Les variations de cette tension traversant le réseau de composants constitué par les condensateurs C14, C12, C9 et les résistances R10, R7 et R9 sont lues par les broches « GPIO1 » et « ISNS ».

Cette dernière correspond à l'entrée du circuit de démodulation et de décodage de la « communication » sans fil, tandis que la broche « GPIO1 » est utilisée en contre réaction afin d'optimiser le fonctionnement du circuit de décodage des données.

Le signal renvoyé sur la broche « HPF » est filtré par la cellule passe-bas constituée des condensateurs C9 et C8. La broche « HPF » correspond à l'entrée du signal du démodulateur.

Dans chaque moitié du pont de sortie pilotant la bobine irradiante, une protection est mise en œuvre afin de

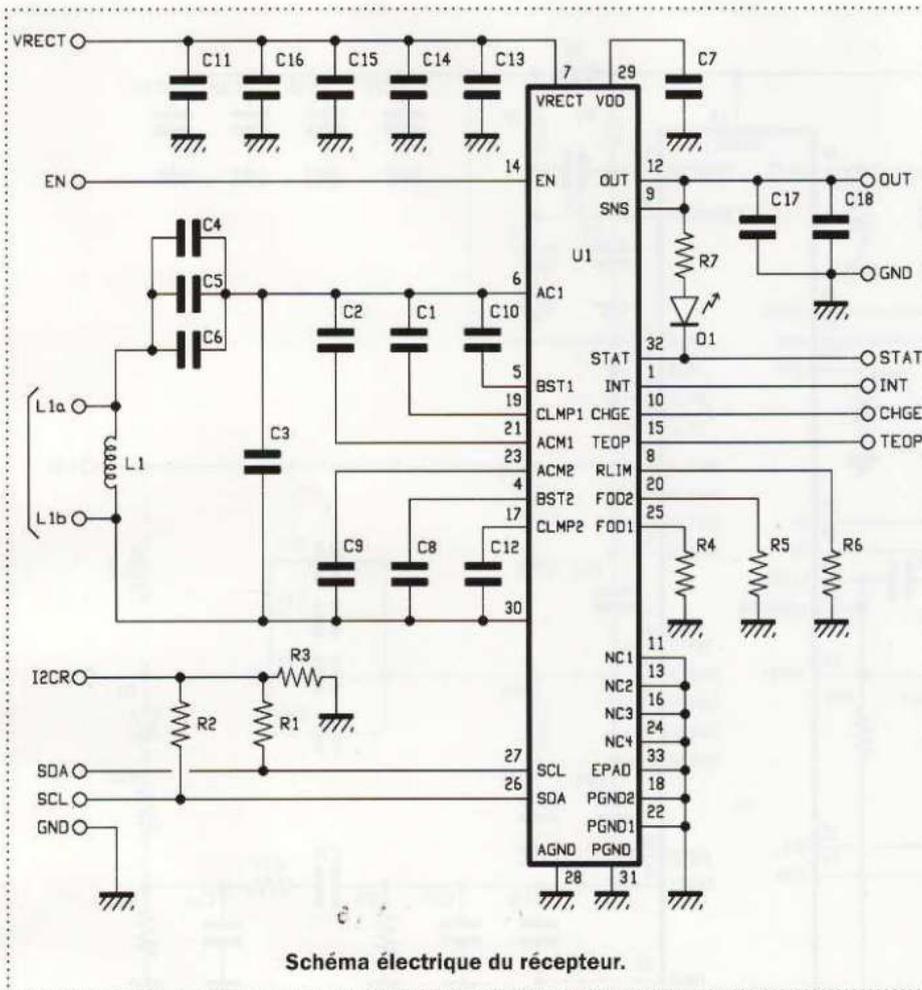


Schéma électrique du récepteur.

NB : une **Interférence électromagnétique** (EMI, Electromagnetic Interference) est un signal ou une émission véhiculé(e) dans l'espace libre ou par des conducteurs électriques traversés par des signaux, qui peut altérer le fonctionnement d'un appareil médical tel qu'un pacemaker ou un dispositif de sécurité ou encore interrompre les communications radio autorisées (bandes AM/FM, TNT, téléphonie GSM, navigation aérienne, etc.).

Le transfert d'énergie du P9038 vers un récepteur compatible WPC, tel que le P9025AC-R monté dans notre récepteur, est géré par ce dernier, grâce à une liaison des données qui exploite la réaction induite qui se produit dans la bobine en raison des variations de courant détectées par le récepteur lors de la communication.

Les paquets des données se superposent au champ électromagnétique rayonné et sont démodulés par le bloc approprié du P9038 (en fait, l'émetteur

et le récepteur communiquent entre eux avant le début du transfert d'énergie).

Si vous souhaitez plus de détails sur le protocole, visitez le site web officiel de WPC :

www.wirelesspowerconsortium.com/

En plus de ces spécifications purement « électriques, IDT a également veillé à assurer la sécurité des communications entre l'émetteur TX et le récepteur RX via un cryptage sur 64 bits.

La tension et le courant en entrée sont échantillonnés par le convertisseur analogique/digital (ou ADC) et sont ensuite traités par le firmware afin de détecter la présence d'un objet externe (FOD) conformément à la norme WPC 1.1.2. Les broches « ISNS_AVG » et « VSNS_AVG » permettent respectivement le filtrage des signaux provenant de la détection du courant et de la tension en entrée.

Le signal d'entrée dérivé de la détection du courant est généré de manière différentielle par les broches « ISNSP_IN » et « ISNSN_IN ». Cette entrée, de détection du courant, est filtrée par un réseau RC constitué par une résistance interne de 50 kΩ et par un condensateur externe relié à la broche « ISNS_AVG ».

Les mesures de la tension d'entrée sont filtrées par une résistance interne de 33 kΩ combinée à un condensateur externe relié sur la broche « VSNS ».

Nous vous recommandons de conserver les valeurs indiquées dans notre schéma afin d'assurer le calibrage des mesures et un calcul précis de la puissance du « FOD ».

Le module émetteur (TX) est alimenté avec une tension pouvant être comprise entre 4,5 VDC et 6,9 VDC. Il est également équipé d'une prise micro USB afin d'être alimenté en 5 V à partir d'un connecteur USB, ce dernier devant être capable de fournir 2 A.

Une fois la **liaison établie**, une **LED verte sur l'émetteur commence à clignoter**, indiquant que la liaison avec le récepteur est correcte. En fait, il existe deux LED sur l'émetteur, en plus de la verte il y a une LED rouge. Le tableau 1 résume l'état de fonctionnement du système, en fonction des indications visuelles fournies par les LED.

Le récepteur

Examinons maintenant le récepteur, qui est « relié » à l'émetteur TX via une **bobine de réception appropriée**. À cet égard, les spécifications « Qi » exigent une distance variant de 1,75 mm à 2,5 mm entre les bobines de l'émetteur et du récepteur.

Afin de garantir cette spécification et **optimiser l'efficacité du transfert de puissance**, nous avons inséré dans

	LED verte	LED rouge
Standby	allumée	allumée
Transfert en cours	clignote à 1 Hz	éteinte
Transfert terminé	allumée	éteinte
Alimenté	clignote à 2 Hz	éteinte

notre prototype un **carré en plexiglass transparent collé sur la grande inductance** filaire plane. La figure 2 montre l'évolution du rendement, avec une distance de 3,7 mm, lorsque le courant fourni au récepteur varie.

Le récepteur est beaucoup plus simple que l'émetteur, il est plus petit. Son rôle se limite à **recevoir la tension induite aux bornes de la bobine réceptrice** (connectée entre les contacts L1a et L1b du circuit).

Cette tension est ensuite redressée afin d'extraire les données et obtenir le courant nécessaire pour alimenter le circuit intégré de charge (entre la broche VRECT et la masse).

La communication (c'est-à-dire la transmission des données) s'effectue sous la forme de pics de courant au niveau de la bobine, obtenus en court-circuitant cette dernière à l'aide de plusieurs MOSFET internes.

Le schéma électrique du récepteur est représenté en figure 3, le circuit intégré U1 étant un **P9025AC**.

L'étage d'entrée obéit aux lois de l'électromagnétisme, c'est-à-dire celui qui détecte le champ électromagnétique où la bobine est reliée, **fournit une tension dont l'amplitude dépend du flux magnétique déterminé par la bobine irradiante** (celle de l'émetteur).

Ce flux dépend de l'angle de l'axe des deux bobines ainsi que de la distance. Le couplage des bobines, comme expliqué précédemment, constitue le maximum de puissance pouvant être obtenue.

La tension aux extrémités de la bobine réceptrice a une forme d'onde alternative pratiquement rectangulaire (les fronts sont un peu émoussés) et traverse les condensateurs de résonance C4, C5 et C6 pour atteindre les broches 6 et 30 de U1 qui sont les entrées alternatives.

À l'intérieur du circuit se trouve un pont de 4 diodes de type Graetz, chacune des diodes est disposée entre le drain et la source d'un MOSFET à canal N.

Ces derniers sont pilotés sur leur grille par l'étage qui fournit les signaux, c'est-à-dire le bloc « **Driver & Control** ».

Ce dernier est synchronisé à l'aide des tensions reçues sur les broches de « **bootstrap** » BST1 et BST2. Les condensateurs C8 et C10 permettent un rendement maximal de la bobine.

La tension ainsi redressée est appliquée aux broches VRECT et GND de U1 et est mesurée périodiquement grâce au convertisseur analogique/digital. Ce dernier, à l'aide d'un multiplexeur, est affecté par la logique de commande à diverses fonctions.

La logique de commande lit la valeur numérique provenant du convertisseur A/N et agit si nécessaire sur l'unité de commande des MOSFET afin de définir la courbe de régulation.

Si la tension sur la broche VRECT, qui est filtrée par les condensateurs C11 et C13 à C16, chute en dessous d'un seuil fixé en interne, la logique envoie à l'émetteur une demande afin d'abaisser la fréquence à une valeur la plus proche possible de celle de la fréquence de résonance.

L'étude de l'électromagnétisme nous apprend que pour un circuit RLC (ici cela correspond à un circuit équivalent à la sortie de l'émetteur et à l'entrée du récepteur), **le courant est maximum lorsque la fréquence de fonctionnement atteint la fréquence de résonance**.

D'autre part, si la tension sur la broche VRECT a tendance à trop augmenter du fait que le circuit fonctionne à une fréquence proche de celle de résonance, les données transmises indiquent à l'émetteur d'augmenter la fréquence de fonctionnement pour sortir de la plage de la fréquence de résonance.

Afin d'augmenter le rendement, la tension sur la broche VRECT est maintenue à une valeur aussi proche que possible de celle de sortie du régulateur interne LDO (qui alimente tous les étages logiques et le convertisseur A/N), en particulier si la charge en sortie du régulateur LDO augmente.

Aux faibles charges, la tension VRECT est paramétrée à une valeur élevée afin de faire face aux augmentations soudaines de la charge. La régulation de VRECT est obtenue grâce aux 4 MOSFET connectés chacun en parallèle à l'une des diodes du pont de Graetz.

Au fur et à mesure que la charge augmente, chacun des MOSFET entre en conduction pendant une période comprise entre le début et la fin de la demi-onde correspondante, fournissant ainsi un courant de charge supplémentaire sur la sortie VRECT à l'aide des condensateurs.

Pour synchroniser la conduction des MOSFET (si, par exemple, ils ne se bloquent pas à la fin de la période de la demi-onde durant laquelle la diode respective conduit), le bloc « **Driver & Control** », qui détecte les impulsions entrantes provenant de la bobine, les synchronise entre elles avant de les envoyer vers les grilles des MOSFET.

Le contrôle ressemble à un signal PWM, plus le courant est important, plus la largeur de chaque impulsion est grande et inversement.

Lorsqu'un dispositif mobile contenant le P9025AC est placé dans le champ électromagnétique produit par l'émetteur, il répond au signal de « **ping** » (interrogation) de l'émetteur en redressant le courant alternatif et en le stockant dans un condensateur connecté à la broche VRECT.

Pendant le « **ping** », le redresseur fournit environ 5 V sur la broche VRECT. Un régulateur de tension linéaire interne fournit la tension d'alimentation à la section numérique afin de permettre la communication WPC.

Le P9025AC communique ensuite son identifiant unique (ID) et ses informations de configuration à l'émetteur.

Après cela, le système autorise le transfert d'énergie (c'est-à-dire la charge). Le **circuit de contrôle** du P9025AC **ajuste ensuite la tension** du redresseur à 7 V, **en envoyant l'instruction « Control Error Packets »** à l'émetteur.

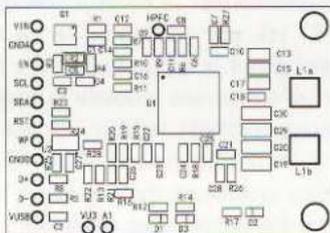
Le standard Qi

Pour assurer la compatibilité entre les émetteurs et les récepteurs, le Wireless Power Consortium (WPC) a été créé. Ce dernier a développé une norme ad hoc appelée standard « Qi » (<http://www.wirelesspowerconsortium.com>). Ce consortium a été créé en 2008 et comprend des sociétés asiatiques, européennes et américaines.

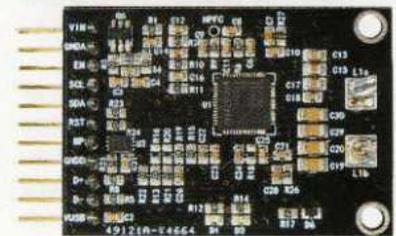
Le WPC définit le type de couplage inductif, c'est-à-dire la configuration des inductances et le protocole de communication à utiliser. Ainsi, chaque appareil conforme à la norme WPC peut fonctionner avec un autre produit standard, même de marque différente. Cela garantit ainsi l'interopérabilité et, surtout, l'utilisation d'une alimentation lorsque vous êtes en déplacement.

En plus du transfert de l'énergie, les inductances initient également le canal de communication entre le récepteur et l'émetteur. Le standard définit les systèmes capables de transférer de l'énergie sans contact, généralement avec des puissances allant jusqu'à 5 W et des distances ne dépassant pas 5 mm. La régulation de la tension de sortie est réalisée grâce à un contrôle numérique qui permet d'utiliser les communications échangées entre le RX et le TX afin d'informer ce dernier de la quantité d'énergie requise par la charge.

[plan de montage de l'émetteur]



Plan de câblage des composants de l'émetteur.



Photographie de l'émetteur.

Liste des composants de l'émetteur

- R1..... 10 MΩ boîtier CMS 0603
- R2..... 0,01 Ω 0 Ω boîtier CMS 0805
- R3..... 22 Ω boîtier CMS 0603
- R4..... 22 Ω boîtier CMS 0603
- R5..... 33 Ω boîtier CMS 0603
- R6..... 10 MΩ boîtier CMS 0603
- R7..... 47 kΩ boîtier CMS 0603
- R8..... 33 Ω boîtier CMS 0603
- R9..... 10 kΩ boîtier CMS 0603
- R10..... 10 kΩ boîtier CMS 0603)
- R11..... 15 kΩ boîtier CMS 0603
- R12..... 4,7 kΩ boîtier CMS 0603
- R13..... 4,7 kΩ boîtier CMS 0603
- R14..... 4,7 kΩ boîtier CMS 0603
- R15..... 4,7 kΩ boîtier CMS 0603
- R16..... NC
- R17..... 1,5 kΩ boîtier CMS 0603
- R18..... NC
- R19..... 10 kΩ boîtier CMS 0603
- R20..... NC
- R21..... 47 kΩ boîtier CMS 0603
- R22..... 47 kΩ boîtier CMS 0603
- R23..... 4,7 kΩ boîtier CMS 0603
- R24..... 4,7 kΩ boîtier CMS 0603
- R25..... 10 kΩ boîtier CMS 0603
- R26..... 0 Ω boîtier CMS 0603
- R27..... 0 Ω boîtier CMS 0603
- R28..... 4,7 kΩ boîtier CMS 0603

- C1..... 6,8 nF céramique boîtier CMS 0603

- C3..... 6,8 nF céramique boîtier CMS 0603
- C3..... 6,8 nF céramique boîtier CMS 0603
- C4..... 6,8 nF céramique boîtier CMS 0603
- C5..... 100 nF céramique boîtier CMS 0603
- C6..... 1 nF céramique boîtier CMS 0603
- C7..... 22 nF céramique boîtier CMS 0603
- C8..... 1 nF céramique boîtier CMS 0603
- C9..... 3,3 nF céramique boîtier CMS 0603
- C10..... 1 nF céramique boîtier CMS 0603
- C11..... 6,8 nF céramique boîtier CMS 0603
- C12..... 3,3 nF céramique boîtier CMS 0603
- C13..... 10 μF céramique boîtier CMS 0603
- C14..... 22 nF céramique boîtier CMS 0603
- C15..... 10 μF céramique boîtier CMS 0603
- C16..... 3,3 nF céramique boîtier CMS 0603
- C17..... 10 μF céramique boîtier CMS 0805
- C18..... 100 nF céramique boîtier CMS 0603
- C19..... 100 nF céramique boîtier

- CMS 0603
- C20..... 100 nF céramique boîtier CMS 0603
- C21..... 100 nF céramique boîtier CMS 0603
- C22..... 1 μF céramique boîtier CMS 0603
- C23..... 100 nF céramique boîtier CMS 0603
- C24..... 1 μF céramique boîtier CMS 0603
- C25..... 6,8 nF céramique boîtier CMS 0603
- C26..... 1 μF céramique boîtier CMS 0603
- C27..... 100 nF céramique boîtier CMS 0603
- C28..... 22 nF céramique boîtier CMS 0603
- C29..... 100 nF céramique boîtier CMS 1206
- C30..... 100 nF céramique boîtier CMS 1206

- D1..... LED rouge boîtier CMS 0603
- D2..... CDSU400B
- D3..... LED verte boîtier CMS 0603
- Q1..... FDC8878
- L1..... Bobine 6,5 μH (Würth 760308111)
- U1..... P9038
- U2..... 24AA64T-I/MNY

La sortie du régulateur interne LDO est activée et la charge est initiée lorsque que VRECT atteint une tension de 7 V.

Pendant le transfert de puissance à la charge, le circuit de commande P9025AC continue à envoyer des données de contrôle à l'émetteur afin de réguler la tension du redresseur au niveau requis pour maximiser l'efficacité du régulateur linéaire.

La transmission vers l'émetteur s'effectue en exploitant le principe de la **réaction induite**, en chargeant opportunément, via les MOSFET connectés aux broches ACM1, ACM2, CLMP1 et CLMP2, le dipôle composé de la bobine de réception L1 et des condensateurs C4, C5 et C6.

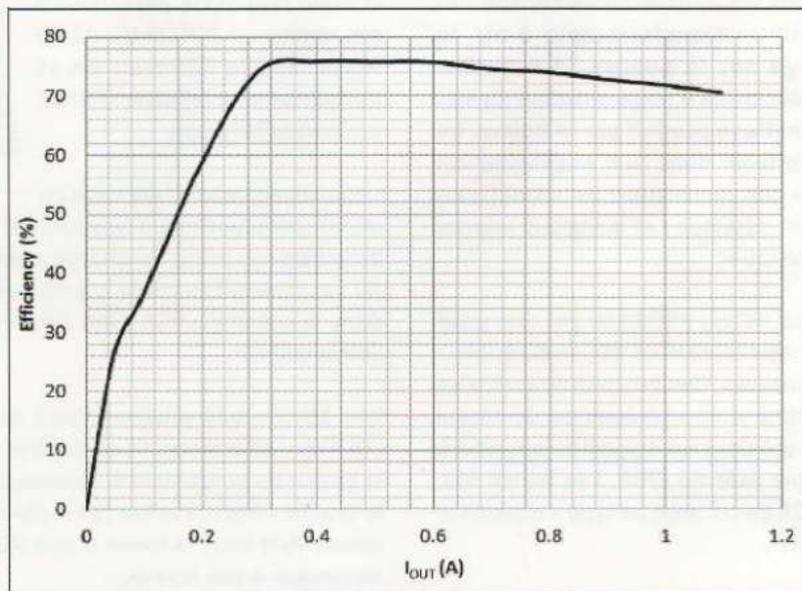


Figure 2 : courbe de transfert faisant référence à une distance de 3,7 mm.

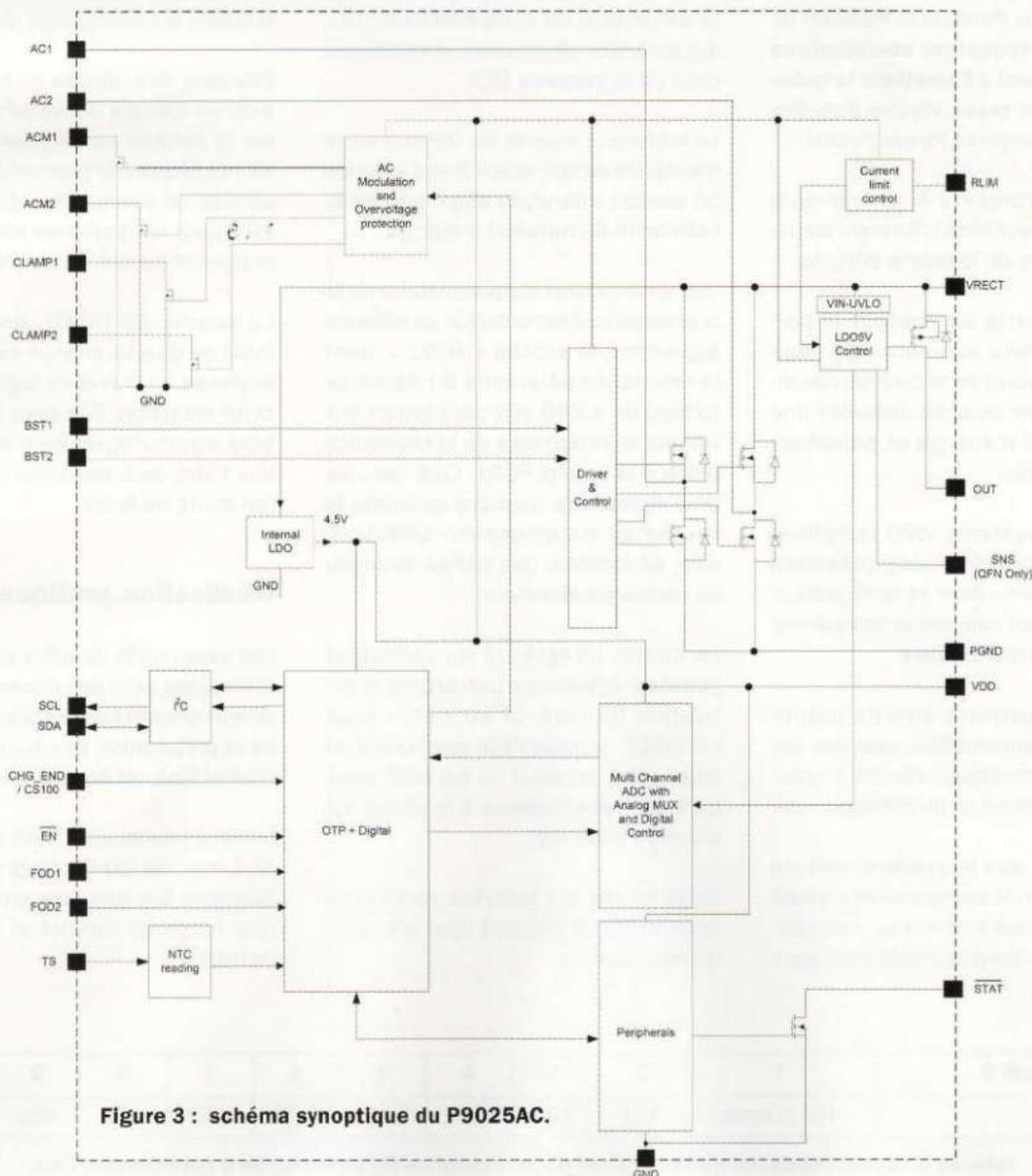


Figure 3 : schéma synoptique du P9025AC.

Comme les deux bobines constituent en fait un **transformateur à air, la charge sur la bobine L1 entraîne une demande d'augmentation du flux magnétique produit par la bobine de l'émetteur**, donc une augmentation de la consommation de l'émetteur, qui est détectée par la logique interne du P9038-R.

Le bus I²C du P9025AC est connecté aux broches SDA et SCL (référencées à la masse). Ces broches sont dotées chacune d'une résistance de tirage (pull-up) dont un des côtés est relié à la ligne externe I2CR. Les lignes SCL et SDA de U1 sont de type « collecteur ouvert ».

Le récepteur prend également en charge la détection des dispositifs avancés (FOD). Pendant le transfert de l'énergie, **le récepteur communique périodiquement à l'émetteur la quantité d'énergie reçue** via des données appelées « Received Power Packet ».

L'émetteur compare la valeur qu'il connaît à la quantité d'énergie transmise au cours de la même période.

En cas de perte de courant importante, l'émetteur interrompt l'alimentation en courant de la bobine, car un corps étranger pourrait absorber une telle quantité d'énergie et provoquer une surchauffe.

Pour qu'un système WPC remplisse cette fonction avec une précision suffisante, l'émetteur et le récepteur doivent chacun calculer et compenser toutes les pertes connues.

Ces pertes peuvent être de nature résistive et peuvent être causées par des pièces métalliques situées à proximité de l'émetteur ou du récepteur, etc.

Étant donné que le système mesure avec précision la puissance et connaît toutes les pertes inhérentes, il est possible de détecter des objets étrangers

et donc tout autre perte due à ces objets. Le P9025AC utilise des techniques FOD avancées et compense avec précision toutes ses pertes connues.

Cette compensation est mise en œuvre via une table d'ajustement de la courbe, qui prend en charge jusqu'à 10 courbes différentes mémorisées dans la mémoire OTP (One Time Programmable).

Les 10 courbes programmées dans l'OTP sont sélectionnées en externe par la valeur de la résistance connectée à la broche « FOD1 » et lue par le convertisseur A/N sous la forme d'une chute de tension à ses bornes.

De plus, l'interface I²C prend en charge le paramètre de compensation FOD, qui peut être sélectionné et remplacer celui de la mémoire OTP.

Le tableau 2 reporte les valeurs de la résistance en correspondance avec les 10 valeurs attendues afin d'améliorer l'efficacité du transfert d'énergie.

Une amélioration supplémentaire de la compensation est obtenue en utilisant également la broche « FOD2 », dont la résistance détermine un décalage (offset) de ± 300 mW par rapport aux valeurs sélectionnées de la résistance reliée à la broche FOD1. Cela est utile pour ajuster de manière optimale la courbe de compensation sélectionnée, en fonction des pertes connues au niveau du récepteur.

Le circuit intégré U1 du récepteur possède également une broche d'activation (broche 14 ou « EN » pour « ENABLE »). Lorsqu'elle se trouve à un niveau bas, le circuit U1 est actif, alors qu'à un niveau logique 1 le circuit est en veille (standby).

Cette broche est reportée vers l'extérieur du circuit imprimé pour une commande externe.



Ici, le récepteur avec la bobine réceptrice connectée.

La broche 15, nommée « TEOP » et dotée d'une résistance interne de tirage, est particulièrement intéressante.

Lorsqu'elle est placée à un niveau logique haut, elle désactive le régulateur interne LDO et envoie à l'émetteur un message « **WPC End-of-Power packet** » avec le code « 0x03 » (surchauffe), cela suspend la transmission d'énergie.

Elle peut être utilisée en combinaison avec un capteur de température placé sur la batterie du dispositif à charger afin de bloquer le processus de charge en cas de surchauffe. Cela est très utile pour les batteries au lithium qui explosent en cas de surchauffe.

La broche 32 (STAT), en revanche, indique que la charge est en cours et passe à un niveau logique 0 dans cette condition. Elle peut être utilisée pour communiquer à un microcontrôleur l'état de la condition de la charge (en cours ou finie).

Réalisation pratique

Les deux unités de notre système sont fabriquées avec des composants CMS et leur assemblage nécessite, en plus de la préparation de circuits imprimés double face, un équipement approprié.

Un fer à souder pour CMS à pointe fine (0,1 mm) de 20 W, de la soudure de diamètre 0,5 mm, une pince à épiler, une loupe de qualité et une bonne dextérité.

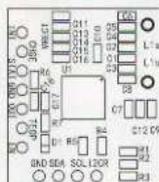
Bank #	1	2	3	4	5	6	7	8	9	10
RFOD1(kΩ)	150 to open	113	97.6	75	64.9	56.2	49.9	45.3	40.2	20 to 33.2

Tableau 2 : correspondance entre la valeur de la résistance R4 et la courbe d'optimisation FOD.

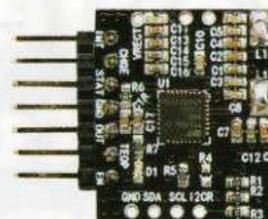
[plan de montage du Récepteur]

Liste des composants du récepteur

R1.....	4,7 k Ω boîtier CMS 0603
R2.....	4,7 k Ω boîtier CMS 0603
R3.....	0 Ω boîtier CMS 0603
R4.....	NC
R5.....	47 k Ω boîtier CMS 0603
R6.....	30 k Ω boîtier CMS 0603
R7.....	4,7 k Ω boîtier CMS 0603
C1.....	470 nF céramique boîtier CMS 0603
C2.....	22 nF céramique boîtier CMS 0603
C3.....	1,8 nF céramique boîtier CMS 0603
C4.....	47 nF céramique boîtier CMS 0603
C5.....	100 nF céramique boîtier CMS 0603
C6.....	100 nF céramique boîtier CMS 0603



Plan de câblage des composants du récepteur.



Photographie du récepteur.

C7.....	1 μ F céramique boîtier CMS 0603
C8.....	10 nF céramique boîtier CMS 0603
C9.....	22 nF céramique boîtier CMS 0603
C10....	10 nF céramique boîtier CMS 0603
C11....	100 nF céramique boîtier CMS 0603
C12....	470 nF céramique boîtier CMS 0603
C13....	4,7 μ F céramique boîtier CMS 0603

C14....	4,7 μ F céramique boîtier CMS 0603
C15....	4,7 μ F céramique boîtier CMS 0603
C16....	4,7 μ F céramique boîtier CMS 0603
C17....	1 μ F céramique boîtier CMS 0603
C18....	4,7 μ F céramique boîtier CMS 0603
D1.....	LED verte boîtier CMS 0603
L1.....	bobine 12 μ H (Würth 760308103202)
U1.....	P9025AC-RNBGI

Heureusement l'émetteur et le récepteur sont disponibles sous la forme de kits déjà montés et testés pour quelques dizaines d'euros.

Pour ceux qui préfèrent fabriquer l'ensemble, nous recommandons l'utilisation d'une station à air chaud pour composants CMS.

Une fois les composants assemblés, il faut connecter les bobines déjà prêtes à l'emploi.

Comme nous l'avons mentionné, il est préférable d'installer l'émetteur entre deux plaques de plexiglas de 3 mm d'épaisseur pour que la distance soit correcte lorsque vous placez le dispositif de réception.

Si vous installez l'émetteur dans un boîtier qui servira de base pour l'appareil à recharger, tenez compte de l'épaisseur de la paroi sur laquelle vous allez disposer la bobine du récepteur, vous pourrez alors vous passer du plexiglas au niveau de la bobine irradiante.

Il est entendu qu'aucune plaque métallique ne doit être interposée entre les bobines émettrice et réceptrice, pas

même un plastique électriquement conducteur, sous peine d'une forte perte d'efficacité ou de l'impossibilité de faire fonctionner le système.

Utilisation

Afin de s'assurer que le dispositif connecté au récepteur a une tension de sortie de 5,3 V avec un courant maximal de 1A, le fabricant IDT recommande d'alimenter l'émetteur avec une alimentation capable de fournir une tension de 5 VDC avec un courant de 2 A.

Une fois la connexion sans fil établie, la LED verte du récepteur s'allume pour indiquer que le transfert d'énergie a lieu correctement et qu'il existe une tension de 5,3 V entre les bornes « OUT » et « GND ». Comme la LED est optionnelle, elle peut même ne pas être montée.

Cependant, comme sa consommation (1 mA) est vraiment faible, pour un système capable de délivrer 5 W, nous suggérons donc de la laisser, ne serait-ce que pour indiquer l'état du système.

En particulier pendant les phases de test et de développement de votre application.

Lors de nos tests effectués en laboratoire, le système a transféré une puissance 5 W avec une tension de sortie au niveau du récepteur de 5 V, avec un courant de 1 A.

L'efficacité du système émetteur/récepteur en fonction du courant consommé par ce dernier (issu de sa bobine) est illustrée en figure 2. Le graphique montre qu'en restant dans la plage de courant recommandée (de 0,2 A à 1 A) par le fabricant, le rendement se situe entre 60 % et 76 %, ce qui est une valeur acceptable.

Les applications possibles du système sont la recharge d'appareils dont la batterie n'est pas amovible ou ayant des contacts difficiles d'accès. Comme le récepteur peut détecter le champ électromagnétique produit par l'émetteur (en le communiquant via la broche 32), le système peut être associé à une tondeuse à gazon autonome ou à un aspirateur qui se met automatiquement en charge lorsqu'il retourne sur sa base.

Roulette Électronique à LED

de Michele Menniti

Nous mettons au goût du jour une version électronique de l'un des jeux de casino les plus célèbres, sans trucage et sans déception.



Le mot « roulette » est évidemment français, mais beaucoup disent que le jeu a été inventé en Italie sous le nom de « giarella ». Quelle que soit son origine, nous pouvons affirmer en toute certitude qu'il s'agit de l'un des jeux de casino les plus célèbres et des plus pratiqués dans le monde et certainement l'un des jeux les plus utilisés, avec le poker, dans les films célèbres comme les westerns ou les films d'espionnage.

Parmi les différentes versions fabriquées dans le monde, connues comme françaises, anglaises et américaines, nous nous sommes inspirés de la première (voir la figure 1), qui est certainement la plus célèbre.



La différence entre les versions concerne principalement la présence ou l'absence du double « 0 » et la disposition des chiffres rouges et noirs. Un très bref résumé du jeu vous évoquera peut-être des souvenirs.

Avec les célèbres phrases en français citées par le Croupier, les temps du jeu sont marqués. Avec le départ de la roue et le lancement de la bille, la phrase « Mesdames et Messieurs, faites vos jeux » indique le début des paris, qui sont effectués en plaçant les jetons sur la table de jeu, généralement un tapis vert, où sont dessinées des cases correspondant aux couleurs et aux numéros de la roulette ainsi que des cases additionnelles (voir la figure 2).

La phrase suivante « Rien ne va plus » marque la fin des paris ou du déplacement des jetons sur la table.

Dès que le numéro sort, le Croupier l'annonce avec les accessoires auxquels il appartient. Par exemple : huit (8), noir, pair et manque (nombre 1 à 18) ou vingt-sept (27), rouge, impair et passe (nombre compris entre 19 et 36).

Voici l'annonce des différents types de gains possibles des joueurs :

- Plein : numéro unique, gain 35 fois le montant de la mise ;
- Cheval : paire de numéros, gain 17 fois le montant de la mise ;
- Transversale Pleine : triplé, gain 11 fois le montant de la mise ;
- Carré : quadruplé, gain 8 fois le montant de la mise ;
- Transversale Simple, quintuple gain 5 fois le montant de la mise ;
- Douzaine : gain 2 fois le montant de la mise ;
- Colonnes : gain 2 fois le montant de la mise.

Bien entendu, notre roulette ne pouvait être qu'électronique. Nous avons donc réfléchi à la façon de créer un circuit capable d'imiter le mouvement rotatif de la bille avec l'allumage séquentiel d'un certain nombre de LED, si possible sans microcontrôleur ni programme afin que le montage soit reproductible par le plus grand nombre de lecteurs.

La solution que nous avons retenue et que nous allons décrire dans ces pages, est que le mouvement de la bille dans le cercle est obtenu en pilotant de manière appropriée 37 LED de couleurs différentes et la sortie du nombre par l'allumage d'une seule d'entre elles.

Schéma électrique

L'électronique de notre roulette est divisée en deux parties : l'unité logique de contrôle et le panneau de LED. Pour rendre la réalisation de ce projet simple mais aussi plus intéressante, nous avons eu recours à d'anciens circuits logiques TTL/CMOS, mais remis au goût du jour en utilisant leurs versions modernes qui sont les HCMOS (74HCXXX).

Pour une analyse approfondie des circuits logiques, il serait très utile de lire les introductions théoriques des articles

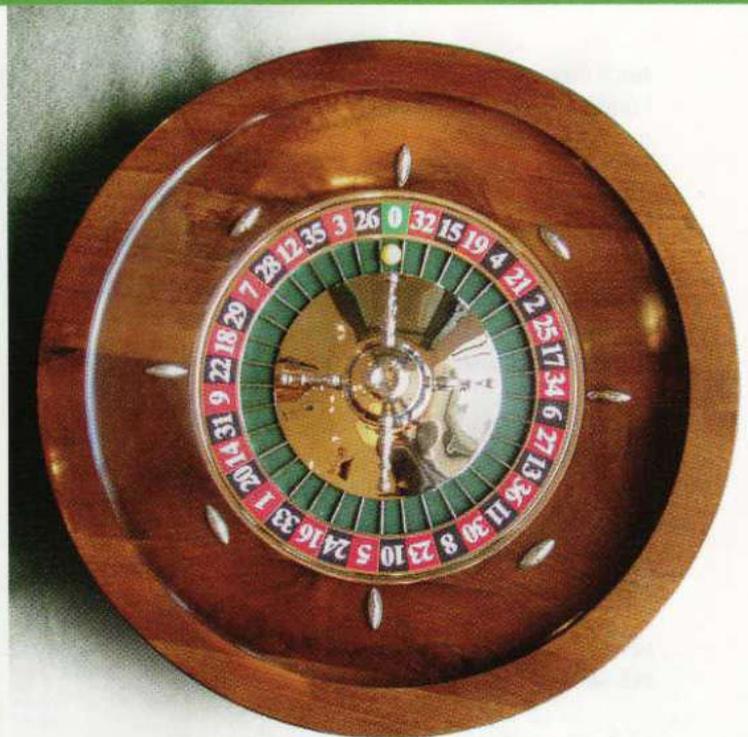


Figure 1 : la roulette française.

présentés dans les numéros 127 et 128 (Capacimètre numérique), où nous avons traité en détail des familles logiques et des composants utilisés dans ce projet.

C'est pour cette raison que nous pouvons également inclure ce nouveau projet dans notre série d'articles « vintage », car tous les composants ont des origines remontant aux années 70 et 80.

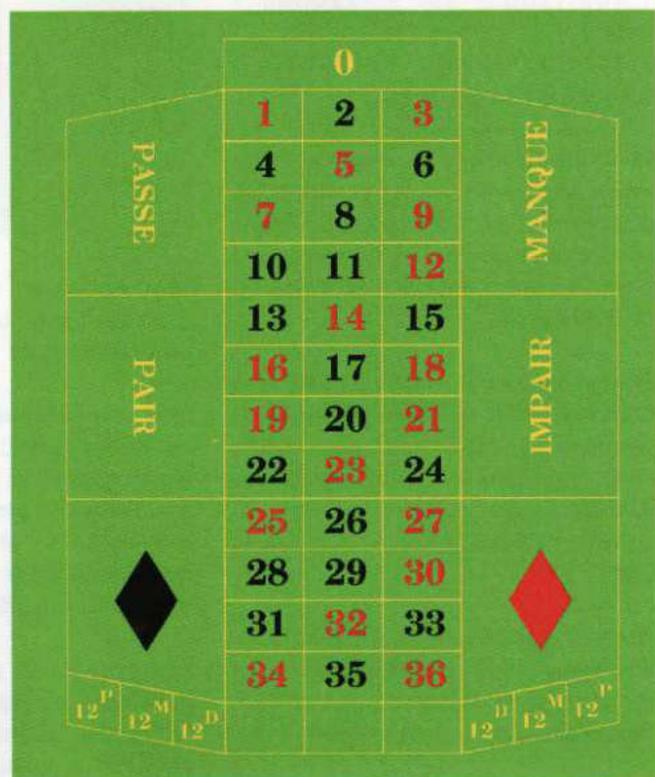


Figure 2 : le tapis vert pour les paris.

Nous commençons par la description de l'unité logique de contrôle en précisant que la partie la plus compliquée de ce projet consiste à **générer**, au moyen de circuits logiques uniquement (sans microcontrôleur), une **base de temps** (horloge ou clock) **qui pilote des compteurs démarrant avec une fréquence d'une valeur donnée** pour diminuer progressivement jusqu'à arriver à 0.

Cette unité logique de contrôle permet d'imiter correctement la vitesse de rotation réelle de la roue qui commence rapidement, puis diminue jusqu'à s'arrêter.

Pour résoudre ce problème, nous avons eu recours à une configuration particulière du circuit intégré toujours d'actualité, le fameux **NE555** (U1).

La configuration de base du circuit est celle du **multivibrateur astable** utilisé comme générateur d'horloge pour les portes logiques, avec les composants R1, R2 et C3 qui définissent la fréquence initiale.

La formule pour calculer la fréquence générée, dans notre cas, est la suivante :

$$F = 1,443 / [C3 (R1 + 2R2)] = 1,443 / [100 * 10^{-9} * (220000 + 200000)] = 1,443 / (0,0000001 * 420000) = 1,443 / 0,042 = 34,37 \text{ Hz}$$

Contrairement à la classique configuration astable, R1 n'est pas connectée directement au +5V, car nous avons interposé le bouton P1 normalement ouvert et ayant pour but de démarrer ou d'arrêter le compteur.

Si vous appuyez et maintenez pressé le bouton P1 (qui se trouve au milieu du panneau de LED et dont les contacts sont raccordés à l'unité logique de contrôle via les broches 5 et 6 de CN1, correspondant aux broches 5 et 6 de CN4).

La sortie du NE555 aura une **fréquence fixe** égale à environ **34 Hz** (la valeur réelle dépend évidemment des tolérances des composants utilisés).

Dans le même temps, le condensateur C1 est chargé au moyen de R3, dont le but principal est d'éviter le choc du

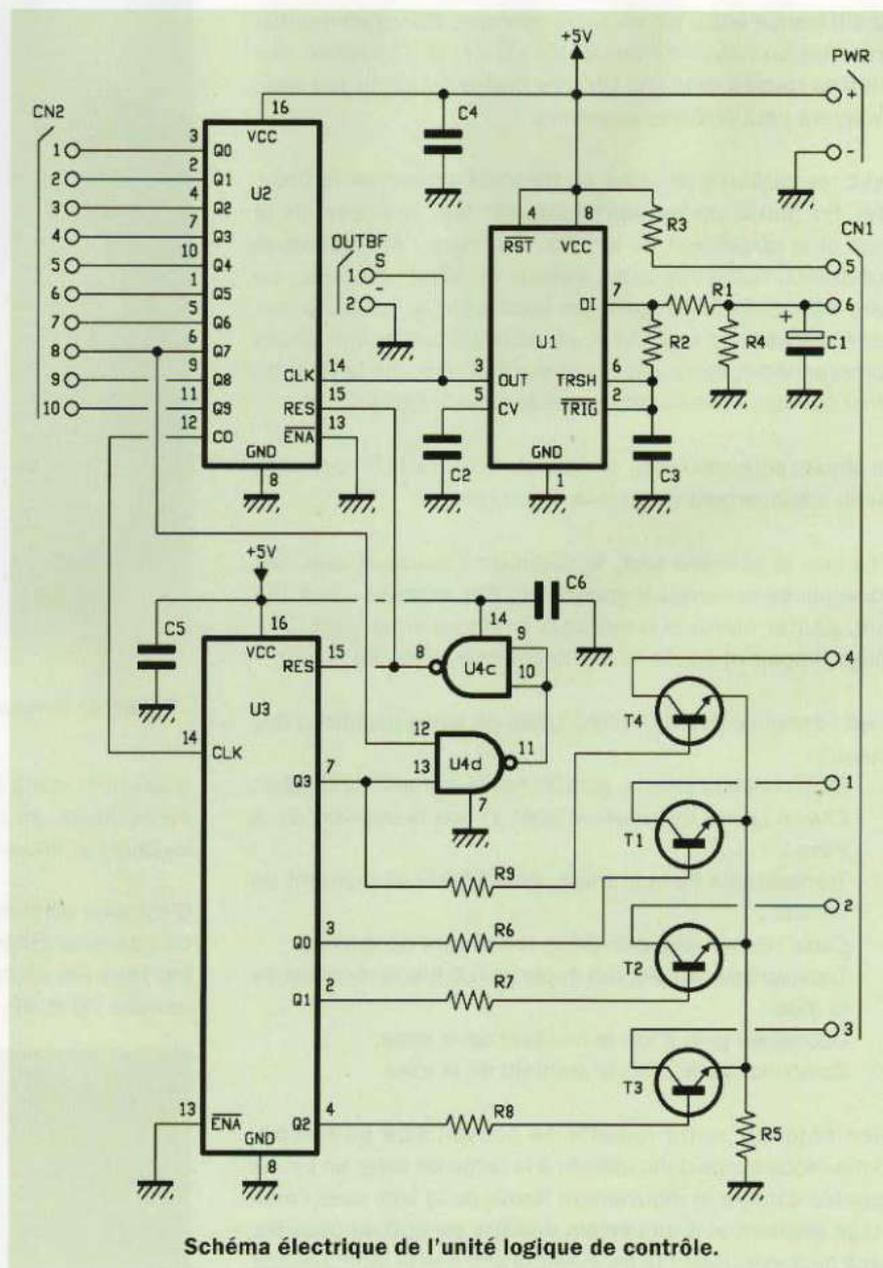


Schéma électrique de l'unité logique de contrôle.

court-circuit qui se crée chaque fois que l'on appuie sur le bouton lorsque le condensateur est déchargé.

Dès que le **bouton est relâché**, le condensateur **C1 commence à se décharger dans R4**, ce qui provoque une diminution progressive de la tension présente sur R1 et, par conséquent, une **diminution de la fréquence en sortie** (broche 3) du NE555 de 34 Hz à 0 Hz.

Par conséquent, il devient clair que **si nous voulons augmenter ou diminuer la fréquence initiale**, nous devons **modifier les valeurs de R1, R2 et C3**. Si nous voulons augmenter ou diminuer le temps de « rotation de la bille », nous devons augmenter ou diminuer la valeur

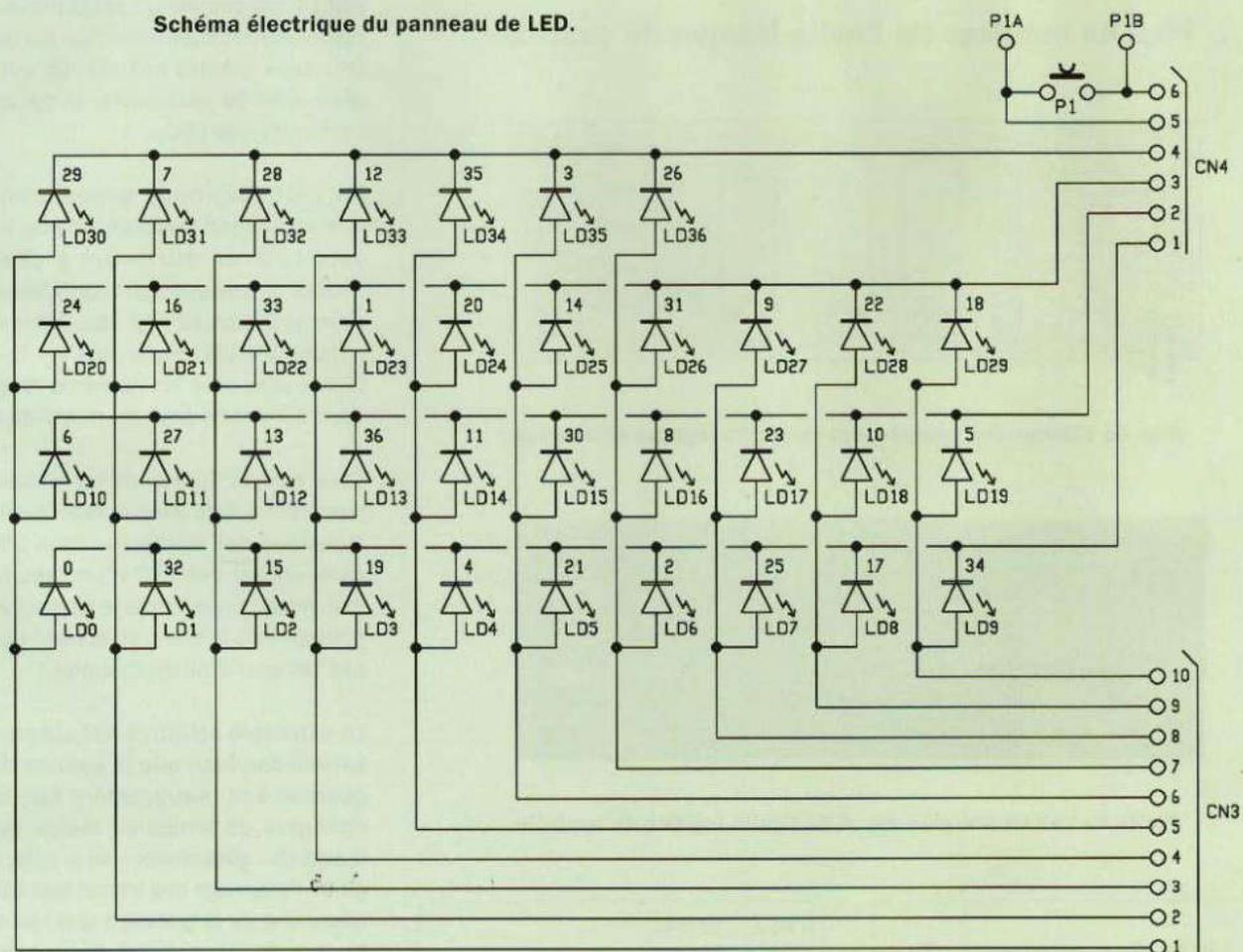
du condensateur C1 ou éventuellement celle de la résistance R4.

Sur la broche 3 du NE555, nous prélevons le signal de cette horloge variable et nous l'appliquons directement à la broche 14 de U2 (74HC4017), qui est un **compteur/diviseur par 10**.

En fait, à **chaque cycle d'horloge** appliqué sur la broche 14, **une seule des sorties Q0 à Q9 passe séquentiellement à un niveau logique haut** (1 ou HIGH).

Les lignes Q0 à Q6 fournissent une alimentation positive à autant de groupes de quatre LED (broche 1 à 7 du connecteur CN2, correspondant au connecteur CN3 du panneau de LED), tandis que les

Schéma électrique du panneau de LED.



lignes Q7 à Q9 alimentent trois groupes de trois LED via les broches 8 à 10 du connecteur CN2.

Cette unité intégrée se charge de faire office de compteur d'unités. Lorsque l'activation de Q9 se produit la séquence recommence à partir de Q0, mais U2 envoie simultanément une impulsion d'horloge (via la broche 12) à la broche 14 du deuxième compteur U3 (74HC4017). Cela active en séquence (comme ce qui vient d'être vu pour le compteur d'unités) les sorties Q0 à Q3, afin de gérer le comptage des dizaines.

Chacune des sorties est bufférisée par les transistors T1 à T4 (BC547) afin de ne pas surcharger le compteur même si, en réalité, les 20 mA délivrés par les sorties des deux circuits intégrés suffisent à piloter les LED haute luminosité que nous avons adoptées.

Il convient de noter que, même si nous utilisons les transistors dans la configuration typique d'émetteur commun, nous

n'avons ajouté qu'une seule résistance de limitation (R5) commune aux émetteurs, au lieu de 4 résistances reliées aux collecteurs (cela simplifie le montage).

La raison qui nous a amenée à utiliser cette « astuce » réside dans le fait qu'une seule LED à la fois sera allumée, sur les 37 du panneau de LED.

Les collecteurs des quatre transistors acheminent ensuite l'alimentation positive vers les quatre groupes respectifs de LED (trois rangées de dix DEL et une rangée de sept).

Le fonctionnement du circuit est le suivant. Supposons, par exemple, qu'au moment du démarrage les deux sorties Q0 sont actives. La première LED à s'allumer sera la LED verte LD0 (numéro 0, vert), car c'est la seule à recevoir une tension positive sur son anode et une tension négative sur sa cathode.

À la prochaine impulsion d'horloge, la sortie Q1 de U1 est activée, tandis que

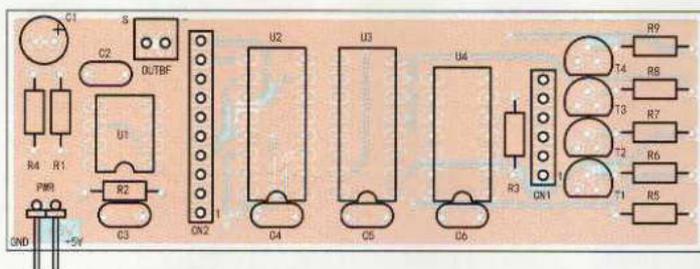
U2 reste sur Q0 et allume la LED rouge LD1 (numéro 32, rouge).

L'impulsion d'horloge suivante implique que la sortie Q2 de U1 est activée, tandis que U2 reste sur Q0 et allume la LED bleue LD2 (numéro 15, noire) et ainsi de suite jusqu'à ce que la LED rouge LD9 (numéro 34, rouge) s'allume.

Lors du signal d'horloge suivant, U1 active de nouveau Q0, mais aussi simultanément, U2 active Q1. La LED bleue LD10 (numéro 6, noir) s'allume et ainsi de suite jusqu'à ce que les sorties Q6 de U1 et Q3 de U2 soient actives. La LED bleue LD36 s'allume (numéro 26, noir).

En l'absence de LED successive après la dernière (37^{ème}), le système doit être réinitialisé en redémarrant à partir de la LED verte LD0. Les sorties Q7 de U1 et Q3 de U2 seront actives au prochain signal d'horloge. Dans cette situation, la porte NAND « D » de U4 (74HC00) aura les deux entrées à un niveau logique haut (HIGH).

Plan de montage de l'unité logique de contrôle



Plan de câblage des composants de l'unité logique de contrôle.

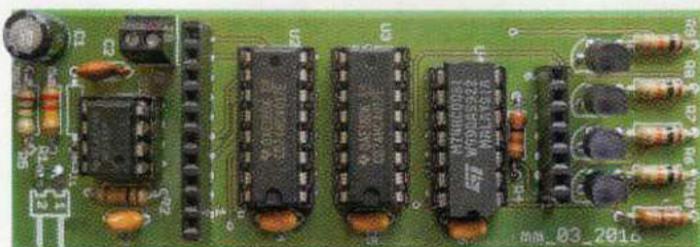


Photo de l'un de nos prototypes de l'unité logique de contrôle.

Liste des composants de l'unité logique de contrôle

- R1..... 220 k Ω
- R2..... 100 k Ω
- R3..... 100 Ω
- R4..... 1 M Ω
- R5..... 180 Ω
- R6..... 10 k Ω
- R7..... 10 k Ω
- R8..... 10 k Ω
- R9..... 10 k Ω
- C1..... 100 μ F/16 V
- C2..... 10 nF céramique
- C3..... 100 nF céramique
- C4..... 100 nF céramique
- C5..... 100 nF céramique
- C6..... 100 nF céramique
- T1 BC547
- T2 BC547
- T3 BC547

- T4 BC547
- U1..... NE555
- U2..... 74HC4017
- U3..... 74HC4017
- U4..... 74HC00N

Divers

- Support circuit intégré 2 x 8 broches (x2)
- Support circuit intégré 2 x 7 broches
- Support circuit intégré 2 x 4 broches
- Bornier 2 pôles
- Barrette femelle 10 pôles
- Barrette femelle 6 pôles
- Barrette mâle 2 pôles coudée 90°

NB : les typons des circuits imprimés à l'échelle 1 sont disponibles en téléchargement dans le sommaire détaillé de la revue.

Sur sa sortie, nous trouverons un niveau logique bas (LOW) qui sera inversé et donc amené à un niveau logique haut par la porte NAND « C » de U4, utilisé en configuration inverseur (c'est-à-dire avec les deux broches d'entrée reliées).

Pour mieux comprendre ces mécanismes, nous vous invitons une nouvelle fois à (re)lire les articles indiqués en début d'article.

Le signal de niveau logique haut (HIGH) appliqué simultanément aux broches

RESET des circuits U1 et U2 provoquera l'activation instantanée des sorties Q0 des deux mêmes circuits, ce qui aura pour effet de redémarrer le cycle avec l'activation de LDO.

En pratique, nous avons créé une matrice simple mais efficace de trente-sept LED, en attribuant à chacune d'elles la couleur correspondante à la roulette française originale. Nous avons adopté une LED verte pour le « 0 », des LED rouges pour les numéros rouges et des LED bleues pour les numéros noirs.

Vous pouvez fabriquer un masque en plastique coloré avec la représentation classique des numéros, vous pourrez alors utiliser des LED d'une seule couleur, nous pensons que le plus approprié pourrait être le blanc ou le bleu, pour ne pas tromper l'œil du Croupier !

Le caractère aléatoire est absolument garanti car, bien que la gamme de fréquences soit pratiquement fixe, ce qui changera de temps en temps sera le temps de « glissement » de la bille, c'est-à-dire l'allumage des trente-sept LED qui dépendra de la pression exercée sur le bouton P1 mais aussi des conditions de fonctionnement du condensateur C1, sachant que la différence de température et la durée de fonctionnement affecteront également ses temps de charge/décharge.

Enfin, comme le montre clairement le schéma, nous avons prévu de **porter le signal d'horloge vers l'extérieur à l'aide de la sortie BF « OUT »**, pour le diriger vers un petit **amplificateur audio**.

Nous avons choisi un petit module amplificateur 2 x 3 W sous 4 Ω en classe D alimenté en 5 VDC. Il est doté d'entrées et de sorties stéréo et d'un potentiomètre de volume avec un interrupteur marche-arrêt.

Il est disponible pour une dizaine d'euros déjà monté. Vous pouvez cependant adopter un autre modèle qui vous conviendra.

En connectant l'amplificateur à un petit haut-parleur, l'effet est exceptionnel. Vous pourrez reproduire le bruit de la bille qui tourne sur le plateau, le bruit qui ralentit de plus en plus parfaitement

synchronisé avec le mouvement des LED, simplement parce que nous utilisons la même horloge.

Il est possible d'**utiliser tout autre système d'amplification** audio, à condition que l'**impédance d'entrée soit élevée**, sinon la sortie du NE555 serait trop chargée et risquerait de ne pas fonctionner, ce qui causerait un dysfonctionnement ou même un blocage complet du circuit.

Le montage consomme environ 70 à 80 mA, de sorte qu'il peut être alimenté indifféremment avec une alimentation continue de 5 VDC ou avec un bloc d'accumulateurs rechargeables (4 x 1,2 V) de type LR06. Dans ce dernier cas, il faudra évidemment repenser la solution du boîtier que nous proposons.

Réalisation pratique

Les composants électroniques de la roulette sont disposés sur deux circuits imprimés, l'un contenant l'unité logique de contrôle et l'autre contenant le panneau d'affichage à LED. Ce dernier comporte donc les 37 LED et le bouton de démarrage de la bille.

Les typons des circuits imprimés à l'échelle 1, nécessaires à la réalisation de ce projet, sont disponibles en téléchargement dans le sommaire détaillé de la revue.

L'assemblage des composants est une opération assez simple, étant donné qu'ils sont tous de type traversant.

Les pastilles des composants sont donc éloignées les unes des autres et ne nécessitent pas d'équipement spécial. Un fer à souder de 20 à 30 W est suffisant, avec de la soudure courante (1 mm de Ø).

Pour ce qui est du circuit de l'unité logique de contrôle, il est préférable, comme d'habitude, de commencer par souder les composants ayant un profil bas comme les résistances, les supports des circuits intégrés, les condensateurs non polarisés, les transistors et de terminer avec le condensateur polarisé C1 en respectant son orientation, puis le connecteur d'alimentation et enfin le bornier pour la sortie audio.

Le son de la bille

Pour reproduire le son de la bille qui tourne sur le plateau de la roulette avant de tomber dans la case d'un numéro, nous prélevons le signal d'horloge et l'amplifions avec un petit amplificateur en classe D de 2 x 3 W.

Les amplificateurs fonctionnant dans cette classe ont des rendements pouvant atteindre de 90 % et plus, contre 60 % typiques pour ceux de la classe AB.

Le rendement élevé signifie que l'amplificateur ne chauffe pratiquement pas, nous n'avons donc pas à nous soucier du refroidissement à l'intérieur du boîtier.

L'amplificateur est basé sur le circuit intégré PAM8403 produit par la firme Diodes Incorporated. Bien qu'il soit stéréo, nous n'utilisons qu'un seul canal.

Le PAM8403 permet d'obtenir, avec seulement 5 VDC, 2 x 3 W sur des haut-parleurs de 4 Ω (avec 10 % de THD). Son rendement est de 90 % avec une charge de 8 Ω et environ 85 % avec 4 Ω.

Le circuit intégré PAM8403 contient un étage préamplificateur stéréo (deux amplificateurs opérationnels par canal) et un double modulateur PWM. Chaque section étant composée d'un comparateur dont les entrées sont comparées au signal BF préamplifié et à une onde triangulaire générée par l'oscillateur interne. Chaque comparateur est doté d'une sortie différentielle qui pilote un amplificateur de puissance MOSFET configuré en pont.

Le PAM8403 contient une logique permettant à la fois l'arrêt contrôlé par la broche « /SHDN », le circuit intégré reste alimenté mais consomme très peu car l'étage de puissance, l'oscillateur et le modulateur PWM sont éteints, et la fonction « silence » des sorties via la broche « /MUTE ». Le gain des étages d'entrée est réduit jusqu'à ce que le signal soit complètement atténué.

Il faut veiller à insérer correctement les supports des circuits intégrés en disposant le détrompeur en « U » à l'emplacement indiqué sur le circuit imprimé.

Pour cela, consultez le plan de câblage des composants de l'unité logique de contrôle. En figure 3, vous pouvez voir l'unité logique de contrôle montée.

Il convient de noter, dans notre cas, que le connecteur d'alimentation est monté du côté soudure. La raison est que les connecteurs JST ont souvent une fiche femelle avec des fils rouges et noirs volants sans polarité standard. Il faudra donc souder le connecteur de manière à faire correspondre le fil rouge avec l'entrée positive +5V et le fil noir avec la masse du circuit.

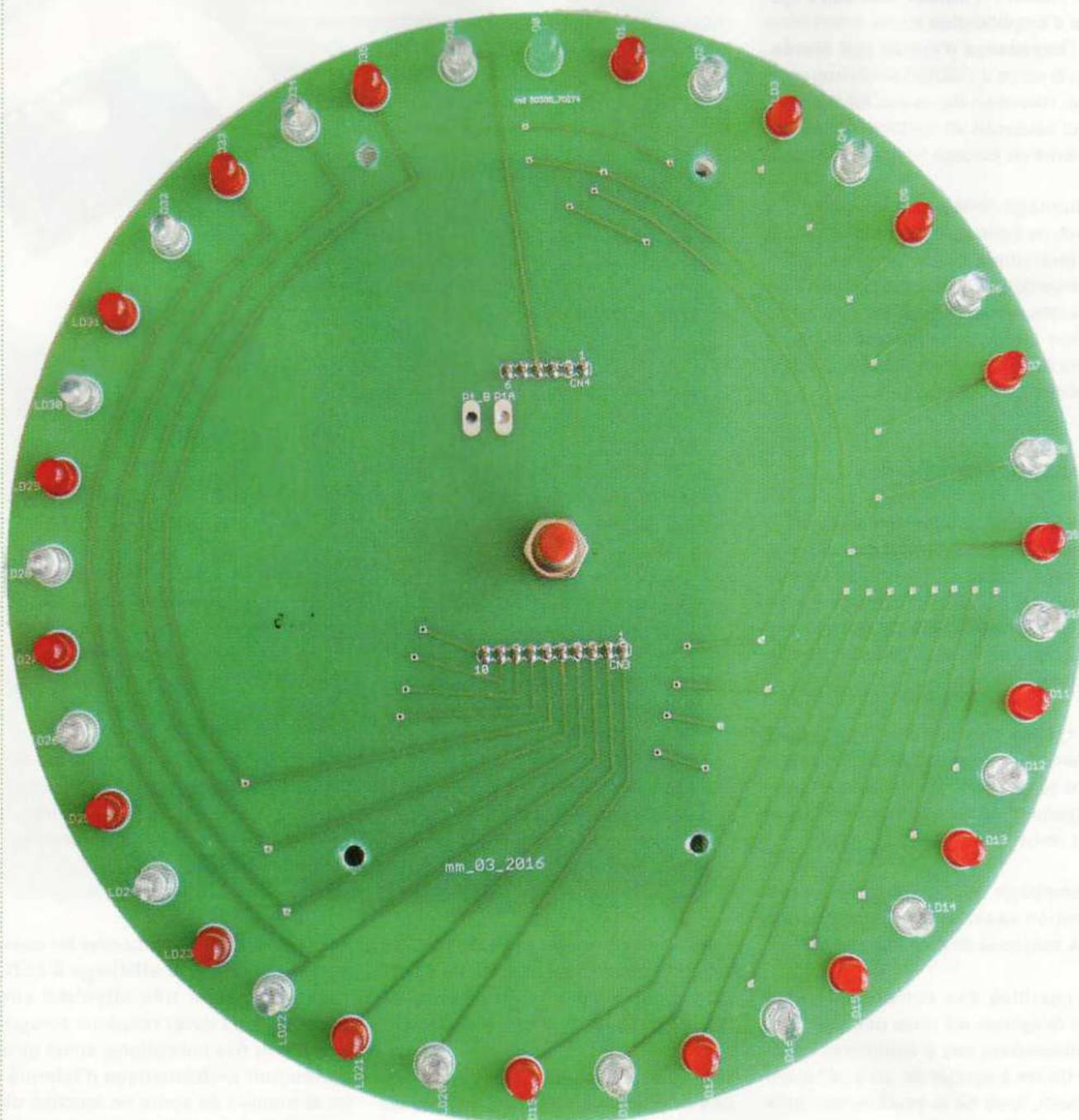
Vous pouvez ensuite procéder au montage du panneau d'affichage à LED. Vous devez faire très attention aux couleurs, vous devez respecter scrupuleusement nos indications, sinon cela deviendrait problématique d'interpréter le numéro de sortie en fonction de l'allumage de la LED correspondante.

Vous devez également faire très attention à la polarité des LED, sinon elles ne s'allumeront pas. Consultez le plan de câblage des composants du panneau d'affichage à LED.

Le critère à adopter est simple. La première LED (LD0) est la seule de couleur verte, toutes les LED avec un nombre impair (LD1, 3, 5, etc.) sont de couleur rouge.



Plan de montage du panneau d'affichage à LED

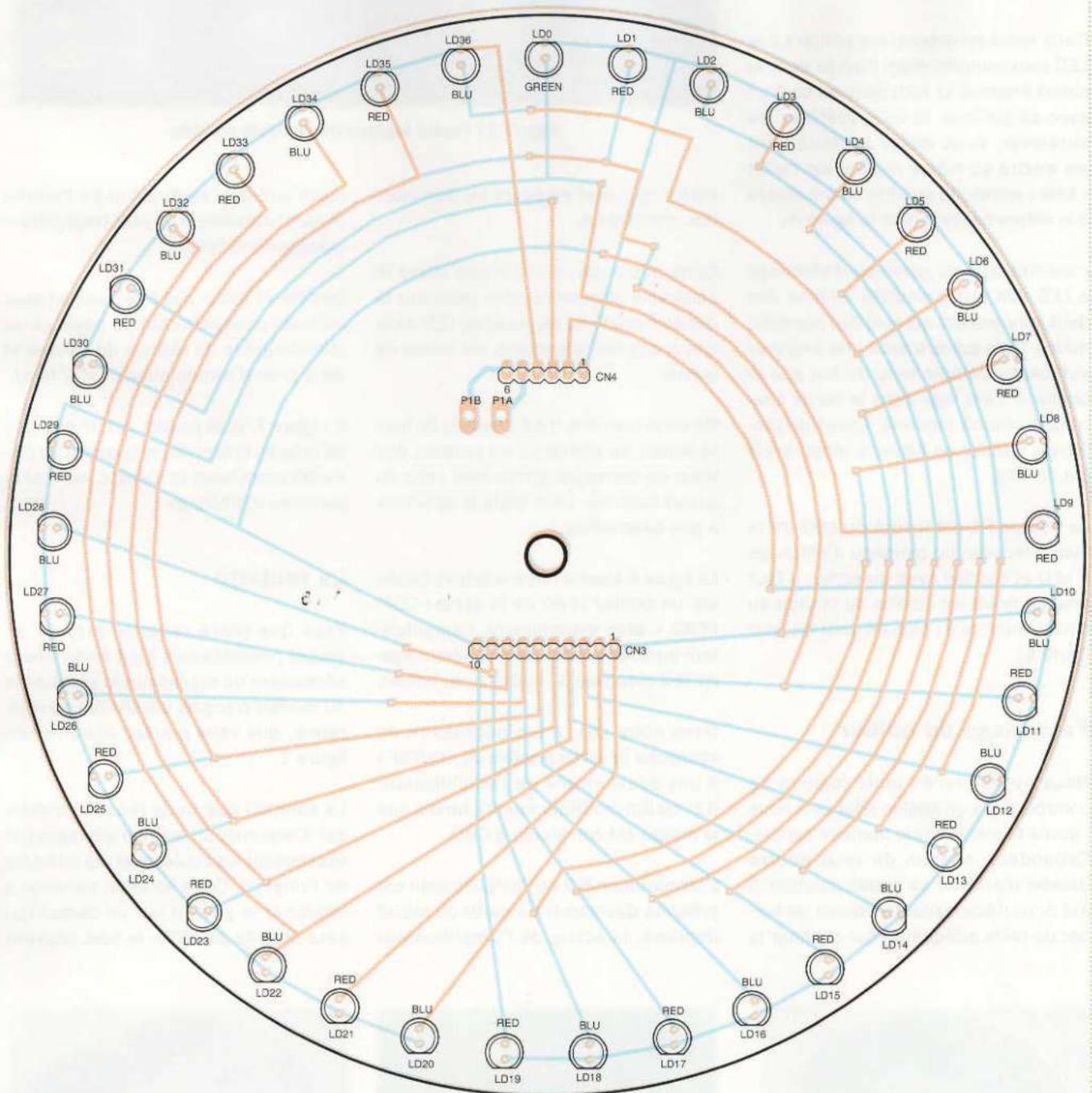


Liste des composants du panneau d'affichage à LED

LD0....LED 5 mm verte
 LD1....LED 5 mm rouge
 LD2....LED 5 mm bleue
 LD3....LED 5 mm rouge

LD4....LED 5 mm bleue
 LD5....LED 5 mm rouge
 LD6....LED 5 mm bleue
 LD7....LED 5 mm rouge
 LD8....LED 5 mm bleue
 LD9....LED 5 mm rouge
 LD10..LED 5 mm bleue
 LD11..LED 5 mm rouge

LD12..LED 5 mm bleue
 LD13..LED 5 mm rouge
 LD14..LED 5 mm bleue
 LD15..LED 5 mm rouge
 LD16..LED 5 mm bleue
 LD17..LED 5 mm rouge
 LD18..LED 5 mm bleue
 LD19..LED 5 mm rouge



LD20..LED 5 mm bleue
 LD21..LED 5 mm rouge
 LD22..LED 5 mm bleue
 LD23..LED 5 mm rouge
 LD24..LED 5 mm bleue
 LD25..LED 5 mm rouge
 LD26..LED 5 mm bleue
 LD27 ..LED 5 mm rouge

LD28..LED 5 mm bleue
 LD29..LED 5 mm rouge
 LD30..LED 5 mm bleue
 LD31..LED 5 mm rouge
 LD32..LED 5 mm bleue
 LD33..LED 5 mm rouge
 LD34..LED 5 mm bleue
 LD35..LED 5 mm rouge

LD36..LED 5 mm bleue

Divers

Barrette mâle 10 pôles
 Barrette mâle 6 pôles

Toutes les LED avec un nombre pair (LD2, 4, 6, etc.) sont de couleur bleue. Les LED rouges et bleues alternent toujours, il n'y a jamais deux LED adjacentes de même couleur.

Dans notre prototype, les boîtiers des LED sont complètement insérés dans le circuit imprimé et sont donc en contact avec sa surface. Si vous préférez les surélever, vous devez parfaitement les mettre au même niveau, car l'effet « bille » nécessite qu'elles soient toutes à la même hauteur (voir la figure 4).

L'assemblage du panneau d'affichage à LED doit être complété à l'aide des deux connecteurs qui sont des barrettes mâles. Elles doivent avoir une longueur suffisante, compte tenu du fait que le bouton P1 est logé dans la partie centrale du circuit imprimé. Celles du prototype, visibles en figure 5, mesurent 3 cm de long.

Le bouton P1 est inséré du côté de la face inférieure du panneau d'affichage à LED et est fixé avec un écrou. Il faut ensuite relier les cosses du bouton au circuit imprimé à l'aide de deux fils (voir figure 5).

Façonnage du boîtier

Nous avons placé l'unité logique de contrôle dans un boîtier, pour cela nous l'avons façonné d'une manière simple. Cependant, chacun de vous pourra décider d'adopter sa propre solution. Il est donc nécessaire de trouver un boîtier de taille adéquate pour contenir la

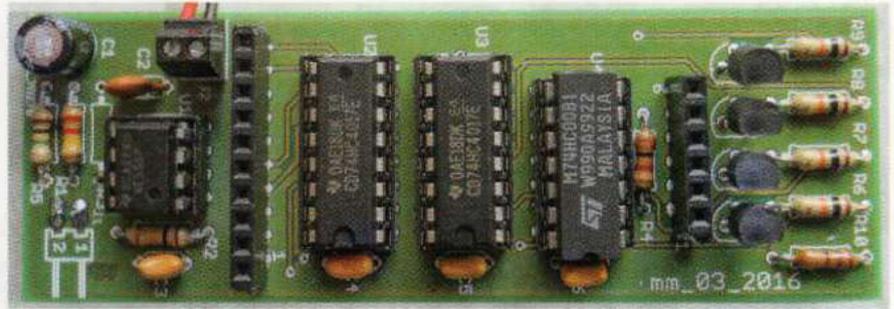


Figure 3 : l'unité logique de contrôle montée.

carte logique et recouvrir les barrettes des connexions.

Dans notre cas, nous avons retiré le couvercle et percé quatre trous sur le circuit imprimé du panneau de LED dans des zones non traversées par pistes de cuivre.

De cette manière, il est possible de fixer le boîtier au panneau, en perçant des trous en correspondance avec ceux du circuit imprimé. Ainsi toute la structure a une base solide.

La figure 6 illustre notre solution, basée sur un boîtier TEKO de la série « COFFER2 ». Bien évidemment, l'amplificateur audio et le haut-parleur seront également disposés à l'intérieur du boîtier.

Dans notre cas, il est nécessaire de connecter la sortie positive de « OUTBF » à une seule entrée de l'amplificateur (Lin ou Rin indifféremment), tandis que la masse est connectée à GND.

L'alimentation 5 V de l'amplificateur est prélevée directement à partir du circuit imprimé, la sortie de l'amplificateur

(Lout ou Rout, en fonction de l'entrée utilisée) est reliée à un petit haut-parleur placé dans le boîtier.

Un côté du boîtier doit être percé en deux endroits pour permettre le passage du potentiomètre de réglage du volume et de la prise d'alimentation (5 VDC max).

En figure 7, vous pouvez voir un exemple de mise en boîtier de l'ensemble des éléments constituant la roulette, excepté le panneau d'affichage.

La roulette

Pour que notre roulette prenne un aspect professionnel, il est évidemment nécessaire de reproduire le graphisme du modèle français utilisé comme référence, que vous pouvez observer en figure 1.

La solution que nous recommandons est d'imprimer un modèle qui reproduit exactement les couleurs et les numéros de l'original. Cette solution consiste à imprimer le gabarit sur un carton qui sera ensuite plastifié, le tout pouvant

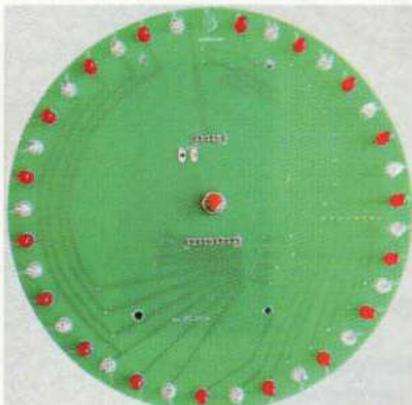


Figure 4 : le panneau d'affichage à LED et le bouton START.

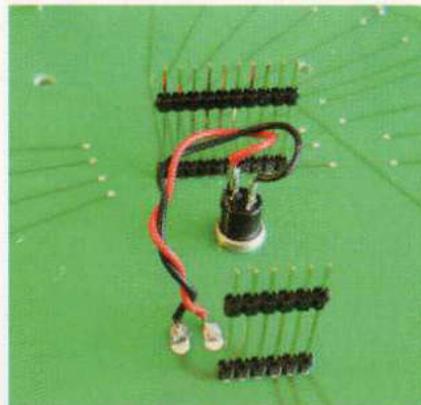


Figure 5 : le panneau d'affichage à LED du côté de sa face inférieure.



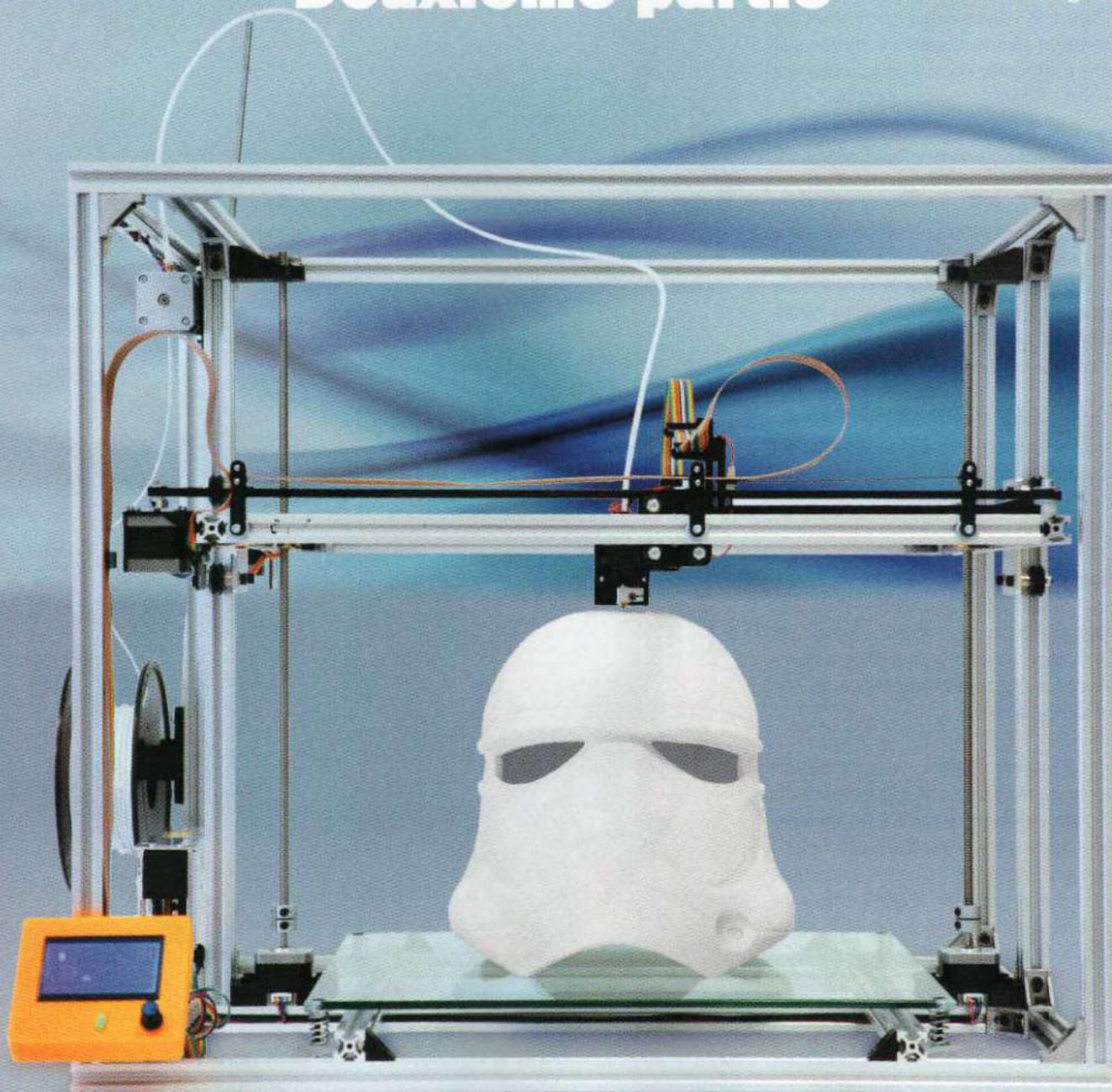
Figure 6 : le boîtier fixé sur la face inférieure du panneau d'affichage à LED.

La 3DRAG+

Imprimez en grand !

Deuxième partie

de Gabriele Daghetta



Après avoir abordé, dans le précédent numéro, le montage de la 3DRAG+ qui est capable d'imprimer des objets de dimensions 40 cm x 40 cm x 40 cm, nous allons compléter notre imprimante d'un plateau chauffant et d'une carte de contrôle autonome, puis nous décrirons la configuration logicielle de l'imprimante. Seconde et dernière partie.

Imprimer des objets de grande taille en plastique est devenu aujourd'hui facilement réalisable et peu coûteux. Nous vous l'avons montré dans le précédent numéro 145 d'Électronique et Loisirs Magazine en vous présentant notre nouvelle imprimante 3D à technologie FDM.

Cette dernière est capable de réaliser des impressions d'objets solides de dimensions maximales de 400 mm x 400 mm x 400 mm dans divers types de matières plastiques (PLA, ABS, etc.). Il est ainsi possible de fabriquer un objet de grandes dimensions à un coût très limité, nous sommes sûr que cela intéressera un grand nombre de nos lecteurs.

Dans le numéro précédent, nous avons présenté et décrit les principaux aspects de la 3DRAG+, tout en vous expliquant



Figure 1 : la CTN avec la gaine thermorétractable.

comment la monter. Dans cette seconde partie, nous allons développer le sujet en expliquant comment implémenter des modules complémentaires qui optimisent l'imprimante.

En particulier, nous allons décrire l'installation du plateau chauffant qui est très utile pour l'impression d'objets volumineux avec des matériaux qui ont tendance à « se rétracter » comme l'ABS.

Puis nous allons décrire l'installation et la configuration logicielle, et comment adapter le firmware Marlin pour démarrer l'impression seulement quand une certaine température est atteinte.

En outre, nous vous proposerons quelques suggestions pour ceux qui souhaitent personnaliser l'imprimante, tels que le mode micro-pas et les paramètres du firmware, mais aussi la description de la carte de contrôle pour une utilisation autonome de l'imprimante 3DRAG+.

L'installation du plateau chauffant

Nous n'allons pas expliquer le rôle du plateau chauffant, car nous avons déjà abordé le sujet lors de la description de la 3DRAG. Disons simplement qu'en maintenant le plateau d'impression à une certaine température, nous sommes en

mesure de rendre la solidification du filament déposé plus progressive et de contrecarrer le phénomène de courbure des premières couches déposées.

Le plateau chauffant de la 3DRAG+ s'adapte aux dimensions plus grandes du plateau d'impression. Ses dimensions sont donc de 40 cm x 40 cm, de manière à fournir une chaleur homogène de l'ensemble de la zone d'impression. La nécessité de chauffer une surface plus grande que celle de la 3DRAG (quatre fois plus, pour être exact ...) implique que la puissance doit être ajustée, c'est-à-dire quadruplée.

Dans notre cas, nous parlons d'un dispositif chauffant en forme de serpentín en cuivre flexible et isolé des deux côtés par une feuille de kapton.



Figure 2 : la CTN placée entre 2 pistes de cuivre de l'élément chauffant.

Il fonctionne sous une tension de 12 V, et permet de fournir une puissance de 240 W, ce qui correspond à une consommation de courant nominale de 20 A ($240 \text{ W} / 12 \text{ V} = 20 \text{ A}$).

Les tests effectués sur le prototype de la 3DRAG+ équipé de ce dispositif chauffant ont montré une consommation à froid de 21,2 A et à chaud une consommation de 20,4 A.

Il est clair que le courant est plus grand à froid car la résistance électrique de tout conducteur (donc aussi celui constituant

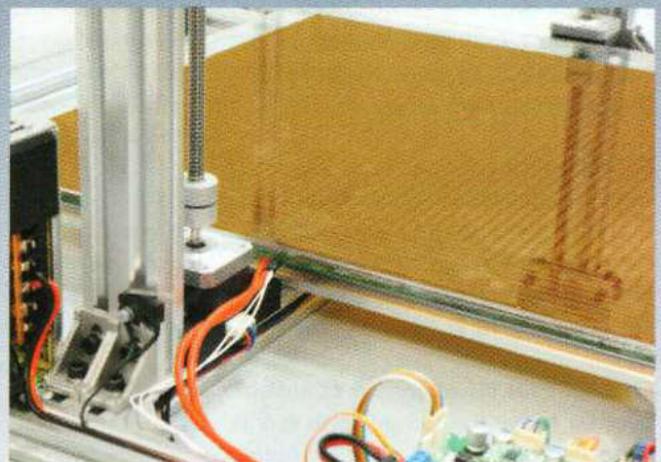


Figure 3 : coupez les 2 fils de l'élément chauffant à environ 20 cm.

le dispositif chauffant à base de cuivre) augmente avec la température.

Le courant à froid est important pour déterminer la puissance de l'alimentation de la 3DRAG+ (qui est de 29 A) et d'évaluer comment cela pourrait affecter le fonctionnement de l'imprimante, tandis qu'à chaud ce qui nous intéresse est le courant consommé par les moteurs pas à pas et le chauffage de l'extrudeuse (fonctionnement en cours d'impression).

Les tests et les mesures effectués ont montré que l'alimentation le permettait, du moins lorsque le contrôle du dispositif chauffant s'effectue sous la forme PWM.

Cependant, le courant du plateau chauffant, comme mentionné dans l'article précédent, doit être commuté par un relais statique et la sortie relative de la carte contrôleur (HEATER 2) ne sera utilisée que pour fournir la tension de commande. Le SSR commutera alors le circuit d'alimentation du chauffage.

Pour chauffer le plateau d'impression, il est nécessaire d'appliquer le dispositif chauffant, qui est un produit adhésif de la société 3M, sur le côté opposé à l'endroit où l'objet sera imprimé, c'est-à-dire sur la face inférieure de la plaque de verre. Mais avant de l'installer, nous devons d'abord fixer la CTN au dispositif chauffant avant de l'appliquer sur la plaque de verre.

Prenez la CTN et repliez délicatement ses broches (sans les casser). Raccourcissez-les avec une pince coupante et laissez environ 10 mm. Soudez ensuite chaque borne à un morceau de câble de 50 cm avec un revêtement en téflon.

Recouvrez chaque soudure avec un morceau de gaine thermorétractable. Rétrécissez le diamètre des gaines à l'aide d'un séchoir ou d'un fer à souder. Vous devez obtenir un résultat semblable à celui de la figure 1.

Fixez la CTN au dispositif chauffant avec du ruban Kapton de sorte que le corps reste entre deux pistes de cuivre, comme indiqué en figure 2. Pour assurer un meilleur transfert thermique, enrobez

Figure 4 : les cosses serties et les fils étamés.



le corps de la CTN avec de la pâte thermique avant de la fixer avec le ruban Kapton. Ensuite, coupez les 2 conducteurs du dispositif chauffant à environ 20 cm (voir la figure 3).

Insérez un capuchon dans chaque fil puis disposez deux cosses FASTON (rouge et noire) et avec une pince à sertir, sertissez les cosses. Puis, dénudez l'extrémité opposée de chaque fil sur environ 5 mm et étamez-la comme indiqué en figure 4.

Placez ensuite la tête d'impression (extrudeuse) en position « HOME ». À l'aide d'un marqueur à pointe fine, marquez le point de contact de l'extrudeuse sur la plaque de verre. Identifiez clairement l'angle de la position « HOME » (voir la figure 5).

Retirez maintenant la plaque de verre de l'imprimante en dévissant les 4 vis de fixation à tête fraisée M5 et nettoyez soigneusement les surfaces opposées à la surface d'impression avec de l'alcool isopropylique.

Placez le dispositif chauffant sous la plaque de verre (avec l'angle de « HOME » sur le coin inférieur gauche) en veillant à ce que les câbles de raccordement sortent du côté gauche, exactement comme indiqué en figure 6. Alignez le coin du dispositif chauffant avec le point de contact de la tête en position « HOME » et fixez-le avec du ruban adhésif (voir la figure 7).

Assurez-vous que les bords du dispositif chauffant sont parallèles à ceux de la plaque de verre, puis fixez la partie centrale du dispositif chauffant avec du ruban adhésif (voir la figure 8).

Découlez le film protecteur du dispositif chauffant aussi loin que possible (voir la figure 9). À l'aide d'un cutter, coupez le film protecteur afin de l'enlever (voir la figure 10).



Figure 5 : marquage sur la plaque de verre du point de contact de l'extrudeuse.



Figure 6 : ici les câbles de l'élément chauffant sortent du côté gauche.



Figure 7 : alignement du coin du dispositif chauffant avec le point de contact de la tête en position « HOME ».

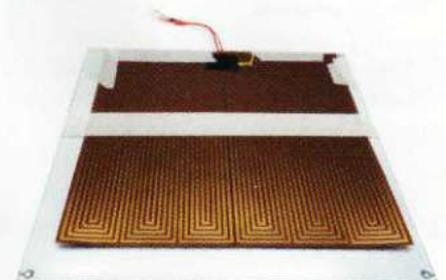


Figure 8 : fixation de la partie centrale de l'élément chauffant avec du ruban adhésif.

NB : lors de l'application de l'élément chauffant adhésif, assurez-vous d'abord que la surface inférieure de la plaque de verre soit parfaitement propre, éventuellement nettoyez-la avec de l'alcool isopropylique, afin d'éviter toute encrassement, ne la touchez pas avec les doigts.

Assurez-vous que le dispositif chauffant adhère peu à peu à la plaque de verre, en veillant à ne pas laisser des bulles d'air se former entre les deux éléments. Par exemple, utilisez le corps d'un surligneur pour faciliter l'opération ou tout autre objet non tranchant (voir la figure 11).

Effectuez les mêmes opérations, décrites ci-dessus, pour la partie restante. À la fin de l'opération, vous devez obtenir un résultat semblable à celui de la figure 12, où vous pouvez voir le dispositif chauffant correctement installé.

Prenez un connecteur bipolaire visible en figure 13 et coupez les 2 fils à environ 10 mm du connecteur. Insérez chaque conducteur de la CTN (câbles blancs) dans un morceau de gaine thermorétractable, puis soudez les extrémités (quelle que soit leur polarité) aux fils du connecteur bipolaire.

Recouvrez les soudures de chaque fil avec les morceaux de gaine et chauffez-les. Il est conseillé d'identifier le connecteur avec l'inscription « TH2 » (THERM2) afin de le distinguer de celui de la CTN de la tête d'impression. Vous devez obtenir un résultat semblable à celui de la figure 14.

Retirez la résistance présente sur les broches du connecteur « THERM2 » de la carte de contrôle, puis insérez à la place le connecteur (que vous venez de fabriquer) de la CTN de l'élément chauffant (voir la figure 15).

Installation du relais statique de l'élément chauffant

Afin de limiter les pertes par effet Joule (inévitables compte tenu du courant mis en jeu), il est recommandé de disposer le relais statique à proximité de l'alimentation afin de limiter la longueur des câbles. Rappelez-vous que le relais doit dissiper une certaine quantité de chaleur, il est donc installé sur un dissipateur dont

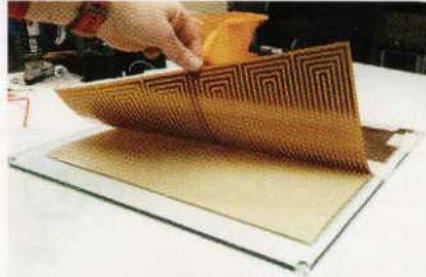


Figure 9 : décollage du film protecteur.



Figure 11 : utilisez un objet non tranchant pour étaler correctement l'adhésif sans laisser des bulles d'air se former.



Figure 14 : soudez les extrémités de la CTN au connecteur bipolaire avec de la gaine thermorétractable. Inscrivez « TH2 ».

la résistance thermique doit être d'au moins 3 °C/W. Cela veut dire que pour 1 W dissipé la température du radiateur s'élèvera de 3 °C par rapport à celle du milieu ambiant.

Le relais statique est fixé au châssis de l'imprimante à l'aide de vis métalliques. La fixation se situe sur l'un des profilés de la base. Evitez de disposer le relais statique sur le montant, car la chaleur produite risque de le déformer. Cela pourrait altérer la précision du mouvement d'élévation de la tête d'impression (axe Z par rapport au cadre X/Y).

Les **connexions du relais statique vers l'alimentation** et vers **l'élément chauffant nécessitent chacune un câble de**

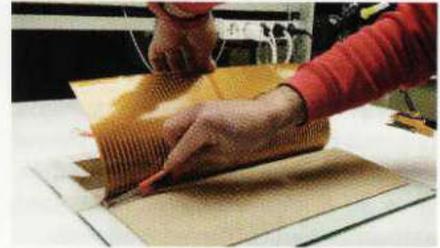


Figure 10 : découpage du film protecteur à l'aide d'un cutter.

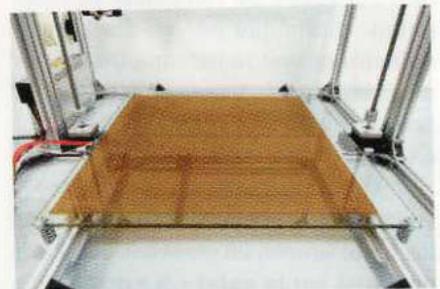


Figure 12 : le dispositif chauffant correctement installé.



Figure 13 : le connecteur bipolaire.

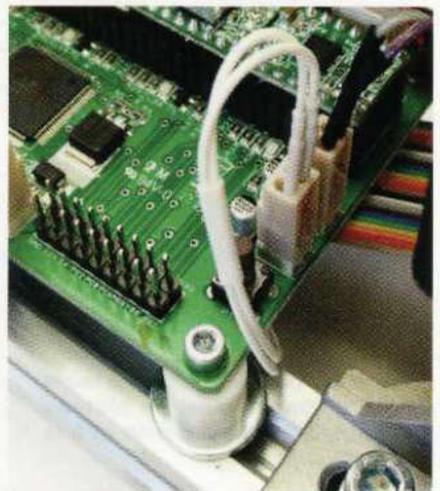


Figure 15 : reliez le connecteur « TH2 » aux broches « THERM2 » de la carte de contrôle.

section d'au moins 1,5 mm². Ne dépassez pas une longueur de câble de 40 cm.

Concernant la commande du relais statique, vous pouvez choisir n'importe quel câble, même plat, d'au moins 0,25 mm² de section. Il faut cependant respecter la polarité, le « + » de « HEATER 2 » doit être

relié au « + » de l'entrée et donc le « - » de « HEATER 2 » au « - » de l'entrée.

Le schéma électrique du relais statique, que vous pouvez voir en figure 16, permet de contrôler un courant maximal d'environ 30 A, même à des températures ambiantes particulièrement élevées.

Si une charge nécessitant un courant plus élevé doit être gérée, il peut s'avérer nécessaire de remplacer le dissipateur thermique par un dissipateur présentant une résistance thermique plus faible, car le transistor MOSFET Q1 pourrait atteindre des températures très élevées.

Le fonctionnement du montage est relativement simple, un **niveau logique bas appliqué sur le point « N »** de l'entrée provoque la **conduction** du transistor **T1**. Ce dernier **polarise la grille du MOSFET Q1 qui se met à conduire** en faisant circuler le courant de « OUT+ » vers « OUT- ».

Le cavalier **JP1** permet de choisir la **logique de fonctionnement du signal de contrôle**, c'est-à-dire soit le **signal qui active** le relais statique est **positif**

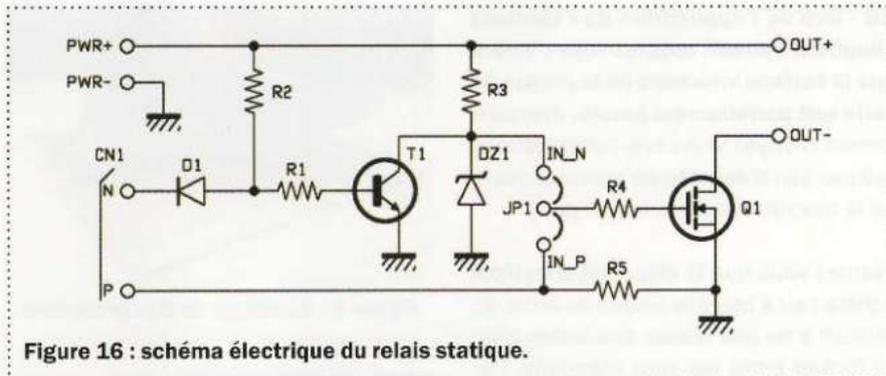


Figure 16 : schéma électrique du relais statique.

(position « IN_N »), soit le signal qui active le relais statique est **positif** (position « IN_P »).

Par conséquent, en fonction du choix effectué, il sera nécessaire d'adapter le signal de commande au niveau du connecteur CN1. Plus précisément, si « IN_N » est sélectionné, le signal de commande doit être relié à la borne « N » du connecteur CN1, dans le cas contraire le signal de commande doit être relié à la borne « P » de CN1 (voir la figure 17).

Coupez les fils de l'élément chauffant à une longueur telle qu'ils soient au niveau

des cosses FASTON du circuit imprimé (« OUT+ » et « OUT- »), puis sertissez les cosses avec des fils de couleur rouge et noire. La cosse FASTON rouge doit être connectée à « OUT+ », tandis que la noire à « OUT- » (voir la figure 17).

De même, à l'aide de deux autres câbles rouge et noir et des cosses FASTON, connectez à la borne +12VDC de l'alimentation le câble de couleur rouge et au -12VDC le câble noir. Reportez-vous aux figures 17 et 18.

Ensuite, fixez le circuit imprimé du relais statique sur le profilé du cadre de l'imprimante (du côté de la carte de

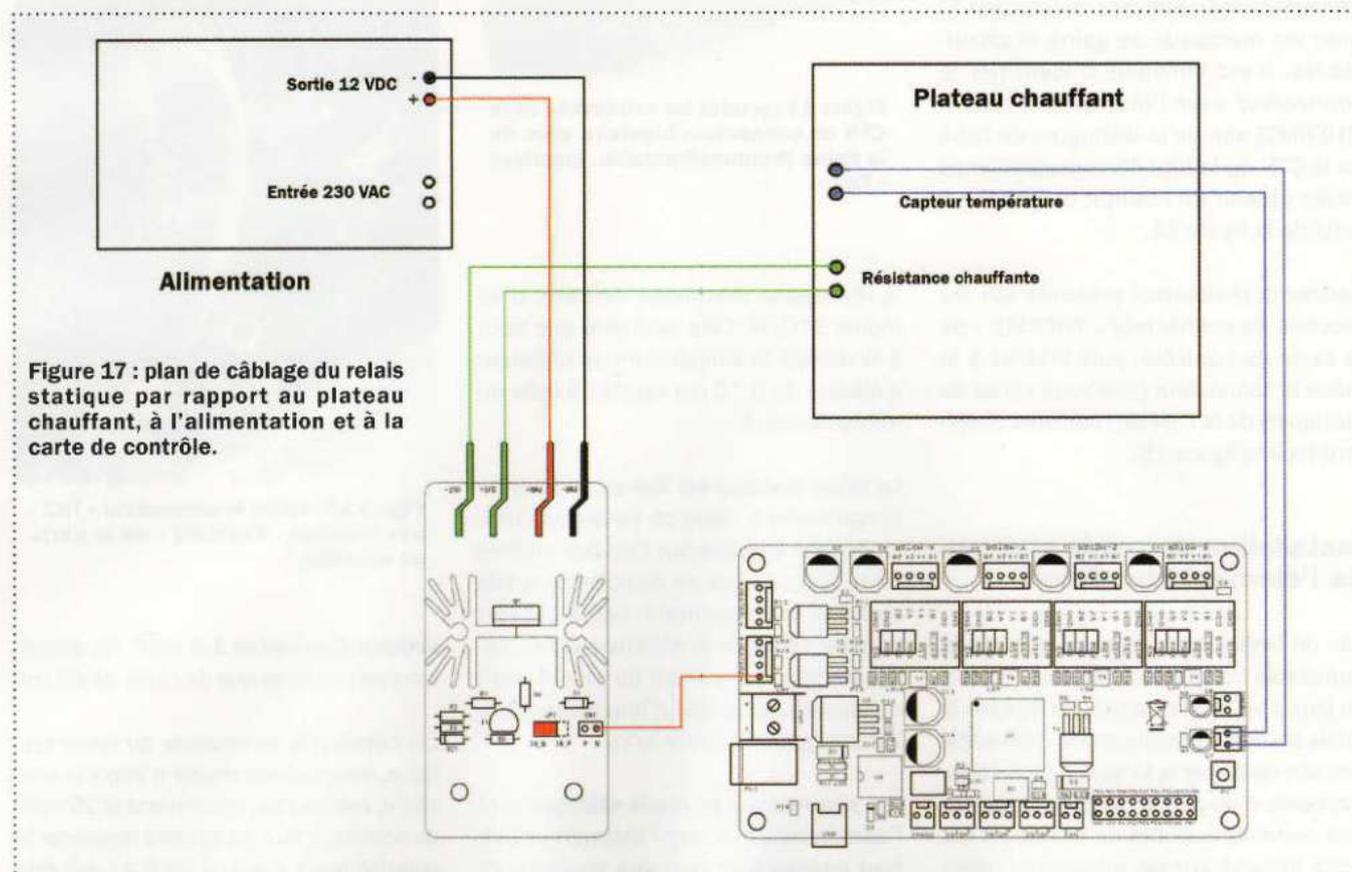


Figure 17 : plan de câblage du relais statique par rapport au plateau chauffant, à l'alimentation et à la carte de contrôle.

contrôle) à l'aide de 2 vis M3 x 6 et de 2 écrous M3 (voir la figure 19).

Il ne vous reste plus qu'à positionner le cavalier en fonction de votre choix. Par exemple, sur la position « IN_N », vous devez connecter la broche « N » de CN1 sur la broche « - » du connecteur « HEATER2 » de la carte de contrôle comme visible en figure 20 (câble marron).

Premiers tests de la DRAG+ et installation du logiciel Repetier Host

Après avoir vérifié le câblage, connectez le câble USB à l'imprimante et le bloc d'alimentation au secteur. Reliez l'autre extrémité du câble USB au PC.

Mettez sous tension, vous devez voir les 2 LED (rouge et verte) s'allumer.

À ce stade, vous devez installer le driver FTDI en créant un nouveau port série. Téléchargez le driver en allant à l'adresse suivante :

<https://www.ftdichip.com/Drivers/VCP.htm>

Il existe plusieurs types de drivers pour différents systèmes d'exploitation, dans

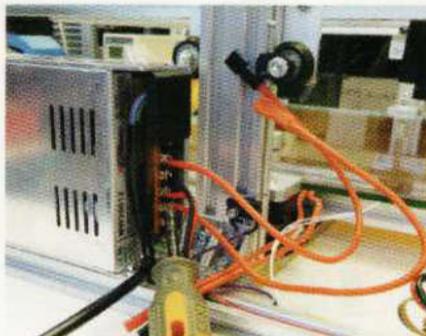


Figure 18 : les cosses FASTON reliées à l'alimentation.

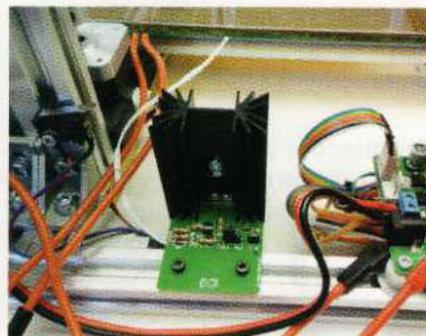


Figure 19 : fixation du relais statique sur le châssis de l'imprimante.

notre cas nous choisissons la version Windows 64 bits 2.12.28.

Dans votre cas, cela peut être différent en fonction de votre ordinateur, cependant la procédure d'installation est globalement identique pour les autres systèmes d'exploitation. Une fois le fichier « zip » téléchargé, décompressez-le dans un dossier que vous nommerez par exemple « FTDI ».

Allez dans le **gestionnaire de périphériques** de Windows, vous devez voir apparaître à l'onglet « Ports (COM et LPT) » un nouveau port dénommé « **USB Serial Port (COM9)** ». Il se peut que dans votre cas à la place de « COM9 » il peut y avoir « COM5 » ou un autre chiffre.

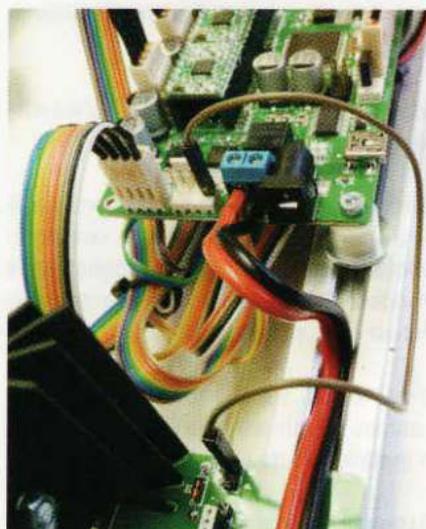
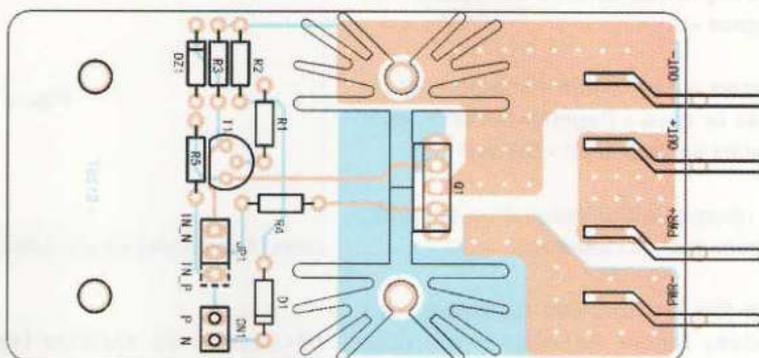


Figure 20 : le câble marron relie la broche « N » de CN1 à la broche « - » du connecteur « HEATER2 ».

Plan de montage du relais statique

Schéma de câblage des composants du relais statique.



Liste des composants du relais statique

R1.....47 kΩ
R2.....47 kΩ
R3.....1 kΩ
R4.....100 Ω
R5.....47 kΩ

DZ1.....zener 15V/400mW
D1.....BAT85
T1BC547
Q1.....IRFPO64NPBF

Divers

Dissipateur 3°C/W
Cavalier

Barrette mâle 3 pôles
Barrette mâle 2 pôles
Cosses FASTON pour circuit imprimé (x4)

Vis 12mm 3 MA
Ecrou 3 MA
Vis pour fixation sur cadre (x2)
Ecrou pour fixation sur cadre (x2)
Pâte thermique

Faites un clic droit sur le « **USB Serial Port** » et choisissez « **Mettre à jour le pilote** » (voir la figure 22).

Sélectionnez alors « **Rechercher un pilote sur mon ordinateur** », puis utilisez le bouton « **Parcourir** » pour chercher votre dossier « **FTDI** ». Ensuite, cliquez sur « **Suivant** », Windows installe automatiquement le driver.

Le driver de l'imprimante est alors correctement installé, vous devez maintenant télécharger et installer le logiciel Repetier Host. Pour cela allez à l'adresse suivante :

<https://www.repetier.com/download-now/>

Choisissez la version correspondant à votre ordinateur, dans notre cas nous cliquons sur le bouton « Repetier-Host Windows 2.1.3 » et nous téléchargeons le fichier « setupRepetierHost_2_1_3.exe ».

Ensuite, nous exécutons ce fichier en tant qu'administrateur. La procédure d'installation de Repetier Host débute.

Attention, pendant la phase d'installation vous devez avoir une connexion internet active.

Sélectionnez la langue (Français par défaut) puis cochez dans la fenêtre suivante l'option « **Je comprends et j'accepte les termes du contrat de licence** ».

Cliquez sur le bouton « Suivant », **décochez la case** « Repetier-Server », puis cliquez sur le bouton « Suivant ».

Le récapitulatif d'installation apparaît, cliquez sur « **Installer** ».

Une fois l'installation terminée, vous pouvez lancer Repetier Host, vous devez obtenir une fenêtre semblable à celle de la figure 23. Vous pouvez cliquer sur le bouton « **Connecter** » en haut à gauche, **il doit changer de couleur en bleu**. Cela prouve bien que le driver de l'imprimante est correctement installé et qu'elle est reconnue. Déconnectez l'imprimante et fermez Repetier Host.

Avant de pouvoir déplacer les axes avec le logiciel Repetier Host, il est

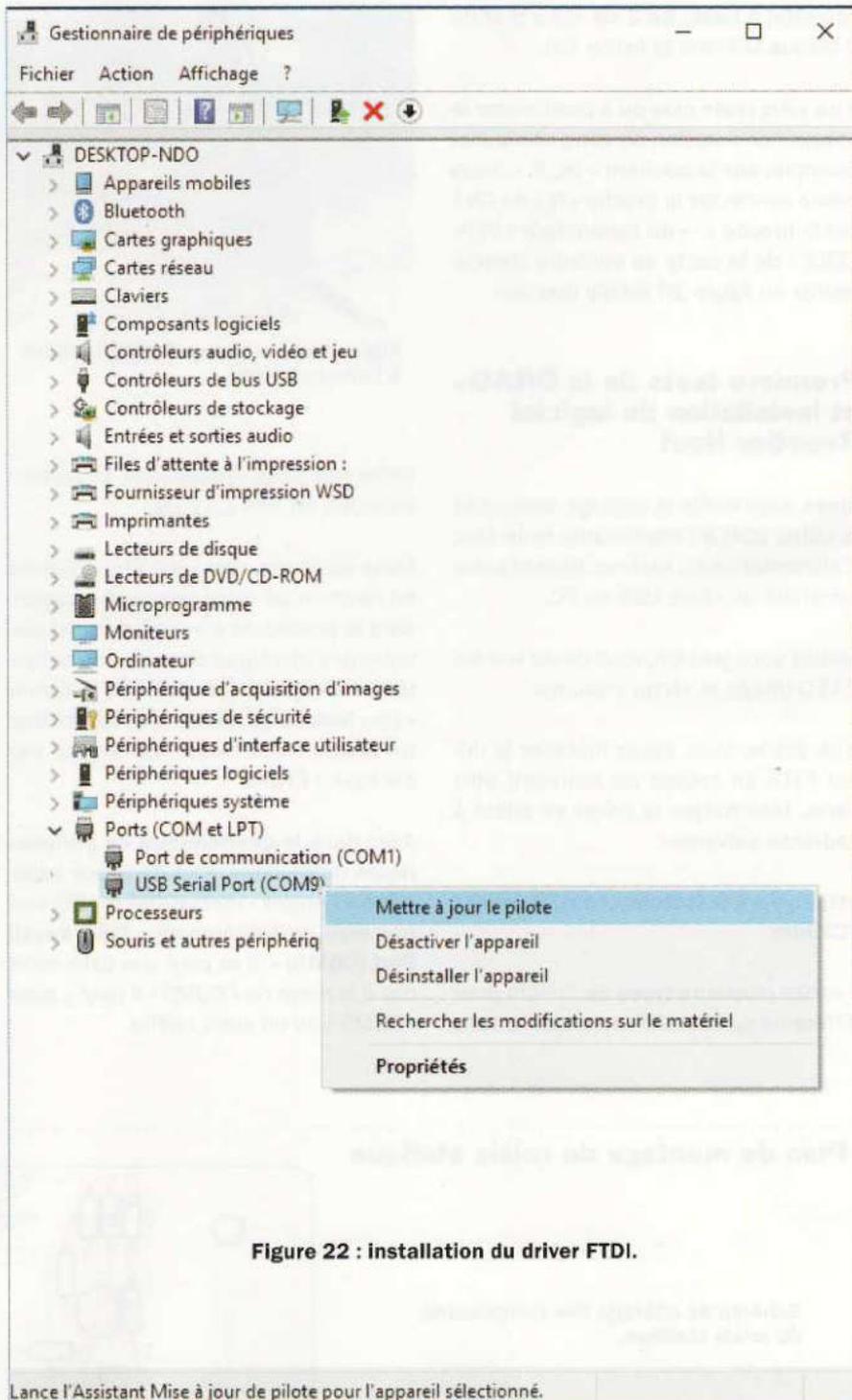


Figure 22 : installation du driver FTDI.

nécessaire de **calibrer les drivers des moteurs** montés sur la carte de contrôle. Pour cette opération, il est nécessaire de disposer d'un multimètre calibré sur l'échelle 2 VDC (continu) et d'un tournevis cruciforme miniature en plastique si possible. Mettez l'imprimante sous tension, reliez le fil noir du multimètre à la masse de la carte de contrôle et le fil rouge sur le point test du **moteur X**, comme visible en figure 24.

À l'aide du tournevis, ajustez le trimmer jusqu'à obtenir une tension d'environ **0,415 V à 0,420 V**.

Assurez-vous que la valeur reste stable, même après avoir retiré le tournevis. Sinon, agissez de nouveau sur le trimmer.

Répétez la même opération avec la même valeur de tension pour les drivers des **moteurs de l'axe Y** et de l'extrudeuse.

La procédure est similaire pour le driver du moteur de l'axe Z (sur la figure 25, le 3^{ème} en partant de la gauche), mais dans ce cas, il est nécessaire de régler une valeur de tension plus élevée, exactement **0,66 VDC**.

Maintenant, lancez **Repetier Host** et cliquez sur l'icône « **Réglages Imprimante** » (en haut à droite) afin de créer le profil de la 3DRAG+.

Dans le champ « Imprimante ; », écrivez le nom 3DRAG+, puis à l'onglet « Connexion » entrez les valeurs suivantes (voir la figure 26) :

- Connecteur : **Connexion Serie** ;
- Port : COM9 (cela peut être différent) ;
- Vitesse : **250 000**
- Protocole de Transfert : Autodetect ;
- Cache de réception : 127 ;
- Erreur de communication : 40 s.

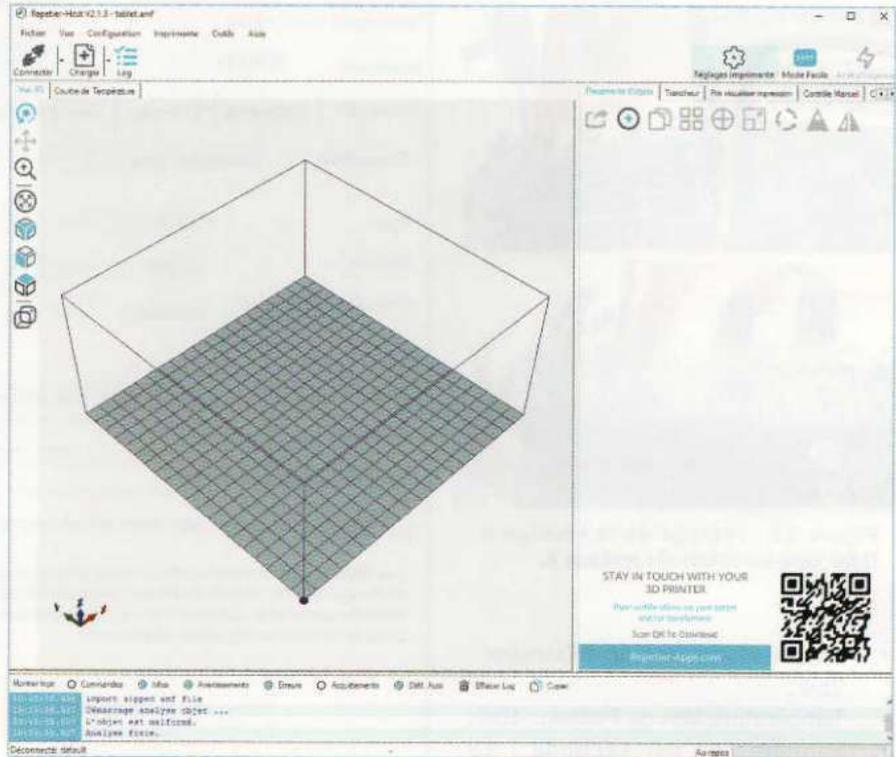


Figure 23 : fenêtre principale de Repetier Host.

Cliquez ensuite sur le bouton « **Appliquer** ». Sélectionnez l'onglet « **Imprimante** » et entrez les valeurs suivantes (voir la figure 27) :

- Firmware Type : Autodetect ;
- Vitesse de déplacement : 4800 mm/min ;
- Vitesse max axe Z : 450 mm/min ;
- Vitesse d'extrusion manuelle : 2 et 20 mm/s (il y a 2 champs à remplir) ;
- Vitesse de rétractation manuelle : 30 mm/s ;
- Température par défaut de l'extrudeur : 205 °C ;
- Température par défaut du plateau : 56 °C.

Décochez les cases « Enlever les requêtes M105 », « Allez à la position de parcage après Fin/Arrêt », « Axe Z » et « Inverser X and Y ».

Les autres cases doivent être cochées, enfin entrez la valeur 22 % dans le champ « Majorer temps d'impr. », puis cliquez sur « Appliquer ».

Ensuite, sélectionnez l'onglet « **Extruder** » et entrez les valeurs suivantes, puis cliquez sur « Appliquer » (voir la figure 28) :

- Nombre d'extruders : 1 ;
- Nombre de ventil : 1 ;

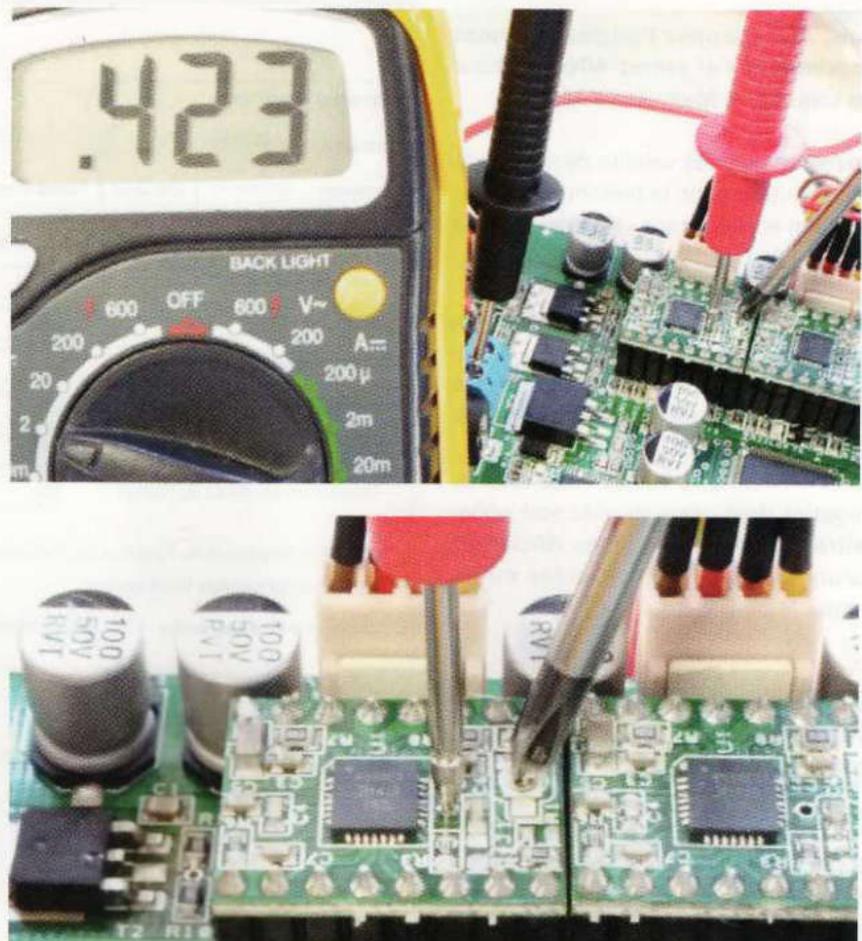


Figure 24 : réglages des tensions à 0,420 V des drivers des moteurs X, Y et de l'extrudeuse.

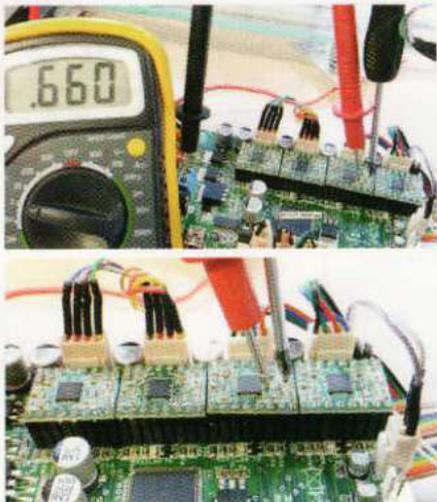


Figure 25 : réglage de la tension à 0,66 VDC du driver du moteur Z.

- Température Max de l'Extruder : 250 ;
- Température Max du Plateau : 100 ;
- Volume Max par seconde : 12 mm³/s ;
- Diamètre extruder : 0,4 mm (le choix de la couleur est à votre discrétion).

Puis, sélectionnez l'onglet « **Format imprimante** » et entrez **400 mm** pour les valeurs « **X Max** » et « **Y Max** ».

De même pour les valeurs de la largeur, de la hauteur et de la profondeur entrez 400 mm et cliquez sur « **Appliquer** » (voir la figure 29).

L'étape suivante consiste à tester la communication entre le logiciel Repetier Host et l'imprimante. Appuyez sur le bouton « **Connecter** » en haut à gauche.

Dans la fenêtre du bas de l'interface de Repetier Host, vous devriez voir apparaître une série de lignes déclarant qu'une 3DRAG+ est connectée au firmware Marlin.

Réglage du plateau d'impression et du fin de course de l'axe Z

Dans un premier temps, vissez la vis de fin de course de l'axe Z pendant environ 10 tours, afin que le chariot Z ne heurte pas la vitre (voir la figure 30).

Utilisez une clé Allen pour régler les 4 vis de fixation de la plaque de verre de sorte

Paramètres imprimante

Imprimante: 3DRAG+

Connexion | Imprimante | Extrudeur | Format imprimante | Scripts | Avancé

Connecteur: Connection Série Aide

Port: COM9

Vitesse: 250000

Protocole de Transfert: Autodetect

Reset en cas d'urgence: Envoyer Commande d'Urgence et Reconnecter

Cache de réception: 127

Erreur de Communication: 40 [s]

Utilisez communication aller/retour (envoi après ok)

Les réglages d'imprimante correspondent toujours à l'imprimante sélectionnée ci-dessus. Ils sont stockés à chaque appui sur OK ou appliquer. Pour créer une nouvelle imprimante, entrez un nom et créer appliquer. L'imprimante démarre avec la dernière configuration sélectionnée

OK Appliquer Annuler

Figure 26 : paramètres de l'onglet « Connexion ».

Paramètres imprimante

Imprimante: 3DRAG+

Connexion | Imprimante | Extrudeur | Format imprimante | Scripts | Avancé

Firmware Type: Autodetect

Vitesse de déplacement: 4800 [mm/min]

Vitesse max axe Z: 450 [mm/min]

Vitesse d'Extrusion Manuelle: 2 20 [mm/s]

Vitesse de Rétraction Manuelle: 30 [mm/s]

Température par défaut de l'extrudeur: 205 °C

Température par défaut du Plateau: 56 °C

Vérifier température du Plateau et de l'Extrudeur

Enlever les requêtes M105 des Log

Vérifier toutes les 3 secondes.

Position de parcage: X: 0 Y: 0 Z min: 0 [mm]

Envoyer le temps restant sur l'afficheur Aller à la position de parcage après Fin/Arrêt d'ur

Désactiver extrudeur après Fin/Arrêt d'urgence Désactiver Plateau chauffant après Fin/Arrêt d'un

Désactiver moteurs après Fin/Arrêt d'urgence L'imprimante possède une carte SD

Majorer temps d'impr. de 22 [%]

Inverser le contrôle de Direction pour Axe X Axe Y Axe Z Inverser X and Y

OK Appliquer Annuler

Figure 27 : paramètres de l'onglet « Imprimante ».

que tous les angles soient distants d'environ 13 mm du profilé en aluminium situé en-dessous. Avec cette distance, le plateau doit être suffisamment ferme.

Si nécessaire, réduisez la valeur pour augmenter la rigidité, mais gardez à l'esprit qu'une contrainte excessive du ressort ne sert à rien et pourrait casser le bord de la plaque de verre.

Lancez Repetier Host et connectez-vous à l'imprimante. Sélectionnez l'onglet « **Contrôle Manuel** » à droite de l'interface. Les commandes de cette section permettent de gérer les mouvements des axes, de l'extrudeuse, de la vitesse ainsi que le chauffage du plateau et de la tête d'impression (voir la figure 31).

Appuyez sur le bouton « X HOME » (le bouton avec le symbole de la maison et la lettre X).

Le chariot X doit se déplacer vers la position « HOME » qui correspond au profilé sur lequel les moteurs X et Y sont fixés (si cela ne se produit pas, vérifiez que l'imprimante est alimentée et éventuellement que la connexion est effective).

Appuyez maintenant sur le bouton « Y HOME », la tête doit se déplacer vers le moteur X.

Appuyez ensuite sur les touches fléchées « X/Y » pour déplacer les axes X et Y jusqu'à la limite de leur course (400 mm). Vérifiez la couverture de la zone d'impression (elle doit être maximale) et que tout peut se déplacer librement sans rencontrer d'obstacles.

Utilisez les touches fléchées pour positionner la tête d'impression au centre de la plaque (+200X, + 200Y).

En supposant que le cadre du chariot Z soit parfaitement parallèle à la base du cadre, appuyez sur la touche « Z Home » pour abaisser le chariot Z jusqu'à sa limite inférieure.

Si la buse ne touche pas la plaque de verre, tournez la vis de fin de course Z de quelques tours, puis appuyez de nouveau sur « Z HOME ».

Si la buse touche la plaque, cela signifie que le réglage du point zéro est correct.

Paramètres imprimante

Imprimante: 3DRAG+

Connexion | Imprimante | Extrudeur | Format imprimante | Scripts | Avancé

Nombre d'extrudeurs: 1

Nombre de ventil.: 1

Température Max. de l'Extrudeur: 250

Température Max. du Plateau: 100

Volume Max. par seconde: 12 [mm³/s]

L'imprimante a un Extrudeur Mixte (Une Buse pour plusieurs couleurs)

Extrudeur 1

Nom: []

Diamètre: 0,4 [mm] Décalage de la température: 0 [°C]

Couleur: []

Décalage X: 0 Décalage Y: 0 [mm]

OK Appliquer Annuler

Figure 28 : paramètres de l'onglet « Extruder ».

Paramètres imprimante

Imprimante: 3DRAG+

Connexion | Imprimante | Extrudeur | Format imprimante | Scripts | Avancé

Type Imprimante: Imprimante classique

Origine X: 0 Origine Y: 0 Origine Z: 0

X Min 0 X Max 400 Décal. Gauche 0

Y Min 0 Y Max 400 Décal. Arrière 0

Larg. zone d'impression: 400 mm

Prof. zone d'impression: 400 mm

Haut. zone d'impression: 400 mm

Les valeurs min et max définissent les plages possibles de coordonnées des extrudeurs. Ces coordonnées peuvent être négatives et en dehors du Plateau. Gauche et face du Plateau déterminent les coordonnées où le Plateau lui-même commence. En changeant les valeurs min/max, vous pouvez déplacer l'origine du centre du Plateau, si supporté par le micro logiciel.

Y Max

E

C

Figure 29 : paramètres de l'onglet « Format imprimante ».

OK Appliquer Annuler



Figure 30 : réglage du fin de course de l'axe Z.

Dans le cas contraire, vous devez agir de nouveau sur la vis jusqu'à ce que vous obteniez cette condition.

Pour plus de commodité, utilisez une feuille de papier pour vérifier que la buse n'est ni trop loin ni trop appuyée sur la plaque en position « HOME ».

En pratique, vous devez faire glisser une feuille de papier entre la plaque et l'extrudeuse sans espace.

À ce stade, relevez la tête de quelques millimètres puis placez-la au-dessus d'un coin de la plaque.

Effectuez de nouveau le « Z HOME » et vérifiez la distance entre la buse et la plaque de verre avec la feuille de papier. Si nécessaire, ajustez la vis de la plaque pour obtenir les conditions optimales. Répétez la même opération pour les autres angles. À la fin de l'opération, le plateau d'impression doit être coplanaire avec la buse (voir la figure 32).

NB : si un ruban adhésif est appliqué sur la plaque de verre pour améliorer l'adhérence de l'objet à imprimer, il est nécessaire d'agir sur la vis du fin de course Z (en le vissant) pour compenser l'épaisseur de l'élément ajouté.

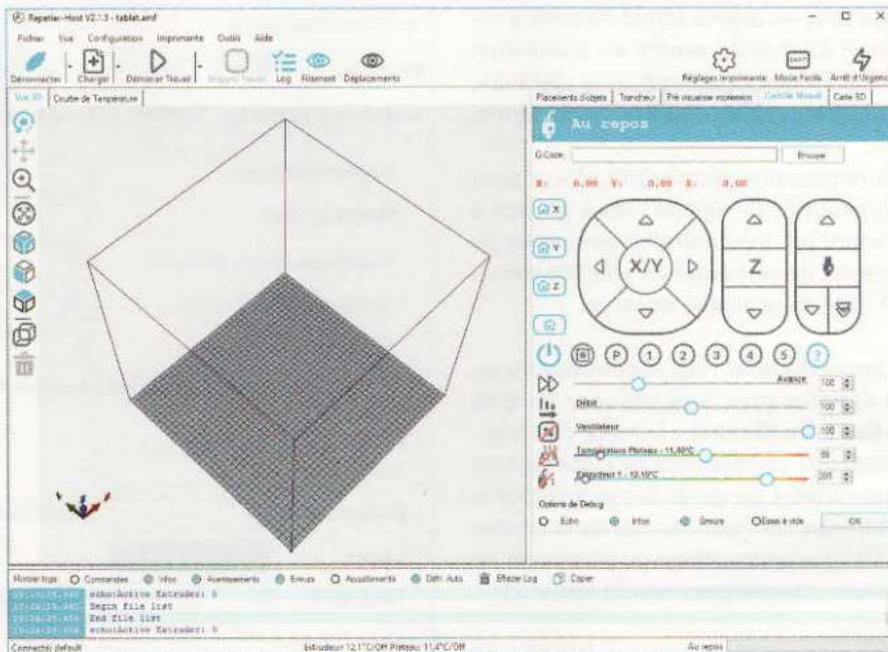


Figure 31 : l'onglet « Contrôle Manuel » de Repetier Host.

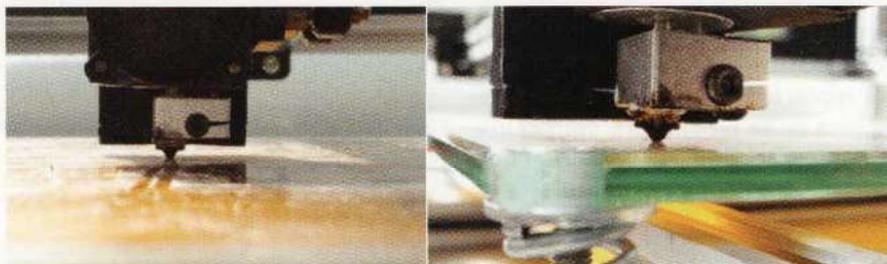


Figure 32 : réglage de la distance entre le plateau d'impression et la buse.

En position « HOME », la buse doit être proche de la vis de fixation de la plaque (comme indiqué en figure 33) et non au-dessus de celle-ci. Si nécessaire, vérifiez le positionnement des fins de course et, dans le cas extrême, celui du plateau d'impression.

Tests d'impression

Lors de la première utilisation, il est nécessaire de vérifier les paramètres enregistrés dans l'EEPROM. Après avoir lancé Repetier Host et connecté l'imprimante, cliquez sur le menu « Configuration » → « Configuration EEPROM » pour ouvrir la fenêtre, visible en figure 34, contenant tous les paramètres du firmware.

Vérifiez la correspondance entre les valeurs stockées dans le microcontrôleur de la carte et celles indiquées dans

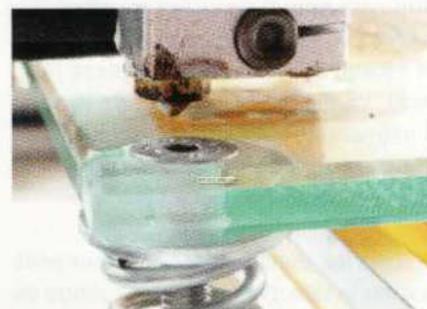


Figure 33 : en position « HOME », la buse doit être proche de la vis de fixation de la plaque mais pas au-dessus.

le tableau et éventuellement apportez les modifications nécessaires. Lorsque vous avez terminé, appuyez sur « OK » pour enregistrer les données.

Insérez une bobine de filament (PLA, ABS, etc.) sur le support approprié, puis desserrez l'écrou du chargeur. Insérez l'extrémité du fil dans la partie inférieure du chargeur, faites-le passer entre le

Réglages EEPROM micro logiciel

X axis resolution	80.00	[Steps/mm]
Y axis resolution	80.00	[Steps/mm]
Z axis resolution	1600.00	[Steps/mm]
E axis resolution	150.00	[Steps/mm]
X axis max. feedrate	500.00	[mm/s]
Y axis max. feedrate	500.00	[mm/s]
Z axis max. feedrate	5.00	[mm/s]
E axis max. feedrate	60.00	[mm/s]
X axis max. acceleration	1000	[mm/s ²]
Y axis max. acceleration	1000	[mm/s ²]
Z axis max. acceleration	100	[mm/s ²]
E axis max. acceleration	10000	[mm/s ²]
Minimum feedrate	0.00	[mm/s]
Minimum segment time	20000	[ms]
Maximum X jerk	20.00	[mm/s]
Maximum Z jerk	0.40	[mm/s]
Maximum E jerk	5.00	[mm/s]
Home offset X	0.00	[mm]
Home offset Y	0.00	[mm]
Home offset Z	0.00	[mm]
PID P	25.09	
PID I	3.08	
PID D	51.18	

Figure 34 : configuration de l' EEPROM.

Exporter valeurs EEPROM

Importer valeurs EEPROM

OK

Annuler

Figure 35 : le filament doit passer entre le roulement et le guide.



roulement et le guide jusqu'à ce qu'il pénètre dans le tube en PTFE (voir la figure 35).

Activez maintenant le chauffage de la tête d'impression à partir de l'onglet « Contrôle Manuel ».

Figure 36 : paramètres de l'onglet « Vitesse et qualité ».

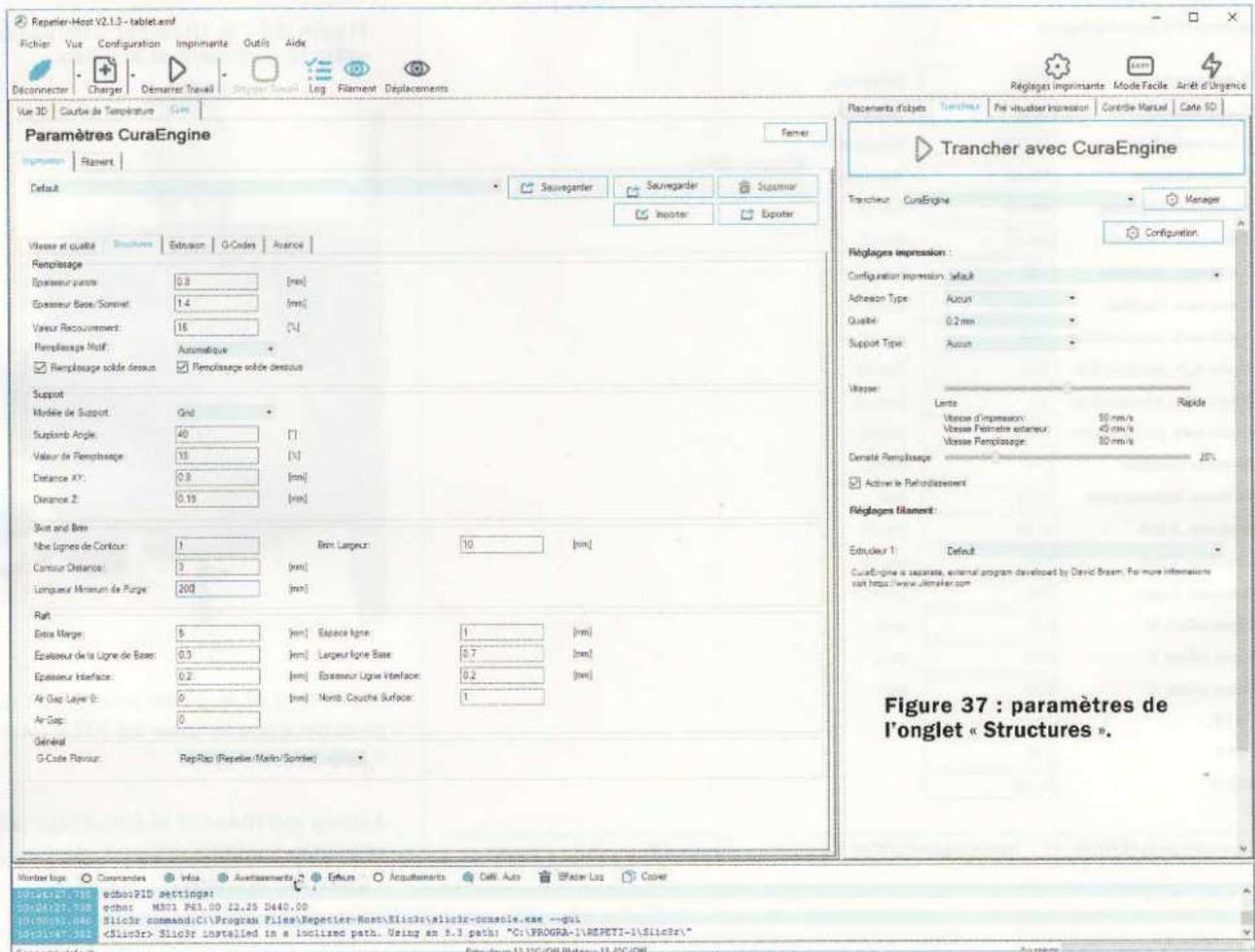


Figure 37 : paramètres de l'onglet « Structures ».

Lorsque l'extrudeuse est à la bonne température, poussez le filament dans la tête d'impression jusqu'à ce que la matière plastique sorte de la buse. Resserrez l'écrou du chargeur sans trop forcer.

Cliquez sur l'onglet « **Trancheur** » situé à droite de la fenêtre. Sélectionnez l'onglet « **Cura** » à gauche et cliquez sur l'onglet « **Impression** » pour définir les paramètres du slicer. La configuration de tous les éléments peut être faite manuellement en insérant dans les champs les différents paramètres des valeurs suivantes. Ces valeurs peuvent être modifiées en fonction de l'objet à imprimer.

Réglez les paramètres suivants pour l'onglet « **Vitesse et qualité** » (voir la figure 36) :

Vitesse :

	Lent	Rapide
Impression	50	70
Déplacement	120	120
Prem couche	30	30

Périmètre extérieur	30	60
Périmètre intérieur	40	60
Remplissage	60	80
Remplissage de surface	30	60

Qualité : 0,2 mm

Réglez les paramètres suivants pour l'onglet « **Structures** » (voir la figure 37) :

- Epaisseur parois : 0.8 mm ;
- Epaisseur base/sommet : 1.4 mm ;
- Valeur recouvrement : 16 % ;
- Surplomb angle : 40° ;
- Valeur de remplissage : 15 % ;
- Distance XY : 0.8 mm ;
- Distance Z : 0.19 mm ;
- Longueur minimum de purge : 200 mm.

Laissez les autres valeurs par défaut.

Réglez les paramètres suivants pour l'onglet « **Extrusion** » (voir la figure 38) :

- Activer rétraction : coché ;
- Vitesse rétraction : 40 mm/s ;
- Distance rétraction : 3.5 mm ;

- Déplacement minimum : 1.5 mm ;
- Rétraction changement extruder : 16 mm ;

Ensuite cliquez sur l'onglet « **G-Codes** », puis sélectionnez « **G- Code de Démarrage** » (voir la figure 39). Dans la fenêtre de droite, vous devez apercevoir une partie du G-Code, complétez le code comme indiqué ci-après :

```

; Default start code
G28 ; Home extruder
G1 Z15 F[Z_TRAVEL_SPEED]
M107 ; Turn off fan
G90 ; Absolute positioning
M82 ; Extruder in absolute mode
{IF_BED}M190 S{BED}
; Activate all used extruder
{IF_EXT0}M104 TO S{TEMPO}
G92 E0 ; reset extruder position
; Wait for all used extruders to reach temperature
{IF_EXT0}M109 TO S{TEMPO}
G1 F200 E10 ; extrusion de 10 mm de filament
G92 E0 ; reset de la position de l'extruder
    
```

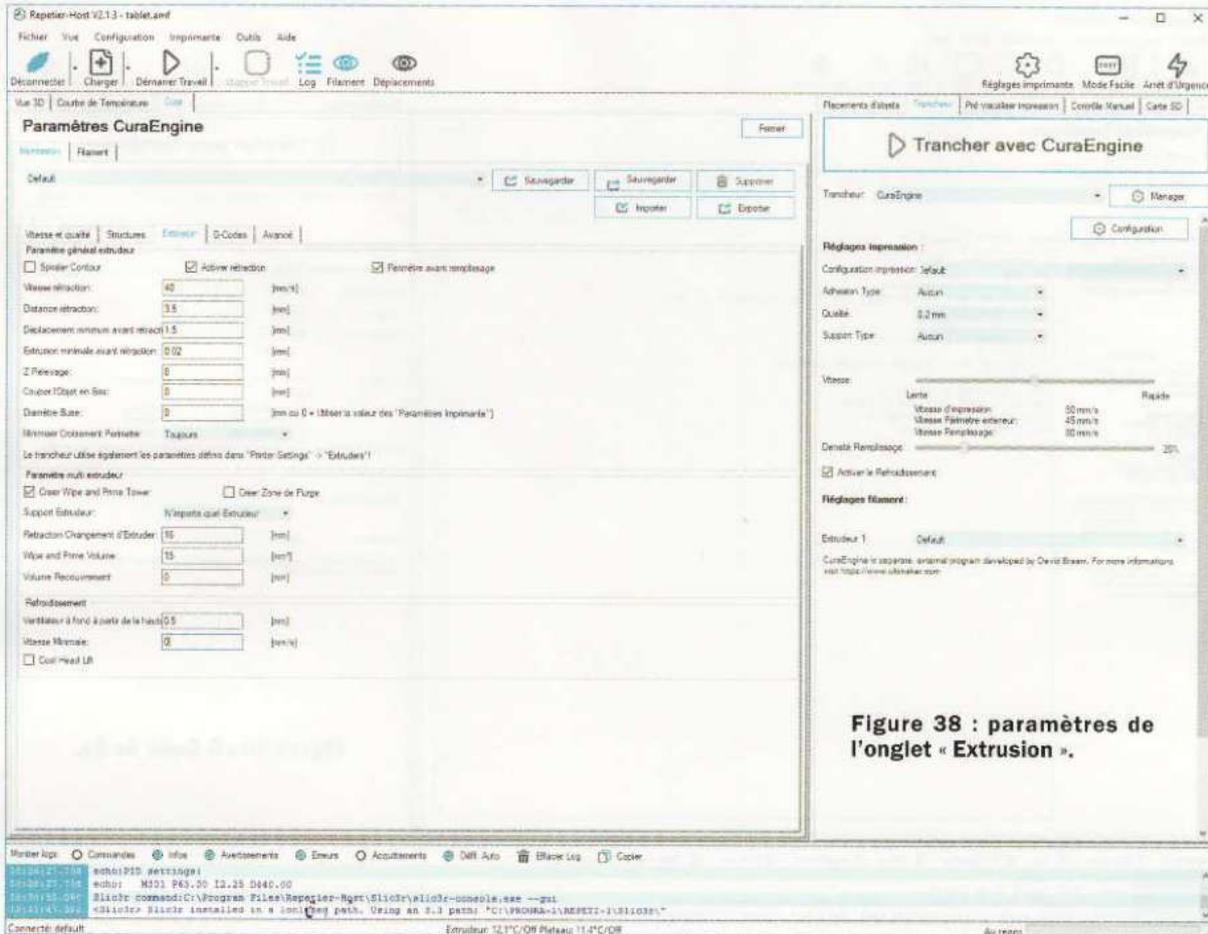


Figure 38 : paramètres de l'onglet « Extrusion ».

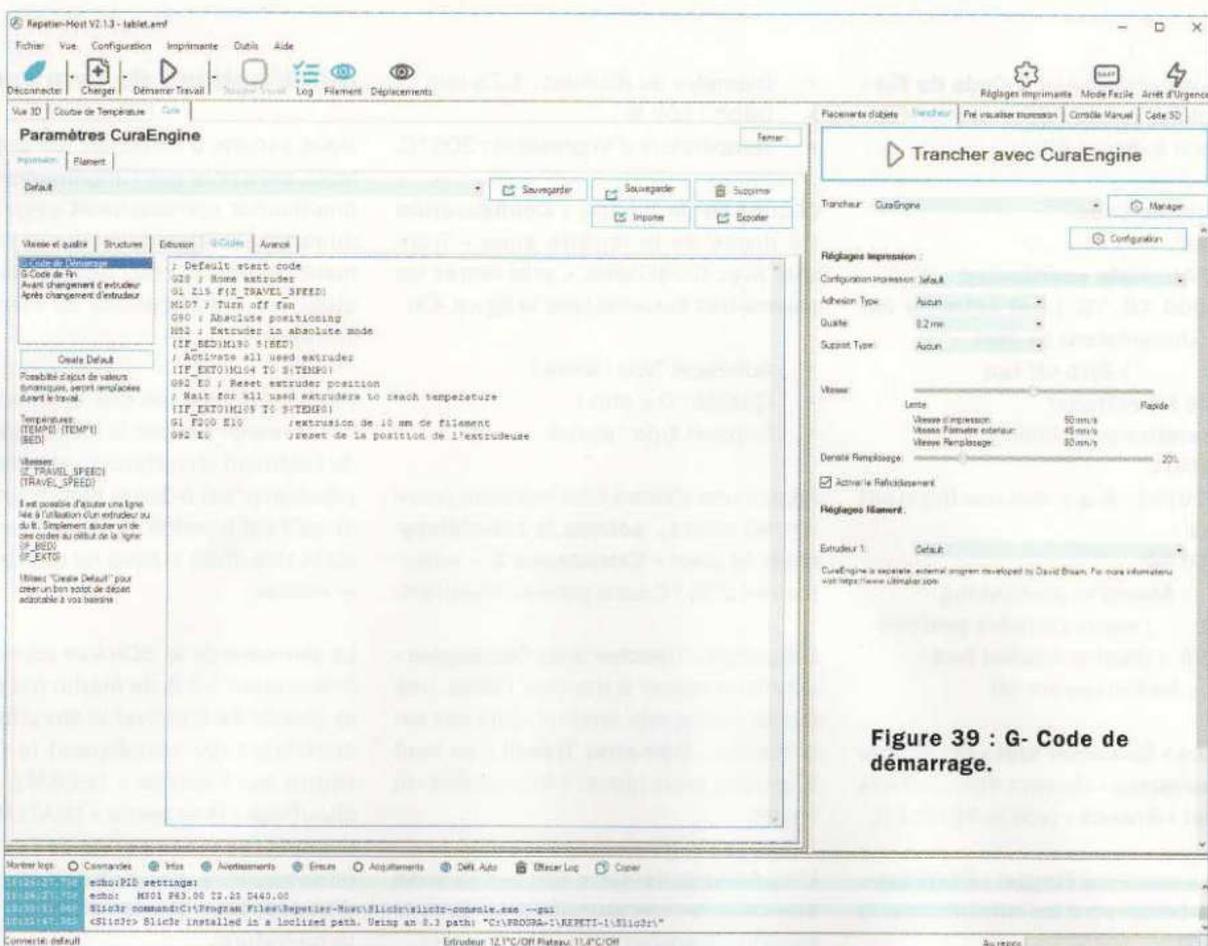


Figure 39 : G- Code de démarrage.

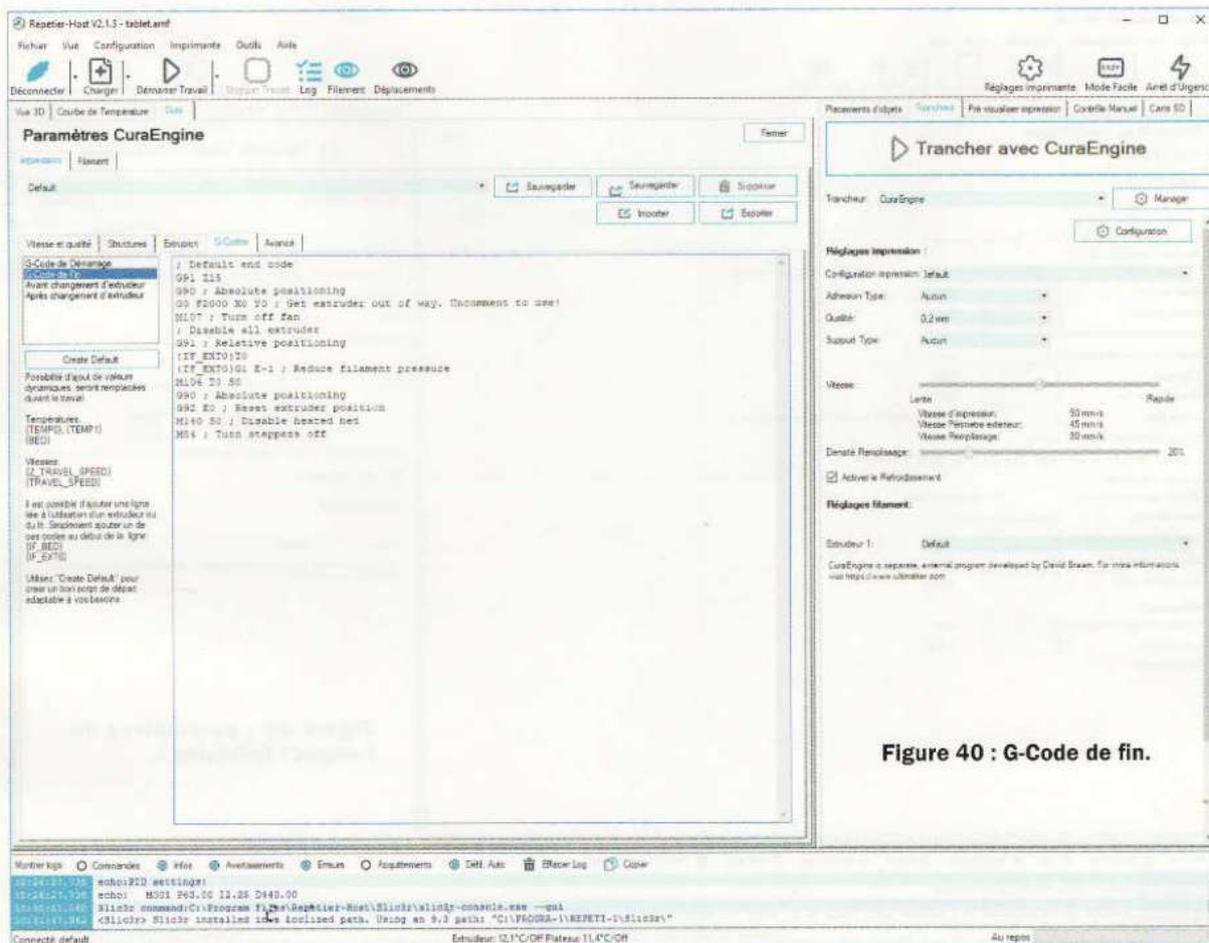


Figure 40 : G-Code de fin.

Ensuite, sélectionnez « **G-Code de Fin** » et complétez le code de la manière suivante (voir la figure 40) :

; **Default end code**

G91 Z15

G90 ; Absolute positioning

G0 F2000 X0 Y0 ; Get extruder out of way. Uncomment to use!

M107 ; Turn off fan

; Disable all extruder

G91 ; relative positioning

(IF_EXT0)T0

(IF_EXT0)G1 E-1 ; reduce filament pressure

M104 T0 S0

G90 ; Absolute positioning

G92 E0 ; reset extruder position

M104 S0 ; disable heated bed

M84 ; Turn steppers off

Les cases « **Combiner tout** » et « **Garder faces ouvertes** » doivent être cochées à l'onglet « **Avancé** » (voir la figure 41).

En ce qui concerne l'onglet « **Filament** », les paramètres sont les suivants (voir la figure 42) :

- Diamètre du filament : 1.75 mm ;
- Débit : 100 % ;
- Température d'impression : 206 °C.

Cliquez sur le bouton « **Configuration** » à droite de la fenêtre sous « **Trancher avec CuraEngine** », puis entrez les paramètres suivants (voir la figure 43) :

- Adhesion Type : **Brim** ;
- Qualité : 0.2 mm ;
- Support type : aucun.

Réglez une vitesse intermédiaire (environ 60 mm/s), **activez le refroidissement** et pour « **Extrudeuse 1** », sélectionnez 206 °C sans plateau chauffant.

Cliquez sur « **Trancher avec CuraEngine** » pour commencer à trancher l'objet. Une fois le découpage terminé, appuyez sur le bouton « **Démarrer Travail** » en haut à gauche pour lancer l'impression de l'objet.

L'imprimante démarre dès que la buse atteint la température définie dans la section « **Trancher avec CuraEngine** ».

Modifications du firmware

Nous venons d'effectuer les premiers tests, mais pour que l'imprimante puisse fonctionner correctement avec l'ajout du relais statique qui commande l'élément chauffant, nous devons effectuer quelques modifications au niveau du firmware.

Partons du principe que le firmware de l'imprimante permet le fonctionnement de l'élément chauffant du plateau d'impression (c'est-à-dire la plaque en verre) et qu'il est possible de désactiver l'élément chauffant si nous ne désirons pas le monter.

Le firmware de la 3DRAG+ correspond à la version 1.1.5 de Marlin qui prévoit la possibilité d'activer le chauffage (en contrôlant par conséquent la température sur l'entrée « **THERM2** » et le chauffage via la sortie « **HEATER2** ») et d'empêcher le démarrage de l'extrusion (impression) jusqu'à ce que le plateau d'impression atteigne une certaine température.

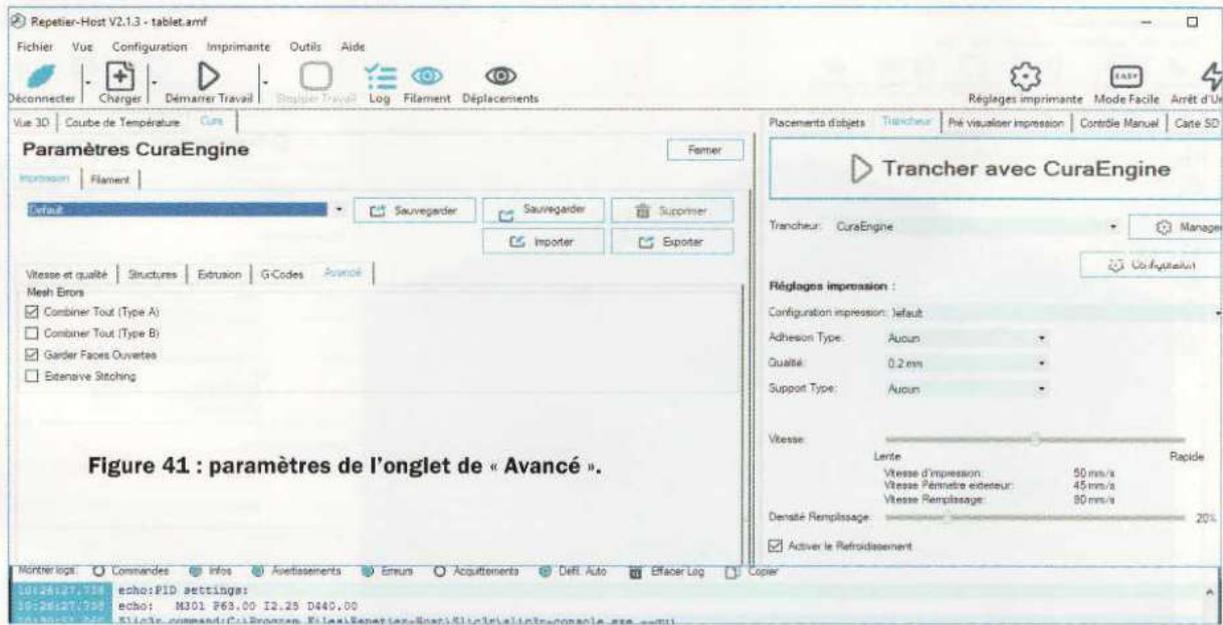


Figure 41 : paramètres de l'onglet de « Avancé ».

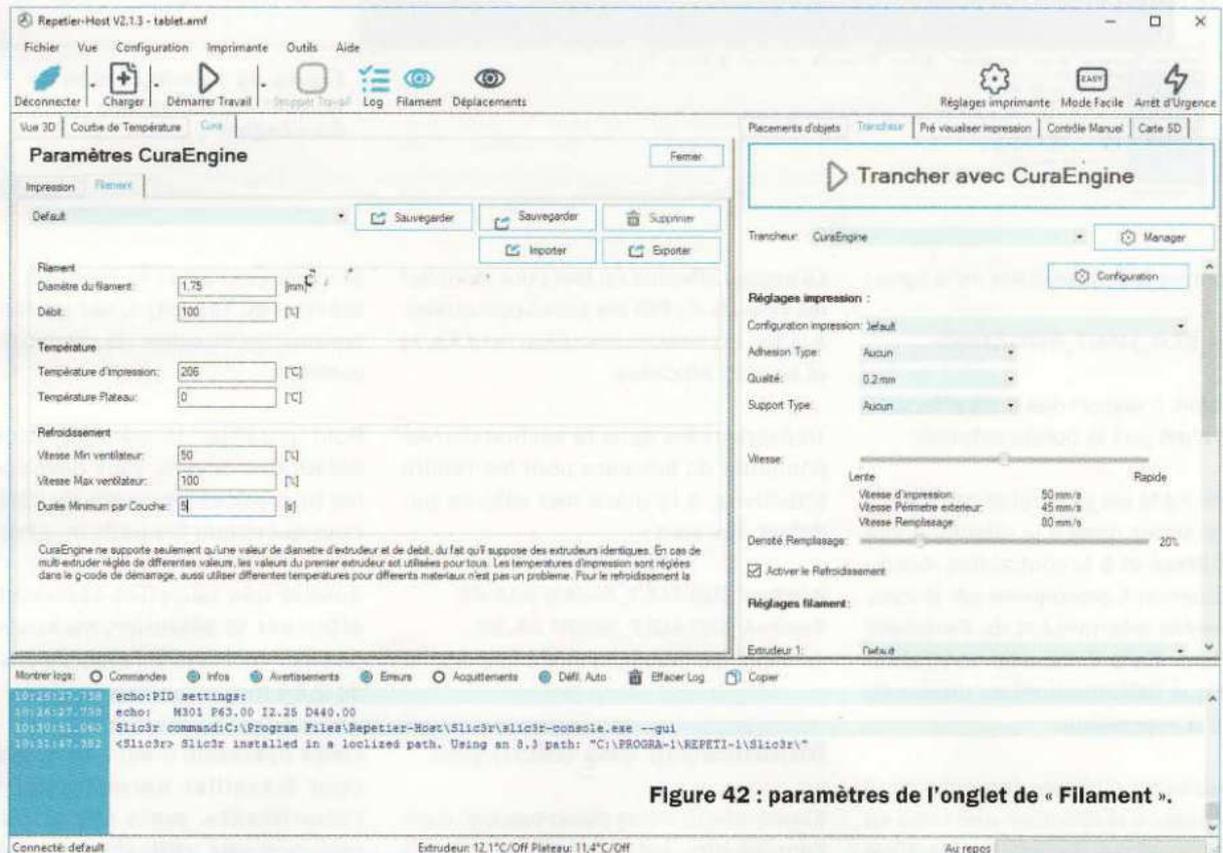


Figure 42 : paramètres de l'onglet de « Filament ».

La température minimale d'extrusion a été fixée à 170 °C, cela signifie que, tant que cette température n'est pas atteinte, l'imprimante ne commence pas à extruder.

L'instruction qui permet ce mode de fonctionnement est : **#define EXTRUDE_MINTEMP 170**

Pour contrôler le plateau à l'aide d'un signal PWM, nous devons supprimer le

commentaire de la ligne suivante :

#define PIDTEMPBED

et placer un commentaire devant la ligne suivante :

// #define BED_LIMIT_SWITCHING

Rappelez-vous que le commentaire signifie que l'on place en début de ligne 2 caractères slash « // », afin que le

compilateur n'exécute pas la ligne (il la voit comme un commentaire).

En ce qui concerne le contrôle du plateau chauffant, il convient de noter qu'il existe une option, à savoir le contrôle du chauffage à l'aide d'un relais traditionnel ou d'un relais statique (SSR) en mode ON/OFF. Pour cela, nous devons mettre en commentaire la ligne :

#define PIDTEMPBED

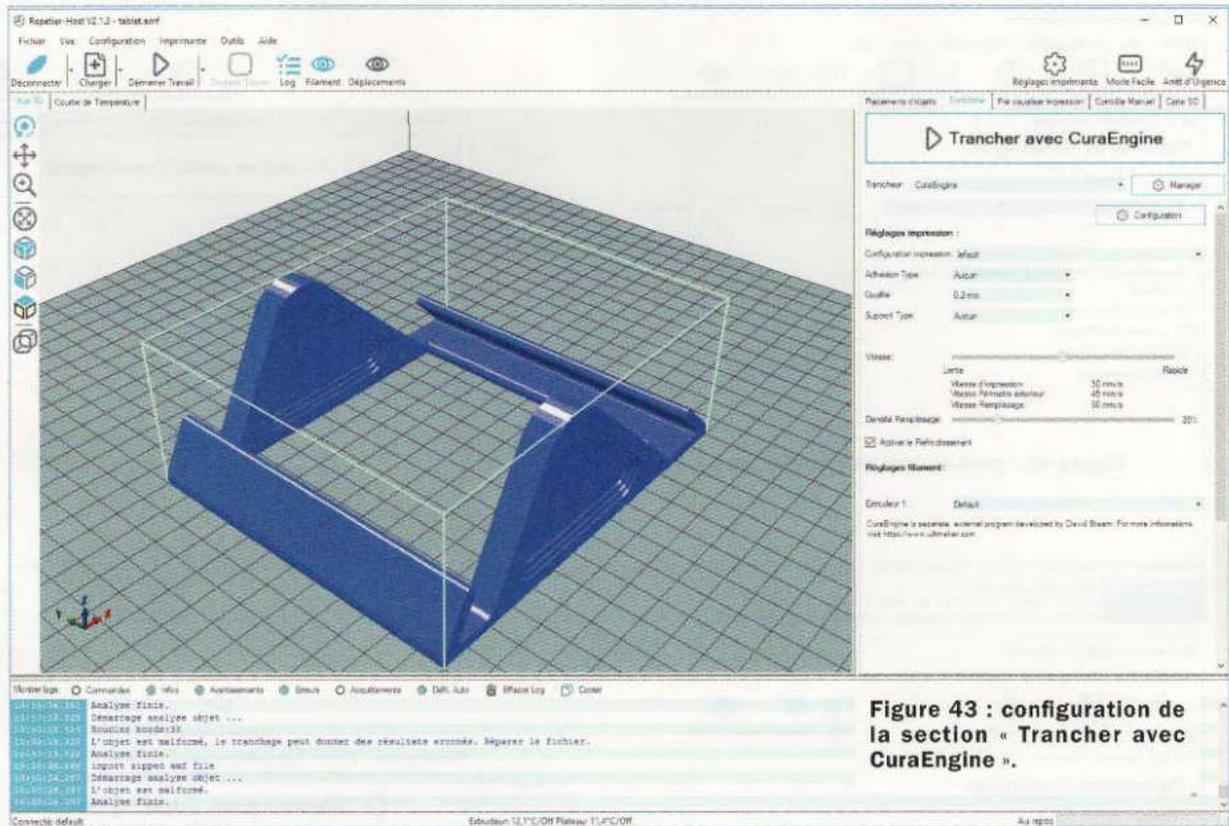


Figure 43 : configuration de la section « Trancher avec CuraEngine ».

et supprimer le commentaire de la ligne :

```
#define BED_LIMIT_SWITCHING
```

Cependant, il ressort des tests effectués que ce n'est pas la bonne solution.

Le mode PWM est préférable car il limite les contraintes dues à la dilatation lors du chauffage et à la contraction lors du refroidissement, provoquées par le fonctionnement intermittent de l'élément chauffant. Cela évite une contrainte mécanique (déformation) au niveau du plateau d'impression.

Notez qu'avant d'utiliser l'imprimante, il est nécessaire d'effectuer une mise au point automatique du plateau chauffant afin de régler le PID.

Vous devez vérifier que le firmware est bien configuré :

```
#define TEMP_SENSOR_BED5
```

sinon, configurez l'instruction.

Ensuite, depuis le logiciel d'impression (Repetier Host), dans la console G-Code, vous devez écrire : **M303 E-1 S60 C8** et appuyer sur « ENTER ».

Le logiciel effectue un test pour identifier les valeurs du PID les plus appropriées. À la fin, les valeurs calculées pour kp, ki et kd sont affichées.

Transcrivez-les dans la section correspondante du firmware pour les rendre effectives, à la place des valeurs par défaut, qui sont :

```
#define DEFAULT_bedKp 334.42
#define DEFAULT_bedKi 25.39
#define DEFAULT_bedKd 1101.30
```

Modification des micro-pas

Examinons le mode micro-pas qui, dans l'imprimante, est pré-réglé à 1:16. Les drivers des moteurs pas à pas sortant d'usine sont pré-réglés dans ce mode.

Si vous souhaitez le modifier, reportez-vous au Tableau 1, qui présente les paramètres du driver via ces broches de contrôle.

Pour modifier la configuration par défaut des drivers, vous devez couper les trois pistes de cuivre du côté inférieur qui relie les pads du cavalier.

Soudez des barrettes standard pour effectuer la sélection manuelle des pas des moteurs à l'aide de cavaliers de 2,54 mm.

Cette opération n'est pas nécessaire pour travailler normalement avec l'imprimante, mais est réservée aux utilisateurs ayant des connaissances avancées dans l'impression 3D.

Tableau 1 : réglage des pas du moteur en fonction des impulsions de commande provenant du driver.

MS1	MS2	MS3	Résolution du micro-pas	Mode de commande
L	L	L	Pas complet	2 Phase
H	L	L	1/2 pas	1-2 Phase
L	H	L	1/4 pas	W1-2 Phase
H	H	L	1/8 pas	2W1-2 Phase
H	H	H	1/16 pas	4W1-2 Phase

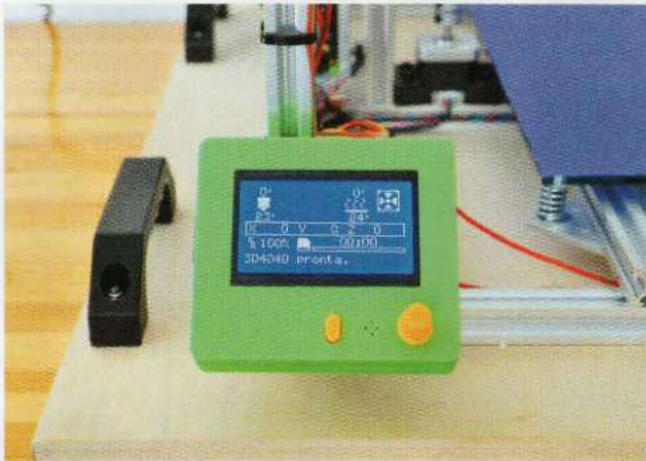


Figure 44 : contrôleur pour impression autonome avec écran graphique, à gauche le bouton de réinitialisation et à droite l'encodeur rotatif pour accéder au menu.



Figure 45 : à gauche se trouve l'emplacement de la carte SD (non fournie).

Il est entendu que si vous modifiez le réglage des micro-pas, il est nécessaire d'adapter le firmware (pour générer les impulsions à la fréquence requise et non pas pour la division par 16), en intervenant sur l'instruction qui définit les impulsions que le microcontrôleur doit générer pour obtenir un certain déplacement de l'axe correspondant en tenant compte du pas de vis trapézoïdal de l'axe Z et des dents de la poulie pour les axes X et Y.

La section correspondante du firmware Marlin est la suivante :

HARDWARE

```
#define DEFAULT_AXIS_STEPS_PER_UNIT {80,80,1600,150} // pas par défaut par unité pour la 3DRAG+
#define DEFAULT_MAX_FEEDRATE {500, 500, 5, 50} // (mm/sec)
```

Les virgules séparent les réglages individuels des moteurs (axes X, Y, Z et le moteur de l'extrudeuse).

Il est à noter que le firmware prévoit que les moteurs de l'axe Z soient maintenus bloqués (ils sont mis en position mais ne sont pas libérés) pendant l'impression, afin d'éviter des mouvements non désirés de la tête.

Si vous souhaitez garder Z actif, recherchez l'instruction « #define DISABLE_Z » et remplacez le paramètre « true » par « false ». Dans le firmware Marlin de base, l'axe Z est toujours actif en raison de l'instruction « false ».

Contrôleur pour impression autonome avec écran graphique

La 3DRAG+ peut être personnalisée en installant un contrôleur pour imprimer de manière autonome et qui est doté d'un écran graphique (voir la figure 44).

Ce contrôleur vous permet de gérer les impressions sans que l'imprimante soit connectée à un ordinateur.

Cette carte contrôleur est identique à celle que nous vous avons présentée à l'époque pour l'imprimante 3DRAG, elle prend également en charge une carte SD (voir la figure 45).

En transférant les fichiers à imprimer sur la carte SD, le microcontrôleur de la carte de l'imprimante gère alors directement l'impression, de manière à pouvoir fonctionner indépendamment sans ordinateur allumé et connecté et cela jusqu'à la fin de l'impression.

Cela représente une économie d'électricité (pas d'ordinateur branché), si l'on considère que certains objets complexes peuvent prendre jusqu'à 3 jours ou plus pour les imprimer !

Le contrôleur pour impression autonome s'interface avec la carte contrôleur de l'imprimante à l'aide d'un connecteur spécial, il dispose d'un encodeur rotatif permettant de sélectionner les fonctions du menu et d'un afficheur graphique.

Le firmware est configuré pour gérer le contrôleur pour impression autonome. Considérez que si vous connectez le PC, ce dernier prendra le pas sur le contrôleur autonome, même si vous apercevez que l'afficheur graphique continue à fonctionner.

Le contrôleur pour impression autonome comporte un bouton de réinitialisation qui sert uniquement si l'imprimante est bloquée. Nous vous conseillons de protéger le bouton de réinitialisation afin d'éviter, par inadvertance, un reset en cours d'impression.

Il ne faut pas oublier qu'une imprimante telle que la 3DRAG+, en raison de la grande taille des objets imprimables, prend des jours pour terminer le travail et qu'une réinitialisation accidentelle pourrait vous faire perdre beaucoup de temps et de matière plastique.

Conclusion

Ici se termine la description de notre imprimante « XXL », nous avons décrit ses points forts et ses caractéristiques, et nous vous avons guidé dans la construction. Nous vous avons aussi proposé des options utiles pour améliorer son utilisation.

Vous trouverez dans le sommaire détaillé de la revue, plusieurs fichiers « .stl » qui vous permettront d'imprimer le boîtier du contrôleur autonome et son support. ■



Ce montage permet de vérifier l'intégrité et la programmation correcte des télécommandes couramment utilisées pour les ouvre-portes, les portails, les volets motorisés, etc.



DECODEUR UNIVERSEL POUR TÉLÉCOMMANDES RADIO

..... de Michele Menniti

De plus en plus de dispositifs et de technologies visent à contrôler des actionneurs de tous types à l'aide de télécommandes sans fil (radio, infrarouge, Bluetooth).

La prolifération des tablettes et des smartphones et la facilité avec laquelle il est possible de développer des applications spécifiques a maintenant déclenché une tendance irréversible à incorporer les fonctions des télécommandes radio dans de tels appareils.

Mais il reste un domaine bien défini, le plus « ancien » de tous, celui de la radiocommande qui contrôle des moteurs électriques conçus pour l'automatisation, à savoir des portes électriques, des volets, des portails automatiques.



Ces systèmes ont toujours été contrôlés par une télécommande radio combinée à un récepteur placé à l'intérieur du système de contrôle.

Ce qui a changé au fil des ans, c'est la fréquence de fonctionnement. Au départ, les télécommandes fonctionnaient dans la gamme de fréquences allant de 27 MHz à 41 MHz (aujourd'hui, les fréquences utilisées sont toujours 40,665 MHz, 40,685 MHz et 30,875 MHz et 30,900 MHz), puis dans la gamme 260 MHz à 350 MHz (aujourd'hui totalement interdite) pour se conformer aux réglementations actuelles autorisant l'utilisation des fréquences 433,92 MHz et 868,35 MHz.

Nous pouvons dire qu'environ 80 % des systèmes actuels fonctionnent à la fréquence de 433,92 MHz.

La commande d'un actionneur est basée sur l'envoi d'un code binaire par la télécommande qui doit être interprété et reconnu correctement par le récepteur, afin que l'opération demandée soit effectuée.

Dans les premiers modèles (mais on en trouve encore aujourd'hui), le codage était défini à l'aide d'une série de 8 à 14 commutateurs DIP. Pour sécuriser d'avantage la transmission, ces dernières années, les télécommandes radio d'origine ont été codées en usine avec les unités de contrôle correspondantes.

Puis sont apparus les télécommandes à auto-apprentissage avec lesquelles il suffisait d'activer le mode de programmation et de transmettre un code compatible (via l'unité de contrôle elle-même ou une autre télécommande similaire) pour définir au nouveau codage.

Un type de codage plus récent, qui était initialement destiné uniquement aux voitures, est apparu dans le domaine de l'automatisation, il s'agit du codage dénommé « rolling code » ou code tournant. Ce codage est basé sur une partie de code fixe, prédéfinie par le fabricant et une partie de code variable, générée à chaque nouvelle pression sur la télécommande à l'aide d'un algorithme spécifique.

Ce dernier étant également présent dans l'unité de réception afin d'interpréter correctement l'ordre reçu. Ce système ne peut en aucun cas être répliqué, sauf en achetant une télécommande d'origine auprès du fabricant.

Pourquoi la télécommande ne fonctionne-t-elle pas ?

Tant que tout fonctionne correctement, personne n'a de problèmes de fréquence, de codage, de contrôle radio ou d'unité de contrôle, mais dès que la pression sur le bouton de la télécommande ne correspond plus à l'exécution de la commande, il faut nécessairement affronter et surmonter la question.

Dans la très grande majorité des cas, il s'agit de la pile qui est déchargée, mais il est fréquent que la télécommande

ait subi des chocs, dus à une ou plusieurs chutes ou, pire, l'unité de réception pose des problèmes.

Et voici que commence le processus nécessaire qui consiste principalement à remplacer la pile, en espérant que tout se passe bien, sinon vous devez essayer une autre télécommande habilitée pour vérifier que l'unité réceptrice n'est plus capable de reconnaître les signaux.

Mais si ce test est négatif, c'est-à-dire que si l'unité réceptrice fonctionne, vous n'avez pas la preuve certaine que la télécommande qui ne fonctionne pas est défectueuse. Il se peut qu'elle ait simplement perdu la programmation du codage, peut-être à cause d'un appui malencontreux sur les touches d'activation de reprogrammation.

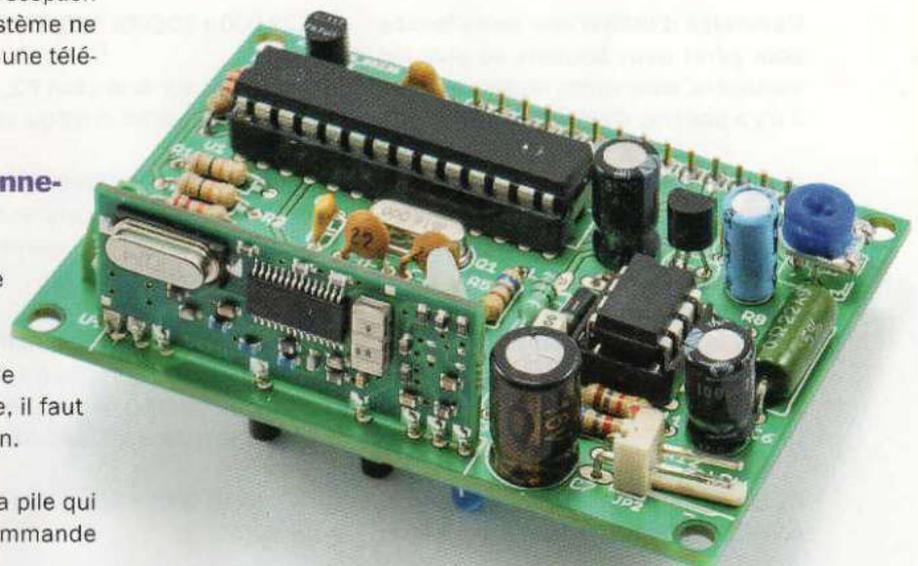
Et si vous arrivez à la conclusion que vous avez besoin d'une nouvelle télécommande, vous devez vous assurer de sa fréquence de fonctionnement et de la compatibilité de son codage, afin d'éviter des dépenses inutiles.

Un autre problème est celui de la soi-disant « dégradation » de la programmation. Si vous n'utilisez pas de radiocommandes programmables fiables et précises au niveau de la synchronisation, lorsqu'elles sont utilisées pour programmer d'autres radiocommandes, il devient de plus en plus difficile de voir les nouveaux clones fonctionner, car les synchronisations imparfaites finissent par conduire à un codage qui n'est qu'une mauvaise copie de l'original. Il est, dans ce cas, très difficile de reconnaître le codage originel du fabricant, voire de ne pas le reconnaître du tout.

En résumé, la technologie évolue, mais les problèmes sont toujours les mêmes depuis des dizaines d'années, s'agissant de ces systèmes de commande basés sur des transmissions codées HF.

Notre décodeur universel

Notre projet s'adresse à ceux qui utilisent souvent des télécommandes radio codées.



Sa fonction est d'analyser un signal et de le décoder, en affichant la séquence relative des bits sur l'afficheur, jusqu'à un maximum de 32 (ce nombre inclut la grande majorité des codes commerciaux standard).

Une autre fonction importante que nous avons mise en œuvre, tirant parti des fonctionnalités du microcontrôleur utilisé, est de pouvoir mémoriser deux codages différents dans son EEPROM, afin de vérifier ensuite le bon fonctionnement des autres radiocommandes. Nous approfondirons cette fonction dans la partie dédiée à l'utilisation de l'instrument.

Schéma électrique

Le cœur de notre décodeur est le très célèbre microcontrôleur **ATmega328P-PU**, dans la version PDIP classique à 28 broches. Il s'agit du même microcontrôleur présent dans les célèbres cartes Arduino UNO. Il dispose d'un quartz externe de 16 MHz et d'une alimentation externe de 5 VDC.

Pour ceux qui ne peuvent pas le programmer, le microcontrôleur est disponible à la vente déjà programmé par la société COMELEC sous la référence **MF1243**.

Nous avons utilisé **une seule broche analogique pour contrôler deux boutons**. Bien que nous disposions des broches nécessaires pour la configuration typique, nous avons préféré adopter cette solution particulière pour proposer de nouvelles idées à nos lecteurs.

L'avantage d'utiliser une seule broche pour gérer deux boutons ou plus est indéniable, dans toutes les situations où il n'y a pas trop d'entrées disponibles.

L'astuce consiste à **créer une série de diviseurs de tension de sorte que chaque combinaison** (pression sur P1, pression sur P2, pression simultanée de P1 et P2) **fournisse sur la broche analogique une tension différente en fonction de la valeur des résistances** constituant les diviseurs de tension reliés aux boutons poussoirs.

En fait, en appuyant sur le bouton poussoir P1, le diviseur de tension formé par

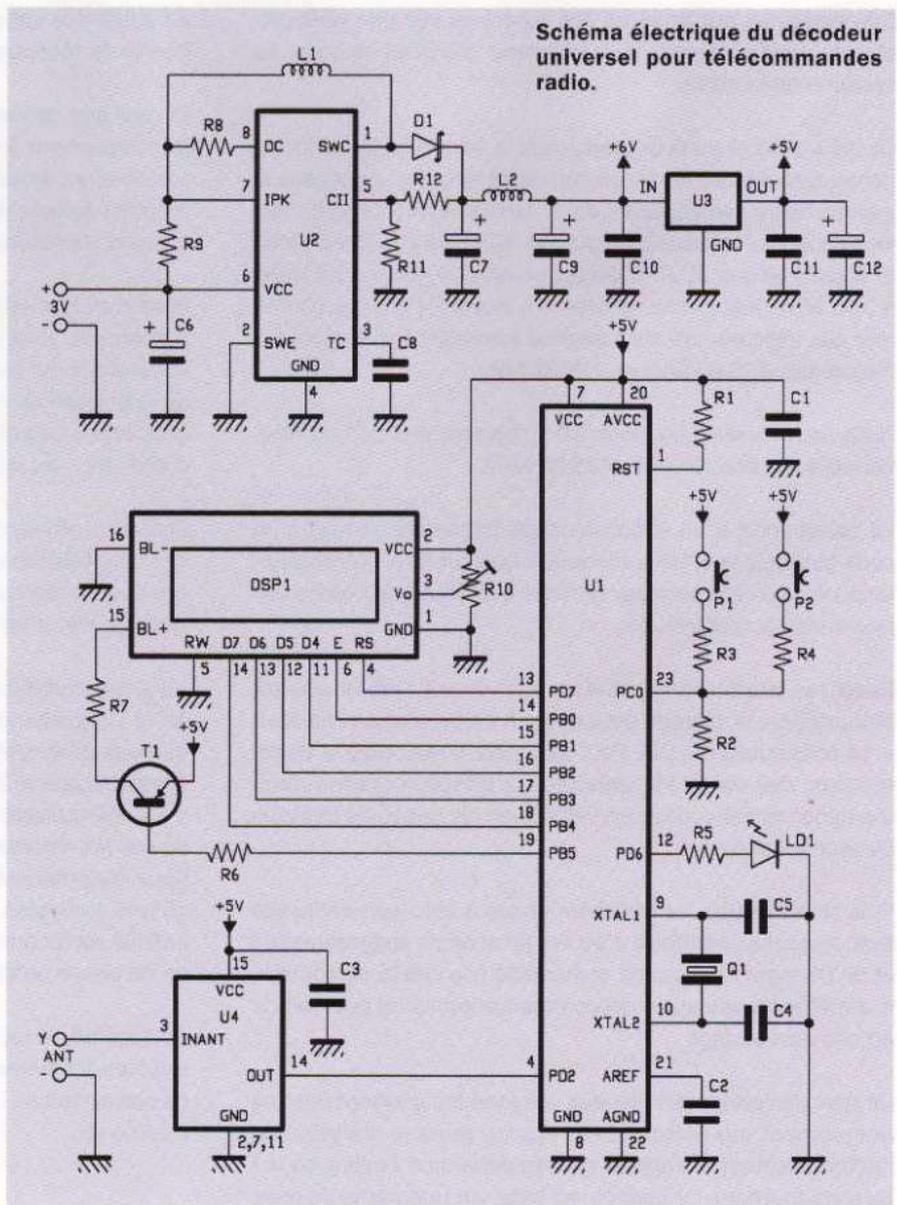


Schéma électrique du décodeur universel pour télécommandes radio.

les résistances R3 et R2 applique sur l'entrée analogique A0 (broche 23 de l'ATmega328P) une tension d'environ :

$$5 / (22\ 000 + 10\ 000) * 10\ 000 = 1,56\text{ V.}$$

En appuyant sur le bouton P2, c'est le diviseur formé par R4 et R2 qui intervient.

Comme le diviseur est composé de deux résistances identiques, cela nous donne une tension de 5/2 soit environ 2,5 V sur l'entrée analogique A0.

Enfin, en appuyant simultanément sur P1 et P2, les résistances R3 et R4 se trouvent en parallèle et se comportent comme une résistance unique de valeur :

$$(22\ 000 * 10\ 000) / (22\ 000 + 10\ 000) = 6875\ \Omega.$$

Le diviseur est alors formé par « R3//R4 » en série avec R2, la tension sur A0 vaut donc :

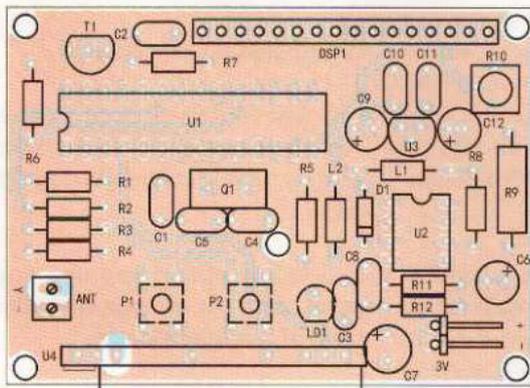
$$5 / (6875 + 10\ 000) * 10\ 000 = 2,96\text{ V.}$$

Au niveau logiciel, il suffit de créer trois plages (ou gammes) sur chacune de ces valeurs de tension et le contrôle des boutons fonctionnera parfaitement.

Naturellement, il sera nécessaire de prévoir une marge de tolérance, en tenant compte à la fois des valeurs d'alimentation et des valeurs des résistances, un pourcentage de $\pm 5\%$ est plus que suffisant.

La **broche 4** du microcontrôleur est utilisée comme entrée numérique pour **recevoir le signal codé** en provenance du

Plan de montage du décodeur universel



Plan de câblage des composants du décodeur universel.

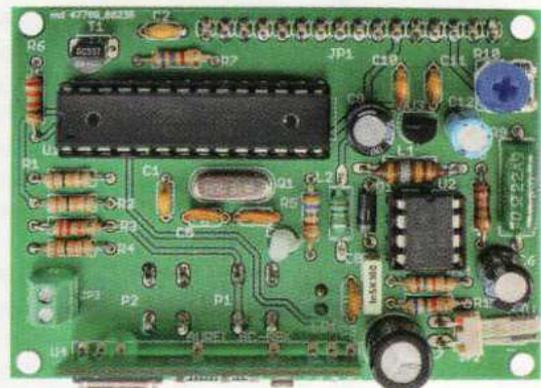


Photo de l'un de nos prototypes du décodeur universel.

Liste des composants du décodeur universel

R1..... 10 k Ω
 R2..... 10 k Ω 1%
 R3..... 22 k Ω 1%
 R4..... 10 k Ω 1%
 R5..... 680 Ω
 R6..... 2,2 k Ω
 R7..... 680 Ω
 R8..... 180 Ω
 R9..... 0,22 Ω 1W
 R10.... trimmer 10 k Ω MO
 R11.... 1,8 k Ω
 R12.... 6,8 k Ω

C1..... 100 nF céramique
 C2..... 100 nF céramique
 C3..... 100 nF céramique
 C4..... 22 pF céramique
 C5..... 22 pF céramique
 C6..... 100 μ F/16 V électrolytique
 C7..... 330 μ F/16 V électrolytique

C8..... 1,5 nF/63 V polyester
 C9..... 100 μ F/16 V électrolytique
 C10.... 100 nF céramique
 C11.... 100 nF céramique
 C12.... 10 μ F/16 V électrolytique

D1..... diode Schottky 1N5819
 L1..... inductance 180 μ H
 L2..... inductance 1 μ H
 T1..... BC557

U1..... ATMEGA328P-PU
 (MF1243)
 U2..... MC33063AP
 U3..... LP2950Z
 U4..... module Aurel
 RX-4MM5++/F

Q1..... quartz 16 MHz
 DISP1 afficheur LCD 16 x 2
 LD1.... LED 3 mm bleue
 P1..... microswitch
 P2..... microswitch
 INT..... interrupteur à levier

Divers :

Barrette femelle 16 pôles
 Barrette mâle 16 pôles
 Bornier 2 pôles au pas de 2,54mm
 Connecteur coudé 90° 2 pôles pour ci
 Porte pile 2 x AA
 Clip pour pile 9 V
 Support circuit intégré 2 x 4 broches
 Support circuit intégré 2 x 14 broches
 Entretoise en plastique hauteur 5 à 7 mm
 Entretoise en plastique hauteur 25 mm
 Boîtier Teko COFFER-2.9

NB : les typons des circuits imprimés à l'échelle 1 sont disponibles en téléchargement dans le sommaire détaillé de la revue.

module récepteur HF et qui sera ensuite analysé et décodé par le firmware.

La broche 12 est configurée en sortie et sert à contrôler la LED de signalisation qui, avec l'afficheur à cristaux liquides, nous donnera des indications utiles sur le fonctionnement du circuit.

Les six broches 13 à 18 permettent de gérer l'afficheur LCD à deux lignes de 16 caractères, en **mode 4 bits** (configuration classique). Le trimmer **R10 permet de régler le contraste** de l'afficheur LCD, tandis que le rétroéclairage

est contrôlé par le microcontrôleur via la broche 19, laquelle commande le transistor PNP T1 à travers la résistance R6. La résistance R7 sert à limiter le courant des LED du rétroéclairage. En fonction des caractéristiques de l'afficheur utilisé, il sera peut-être nécessaire de diminuer la valeur de R7 pour obtenir une plus grande luminosité.

Une attention particulière doit être portée au module U4, il s'agit du circuit hybride qui **capture le signal HF** provenant de la radiocommande (via l'antenne reliée à sa broche appelée « ANT »).

Ce module **génère sur sa sortie « OUT » un signal numérique correspondant au code binaire reçu**, et qui est directement décodé par le firmware de l'ATmega328P.

L'utilisation de ce module est relativement simple, mais, pour qu'il fonctionne dans des conditions optimales, il est essentiel de suivre les indications fournies par le constructeur AUREL. Ce module fonctionne à la fréquence standard de **433,92 MHz**.

Complétons la description de notre schéma avec l'étage d'alimentation.

Pour rendre notre décodeur aussi léger et économique que possible, nous avons jugé utile de l'alimenter à partir de deux piles de type « 1,5 V AA », placées en série afin d'obtenir 3 VDC.

Cependant cette tension est insuffisante pour faire fonctionner correctement le microcontrôleur et l'afficheur, car ils nécessitent au moins une tension d'alimentation de 5 VDC.

Par conséquent, nous avons utilisé un **circuit à commutation** (découpage) qui permet **d'élever** (step-up) la **tension en sortie**, basé sur le circuit intégré MC33063AP (U2). Il permet d'obtenir en sortie une tension d'environ 6 VDC (à partir de 3 VDC).

Pour comprendre la manière d'utiliser ce circuit intégré pour d'autres tensions d'entrée et/ou de sortie, il suffit de consulter la documentation technique disponible à l'adresse :

<https://www.onsemi.com/pub/Collateral/MC34063A-D.PDF>

Dans notre cas, pour le dimensionnement de l'alimentation, la formule de calcul de la tension de sortie est correcte et indépendante de la tension d'entrée (dans des grandeurs acceptables en mode « step-up », c'est-à-dire une tension d'entrée inférieure à celle de sortie), soit :

$$|V_{out}| = 1.25 * (1 + R2/R1)$$

Donc tout dépend des valeurs des résistances R2 et R1. Dans notre montage, il s'agit respectivement de R12 et R1.

En fait : $V_{out} = 1,25 * (1 + 6800/1800) = 1,25 * 4,78 = 5,98$ valeur très proche de 6 V.

Expérimentalement, nous sommes partis d'une valeur de R1 proche de 2200 Ω, puis nous avons cherché la valeur idéale de R2. Si vous n'atteignez pas un résultat satisfaisant, augmentez R1 à 2700 Ω ou réduisez-la (comme dans notre cas) à 1800 Ω et recherchez de nouveau la valeur de R2.

Les autres formules servent essentiellement à calculer les deux condensateurs « CO » et « CT » (dans notre cas, C7 et

C8), qui agissent respectivement sur la fréquence de fonctionnement et sur l'ondulation résiduelle en sortie.

Normalement, les valeurs indiquées peuvent être adoptées. Il est vivement conseillé d'utiliser le filtre de sortie, composé d'une impédance de 1 μH (L2) et d'un condensateur de 100 μF (C9), cela réduit d'environ 90 % l'ondulation résiduelle originelle.

La tension fournie par le régulateur à découpage est ensuite stabilisée par un régulateur linéaire de 5 V, un LP2950Z (U3) qui est équivalent au classique LM78L05, mais avec l'avantage d'être un LDO (faible chute de tension) et de fonctionner correctement avec seulement une tension d'entrée de 6 V.

Nous aurions pu prévoir un régulateur à découpage qui fournisse directement 5 V en sortie, ce qui aurait simplifié le montage. La réponse réside dans le fait qu'**il n'est jamais commode d'alimenter un microcontrôleur avec une tension commutée** qui, comme nous le savons, oscille à une fréquence relativement élevée et risquerait de perturber facilement le fonctionnement d'un microcontrôleur (qui lui oscille à 16 MHz).

Dans notre montage, U3 est utilisé dans une configuration classique avec 4 condensateurs, deux de filtrage et deux d'anti-oscillation, idéal pour alimenter les microcontrôleurs.

Avec ce type d'alimentation, le décodeur consomme environ 65 mA en veille et environ 115 mA lors de la lecture et de l'affichage du codage. En réalité, étant donné que le montage est doté d'un régulateur « step-up » et l'utilisation qui

peut être faite en général, avec deux piles rechargeables, la consommation ne posera pas de problème.

Réalisation pratique

Les typons des circuits imprimés à l'échelle 1 sont disponibles en téléchargement dans le sommaire détaillé de la revue.

Une fois le circuit imprimé gravé et percé, vous pouvez passer à l'étape de soudure des composants.

Grâce à la conception soignée du circuit imprimé, l'assemblage des composants est une opération très simple. Comme il s'agit d'un circuit imprimé double face, il est nécessaire d'assembler les composants des deux côtés.

Reportez-vous au plan de montage dans ces pages pour visualiser les composants se trouvant sur la face supérieure et inférieure (côté soudures).

Certains composants montés sur la face inférieure doivent dépasser du couvercle du boîtier, c'est le cas notamment de l'afficheur LCD, des deux boutons P1 et P2 et de la LED LD1 (faites attention à la polarité de cette dernière !). De plus le connecteur CN1, une barrette femelle à profil bas, doit être soudé sur la face inférieure car il est prévu pour accueillir le connecteur mâle correspondant qui doit être soudé sur l'afficheur LCD.

Comme d'habitude, commencez par souder les composants ayant un profil bas pour continuer avec les composants de plus grande taille.

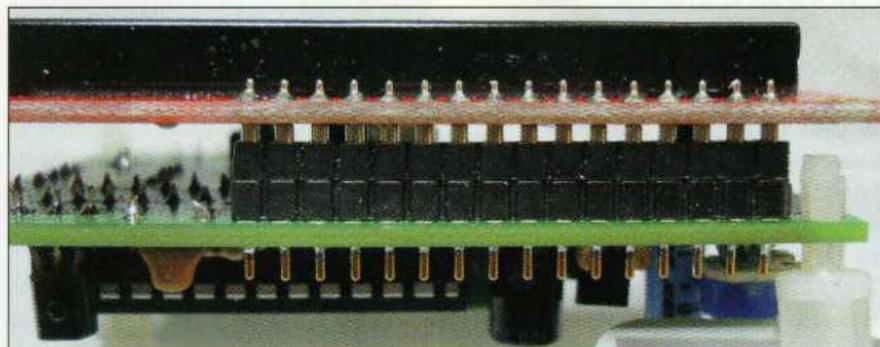


Figure 1 : Ici vous pouvez voir le double connecteur mâle/femelle entre le circuit imprimé et l'afficheur LCD.

Commencez par les résistances, le trimmer, la diode D1 en respectant l'orientation de sa bague.

Continuez avec les selfs, les supports de circuits intégrés en respectant leur orientation (détrompeur en « U »), les condensateurs non polarisés (céramique et polyester), le quartz.

Ensuite soudez les condensateurs électrolytiques en respectant leur polarité (le « - » est indiqué sur le boîtier et la patte la plus longue correspond au « + »),

De même pour le transistor T1, orientez correctement son méplat (vers le bord du circuit imprimé), de même pour le régulateur U3, son méplat fait face aux condensateurs C10 et C11.

Enfin, insérez dans les supports les deux circuits intégrés (U1 qui peut être acheté déjà programmé) et U2 en respectant leur orientation à l'aide du détrompeur en forme de « U » (voir le plan de montage).

Ensuite soudez le module HF, sa disposition étant asymétrique il ne s'insère que dans un sens. Le module dispose de broches très longues, dans notre cas, nous l'avons complètement inséré dans le circuit imprimé et, après le soudage nous avons coupé les broches.

Ensuite, vous pouvez souder le connecteur CN1 et les trois composants venant sur la face inférieure, en vous assurant que la LED arrive à la même hauteur que les deux boutons, et si possible à fleur du boîtier.

À cet égard, soulignons que les deux boutons P1 et P2 doivent avoir une hauteur de 13 mm (corps de 3 mm et la tige de 10 mm), sinon vous ne pourrez pas les faire dépasser du boîtier que nous avons utilisé pour notre prototype.

Nous avons pris un boîtier non percé de dimensions externes 125 mm x 70 mm x 39 mm (internes 120 mm x 63 mm). Il faut donc consacrer un peu de temps au perçage de la face avant du boîtier, notamment effectuer la découpe dans laquelle l'afficheur doit passer et les trois trous pour les boutons et la LED. Référez-vous au plan de perçage du boîtier.

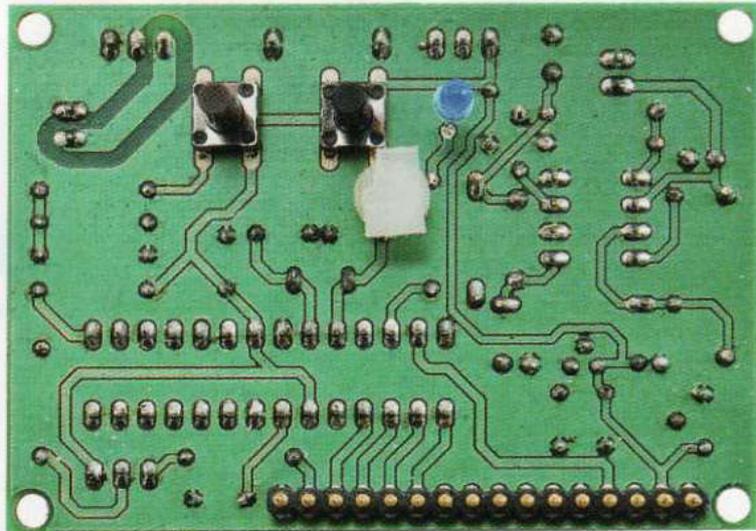


Figure 2 : l'entretoise en plastique permet de maintenir l'afficheur.

Notez que le connecteur CN1 femelle qui, comme indiqué, est monté sur la face inférieure du circuit imprimé, doit être de profil très bas, il en est de même pour le connecteur mâle correspondant à souder sur l'afficheur.

En fait, la distance entre le circuit imprimé et l'afficheur ne doit pas dépasser 7 mm (voir la figure 1).

Il est vivement recommandé d'utiliser un connecteur double mâle/femelle, car l'afficheur doit pouvoir être retiré facilement en cas de réparation du circuit principal.

Toutefois, si vous ne trouvez pas cette paire de connecteurs, il est possible d'utiliser une barrette mâle normale, en la soudant à la fois sur le circuit imprimé et sur l'afficheur, l'important étant de maintenir une distance entre les deux circuits imprimés de 6 à 7 mm.

La distance minimale évite tout court-circuit entre les soudures du circuit imprimé principal et les crochets métalliques qui bloquent le corps de l'écran LCD sur son circuit imprimé.

Quelle que soit la solution adoptée pour relier le circuit principal à l'afficheur, il sera nécessaire d'ajouter une entretoise en plastique, qui a la même épaisseur que la barrette, sur le côté « flottant » de l'afficheur. C'est pour cette raison que le circuit imprimé dispose d'un trou entre C4 et R5, afin de maintenir l'afficheur (voir la figure 2).

Comme nous le verrons un peu plus loin, l'un des quatre trous de fixation de l'afficheur doit correspondre à l'un des trous du couvercle du boîtier. Cela procure une stabilité mécanique à l'ensemble du circuit. En plus de percer le couvercle, à certains endroits, il faudra également travailler la découpe de l'afficheur.

Façonnage du boîtier

Nous réitérons que ce type de boîtier est optionnel, vous pouvez effectuer une mise en boîtier selon vos désirs.

Cependant, nous recommandons vivement ce modèle, car nous le considérons comme idéal pour loger l'ensemble du circuit, y compris le support de pile.

Si vous voulez utiliser le boîtier préconisé, procurez-vous un boîtier de la marque **Teko** référence « **COFFER-2.9** » et procédez de la manière suivante.

En figure 3, vous pouvez voir le plan de perçage du boîtier, c'est-à-dire les mesures et les coordonnées précises pour le perçage du couvercle.

L'usage idéal serait d'utiliser l'imprimante 3DRAG transformée en CNC, sinon vous pouvez découper délicatement le couvercle à l'aide d'une petite scie à métaux, puis réaliser les finitions à l'aide d'une lime fine jusqu'à obtenir une découpe précise dans laquelle l'afficheur se loge avec une certaine pression.

En ce qui concerne les trous pour la LED et les boutons, il est préférable de commencer par un petit foret de 1 à 2 mm de \varnothing , puis de terminer les trous avec une mèche de 3 mm pour la LED et de 3,5 mm pour les deux boutons.

La position de l'interrupteur n'est pas critique, mais nous recommandons de le fixer sur le couvercle, au-dessus du support de pile. Le trou sera ajusté au diamètre du modèle choisi.

À ce stade, il est nécessaire d'agrandir délicatement le trou de fixation de l'afficheur LCD pour qu'il coïncide avec la vis de fixation du couvercle. L'objectif est de faire coïncider le circuit imprimé de l'afficheur et le bas du couvercle comme illustré en figure 4.

Le support de pile s'intègre parfaitement dans le boîtier et il n'est donc pas nécessaire de le fixer. Pour plus de sécurité, vous pouvez mettre quelques gouttes de colle.

Maintenant, coupez et dénudez le fil rouge provenant du support de pile, pour le souder aux 2 pôles de l'interrupteur. Reliez ensuite le connecteur dans la prise mâle CN2 du circuit imprimé.

Il est préférable de vérifier avec un testeur la correspondance de la polarité de la pile avec celle du circuit, une inversion malheureuse pourrait causer de graves dommages !

Enfin, disposez 4 entretoises en plastique d'environ 25 mm dans les 4 trous du circuit imprimé. Si elles ne possèdent pas de partie adhésive, mettez de la colle sur les embases des entretoises afin qu'elles soient solidaires du boîtier.

Le fonctionnement du décodeur

Sur un côté du circuit imprimé, près du module HF, se trouve le connecteur CN3, destiné au raccordement d'une antenne adaptée à la fréquence de 433 MHz. Le circuit étant très sensible, vous pouvez vous passer de cet accessoire pour les petites distances (quelques mètres).

Cependant, pour augmenter la portée sans placer d'antenne externe, il est

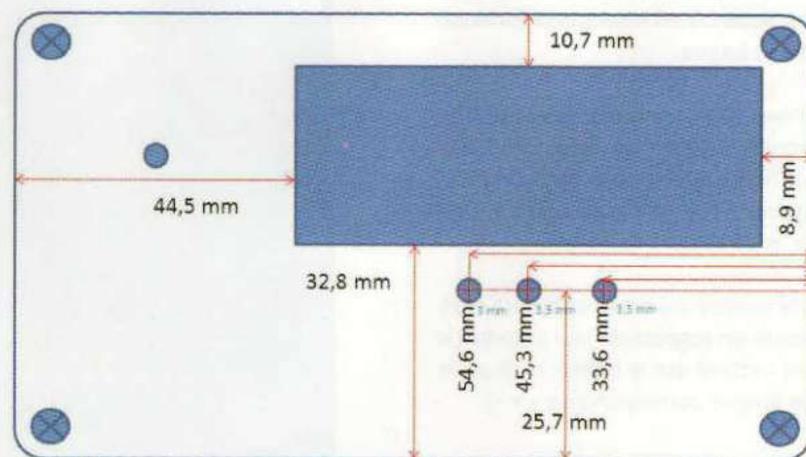


Figure 3 : Plan de perçage du boîtier

possible d'utiliser un morceau de fil de cuivre rigide émaillé d'environ 8 cm de long, à relier à la broche « Y » du connecteur « ANT ».

Vous devez gratter à l'aide d'un cutter l'une des extrémités du fil de cuivre et l'insérer dans le bornier. Ensuite, vous pouvez plier le fil rigide à l'intérieur du boîtier le long d'une paroi.

Un seul réglage doit être effectué au premier allumage. Cela consiste à ajuster le potentiomètre R10 jusqu'à ce que vous visualisiez les caractères sur l'afficheur LCD.

Dans la pratique, le contraste maximum garantit la meilleure visibilité.

À ce stade, le décodeur est prêt à fonctionner.

Lors de l'allumage, l'écran affiche pendant quelques secondes les deux codes stockés dans la mémoire EEPROM du microcontrôleur et passe immédiatement en mode veille pour économiser les piles. Il est déconseillé de laisser le mode actif trop longtemps.

Étant donné que l'afficheur est éteint la plupart du temps, pour mettre en évidence le mode veille, la LED LD1 s'allume pendant un instant toutes les 10 secondes. Vous pouvez ensuite l'éteindre complètement à l'aide de l'interrupteur.

Si le décodeur est en mode veille, **quatre opérations** sont possibles :

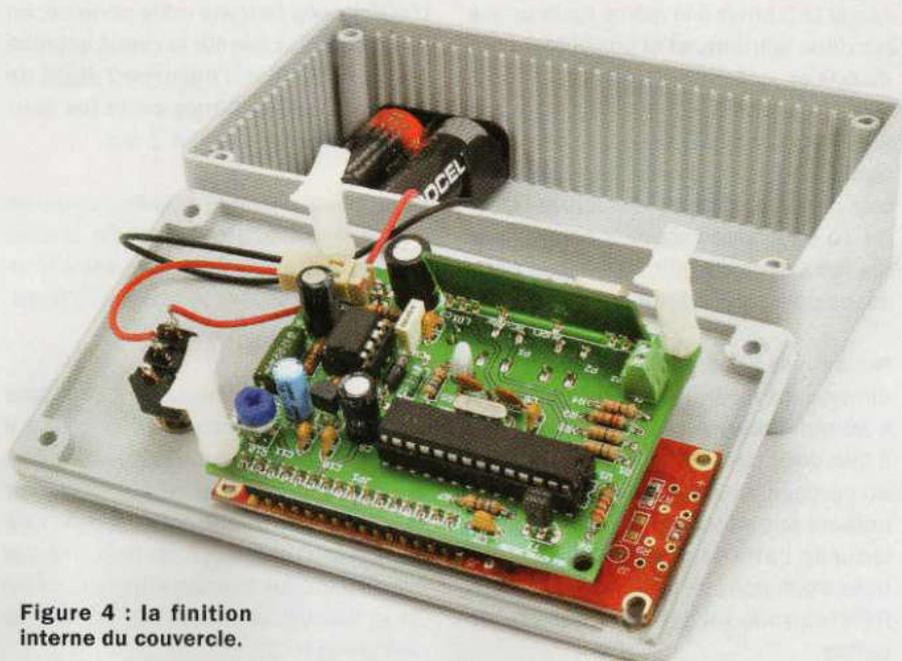


Figure 4 : la finition interne du couvercle.

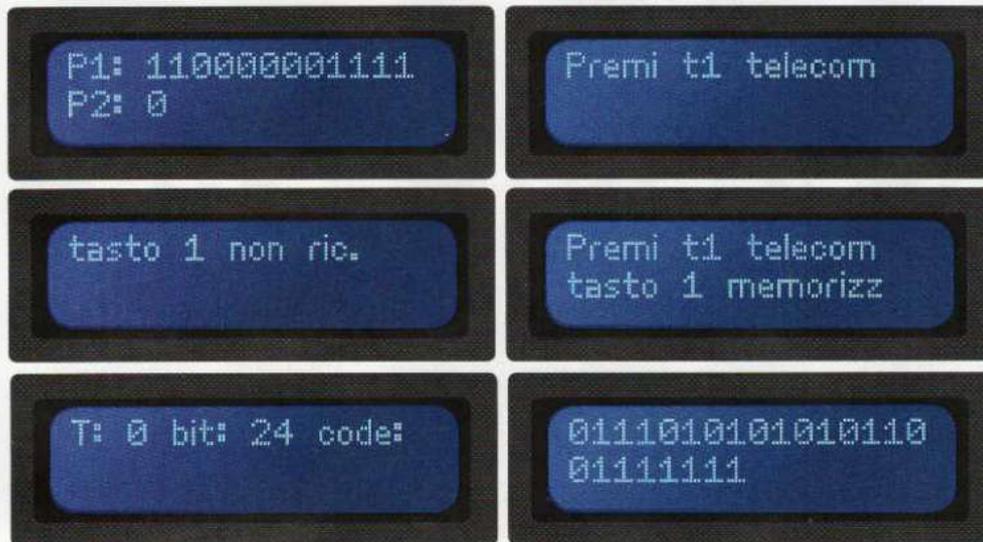


Figure 5 : les différents messages de fonctionnement visualisés sur l'afficheur LCD.

- décodage d'une radiocommande**
 en effectuant simplement une transmission. L'afficheur LCD visualise le code TO (cela signifie que le code n'est pas stocké dans l'un des deux emplacements de l'EEPROM), le nombre de bits du code et le code lui-même, représenté sous la forme binaire typique d'une suite de « 0 » et de « 1 ». Si la télécommande a plusieurs boutons, il y aura probablement plusieurs codes différents associés à chacun des boutons ;
- mémorisation d'un code en position « 1 »** : appuyez sur le bouton P1 (le plus proche de la LED), la LED commence à clignoter rapidement et l'affichage vous invitera à appuyer sur le bouton 1 de la télécommande. L'attente dure environ dix secondes, puis, en l'absence d'un code reconnaissable, le message « bouton 1 non reconnu » s'affiche. Inversement, si un code identifiable est reçu durant cette période, le message « bouton 1 mémorisé » apparaît. Dans les deux cas, comme toujours, le décodeur passera en mode veille, dans l'attente d'une nouvelle opération ;
- mémorisation d'un code en position « 2 »** : la procédure est identique à la précédente, mais est activée par le bouton P2. Le code est mémorisé dans le deuxième emplacement de la mémoire EEPROM ;
- effacement des deux emplacements de l'EEPROM** : appuyez

simultanément sur les boutons P1 et P2, les deux emplacements sont effacés instantanément et un message de confirmation est affiché sur l'écran LCD.

Une fois qu'un code est mémorisé dans l'un des emplacements de la mémoire EEPROM, si le décodeur l'identifie à nouveau, l'écran affiche le code initial « T : 1 » ou « T : 2 ».

La LED clignote alors respectivement une ou deux fois, confirmant de nouveau la correspondance avec le canal 1 ou 2. Cette fonction est particulièrement utile lorsque vous devez vérifier la validité du code d'un certain nombre de télécommandes identiques (par exemple celles à distribuer dans des immeubles). En fait, il ne sera pas nécessaire de marquer le code et de le comparer, cela évite des dépenses inutiles, des pertes de temps et des erreurs possibles, car la confirmation sera clairement visible.

Généralement, les codages sont composés de 12, 16, 24 ou 32 bits.

Dans les deux premiers cas, le code est visualisé sur une seule ligne.

Dans les deux autres cas, il est affiché sur 2 lignes. Les 16 premiers bits sur la première ligne et les 8 bits sur la deuxième ligne (24 bits), ou les 16 premiers bits sur la première ligne et les 16 derniers bits sur la deuxième ligne (32 bits). Lorsque le décodeur est mis sous tension, les deux codes T1 et T2 sont affichés simultanément.

Si l'un d'eux ou les deux dépassent 16 bits, un défilement aura lieu, c'est-à-dire qu'après avoir affiché les 16 premiers bits, l'affichage défilera vers la gauche pour visualiser la suite des 8 ou 16 bits supplémentaires.

La figure 5 représente un aperçu des différents messages à l'écran.

Une dernière remarque, lors de nos tests, nous avons très facilement décodé des dizaines de télécommandes de différentes marques, mais pour y parvenir, nous avons travaillé longtemps pour trouver un timing commun.

Il peut parfois arriver que certaines télécommandes soient particulièrement difficiles à décoder. Pour cette raison, nous ne pouvons pas garantir un succès à 100 %, mais nous vous assurons que nous sommes très proches de ce pourcentage.

Il convient toutefois de rappeler qu'en aucun cas il ne sera possible de décoder des télécommandes de type « rolling code »

La fréquence des télécommandes doit obligatoirement être de 433,92 MHz, faute de quoi le signal ne sera pas reçu par le module HF. ■



ANALYSEUR DE SEMI-CONDUCTEUR

Nous vous présentons dans cet article un ensemble d'instruments de laboratoire vous permettant d'identifier, de vérifier les caractéristiques et de tester les transistors BJT, les transistors FET, les thyristors, les diodes, etc., à un coût abordable et simple d'utilisation.



de Vincenzo MENDOLA

Il y a encore quelques années, l'idée de disposer d'un instrument portable, relativement bon marché et universel pour l'identification automatique et la vérification de pratiquement tous les semi-conducteurs, était presque une utopie.

Mais aujourd'hui, grâce aux progrès technologiques remarquables, il est possible de trouver sur le marché des appareils ayant des hautes performances avec un faible coût, indispensables pour le laboratoire de tout amateur d'électronique.

Les expérimentateurs, les étudiants, mais aussi les professionnels (y compris les réparateurs et les bureaux d'études) des départements de recherche et développement fabriquant des prototypes électroniques ont besoin de ces types d'instruments.

Nous souhaitons vous présenter deux de ces analyseurs de semi-conducteurs, fabriqués par une société britannique leader dans le monde pour ce type de produits, il s'agit de la firme **Peak Electronic Design Ltd**. Ces deux instruments sont l'**Atlas DCA55** et l'**Atlas PRO DCA75**. Ces produits sont disponibles auprès de la société COMELEC. Le DCA55 peut être considéré comme la version d'entrée de gamme, tandis que le modèle « PRO » ajoute des fonctionnalités supplémentaires à celles du DCA55.

Les deux instruments sont capables d'identifier les broches et de vérifier le bon fonctionnement des **transistors BJT**, des **Darlington**, des **MOSFET** (les types à enrichissement et à appauvrissement), des **JFET**, des **triacs**, des **LED**, des **diodes**. Le **DCA75 PRO**, en plus des types de semi-conducteurs précédents, est également capable d'identifier les

IGBT, les diodes zener et de nombreux **régulateurs** de tension à 3 broches ayant une tension de sortie inférieure à 8 V.

La caractéristique la plus intéressante de la version « PRO », qui justifie également la différence de prix, est la possibilité d'être **connecté à un port USB** d'un ordinateur (compatible à partir de Windows XP ou supérieur). À partir de là, vous pouvez **tracer les courbes caractéristiques des semi-conducteurs** grâce au logiciel fourni sur une clé USB.

Une autre différence entre les deux versions réside dans la possibilité de mettre à jour le firmware de la version « PRO », afin d'implémenter des fonctionnalités supplémentaires au fur et à mesure de leur développement par la société Peak Electronic Design Ltd.

Les deux analyseurs de semi-conducteurs sont contenus dans un petit boîtier élégant en plastique, et sont alimentés par une pile GP23A de 12 V pour le DCA55 et par une pile AAA pour le DCA75. Ils disposent de 3 contacts plaqués or auxquels les broches servant à tester les composants sont connectées.

Regardons comment les deux modèles se comportent lors d'un test d'un transistor bipolaire (BJT). En reliant les trois broches du transistor à l'appareil, l'afficheur LCD indique « Unknown/Faulty component » en cas de court-circuit d'une ou plusieurs jonctions.

L'Atlas DC55 affiche « Short circuit on Red Green Blu ». En fonction des jonctions impliquées, le DCA75 « PRO » affiche « Red, Green & Blue leads shorted » en cas d'une défaillance similaire.

Si le transistor n'est pas endommagé, le DCA55 indique le type : germanium, silicium, Darlington, NPN ou PNP et la présence éventuelle d'une diode de protection en parallèle sur la jonction « collecteur-émetteur ».

En faisant défiler l'affichage à l'écran à l'aide du bouton « scroll/off », la broche du composant et la couleur du câble à laquelle ce dernier est reliée sont affichées.

En faisant défiler l'affichage sur l'écran LCD à l'aide du bouton « scroll/off », les fonctions des broches du composant reliées aux pinces de couleur sont affichées (RED = GATE, etc.).

En continuant d'appuyer sur la touche de défilement, le gain en courant « h_{fe} », le courant de test relatif, le V_{BE} (avec le courant qui lui est référé) et le courant de fuite I_{cbo} (leakage) sont affichés.

La version « PRO » dispose d'un afficheur qui visualise les informations sur trois lignes au lieu de deux, ainsi que le symbole graphique du composant connecté à l'appareil, ce qui est utile pour identifier immédiatement le semi-conducteur.

Dans ce cas également, pour visualiser les différentes informations relatives au transistor testé, il est essentiel de faire défiler l'affichage à l'aide de la touche « scroll ».

Dans les deux modèles analyseurs, le test complet d'un composant nécessite quelques secondes après avoir appuyé sur le bouton « on - test ».

Comme pour un transistor BJT, l'analyse d'un transistor MOSFET affiche le type

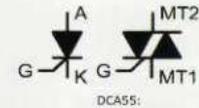


Figure 1

(enrichissement ou déplétion) avec le symbole relatif (pour la version « PRO »), le brochage, la tension de seuil et le courant de test.

En ce qui concerne les JFET, la structure étant essentiellement symétrique par rapport à la grille (gate), les deux instruments ne sont généralement pas en mesure de distinguer le drain de la source et donc ils identifient uniquement la broche « GATE ».

Le DCA75 est également capable de déterminer la tension de pincement (V_{GS} (off)), la tension V_{GS} (ON) et le courant relatif du DRAIN ainsi que la transconductance et l' ID_{SS} .



Figure 2

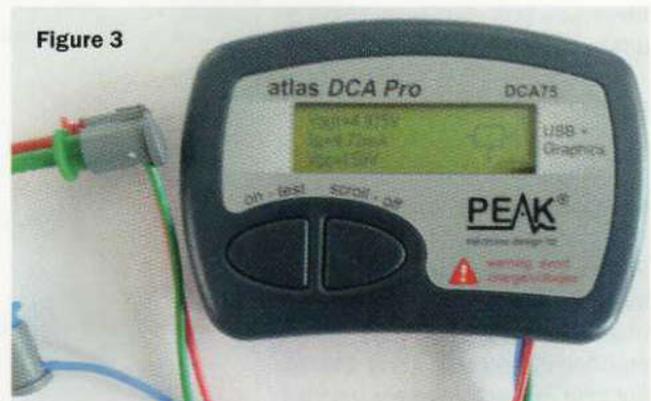


Figure 3

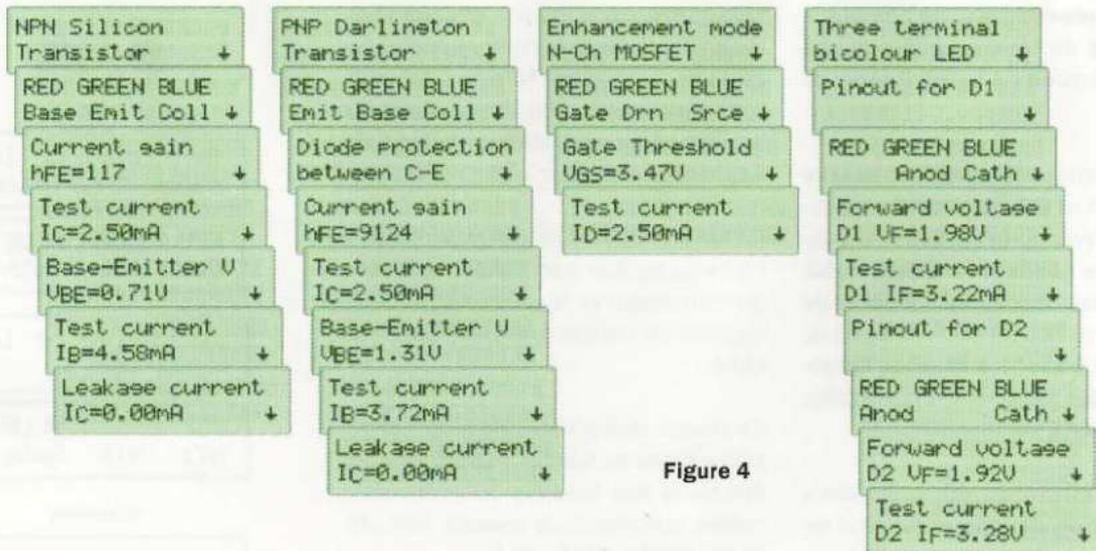


Figure 4

Une autre fonction utile présente sur les deux modèles est le test des LED et des doubles LED, qui permet d'identifier rapidement, en plus de la polarité, la tension de seuil (V_f) et le courant auquel elle est mesurée. De plus, pendant le test, la LED s'allume brièvement pour que vous puissiez également voir sa couleur (ou les couleurs dans le cas de LED bicolores).

Comme nous l'avons mentionné, les deux instruments Atlas sont capables de tester les composants de type SCR et TRIAC (avec un courant d'amorçage ou de gâchette inférieur à 10 mA). La figure 1 montre comment les deux composants sont identifiés par les deux modèles d'analyseurs.

Les figures 2 et 3 illustrent l'Atlas DCA75 « PRO » qui identifie un régulateur 7805 et met en évidence, en plus du brochage, la tension de sortie et la chute de tension entre l'entrée et la sortie.

La figure 4 montre une série de captures d'écrans du modèle DCA55 obtenues lors de divers tests de différents types de composants.

Le modèle « PRO » est capable d'identifier une diode zener avec une tension de fonctionnement ne dépassant pas 11 V.

Si vous souhaitez tester des diodes avec une tension plus élevée (jusqu'à 50 V), nous vous recommandons le testeur **Atlas ZEN50**, qui peut également tester les **VDR**,

TVS et **Transzorb**, avec une résolution maximale de 20 mV et un courant de test sélectionnable parmi les valeurs suivantes : 2 mA, 5 mA, 10 mA et 15 mA.

L'Atlas ZEN50, spécifique aux diodes zener, permet également de mesurer les caractéristiques de conduction de nombreux autres types de composants, tels que les diodes normales, les LED, les supprimeurs de transitoires tels que les VDR, TVS et Transzorb.

Il est aussi capable de mesurer la résistance dynamique du composant (en effectuant trois groupes de lectures du courant et de la tension afin d'obtenir une courbe « tension/courant » passant par trois points). Cela est utile pour identifier les capacités de redressement d'une diode (voir la figure 5).

$V_z=47.17V @ 10mA$
 $Slope R=196\Omega$

Figure 5

Il existe une différence importante par rapport aux modèles DCA55 et DCA75 « PRO », le **ZEN50 a été conçu pour polariser la diode zener en inverse**.

Par conséquent, ses deux pinces de test sont identifiées par les couleurs rouge et noire et **leur positionnement doit être respecté** (le fil rouge sur la cathode de la zener et le fil noir sur l'anode).

Cette observation est particulièrement importante afin de ne pas endommager les LED connectées à l'instrument, dont

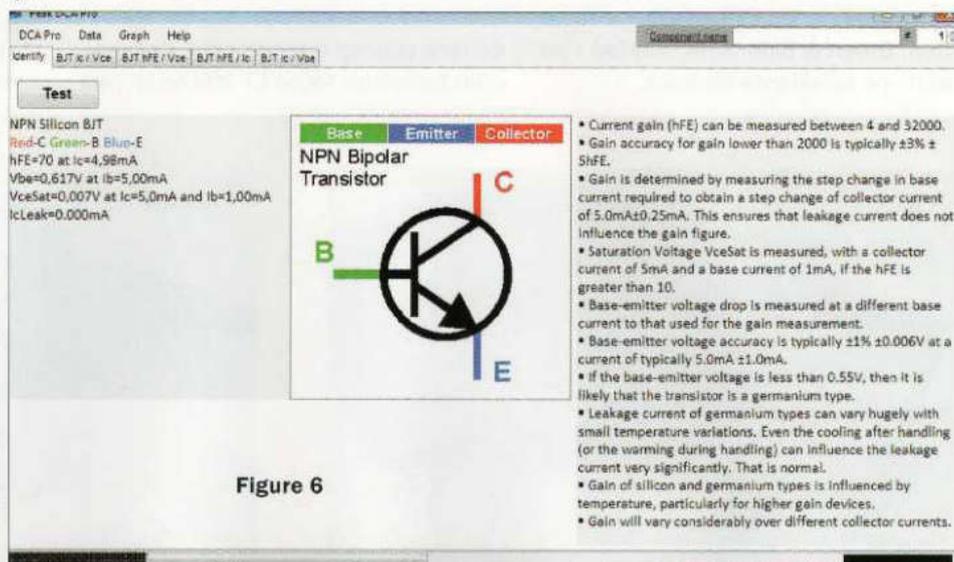


Figure 6

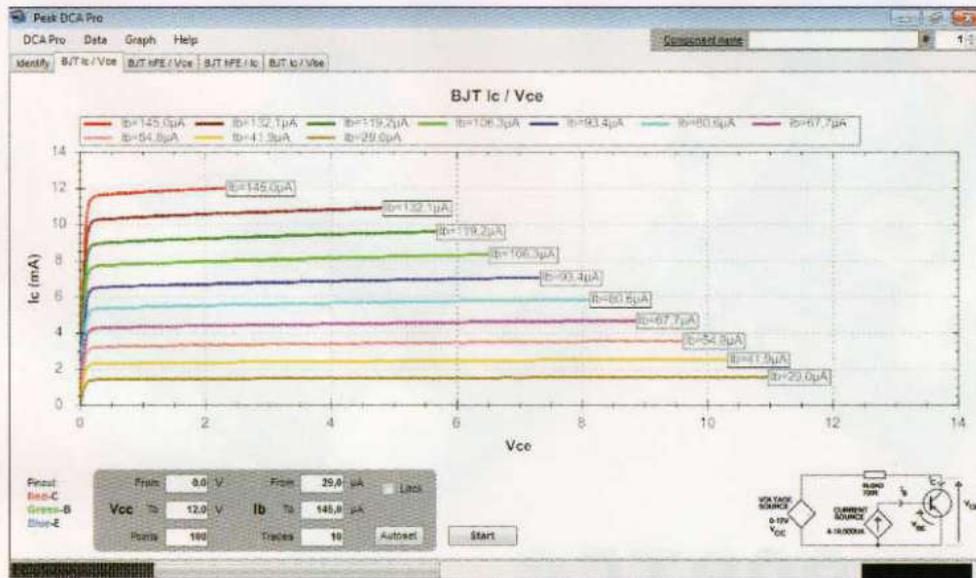
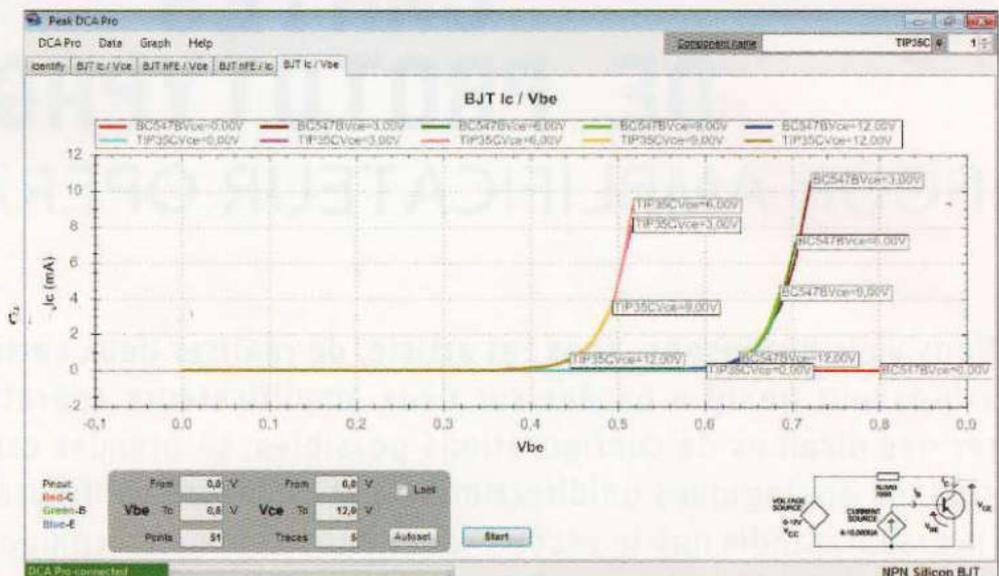


Figure 7

Figure 8



le test est prévu pour les spécifications du composant. Si la polarité n'est pas respectée, la LED sera irrémédiablement endommagée par la haute tension inverse, pouvant atteindre 60 V.

Le ZEN50 dispose du même type de boîtier plastique compact, dont seule la couleur change (vert pour le ZEN50, bleu pour le DCA55 et gris foncé pour le DCA75) ainsi que l'inscription d'identification.

Une fonction qui distingue le DCA75 du DCA55, est sa capacité à tracer les courbes des composants une fois connecté au port USB d'un ordinateur avec un système d'exploitation Windows (à partir de XP), grâce au logiciel fourni (voir la figure 6). En plus de la visualisation graphique des courbes, il est également possible d'exporter les données

sous la forme d'une feuille de calcul à l'aide du menu « Data ».

Bien que l'utilisation de ces outils soit vraiment intuitive et immédiate, pour exploiter tout le potentiel offert et en faire une utilisation correcte, nous vous conseillons de lire attentivement le manuel d'utilisation fourni par le fabricant.

Visitez également le site officiel à l'adresse :

<https://www.peakelec.co.uk/>

Conclusion

Au fil des ans, nous vous avons proposé en kit divers instruments de test pour les semi-conducteurs, mais aucun

n'était vraiment complet et n'a pu tester une variété de composants, comme ceux de la firme Peak.

La qualité des instruments de mesure, que nous avons testés lors de nos manipulations en laboratoire, nous permet de vous les conseiller.

Ces instruments de petite taille, mais à un prix vraiment intéressant, conviennent à tous ceux qui travaillent dans l'électronique, à la fois pour les amateurs et les professionnels.

Une fois que vous aurez commencé à les utiliser, ils deviendront rapidement incontournables pour votre laboratoire.

Essayez-les et vous ne pourrez qu'être d'accord avec nous sur leur utilité et leur qualité !



CARTES DE PROTOTYPAGE POUR AMPLIFICATEUR OPÉRATIONNEL

de Boris Landoni

Nous vous proposons, dans cet article, de réaliser deux cartes de prototypage ou « breakout board » basées sur deux amplificateurs opérationnels, afin de tester des dizaines de configurations possibles. Le premier est un amplificateur de signaux analogiques unidirectionnels et peut être configuré en inverseur ou non inverseur, tandis que le second fonctionne comme un comparateur et peut piloter des charges allant jusqu'à 2 A.

Les amplificateurs opérationnels sont utilisés depuis des décennies dans l'électronique analogique et numérique, car, bien qu'ils soient des amplificateurs linéaires et qu'ils se prêtent donc à l'amplification de signaux audio ou de radiofréquences, ils ont la caractéristique d'effectuer des fonctions simples.

En fait, **ils amplifient la différence entre les tensions appliquées sur leurs entrées inverseuses** (le potentiel appliqué est amplifié et inversé) et **sur leurs entrées non inverseuses** (le potentiel appliqué est amplifié et ne subit aucune inversion).

Le **gain élevé**, théoriquement infini, qui les caractérise, permet de **réaliser de véritables opérations logiques**. Il suffit de ne pas leur appliquer de contre réaction ou de leur appliquer une rétroaction positive (c'est-à-dire injecter une fraction de la tension de sortie sur l'entrée non inverseuse) pour que l'état de la sortie dépende de la différence entre les tensions appliquées sur les deux entrées.

Ces tensions peuvent également être des signaux logiques. C'est précisément en raison de leur importance que les amplificateurs opérationnels sont souvent utilisés et c'est pourquoi nous avons décidé de réaliser deux cartes de prototypage pouvant être intégrées dans n'importe quel circuit existant.

La première carte comporte un amplificateur opérationnel configuré en tant qu'**amplificateur linéaire**, tandis que l'autre carte utilise un AOP (par la suite non utiliserons ce sigle pour désigner un amplificateur opérationnel) configuré en tant que **comparateur**.

L'amplificateur linéaire

Dans la première carte, l'AOP est utilisé en tant qu'amplificateur en tension. Il permet d'élever (augmenter) le niveau électrique de tout signal, qui dans notre cas peut être continu ou variable, **mais doit être unidirectionnel**.

Grâce au double commutateur **SW1**, l'AOP peut être **configuré en tant qu'inverseur** (position INV) ou **en tant que non inverseur** (position NINV). En effet, SW1 permet de diriger le signal d'entrée soit sur l'entrée inverseuse, soit sur l'entrée non inverseuse. Le commutateur doit être double car une partie commute l'entrée, l'autre gère la liaison de rétroaction (contre réaction).

Nous pouvons en comprendre la raison en sachant comment est appliquée la contre réaction à l'AOP lorsqu'il fonctionne en amplificateur linéaire. Dans cette configuration, la **rétroaction est toujours négative** (c'est-à-dire qu'un réseau de composants réinjecte une partie du signal de sortie sur l'entrée inverseuse).

Cela permet de réduire le gain en tension par rapport à un fonctionnement en boucle ouverte (sans contre réaction) afin de pouvoir régler le gain à une valeur acceptable et éviter ainsi une distorsion du signal en sortie, tout en conférant une stabilité au montage.

Si l'AOP est configuré en inverseur, il doit amplifier le signal en inversant la polarité. Une résistance de contre réaction doit être connectée entre la sortie et l'entrée non inverseuse (dénommée R_f) et une autre résistance entre R_f et le point « IN » auquel est appliqué le signal d'entrée (nous l'appelons R_i). Le gain en tension est alors égal à :

$$G = - R_f / R_i$$

Cela signifie qu'un signal (V_i) appliqué sur l'entrée présente en sortie une valeur égale à :

$$V_u = - V_i * R_f / R_i$$

Si V_i vaut 1 mV_{eff}, et si nous prenons R_f égale à 1 M Ω et R_i égale à 1 k Ω , l'AOP aura un gain de 1 000 et la tension de sortie V_u aura comme valeur :

$$V_u = - 1 \text{ mV} * 1 \text{ M}\Omega / 1 \text{ k}\Omega \\ = - 1 \text{ mV} * 1000 = - 1 \text{ V}$$

Dans une configuration non inverseuse, cependant, même si la résistance de contre-réaction est présente et que R_i est reliée à la masse, le signal d'entrée est directement appliqué sur l'entrée non inverseuse.

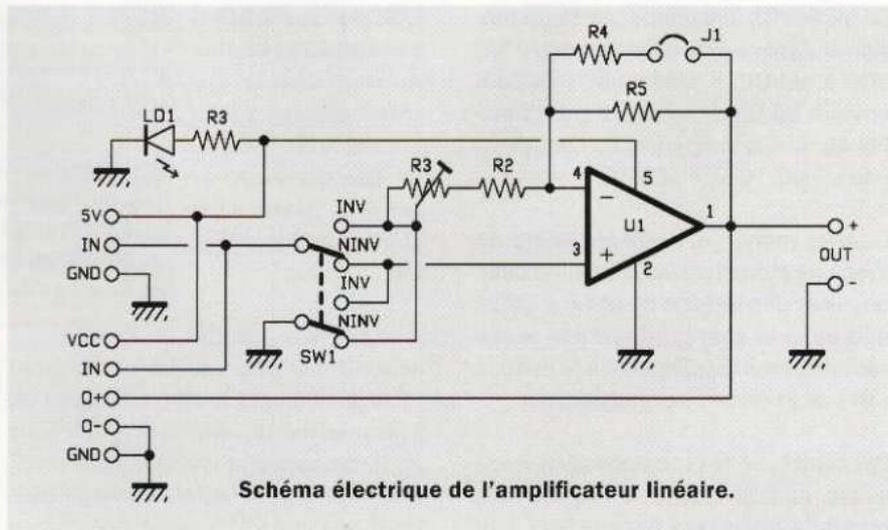


Schéma électrique de l'amplificateur linéaire.

Le gain en tension G est alors égal à :

$$G = (R_f + R_i) / R_i$$

En utilisant les mêmes valeurs que dans le cas précédent, c'est-à-dire $R_f = 1 \text{ M}\Omega$ et $R_i = 1 \text{ k}\Omega$, le gain en tension vaut **1 001**, il est donc légèrement supérieur à celui de la configuration inverseuse.

L'écart de gain entre les deux configurations dépend du rapport de proportion entre les résistances R_f et R_i , en ce sens que si cette dernière est beaucoup plus petite que R_f , le gain est pratiquement égal dans les deux configurations. Si par contre, R_i a une valeur similaire à celle de R_f , la différence de gain entre les deux configurations augmente.

L'interrupteur SW1 permet de réaliser les deux configurations en reliant simultanément, pour la configuration non inverseuse, le signal d'entrée « IN » vers la broche 3 de U1 et l'extrémité de la résistance R_i vers la masse et, pour la configuration inverseuse, en reliant « IN » vers R_i et la broche 3 de U1 vers la masse.

Notez que dans notre montage, la résistance R_i est en réalité composée de deux résistances (R_2 et R_3) dont l'une est un trimmer (R_3) pouvant faire varier le gain. Le gain peut également être réduit en fermant le cavalier J1, la résistance R_4

se trouve alors en parallèle avec R_5 (qui dans le montage correspond à R_f).

Dans ce cas, pour le calcul du gain, nous devons considérer la résistance résultante de R_4 en parallèle avec R_5 :

$$R_f = R_4 * R_5 / (R_4 + R_5)$$

Une autre chose à noter est que dans la configuration inverseuse, la résistance d'entrée du circuit est théoriquement déterminée par R_i , c'est-à-dire la somme de R_2 et R_3 . En faisant varier le trimmer, la résistance d'entrée varie et par conséquent le gain de l'AOP.

Le circuit utilisé dans le montage est le MCP6L01 de Microchip, disponible en boîtier CMS de type SC70, SOT-23 et SOIC 8. Le produit gain/bande passante est de 1 MHz, c'est-à-dire qu'en amplifiant un signal de 100 kHz, le gain en tension peut être au maximum de 10.

Nous pouvons dire que pour un AOP, **l'augmentation du gain provoque une diminution de la fréquence maximale amplifiable et inversement.**

Caractéristiques techniques de l'amplificateur linéaire

- Tension d'alimentation : 5 VDC ;
- Gain en configuration inverseur : de 0,3 à 10 ;
- Gain en configuration non inverseur : de 1,3 à 11 ;
- Impédance d'entrée : 2 M Ω .

Le MCP6L01 fonctionne avec une tension d'alimentation comprise entre 1,8 VDC et 6 VDC. Il consomme très peu, environ 85 μ A au repos, et peut fonctionner à des températures comprises entre - 40 °C et + 125 °C.

Dans le montage, l'AOP est alimenté avec une mono tension, il ne peut donc amplifier une tension d'entrée positive que dans la configuration non inverseuse (tension appliquée sur le contact « IN » de la carte de prototypage).

Par contre, dans la configuration inverseuse, peut uniquement amplifier des tensions négatives par rapport à la masse, peu importe si elles sont continues ou variables.

L'amplitude du signal d'entrée doit être déterminée en tenant compte du gain choisi, car la tension de sortie ne peut pas dépasser la tension d'alimentation (notez que, comme il s'agit d'un AOP « rail to rail », sa tension de sortie peut atteindre celle d'alimentation appliquée entre les broches 5 et 2).

Notez que dans la configuration non inverseuse, si des tensions inférieures à zéro sont appliquées elles seront « coupées » car la sortie ne peut pas chuter en dessous de 0. De même, dans la configuration inverseuse, si des tensions supérieures à zéro sont appliquées, elles subiront le même sort.

Complétons la description de la carte en indiquant la présence d'une LED avec sa résistance de limitation du courant en série, qui permet de signaler le fonctionnement du montage.

Tous les contacts d'entrée et de sortie sont reportés sur un côté du circuit imprimé, vous pouvez ainsi souder une barrette mâle ou femelle afin d'insérer le montage dans un système existant. De plus, tous les contacts sont répétés sur deux connecteurs de type « JST » au pas de 1,25 mm, un à deux pôles pour la sortie et l'autre à trois pôles pour le signal d'entrée et l'alimentation.

Le comparateur

Passons maintenant à la deuxième carte de prototypage qui utilise un

AOP de type **ADCM-P371AKSZ** de la firme **Analog Device**. Ce circuit est conçu pour fonctionner comme un comparateur de tension plutôt que comme un amplificateur linéaire.

Il fonctionne plutôt comme une **bascule**, sa sortie peut prendre deux niveaux, soit celui de la **masse** (pratiquement 0 V) soit celui de l'**alimentation** en atteignant pratiquement le potentiel de l'alimentation positive, car il s'agit aussi d'un AOP de type « rail to rail ».

L'ADCM-P371AKSZ fonctionne dans une plage de tension comprise entre 2,25 VDC et 5,5 VDC et ne consomme que 4 μ A au repos. Ce circuit est prévu pour fonctionner à des niveaux de tensions sur ses entrées plus élevés que la tension d'alimentation, le circuit peut supporter sur ses entrées des tensions allant jusqu'à 22 V. Sa sortie est de type « push-pull ».

Ce qui caractérise le comparateur par rapport à l'amplificateur linéaire, c'est qu'**il n'y a pas de contre réaction**, c'est-à-dire qu'elle existe, mais elle est positive, comme nous l'expliquerons plus loin. L'absence de contre réaction garantit que la commutation de la sortie se produit même pour des différences de tensions infinitésimales entre les entrées non inverseuses et inverseuses.

Le comparateur fonctionne, en ce qui concerne la corrélation entre les tensions d'entrée et de sortie, comme un AOP classique.

Caractéristiques techniques du comparateur

- Tension d'alimentation : 5 VDC ;
- Seuil sans hystérésis : de 0,1 à 4,9 V ;
- Hystérésis (inverseur) : de 0 à $\pm 0,4$ V ;
- Hystérésis (non inverseur) : ± 5 mV ;
- Courant de sortie : 2 A max.

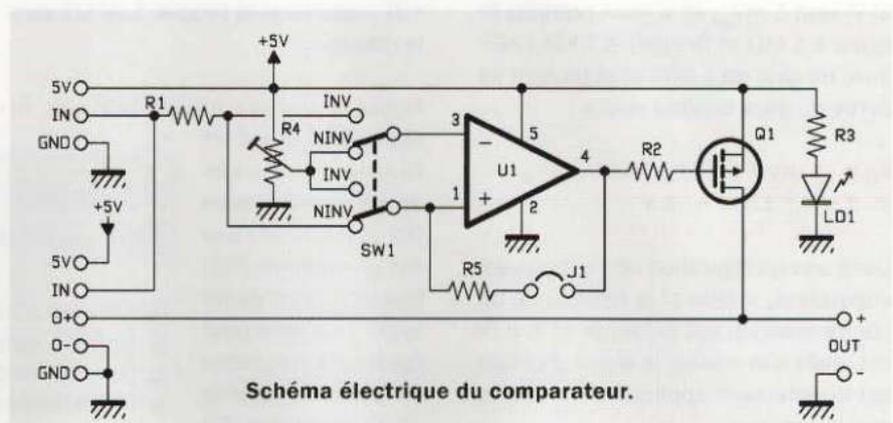
Ainsi, ce qui a été décrit pour la breakout board précédente reste valable ici.

La différence par rapport au schéma précédent est que la sortie du comparateur pilote la grille d'un **MOSFET** de type canal P à enrichissement. Ce dernier se comporte comme un commutateur statique et permet de faire passer un courant d'environ **2 A** au maximum.

Le MOSFET a été ajouté car, normalement, les comparateurs sont utilisés pour contrôler des systèmes lorsqu'une certaine condition de tension survient sur les entrées. Dans notre cas, le MOSFET permet de transformer la carte de prototypage en un interrupteur crépusculaire ou un thermostat. Il suffit de connecter sur l'entrée « IN » un diviseur de tension (alimenté d'un côté par le + 5 V et de l'autre relié à la masse) constitué d'une résistance et, respectivement, d'une photorésistance ou d'une thermistance.

La résistance **R5**, qui peut être insérée avec le cavalier **J1**, permet **d'appliquer ou de ne pas appliquer une rétroaction positive** au comparateur afin de disposer d'une hystérésis.

Ainsi, nous obtenons un **comparateur à deux seuils ou trigger de Schmitt**



(on dit aussi comparateur à hystérésis) comprenant un AOP en mode non-linéaire. Sa sortie passe à 1 lorsque son signal analogique en entrée franchit le seuil supérieur, mais elle ne retombe à 0 que lorsqu'il franchit le seuil inférieur, qui est différent.

Examinons le cas où **J1 ouvert**, donc **sans contre réaction**. Dans ces conditions, la sortie du comparateur est à un niveau haut lorsque la broche 1 sera plus positive que la broche 3 et prendra un niveau bas dans le cas contraire, c'est-à-dire lorsque la broche 3 sera plus positive que la broche 2.

Le double déviateur SW1 permet de faire fonctionner le comparateur en mode inverseur (si la tension d'entrée dépasse le seuil, la sortie passe à un niveau bas et provoque la conduction du MOSFET) ou en mode non inverseur (la sortie prend un niveau haut lorsque la tension appliquée sur l'entrée « IN » dépasse le seuil prédéfini).

Pour régler la tension de seuil du comparateur (celle à laquelle se produit la commutation de la sortie), nous utilisons le trimmer R4 qui, lorsque SW1 est en position « INV », voit son point milieu relié à la broche 1 de U1. Ainsi la tension de référence du comparateur est appliquée sur l'entrée non inverseuse.

Par contre, lorsque SW1 est sur la position « NINV », le point milieu de R4 se trouve relié à la broche 3, de sorte que la référence de tension est appliquée sur l'entrée inverseuse.

Pour résumer, SW1 commute en même temps l'entrée « IN » sur l'entrée non inverseuse et le point milieu du trimmer sur l'entrée inverseuse et inversement.

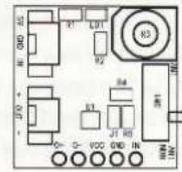
Dans le premier cas, lorsque la tension appliquée sur l'entrée « IN » dépasse celle dérivée par le trimmer R4 (il s'agit de la tension de référence qui dépend du réglage du trimmer), l'entrée inverseuse du comparateur se trouve à un potentiel supérieur à celui de l'entrée non inverseuse, ainsi la sortie de U1 passe d'un niveau logique haut à un niveau logique bas, ce qui provoque la conduction (ON) du MOSFET Q1. Le drain porte la sortie « OUT » à 5 V.

Plan de montage de l'amplificateur linéaire

Plan de câblage des composants de l'amplificateur linéaire.



Photo de l'un de nos prototypes de l'amplificateur linéaire.



Liste des composants de l'amplificateur linéaire

R1..... 470 Ω boîtier CMS 0805
 R2..... 1 k Ω boîtier CMS 0805
 R3..... trimmer 10 k Ω
 R4..... 4,7 k Ω boîtier CMS 0805
 R5..... 10 k Ω boîtier CMS 0805
 LD1.... LED verte boîtier CMS 0805
 U1..... MCP6L01T-E/LT

Divers

Barrette mâle soudée à 90° 5 pôles

Double déviateur coulissant coudé à 90°

Connecteur JST 1,25mm 2 pôles pour ci

Connecteur JST 1,25mm 2 pôles fils volants

Connecteur JST 1,25mm 3 pôles pour ci

Connecteur JST 1,25mm 3 pôles fils volants

NB : les typons des circuits imprimés à l'échelle 1 sont disponibles en téléchargement dans le sommaire détaillé de la revue.

Dans le cas contraire, lorsque la tension appliquée sur l'entrée « IN » est inférieure à celle de référence du potentiomètre R4, la sortie du comparateur se retrouve à 5 V (niveau haut) et le MOSFET se trouve bloqué, isolant ainsi la sortie « OUT » (0 V).

Dans le second cas, c'est-à-dire le comparateur en mode non inverseur, lorsque la tension appliquée sur l'entrée « IN » dépasse le seuil de celle de référence, la sortie de U1 passe à 5 V et maintient Q1 bloqué, tandis que si elle est inférieure au seuil, la sortie de U1 (broche 4) passe à 0 V et fait conduire le MOSFET, dont le drain porte le potentiel de 5 V sur la sortie de U1.

Notez qu'en fonctionnement avec J1 ouvert (sans hystérésis), la résistance R1 est sans importance, en ce sens que le comparateur de tension n'absorbe théoriquement aucun courant sur ses entrées, il n'y a pas de chute

de tension aux bornes de R1. Donc, la tension appliquée sur l'entrée « IN » se retrouve sur l'entrée correspondante de U1 en fonction de la position de SW1.

Examinons maintenant le fonctionnement du montage avec **J1 fermé**. La résistance **R5 introduit alors une contre réaction positive** sur le comparateur.

Cela détermine une hystérésis minimale qui permet d'éviter toute commutation instable, par exemple dans des applications telles que l'interrupteur crépusculaire où le passage du jour à la nuit (ou vice versa) se ferait dans des conditions météorologiques nuageuses.

La commutation du comparateur deviendrait instable et le MOSFET inverserait continuellement son état jusqu'à ce qu'une luminosité suffisante soit présente.

Dans ce cas, supposons que nous connectons sur l'entrée « IN » un diviseur de tension constitué d'une photorésistance reliée entre le 5 V et l'entrée « IN ». Positionnons SW1 sur « NINV », dès que le jour commence à s'obscurcir, la tension sur l'entrée « IN » diminue ce qui rend l'entrée non inverseuse moins positive que l'entrée inverseuse.

En raison des variations dues par exemple à un passage nuageux, la différence de potentiel sur les entrées de U1 peut varier suffisamment pour activer la sortie (par exemple, le potentiel sur l'entrée non inverseuse croît ou celui sur l'entrée inverseuse diminue).

Pour comprendre le fonctionnement de l'hystérésis, émettons l'hypothèse que l'entrée non inverseuse du comparateur soit plus positive que l'entrée inverseuse et que, pour cette raison, la sortie se trouve à un niveau haut.

La résistance de contre réaction forme avec R1 un diviseur de tension, celui-ci augmente légèrement le potentiel de l'entrée non inverseuse par rapport au diviseur de tension comportant la photorésistance.

Supposons maintenant que la tension sur ce dernier diminue suffisamment pour rendre le potentiel de la broche 3 supérieur à celui de la broche 1 de U1. La sortie de ce dernier passe à un niveau logique bas et donc R5 se retrouve à la masse, ce qui a pour effet de diminuer d'avantage le potentiel sur l'entrée non inverseuse, de sorte que même en cas de légère variation de la tension, cela n'a aucun effet sur le comportement du montage.

Terminons la description du schéma avec la présence de la résistance R2, qui placée en série avec la grille du MOSFET Q1, permet d'atténuer le comportement capacitif de cette dernière. Lorsque la tension qui lui est appliquée varie, la résistance absorbe le pic de courant, ce qui a pour effet de limiter et de ne pas surcharger la sortie de l'AOP.

De plus, la carte comprend comme la précédente des contacts au pas de 2,54 mm répétés à l'aide de deux connecteurs.

Plan de montage du comparateur

Plan de câblage des composants du comparateur.

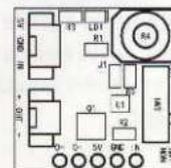


Photo de l'un de nos prototypes du comparateur.

Liste des composants du comparateur

- R1..... 100 Ω boîtier CMS 0805
- R2..... 1 kΩ boîtier CMS 0805
- R3..... 470 Ω boîtier CMS 0805
- R4..... trimmer 10 kΩ
- R5..... 100 kΩ boîtier CMS 0805)
- LD1.... LED rouge boîtier CMS 0805
- U1..... ADCMP371AKSZ
- Q1..... FDN340P

- Divers
- Barrette mâle 5 pôles coudée à 90°

- Double déviateur coulissant coudé à 90°
- Connecteur JST 1,25mm 2 pôles pour ci
- Connecteur JST 1,25mm 2 pôles fils volants
- Connecteur JST 1,25mm 3 pôles pour ci
- Connecteur JST 1,25mm 3 pôles fils volants

NB : les typons des circuits imprimés à l'échelle 1 sont disponibles en téléchargement dans le sommaire détaillé de la revue.

Réalisation pratique

Passons maintenant à la construction des deux cartes de prototypage, qui nécessitent chacune un circuit imprimé double face. Vous pouvez réaliser ces deux cartes par gravure chimique à l'aide des typons qui sont téléchargeables dans le sommaire détaillé de la revue. Notez que les deux cartes sont disponibles à la vente déjà montées et testées pour une vingtaine d'euros chacune.

Une fois les circuits imprimés fabriqués, vous devez percer les trous des trimmers et des déviateurs, qui sont des composants traditionnels classiques (traversant). Tous les autres composants sont des CMS, ils nécessitent un minimum d'expertise. Vous aurez besoin d'un fer à souder de 20 W avec une pointe très fine (0,1 ou 0,2 mm) ainsi que de la soudure de diamètre 0,5 mm. Une pince à épiler vous permettra de positionner les composants et une loupe vous permettra de

vérifier que chaque composant est correctement placé et soudé sans bavure.

Il est également conseillé d'utiliser un flux de soudure à faible densité, que vous pouvez étaler avec un pinceau sur les pastilles des composants (pads) avant de les placer et les souder.

Lors du placement des composants, faites attention à l'orientation des deux LED, du circuit intégré U1 et du MOSFET (pour la carte comparateur) qui se présente sous la forme d'un boîtier SOT-23. Il ne peut donc pas être disposé dans le mauvais sens à cause de la disposition asymétrique de ses broches.

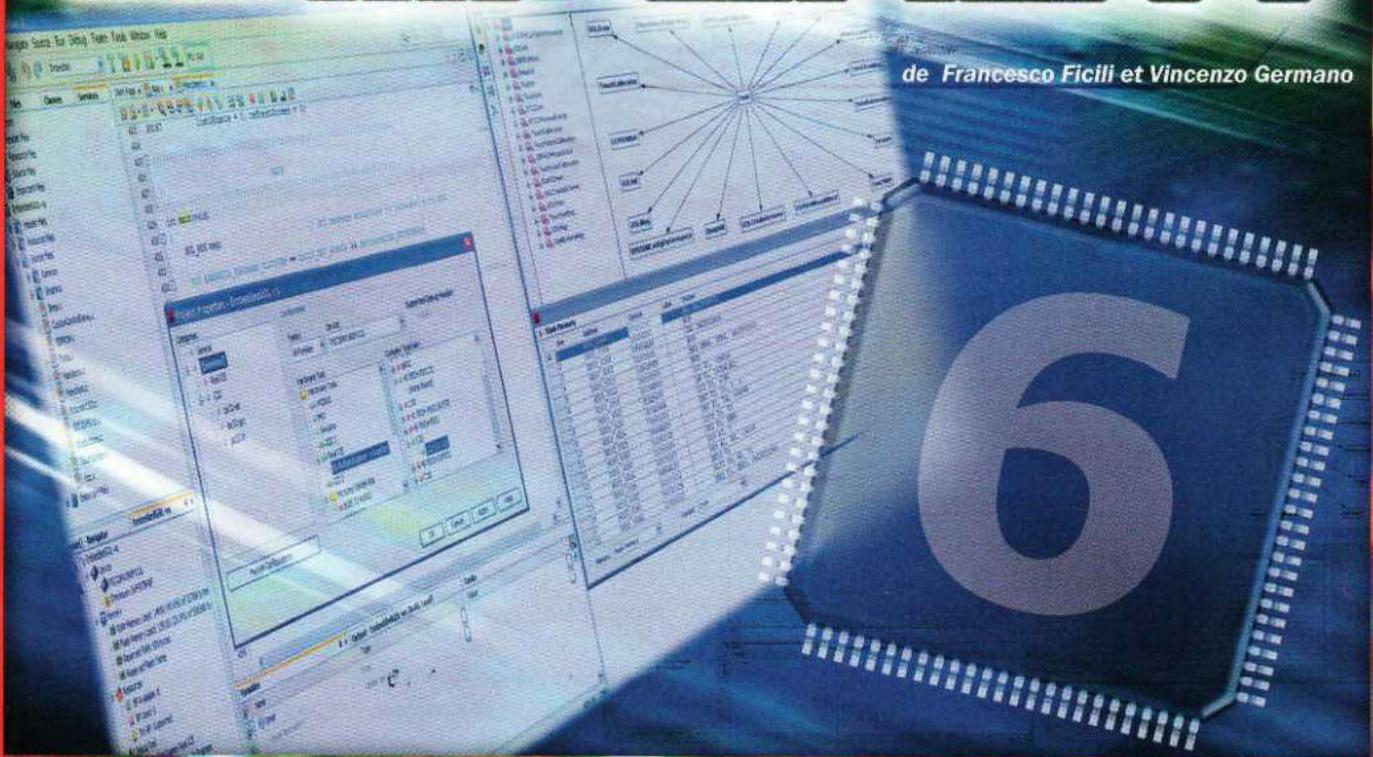
Les potentiomètres et les interrupteurs à glissière doivent être soudés en dernier.

Si vous avez l'intention d'utiliser ces cartes dans d'autres systèmes déjà existants, montez sur le côté de chaque circuit imprimé les barrettes au pas normalisé de 2,54 mm

COURS

MPLAB X

de Francesco Ficili et Vincenzo Germano



Cours MPLAB X

Sixième partie

Nous continuons notre cours sur MPLAB-X, le nouvel environnement de développement intégré produit et distribué par Microchip et qui remplace l'ancien MPLAB IDE. Dans cet article, nous allons créer des applications, telles qu'un serveur Web en utilisant la pile TCP/IP de Microchip. Sixième leçon.

Dans les dernières leçons de ce cours, nous avons concentré notre attention sur les piles (stack) mises à disposition par Microchip afin d'aider les développeurs qui souhaitent créer des applications avec un PIC32 et l'environnement de développement MPLAB-X. En particulier, nous avons analysé la pile USB, qui comprend une partie périphérique (device) et une partie hôte (host).

Dans cette leçon, nous allons analyser une autre pile puissante et complexe de Microchip.

Il s'agit de la **pile (stack) TCP/IP**. Comme exemple pratique, nous allons créer un simple serveur Web embarqué.

La stack TCP/IP de Microchip

Le package Microchip Libraries for Applications (MLA) contient la pile USB (déjà utilisée) et d'autres piles et bibliothèques qui ne seront pas abordées dans ce cours. Nous allons utiliser dans ce package la pile TCP/IP, qui

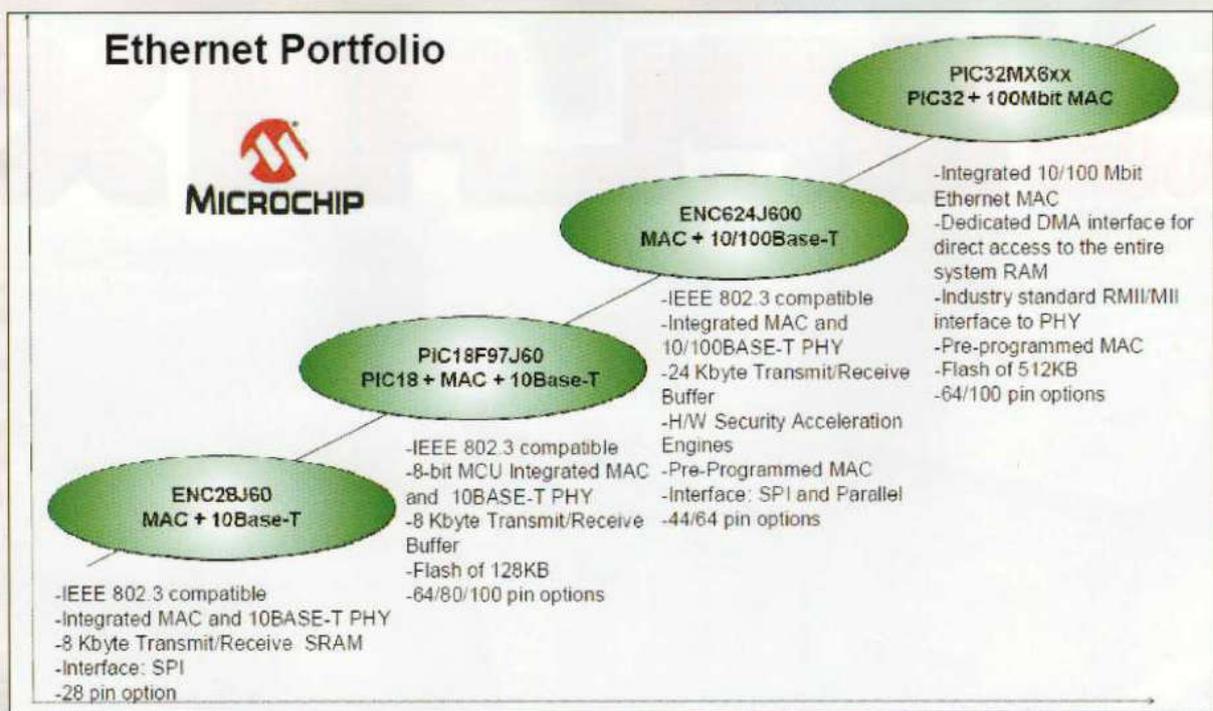


Figure 1 : solutions matérielles (hardware) prises en charge.

est complète et peut être intégrée à des projets utilisant des microcontrôleurs des familles PIC18, PIC24/dsPIC et PIC32 de Microchip.

La pile prend en charge diverses configurations matérielles, ce qui permet d'utiliser les contrôleurs internes présents dans les PIC18 et PIC32, ainsi que les contrôleurs externes ENC28J60 (10Base-T) et ENC624J600 (10/100Base-T) de Microchip. La figure 1 illustre les solutions matérielles (hardware) prises en charge par la pile, en mettant en évidence les caractéristiques de chacune.

Dans cette leçon, nous utiliserons la solution représentée en haut à droite dans le diagramme de la figure 1, c'est-à-dire celle qui prend en charge le contrôleur interne du PIC32 avec une adresse MAC intégrée.

Une adresse MAC (Media Access Control), parfois nommée adresse physique, est un identifiant physique stocké dans une carte réseau ou une interface réseau similaire.

Dans la figure 2, vous pouvez apercevoir le schéma synoptique d'un PIC32, où le module Ethernet interne est présent en haut à droite. Il est connecté à deux canaux DMA. Dans cette configuration, le niveau physique n'est pas présent à l'intérieur du microcontrôleur, il doit être ajouté à l'extérieur.

Par exemple, il est possible d'utiliser un circuit intégré DP83848C de National Semiconductor connecté à un PIC32MX795F512L selon la norme « RMII ». Cette configuration était utilisée dans la Demoboard qui accompagne ce cours.

Architecture de la stack TCP/IP

L'architecture de la pile TCP/IP de Microchip fonctionne selon une approche en couches, comme le suggère la théorie d'implémentation d'une stack de communication.

Le logiciel est divisé en couches qui se superposent et chaque couche utilise des services provenant des couches sous-jacentes et fournit des services aux couches supérieures.

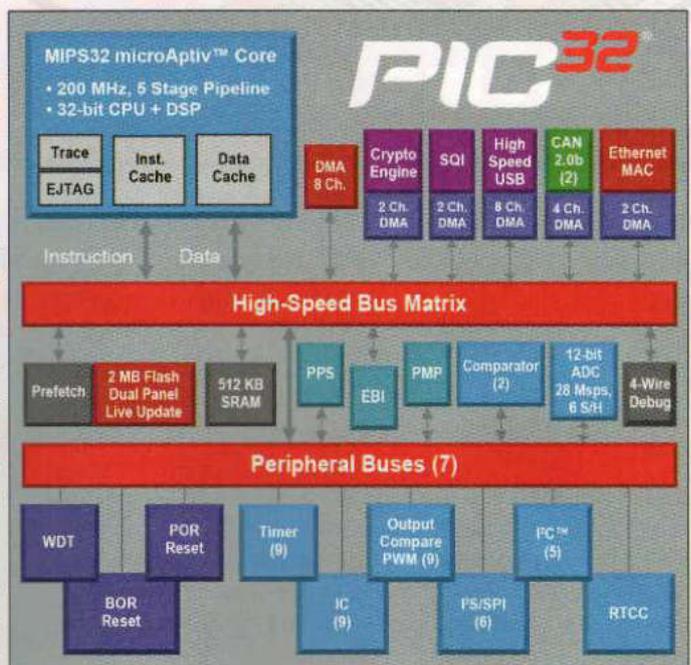


Figure 2 : schéma synoptique de l'architecture matérielle (hardware) interne d'un PIC32.

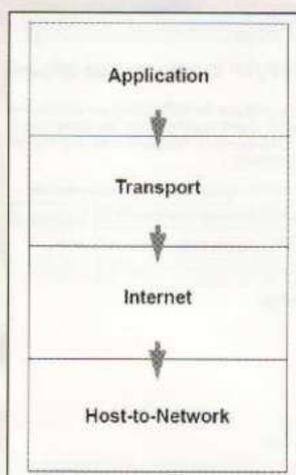


Figure 3 : modèle simplifié de la pile TCP/IP.

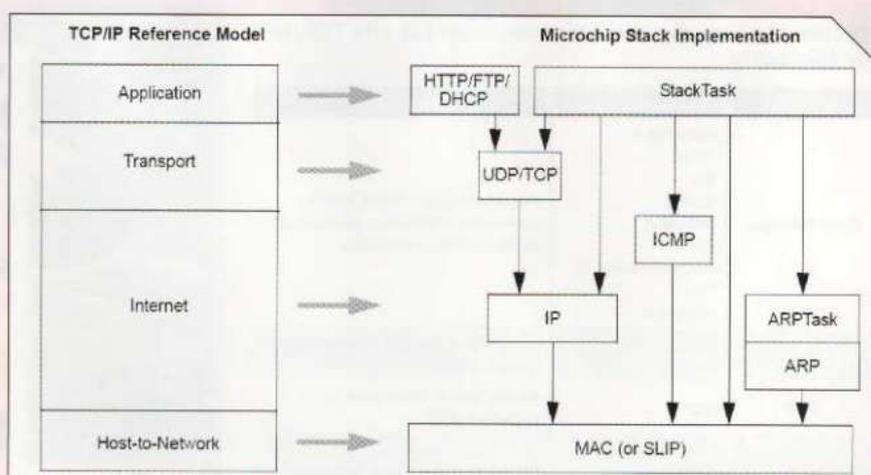


Figure 4 : comparaison entre l'implémentation de Microchip et le modèle de référence pour le protocole TCP/IP.

Un modèle simplifié de la pile TCP/IP est représenté en figure 3. Bien évidemment, l'implémentation de la pile n'est pas conçue comme pour un ordinateur de bureau, avec des ressources pratiquement infinies et une capacité informatique élevée. L'implémentation de Microchip contient un sous-ensemble des différents niveaux pouvant composer la pile.

La figure 4 illustre la mise en œuvre de Microchip par rapport au modèle de référence du protocole TCP/IP. L'implémentation de la pile **suit le paradigme du multitâche coopératif** et peut donc être intégrée à tout autre système exploitant la même approche, comme dans l'ordonnanceur que nous avons développé dans ce cours et que nous avons déjà utilisé pour l'intégration de la pile USB.

Comme vous pouvez le voir sur le diagramme, l'implémentation de la pile est modulaire et chaque module implémente un niveau ou une partie d'un niveau.

Le tableau 1 présente les modules principaux constituant la pile avec la description relative et la liste des fichiers nécessaires à l'inclusion du module dans le projet.

Si vous souhaitez plus de détails sur la pile TCP/IP, reportez-vous à la note d'application AN833 nommée « The Microchip TCP/IP Stack ».

Configuration de la pile

Comme nous l'avons mentionné précédemment, la pile TCP/IP de Microchip est composée de plusieurs modules.

En outre, chaque module peut être configuré de manière optimale en fonction d'une l'application spécifique et cette configuration réside dans un fichier particulier, appelé « **TCPIPConfig.h** ». Celui-ci doit être inclus dans le projet.

La configuration du fichier « **TCPIPConfig.h** » peut être assez complexe, c'est la raison pour laquelle Microchip a créé une application, dénommée « **TCPIPConfig.exe** », qui permet de configurer facilement ce fichier via une interface graphique, comme nous l'avons fait pour la pile USB.

Examinons donc comment cela fonctionne, en l'utilisant pour configurer le fichier « **TCPIPConfig.h** » de l'exemple pratique proposé dans cette leçon.

Commençons d'abord par lancer l'application « **TCPIPConfig.exe** », qui présente la fenêtre de la figure 5, dans laquelle vous pouvez modifier le chemin du fichier « **TCPIPConfig.h** ». Si vous utilisez également le Wi-Fi, vous devez également entrer le chemin du fichier « **WF_Config.h** ».

Ce dernier contient les options de configuration du module Wi-Fi. À partir de la fenêtre suivante, illustrée en figure 6, vous pouvez sélectionner les modules de base à inclure dans l'application.

Dans notre cas, nous voulons créer un serveur Web, nous cochons simplement la case « **Web Sever** » sous la ligne « **Application Protocols** » et nous **désélectionnons les autres cases** à cocher. La fenêtre suivante permet de sélectionner des exemples de modules. Dans notre cas, cela ne nous intéresse pas, alors nous désélectionnons tout et continuons.

Ensuite, la fenêtre suivante est illustrée en figure 7, elle permet de sélectionner des modules de support. Sélectionnons les modules suivants, puis cliquons sur le bouton « **Next** » :

- DHCP Client et Server ;
- ICMP Server ;
- DNS Client ;
- NetBIOS Name Service ;
- AutoIP Client.

Tableau 1 : liste des modules constituant la pile TCP/IP de Microchip.

Module	Fichiers requis	Description
Stack Manager	StackTsk.c TCP.c IP.c ICMP.c ARPTsk.c ARP.c MAC.c or SLIP.c Tick.c Helpers.c	Stack Manager ("StackTask"), coordonne l'exécution de tous les autres modules de la pile.
MAC	MAC.c Delay.c	Couche d'accès aux médias (Media Access Layer).
SLIP	SLIP.c	Media Access Layer pour le protocole SLIP.
ARP	ARP.c ARPTsk.c MAC.c or SLIP.c Helpers.c	Protocole de résolution d'adresse (Address Resolution Protocol).
IP	IP.c MAC.c or SLIP.c Helpers.c	Protocole Internet (Internet Protocol).
ICMP	ICMP.c StackTsk.c IP.c MAC.c or SLIP.c Helpers.c	Protocole de contrôle des message Internet (Internet Control Message Protocol).
TCP	StackTsk.c TCP.c IP.c MAC.c or SLIP.c Helpers.c Tick.c	Protocole de contrôle de la transmission (Transmission Control Protocol).
UDP	StackTsk.c UDPC.c IP.c MAC.c or SLIP.c Helpers.c	Protocole de datagramme (paquet de données) utilisateur (User Datagram Protocol).
HTTP Server	HTTP.c TCP.c IP.c MAC.c or SLIP.c Helpers.c Tick.c MPFS.c	Serveur du protocole de transfert HyperText (HyperText Transfer Protocol Server).
FPT Server	FTP.c TCP.c IP.c MAC.c or SLIP.c	Serveur du protocole de transfert des fichiers (File Transfer Protocol Server).
DHCP	DHCP.c UDP.c IP.c MAC.c Helpers.c Tick.c	Protocole de configuration dynamique de l'hôte (Dynamic Host Configuration Protocol).

À partir de la fenêtre suivante, nous pouvons **configurer différentes adresses**. Laissons l'adresse MAC et l'adresse IP inchangées, ainsi que celles du masque de sous-réseau, de la passerelle et des serveurs DNS.

Nous **modifions le nom de l'hôte** selon notre convenance, vous pouvez en choisir un autre. Dans notre exemple, nous avons utilisé le nom « EINBOARD ».

Considérez qu'après avoir inclus le module DHCP, les valeurs insérées pour l'adresse IP (IP Address), le masque de sous-réseau (Subnet Mask), l'adresse de la passerelle (Gateway Address) et l'adresse du serveur DNS (DNS Server) ne sont



Figure 5 : écran initial de l'outil « TCPIPConfig ».

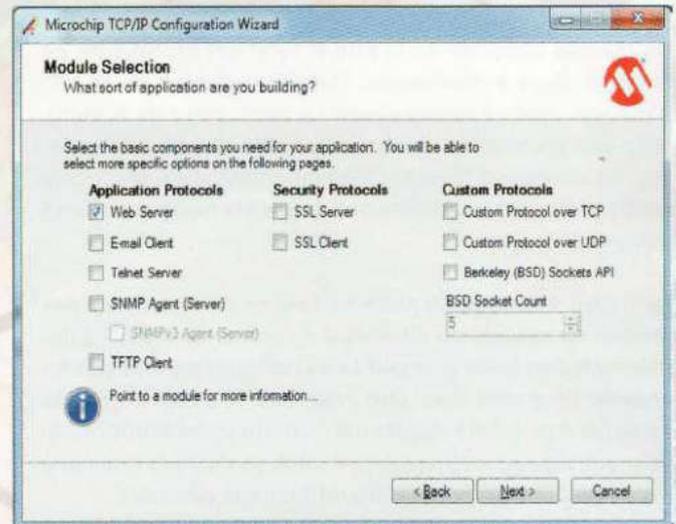


Figure 6 : sélection des modules de base à inclure dans l'application.

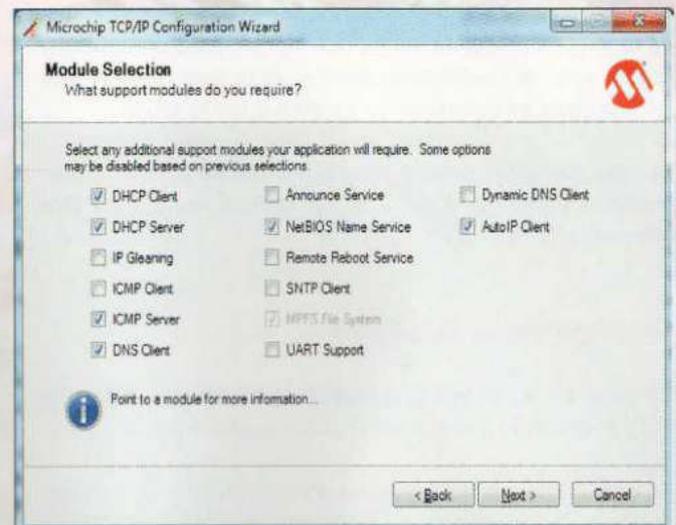


Figure 7 : sélection des modules de support à inclure dans l'application.

LE LANGAGE HTML

Le langage HTML, qui signifie « HyperText Markup Language », est le langage de base utilisé pour la création des pages web. C'est un langage Open Source, dont la syntaxe a été définie par le World Wide Web Consortium (W3C), et qui est dérivé d'un autre langage à vocation plus générale, le SGML. Il est basé sur l'utilisation de balises permettant de mettre en forme le contenu des pages web de la manière souhaitée, d'inclure des ressources multimédias (images, vidéo, etc.) et des formulaires de saisie.

Le langage HTML ne peut pas être considéré comme un langage de programmation car certaines variables et boucles de contrôle ne peuvent pas être définies. Il s'agit plutôt d'un langage de formatage textuel, il est généralement appelé langage de balisage (markup). Toutefois, pour éviter cette limitation, il est possible d'insérer dans la page web du contenu externe tel que des images, des vidéos ou des scripts dans d'autres langages de programmation tels que le langage JavaScript. Les balises sont placées entre des chevrons et chaque fois qu'une balise est ouverte, elle doit également être fermée en insérant une barre oblique avant le nom de la balise.

Les balises HTML respectent une syntaxe simple et stricte :

- Un chevron ouvrant (<)
- Le nom de la balise
- Des attributs (optionnels). Un espace, suivi du nom de l'attribut, d'un signe égal (=) et d'une valeur entre doubles quotes (« »).
- Un chevron fermant (>)

Voici quelques exemples :

```
<texte>
<meta charset="utf-8">

```

Les balises fonctionnent par paires, la première balise est la balise ouvrante et la seconde est la balise fermante. Une balise fermante doit avoir le même nom que la balise ouvrante correspondante. De plus, une balise fermante doit contenir une barre oblique entre le chevron initial et le nom de la balise. Ainsi, si <p> est une balise ouvrante, </p> sera la balise fermante correspondante. Par exemple :

```
<p>
début du paragraphe
  <strong>
    ici un texte important en gras
  </strong>
ici la fin du paragraphe.
</p>
```

La structure de base d'un fichier HTML est illustrée en figure A. Elle commence par l'en-tête du fichier qui contient des informations sur le document et le corps du document qui contient tout le contenu formaté de la page.

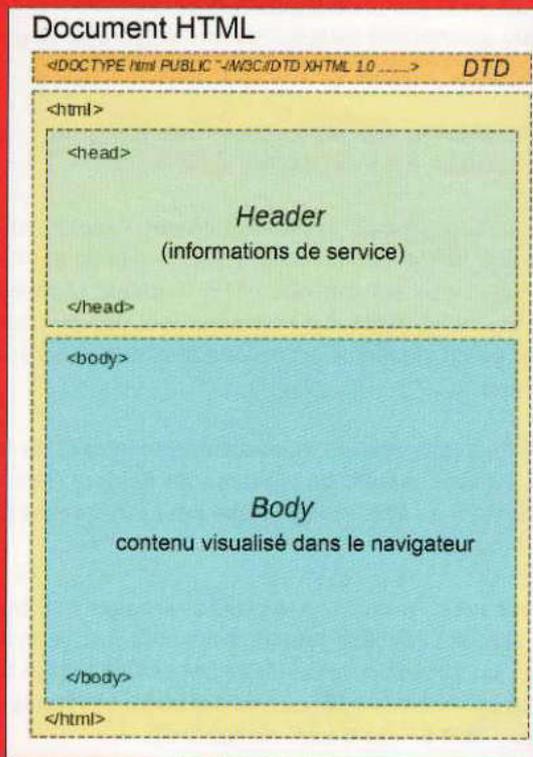


Figure A : structure de base d'un fichier HTML.

que des valeurs par défaut, car le module DHCP utilisera des valeurs dynamiques lors de la connexion de la carte.

À ce stade, nous devons configurer le serveur web. Pour cela, à partir de la fenêtre de la figure 9, nous insérons le nom de la page par défaut, celle qui contiendra la page d'accueil de notre serveur Web. Ensuite, nous entrons le nombre maximal de sockets pouvant être maintenus simultanément (dans notre exemple 2).

Un **socket est un modèle permettant la communication inter processus afin de permettre à divers processus de**

communiquer aussi bien sur une même machine qu'à travers un réseau TCP/IP. Dans les sections « Additional Features » et « Demo Applications », désélectionnez tout.

Dans la fenêtre suivante (voir la figure 10), il vous suffit de **définir l'emplacement de stockage de la ou des pages web du serveur Web**. Il y a plusieurs options, nous choisissons la plus simple, à savoir « **Internal Program Memory** ».

Avec cette option, la page web est directement chargée dans la mémoire Flash du PIC32 que nous utilisons, et nous verrons plus tard comment cette opération est effectuée.

Il n'est pas toujours possible d'utiliser cette option, étant donné que si les pages web prennent beaucoup d'espace, le mémoire Flash du microcontrôleur risque de ne pas pouvoir les contenir.

Dans ce cas, il existe d'autres possibilités, telles que l'utilisation de mémoires EEPROM ou de mémoires Flash externes, voire d'un support FAT (tel qu'une carte SD ou une clé USB).

La figure 11 montre la fenêtre de configuration des sockets TCP. Vous pouvez conserver les valeurs par défaut, mais si vous souhaitez effectuer des modifications, consultez la section relative au module TCP de la note d'application AN833. Une fois la configuration du socket terminée, la fenêtre permettant de terminer la configuration s'affiche.

Le système de fichiers Microchip ou Microchip File System (MPFS)

Comme nous l'avons vu précédemment, l'implémentation de la pile TCP/IP de Microchip implique également l'implémentation d'un serveur Web HTTP. Toutefois, si un serveur Web est implémenté, il y aura sûrement une ou plusieurs pages web à afficher, elles doivent être stockées dans une mémoire.

Dans l'implémentation du serveur Web intégré de Microchip, **un type simple de système de fichiers (Microchip File System ou MPFS) est utilisé pour stocker les pages web** à publier.

Comme nous l'avons déjà expliqué, les pages web peuvent être stockées dans la mémoire flash du microcontrôleur ou sur un support externe (EEPROM externe, Flash SPI ou même une mémoire SD ou une clé USB). Le format d'une image « MPFS » est illustré en figure 12.

Après un bloc initial réservé, dont la taille est fixée par la macro « MPFS_RESERVE_BLOCK », l'image contient une série d'entrées FAT (File Allocation Table) suivies par autant de blocs de données. Chaque entrée FAT a un format représenté en figure 13. Comme vous pouvez facilement le constater, l'ensemble des entrées FAT constitue une table qui associe une adresse au nom d'un bloc de données. L'adresse peut être au format 16 ou 24 bits selon le support de stockage utilisé.

Par exemple, si les pages sont stockées dans la mémoire Flash et que le modèle de mémoire « SMALL » est utilisé pour compiler un projet, les adresses sont au format 16 bits. En ce qui concerne le nom du fichier, le type « short » (court) au format « 8 + 3 » est utilisé.

L'adresse de chaque entrée FAT pointe vers le bloc correspondant contenant les données binaires de la page, dont le format est indiqué en figure 14.

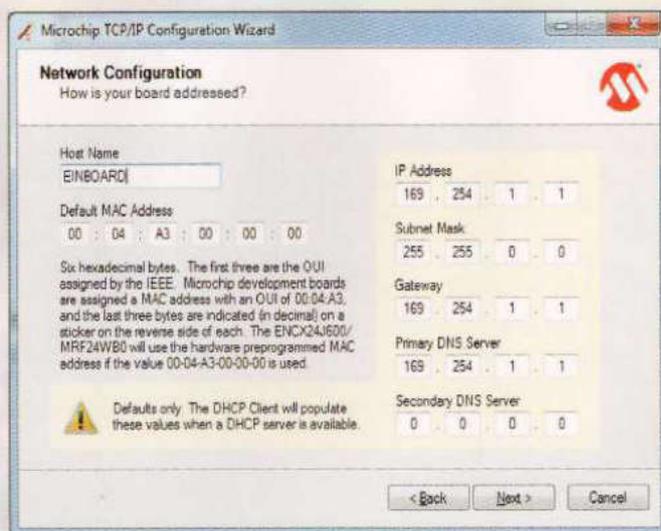


Figure 8 : configuration des adresses.

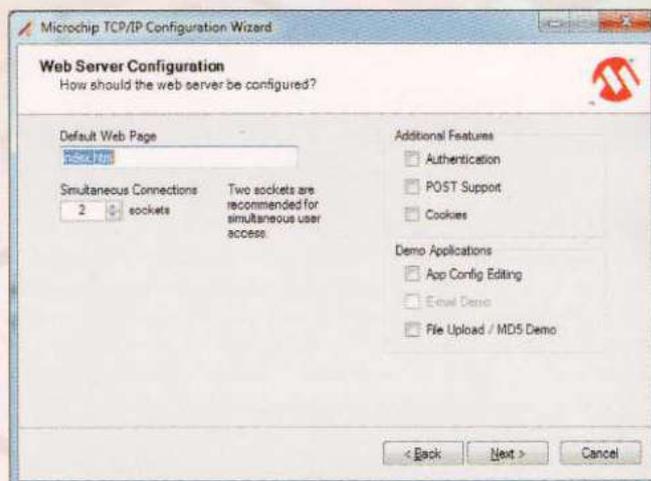


Figure 9 : configuration du serveur Web.

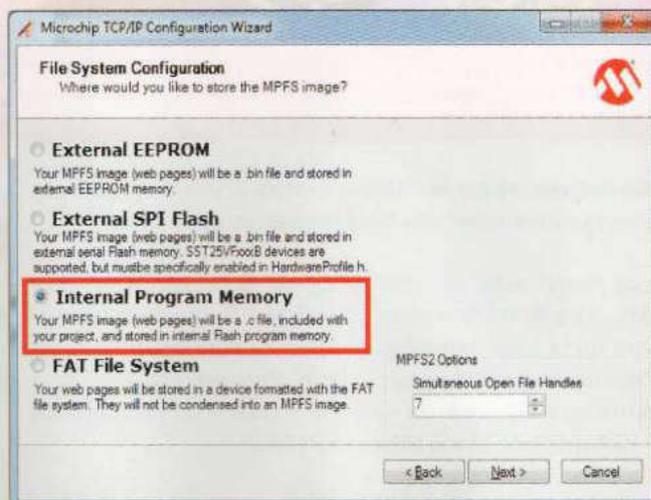


Figure 10 : options de stockage des pages web.

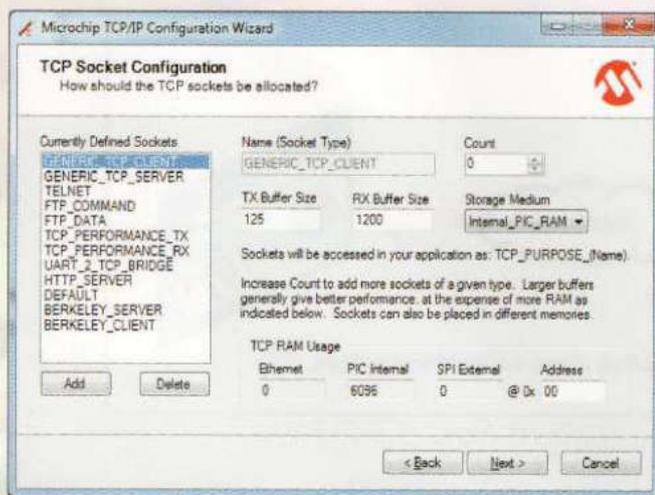


Figure 11 : configuration des sockets TCP.

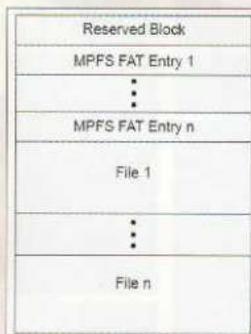


Figure 12 : format d'une image MPFS.

Figure 13 : format d'entrée de la FAT.

MPFS FAT ENTRY FORMAT

Flag	Address	File Name
(8 bits)	(16 or 24 bits)	(8-byte + 3-byte format)

MPFS DATA BLOCK FORMAT

Data (variable length)	EOF (8 bits)	FFFFh or FFFFFFh (16 or 24 bits)
---------------------------	-----------------	--

Figure 14 : format d'un bloc de données.

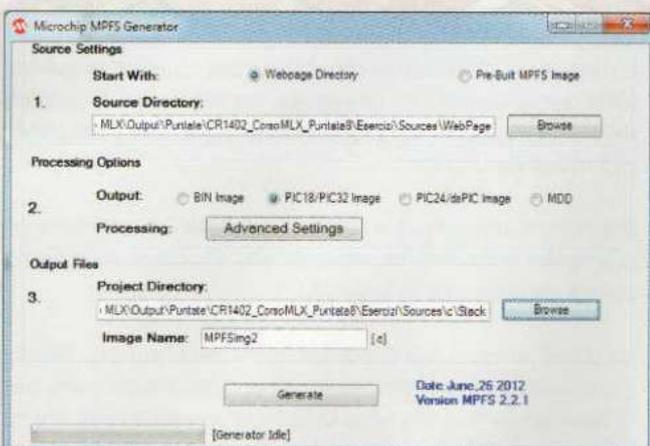


Figure 15 : fenêtre principale de l'application « MPFS2.jar ».

La génération d'une image de type MPFS est indispensable pour la réalisation d'une application utilisant le serveur Web (donc semblable à celle de notre exemple pratique).

Pour simplifier le processus, Microchip a développé une application appelée « MPF2 » qui, à partir d'une page web normale créée en HTML, génère l'image MPFS correspondante à utiliser dans notre projet embarqué.

L'application, dont la fenêtre est représentée en figure 15, permet de générer des images binaires (à stocker sur un support externe), le fichier « .c » (dans le cas du stockage dans la Flash du microcontrôleur), ainsi que le code pour les systèmes de fichiers de type FAT (dans le cas d'une carte SD ou d'une clé USB). Dans notre application, nous utiliserons cet outil pour la conversion de notre page web.

Variables dynamiques

Une application qui implémente un serveur web dans notre PIC32 serait plutôt inutile si elle n'était pas capable d'interagir avec le reste du système pour échanger des données. Microchip a réfléchi à la question et, pour résoudre le problème, a introduit le concept de **variable dynamique**.

Ces variables permettent une interaction entre le module « Web Server » et le reste de l'application qui s'exécute dans le microcontrôleur, ce qui permet de partager les variables entre le module lui-même et le reste du code.

Un exemple de l'utilité de cette solution est schématisé en figure 16. Dans ce schéma, une application a été créée. Elle implémente un serveur Web et est également connectée à un capteur particulier.

En utilisant le concept de variable dynamique, les lectures du capteur sont combinées dans le serveur Web afin d'être affichées sur une page web, puis transmises via Internet, à partir duquel elle est disponible pour chaque client qui en fait la demande.

Les variables dynamiques peuvent être utilisées, en combinaison avec le serveur Web, de différentes manières. Nous allons examiner deux d'entre elles qui correspondent aux deux cas utilisés dans notre exemple pratique, à savoir l'**affichage des données** sur la page web et l'utilisation de la méthode « GET ».

Dans le premier cas, nous avons la possibilité d'afficher sur notre page une variable système (c'est-à-dire une variable générique présente dans notre projet embarqué).

Pour effectuer cela, deux étapes sont nécessaires : du côté de la page web, la **variable à afficher doit être placée entre deux symboles « tilde » (~)**, tandis que du côté embarqué, un rappel approprié doit être implémenté.

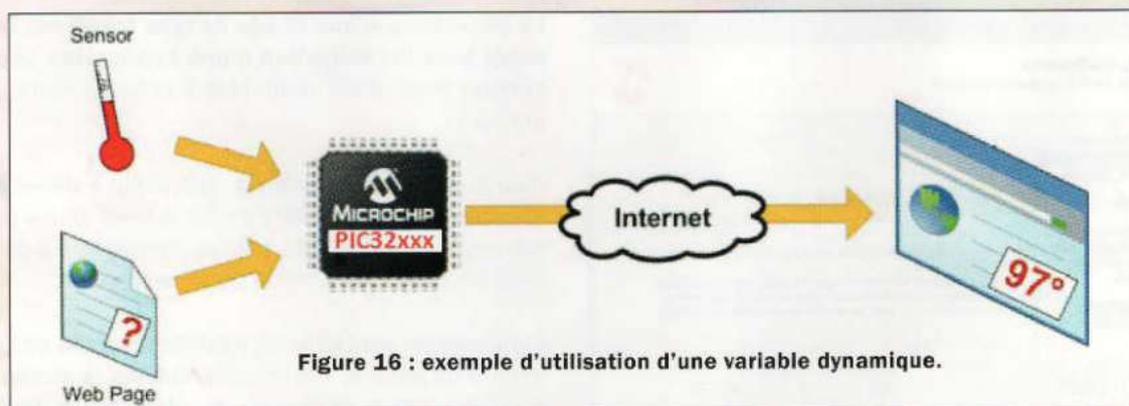


Figure 16 : exemple d'utilisation d'une variable dynamique.

Ce rappel est généré par l'outil « MPFS2 » lorsque la page web est traduite en une image binaire.

Par exemple, si nous voulons afficher une variable générique « var » présente dans notre projet, nous devons insérer la variable entre les symboles « tilde » dans la page web, comme ci-après : `~var~`

Ensuite, du côté embarqué, nous devons implémenter le rappel suivant : `HTTPPrint_var()`

La pile effectue d'elle-même le reste de la procédure, nous aborderons plus clairement cette procédure dans un exemple concret.

Dans le second cas, c'est-à-dire l'exécution de la méthode « GET », cela entraîne du côté web l'accumulation de toutes les données transmises avec cette méthode dans la mémoire tampon « HTTPcurr.data » et l'appel du rappel (callback) « `HTTPExecuteGet()` ».

En combinant ces deux événements, il est possible de gérer l'occurrence d'un GET de manière appropriée du côté embarqué. Cet aspect deviendra également plus clair avec la description du projet pratique.

Exemple pratique : un serveur Web embarqué

Passons maintenant à la partie pratique, dans laquelle nous vous proposons de créer un simple serveur Web embarqué, qui vous permettra de contrôler à distance une LED. L'application elle-même est très simple, mais elle nous permet de valider de manière pratique ce qui a été présenté dans cette leçon.

Commençons par la description de l'architecture logicielle du serveur, dont le synoptique simplifié est visible en figure 17. Comme vous pouvez le constater, dans ce cas, nous n'utilisons que deux tâches pour implémenter notre application.

Une tâche dédiée au protocole TCP/IP et une autre réservée à la gestion de la LED (Led task) qui nous permet d'obtenir un

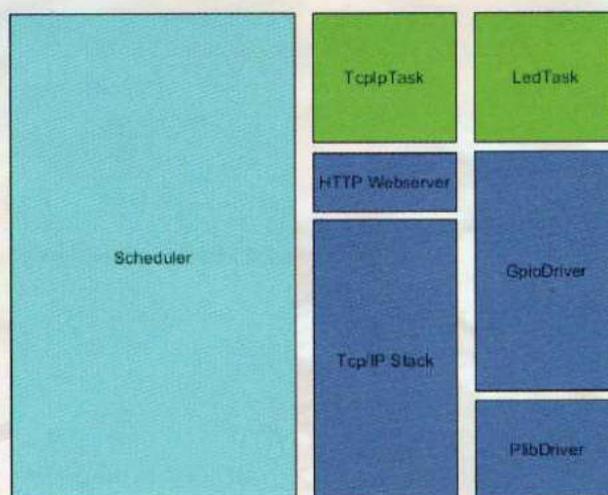


Figure 17 : architecture logicielle (software) du serveur.

retour de notre application (dans notre cas, il s'agit simplement d'un clignotement de la LED à une fréquence de 1 Hz).

Il est évident que « `TcpIpTask` » repose sur la pile TCP/IP et sur l'implémentation du serveur Web HTTP de Microchip. En figure 18, vous pouvez visualiser le « Project manager » de MPLAB-X qui gère le projet pratique (nous l'avons appelé `MLX_Ex11`).

Le tableau 2 résume la périodicité des deux tâches en cours d'exécution. L'infrastructure utilisée est toujours la même, ainsi que les définitions des structures des données prenant en charge l'exécution du planificateur (scheduler), elles sont reportées au Listing 1.

Examinons maintenant la tâche « `TcpIpTask` », qui correspond à la tâche principale de cet exemple. Étudions sa mise en œuvre en analysant le Listing 2.

La tâche se compose d'une section d'initialisation, utilisée pour initialiser correctement les structures des données des différentes tâches des piles utilisées dans cette application, et d'une section « running » (exécution) permettant aux machines à état de s'exécuter sur la pile correspondante.

Listing 1 : structure des données

```

/* Nombre maximal de tâches */
#define ACTIVE_TASK_NUMBER ((UINT16)(2))

/* Tableau des tâches à appeler */
void (*TaskArray[ACTIVE_TASK_NUMBER]) (void) =
{
    LedTask,
    TcplpTask,
};

/* Tableau des délais d'expiration des tâches */
UINT16 TaskPeriodTimeoutMs[ACTIVE_TASK_NUMBER] =
{
    LED_TASK_PERIOD_MS,
    TCPIP_TASK_PERIOD_MS,
};

```

Tâches	Périodicité en ms
LedTask	100
TcplpTask	10

Tableau 2 : périodicité des tâches.

Analysons la structure de la fonction d'initialisation, dont le code est reporté au Listing 3. Dans cette fonction, sont initialisés en séquence :

- le composant « **Tick** », qui est utilisé pour générer tout le timing nécessaire pour la pile entière ;
- le composant « **MPFS** », utilisé pour l'implémentation des pages web du serveur ;
- la structure des données « **AppConfig** ».

Les initialisations plus générales de la pile sont également effectuées, en particulier, l'initialisation de la structure des données « **AppConfig** » permet de charger la plupart des données via l'outil de configuration, telles que les valeurs d'adresse, le nom du NetBios, etc.

Examinons maintenant la partie opérationnelle de la tâche, dont le code est reporté au Listing 4. Deux tâches sont appelées dans ce cas, la première dénommée « **StackTasks** » exécute toutes les machines à états principales de la pile. Il s'agit d'une série de services généraux et de temporisations nécessaires pour toute tâche de niveau d'application de la pile.

Par contre, la tâche « **StackApplications** » exécute une série de tâches de niveau de l'application et dépend fortement de la configuration de la pile contenue dans « **TCPIPConfig.h** ». Dans ce cas spécifique, par exemple, les tâches liées au serveur web HTTP, au serveur DHCP et au serveur de noms NetBIOS sont exécutées.

Maintenant, notre pile TCP/IP est correctement initialisée et fonctionne, mais si nous démarrons l'application, notre

Listing 2 : implémentation de la tâche « TcplpTask »

```

/*****
* Fonction :      TcplpTask
* Input :        None
* Output :       None
* Auteur :       F.Ficili
* Description :   Manage TCPIP Task.
* Date : 30/11/18
*****/

void TcplpTask (void)
{
    switch (SystemState)
    {
        /* Phase d'initialisation du système */
        case InitializationState:

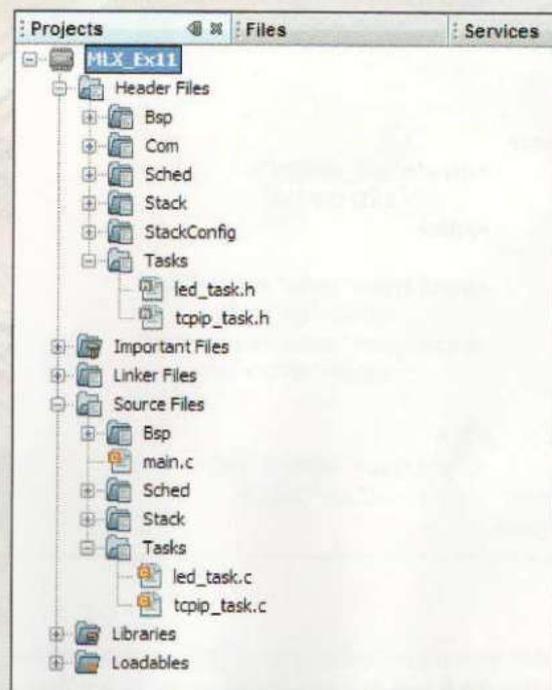
            /* Initialise la pile tcpip */
            TcplpInitStack();
            break;

            /* Phase de fonctionnement normal du système */
        case RunningState:

            /* Appelle la tâche tcpip */
            TcplpStackTasks();
            break;

        /* Défaut */
        default:
            break;
    }
}

```

**Figure 18 : notre projet ouvert dans le « Project manager » de MPLAB-X.**

Listing 3 : implémentation de la fonction « TcplpInitStack ».

```

/*****
* Fonction : TcplpInitStack
* Input : None
* Output : None
* Auteur : F.Ficili
* Description : Initialise la pile TCPIP
* Date : 30/11/18
*****/

void TcplpInitStack (void)
{
    /* Initialise le compteur tick */
    TickInit();
    /* Initialise MPFS */
    MPFSInit();
    /* Initialise AppcConfig */
    InitAppConfig();
    /* Initialise la pile */
    StackInit();
}
    
```

Listing 4 : implémentation de la fonction « TcplpStackTasks ».

```

/*****
* Fonction : TcplpStackTasks
* Input : None
* Output : None
* Auteur : F.Ficili
* Description : Gestion des tâches nécessaires à TCPIP.
* Date : 30/11/18
*****/

void TcplpStackTasks (void)
{
    /* Exécute des tâches de la pile */
    StackTask();

    /* Cette tâche appelle chacune des tâches principales
    de l'application*/
    StackApplications();
}
    
```

Listing 5 : code HTML de la page web à télécharger sur le serveur Web.

```

<h1>Welcome !</h1>
<h1> Ceci est une page de test !</h1>
<h3> Cette page de test montre la capacité de la pile TCP/
IP de Microchip </h3>

<div id="location">
    Machine NetBios Hostname: ~hname~
</div>

<br>

<form>
    <div id="led_control">
        LED Control:
    </div>

    <input type="radio" name="led"
        value="on"/> On
    <input type="radio" name="led"
        value="off"/> Off

    <br>
    <input type="submit" class="btn"
        value="Set"/>
</form>
    
```

serveur Web ne permettra pas d'afficher quoi que ce soit, car nous n'y avons chargé aucune page. Pour afficher une page web, nous devons créer une image MPFS adaptée au serveur Web HTTP, comme décrit précédemment.

Listing 6 : implémentation du callback (rappel) « HTTPPrint_hname ».

```

/*****
* Fonction : HTTPPrint_hname
* Input : None
* Output : None
* Auteur : F.Ficili
* Description : Affiche le nom du rappel de l'hôte.
* Date: 30/11/18
*****/

void HTTPPrint_hname(void)
{
    TCPPutString(skTHTTP,AppConfig.NetBIOSName);
}
    
```

La page que nous voulons afficher est une page HTML normale, dont le code est présenté au Listing 5.

Analysons brièvement le code HTML de la page. La section initiale comprend uniquement du texte placé dans des en-têtes différents (h1, h2 et h3) et, par conséquent, seule une partie du code HTML affiche un texte statique (en particulier le mot « Welcome ! » et quelques lignes de description).

La section suivante utilise le concept de variable dynamique pour afficher le nom du NetBios associé à la carte (comme nous l'avons vu ci-dessus, la variable `~hname~` invoque le rappel « `HTTPPrint_hname` », grâce auquel, en utilisant

Listing 7 : implémentation du callback (rappel) « HTTPExecuteGet ».

```

/*****
Function :      HTTPExecuteGet
* Input :      None
* Output :     None
* Auteur :     F.Ficili
* Description : Exécute le callback de GET.
* Date :      30/11/18
*****/

HTTP_IO_RESULT HTTPExecuteGet(void)
{
    BYTE *DataPtr;

    /* Obtient les données */
    DataPtr = HTTPGetROMArg(curHTTP.data,(ROM BYTE
*)"led");

    /* Vérifie si les données sont présentes à l'intérieur
du pointeur */
    if(DataPtr)
    {
        /* Si « on », les données sont transmises*/
        if(strncmp(DataPtr,(ROM char*)"on")
== 0)
        {
            /* Envoie l'évènement « on » */
            GenerateEvt(&Led2On);
        }

        else
        {
            /* Envoie l'évènement « off » */
            GenerateEvt(&Led2Off);
        }
    }
}

```

une fonction simple, il est possible d'afficher à la place de la variable le nom du NetBios.

Enfin, la section comprise entre les balises **<form>** et **</form>** utilise la méthode « GET » pour générer une entrée pour le serveur Web afin de contrôler la LED sur la carte (pour interagir avec les données transmises par la méthode « GET », il est nécessaire d'utiliser le callback (rappel) « HTTPExecuteGet »).

D'un point de vue graphique, cette section génère deux boutons radio marqués respectivement par « On » et « Off » et un bouton poussoir (Set).

Avant de procéder à la création des rappels (callback), voyons comment générer le code exécutable de la page web, afin de l'intégrer au projet MPLAB-X.

Comme nous l'avons abordé précédemment, le code de la page est généré à l'aide de l'outil MPFS2.

Nous pouvons soit utiliser l'interface graphique de l'outil, soit utiliser un script batch, comme celui présenté ci-après :

```

java -jar "..\Utilities\MPFS2.jar" /mpfs2 /C18_C32 "..\
WebPage" "." "MPFSImg2.c"

```

Ce script appelle l'outil MPFS2, en passant les paramètres corrects pour générer le code C correspondant à la page. Le résultat de l'exécution du script comprend les deux fichiers suivants :

- **MPFSImg2.c** : constitue le code C qui implémente la page web à afficher ;
- **HTTPPrint.h** : qui est le fichier d'en-tête contenant les prototypes de rappel (callback) à inclure dans le projet.

Les deux fichiers doivent être inclus dans le projet MPLAB-X. La page web, configurée par l'outil « TCPConfig », sera compilée afin de résider directement dans la mémoire Flash du PIC32.

Nous avons presque terminé, il suffit d'implémenter les callbacks liés à la variable dynamique « hname » et à la méthode « GET » utilisée dans la page web et de compiler notre projet.

Commençons par le callback de l'affichage du nom du NetBios.

Comme nous l'avons vu précédemment, la présence de la variable dynamique ~ hname ~ dans la page web entraînera, lors du chargement de la page, l'appel du callback « HTTPPrint_hname(); ». Tout ce que nous avons à faire est d'implémenter ce rappel afin qu'il affiche le nom du NetBios de la carte, une fois invoqué.

Le **nom du NetBios**, c'est-à-dire le **nom de l'hôte** que nous avons donné lors de la configuration (dans notre exemple c'est « EINBOARD »), est facilement récupérable à partir de la structure des données « AppConfig » et le code qui implémente le callback est reporté au Listing 6.

Dans ce cas, la pile nous est très utile car elle nous fournit la fonction « TCPPutString ». Celle-ci écrit dans le socket transmis en tant que premier paramètre les données présentes dans le tableau transmis en tant que second paramètre. Il suffit de passer dans la fonction le socket « http » (sktHTTP) et les données à afficher (AppConfig.NetBIOSName). Cela est relativement simple, comme vous pouvez le constater.

En ce qui concerne le contrôle de la LED, pour intercepter les données demandées au serveur via la méthode « GET », la pile nous fournit le callback « HTTPExecuteGet ». Ce dernier est appelé pour exécuter cette méthode. En outre, les données transmises sont automatiquement chargées dans le buffer « currHTTP.data ».

Étant donné que le callback est unique pour toutes les instances de la méthode, il sera nécessaire d'examiner les données via une structure de sélection appropriée pour exécuter des actions spécifiques.

Les implémentations possibles, dans ce cas spécifique, sont multiples. Nous proposons celui reporté au Listing 7.

Tout d'abord, via la fonction « HTTPGetROMArg », les données associées à l'argument passé en tant que paramètre sont récupérées (dans ce cas c'est la chaîne « led »).

Le résultat est transmis à un pointeur approprié.

Ensuite, nous devons vérifier si le pointeur contient des données valides, puis si cela est vérifié, l'opération précédente fournit une correspondance (match). Dans le cas affirmatif, en comparant la chaîne des données à la chaîne « on », nous déterminons si la LED doit s'allumer ou s'éteindre.

Notez que l'activation (ou la désactivation) réelle de la LED n'est pas effectuée directement dans le callback, mais par la fonction « LedTask », en envoyant un événement approprié (GenerateEvt(&Led2On) et GenerateEvt(&Led2Off)).

L'extrait de code de la fonction « LedTask », qui intercepte l'événement, est reporté au Listing 8.

L'opération peut sembler un peu lourde, mais délègue correctement les tâches à une tâche spécifique, ce qui nous permet de maintenir une modularité correcte de l'implémentation.

Si le clic sur le bouton « Set » doit effectuer une action plus complexe qu'une simple activation/désactivation, comme par exemple un clignotement, la tâche « LedTask » conviendra pour effectuer cette opération, mais ne pourrait pas être correctement implémentée par « HTTPExecuteGet ». Cela nous obligerait à modifier le code ou à ajouter une fonction spécifique.

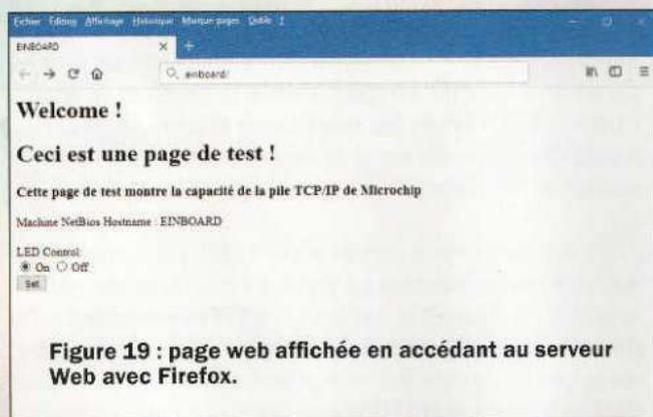


Figure 19 : page web affichée en accédant au serveur Web avec Firefox.

Listing 8 : extrait du code de la tâche « LedTask » qui intercepte les événements d'allumage/extinction de la LED.

```
/*Si l'évènement « Led on » est reçu*/
if (ReceiveEvt(&Led2On))
{
    /*Allume la LED */
    LED_2 = LED_ON;
}
else if (ReceiveEvt(&Led2Off))
{
    /*Eteint la LED */
    LED_2 = LED_OFF;
}
else
{
    /* Ne fais rien*/
}
```

À ce stade, notre application est terminée, nous devons simplement la tester.

Après avoir chargé le code, nous devons alimenter la carte et connecter le câble Ethernet au PC ou à un point d'accès.

Nous devons simplement nous connecter, avec n'importe quel navigateur, à l'adresse « <http://einboard> ». La page web devra ressembler à celle de la figure 19.

Comme nous pouvons le constater, en plus de la chaîne statique, la page fournit le nom du NetBIOS (EINBOARD) associé à la carte.

Pour contrôler la LED, nous devons sélectionner simplement le bouton radio correspondant à l'action que nous voulons effectuer (LED allumée ou éteinte) et cliquer sur le bouton « Set ». La LED doit s'allumer ou s'éteindre en conséquence.

Rappelons que, comme le module DHCP est inclus, il n'est pas nécessaire de modifier les attributs de la carte réseau.

Conclusion

Dans cette sixième leçon, nous avons étudié la pile TCP/IP Open Source de Microchip, qui fournit un support à l'environnement de développement MPLAB-X.

Nous avons, grâce à cet outil puissant, implémenté une application assez complexe qui est un serveur Web embarqué.

Cette application met en évidence la puissance et la polyvalence de l'environnement de développement, la plate-forme utilisée (les microcontrôleurs PIC32), ainsi que les piles et les infrastructures logicielles fournies. ■

Ceux d'entre vous qui impriment des objets en 3D, à l'aide par exemple de la 3DRAG+, avez besoin de logiciels de modélisation et d'édition en trois dimensions pour créer et éditer des fichiers 3D. Le logiciel OpenSCAD en est un bon exemple. Il permet entre autres de gérer la plupart des fichiers disponibles sur le site Thingiverse.

Tutoriel OpenSCAD

Première partie

du Laboratoire informatique Free&Open

Par modélisation 3D, nous entendons la conception d'une forme tridimensionnelle, la génération du modèle et sa représentation dans un format pouvant être utilisé par un ordinateur. C'est généralement la première étape de procédures très différentes.

Par exemple, un tel logiciel peut être utilisé pour fabriquer des objets physiques à l'aide de machines numériques, mais aussi pour simplement visualiser des objets purement « virtuels » ou encore des images ou des vidéos en 3D.

Il existe **deux techniques de modélisation 3D** : une **manuelle** et l'autre **algorithmique**. Une modélisation peut également être obtenue par numérisation ou à l'aide d'un scanner 3D (que nous vous présenterons dans les prochains numéros).

Les techniques manuelles sont inspirées des techniques traditionnelles des arts plastiques, dans lesquelles un matériau est façonné pour obtenir la forme souhaitée.

Dans les techniques algorithmiques, par contre, il s'agit de programmation (lignes de code) qui permettent de décrire l'objet. Ces techniques algorithmiques conviennent parfaitement pour obtenir une précision et pour reproduire plusieurs exemplaires d'un objet à l'identique, ce qui est difficile manuellement. Il existe plusieurs méthodes de représentations informatiques.

L'une des plus courantes, par exemple, est celle utilisée par divers programmes célèbres tels que « Blender » ou « Google Sketchup ». Cette méthode consiste à utiliser un **maillage polygonal qui décrit la surface externe du modèle ou de l'objet**.

Cependant, il en existe d'autres, telles que la « **géométrie de construction de solides** » (Constructive Solid Geometry ou CSG) dans laquelle **tout le volume du modèle est décrit à partir de primitives (formes) simples** telles que des cubes, des cylindres, des sphères, combinées entre elles à l'aide d'unions, d'intersections et ou de soustractions.

Pour résumer, nous pouvons dire que **cette technique consiste à représenter un objet solide complexe comme la combinaison d'objets solides simples** (cylindre, sphère, cône, etc.) à l'aide d'opérateurs géométriques booléens tels que des unions, des intersections ou des soustractions.

Comme vous pouvez l'imaginer, obtenir une représentation réaliste de formes organiques avec cette méthode nécessite beaucoup de travail. Par contre, cette dernière est très appropriée pour décrire des pièces mécaniques.

En fait, les pièces mécaniques peuvent être facilement décomposables en primitives géométriques. Elles sont plus facilement représentées par des objets solides que par des surfaces dont nous devons nous préoccuper de définir ce qui doit être considéré comme « interne ».

En fonction de l'objectif à atteindre, il convient de prêter attention aux différents aspects de la modélisation.

Si l'aspect extérieur du modèle et les propriétés visuelles de ses surfaces sont importants pour les représentations graphiques en 3D, travailler sur des modèles d'objets qui doivent être reproduits physiquement implique que la structure réelle de l'objet soit d'une grande importance.

Dans ce cas, une représentation basée sur des solides plutôt que sur des surfaces est plus appropriée.

OpenSCAD est un logiciel gratuit permettant de créer des modèles 3D basés sur une « **géométrie de construction de solides** ». Il excelle dans la modélisation d'objets dans lesquels la précision est requise, du plus simple comme une boîte rectangulaire à des structures ou mécanismes complexes.



Figure A : le site www.openscad.org.

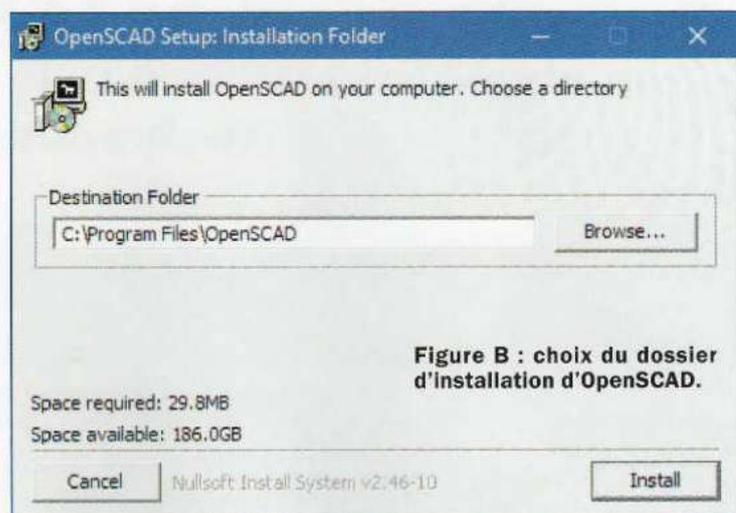


Figure B : choix du dossier d'installation d'OpenSCAD.

L'autre caractéristique essentielle d'OpenSCAD est que les modèles ne sont pas créés via une interface basée sur l'utilisation de la souris, mais sont décrits de façon algorithmique dans un langage de programmation simple.

Il s'agit de l'une des interfaces utilisateur les plus simples permettant d'obtenir la

précision requise par les applications CNC. Elle fournit également des outils utiles pour la réalisation de pièces composées de parties répétées ou pouvant être décrites à l'aide de formules mathématiques.

Ces caractéristiques, associées à la facilité d'utilisation, font d'OpenSCAD

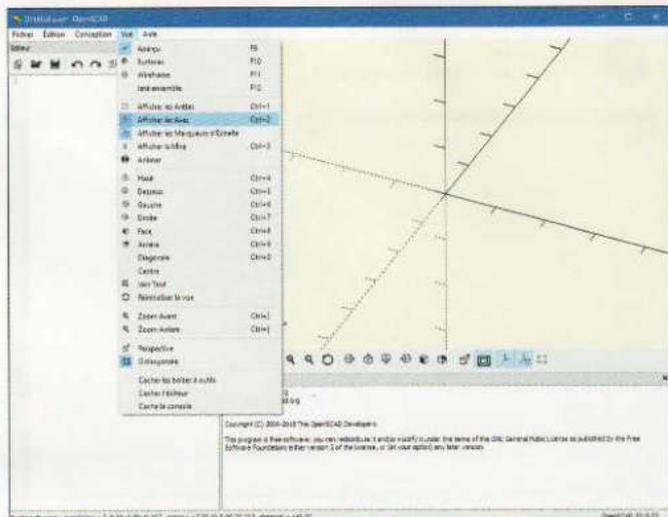


Figure 1A : pour afficher les 3 axes, sélectionnez dans le menu « Vue » → « Afficher les Axes ».

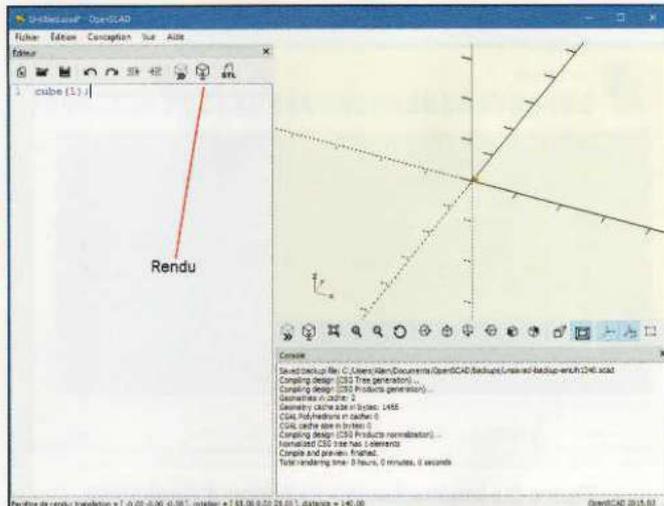


Figure 2 : rendu du cube.

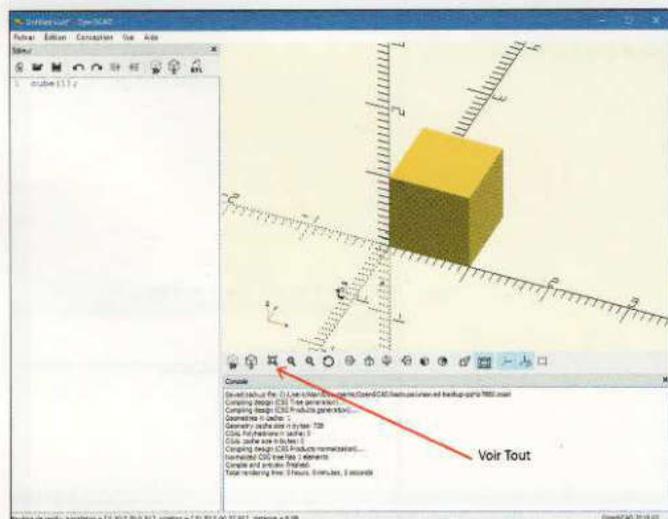


Figure 2A : l'icône « Voir tout » permet d'effectuer un zoom.

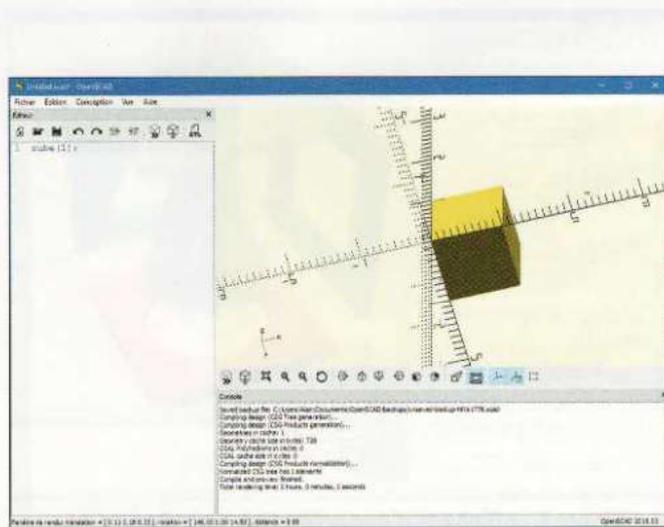


Figure 3 : orientation de l'objet à l'aide des boutons de la souris.

chemin d'installation en cliquant sur le bouton « Browse ».

Cliquez ensuite sur le bouton « Install » (en bas à droite), le processus d'installation commence jusqu'à l'apparition de la ligne « Completed » (il faut auparavant avoir cliqué sur le bouton « Show details » durant l'installation).

Pour fermer la fenêtre d'installation, cliquez sur le bouton « Close ». Vous devez ensuite chercher dans le menu « Démarrer » de Windows le programme « OpenSCAD ». Eventuellement, créez un raccourci sur votre bureau.

Notez que le processus d'installation détecte la langue de votre système d'exploitation et donc le logiciel sera automatiquement en français.

En cliquant sur l'icône du programme « OpenSCAD », ce dernier s'ouvre en affichant la fenêtre de « Bienvenue » visible en figure D. Dans la partie droite de la fenêtre, vous pouvez voir des exemples de base, avancés, et de fonctions. Cliquez sur les flèches pour développer les dossiers.

Par exemple si vous cliquez sur « Bases » → « Projection », vous devez voir apparaître la fenêtre de la figure E. Attention toutefois, il faut disposer d'une carte graphique relativement puissante, l'affichage peut prendre plusieurs dizaines de secondes.

Dans la partie de gauche se trouve le code de la description algorithmique de l'objet et à droite l'affichage en 3 dimensions de l'objet.

En dessous se trouve la console dans laquelle sont reportées les actions et les éventuelles erreurs.

À noter que la dernière ligne indique le temps mis pour afficher l'objet, ici cela a pris **15 secondes** avec une carte graphique NVIDIA GTX970, un Core I7 et 16 Go de RAM avec un disque SSD. Sur un PC relativement ancien, cela peut prendre plusieurs minutes pour afficher l'objet. Nous allons étudier maintenant les fonctionnalités d'OpenSCAD.

Le cube unitaire et les autres solides

Comme nous l'avons mentionné précédemment, lorsque vous lancez OpenSCAD, vous remarquez immédiatement

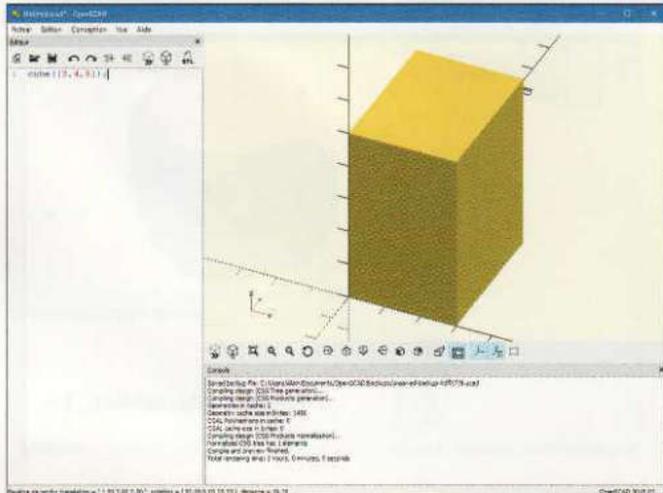


Figure 4 : indication des mesures sur les trois axes.

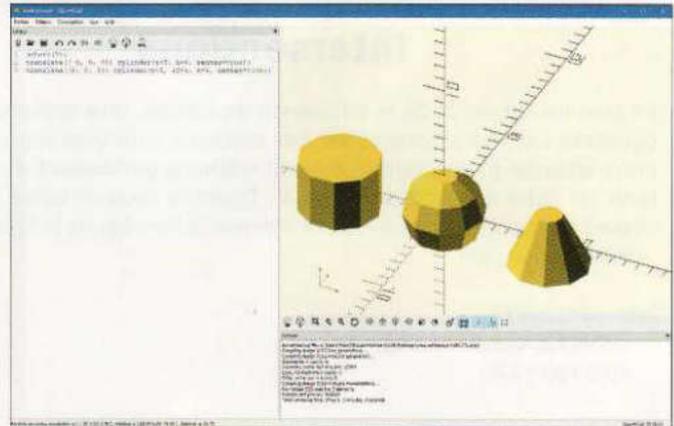


Figure 5 : ici différentes primitives.

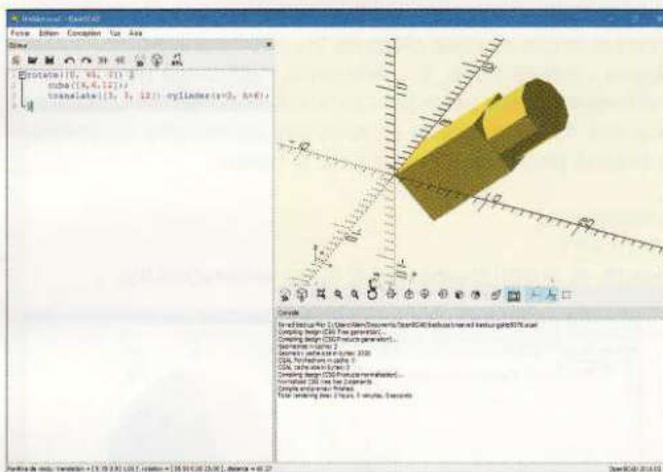


Figure 6 : translation d'un cylindre et rotation de 45°.

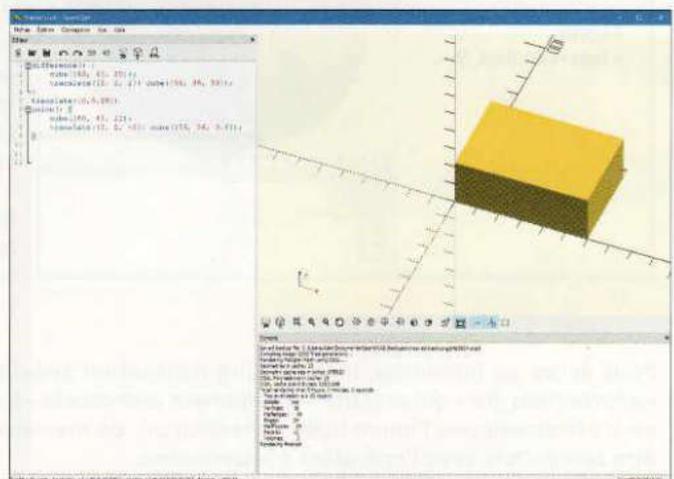


Figure 7 : construction d'un boîtier rectangulaire simple.

que l'interface est divisée en trois parties : à gauche, la zone où se trouve le code source, à droite, la zone dédiée à l'affichage du modèle rendu et en dessous une zone appelée « Console » où sont affichés les logs (événements) du programme, ainsi qu'un rappel de la licence GNU/GPL comme visible en figure 1.

NB : la figure 1 est obtenue en lançant le programme et en cliquant sur le bouton « Nouveau » à partir de la fenêtre de « Bienvenue ». Si les axes x, y et z ne sont pas déjà visibles, cliquez sur le menu « Vue » → « Afficher les Axes », comme indiqué en figure 1A.

Vous allez maintenant décrire votre premier modèle, pour cela dans la partie gauche, écrivez la ligne suivante :

cube(1);

Ensuite sélectionnez le menu « Conception » → « Rendu » ou cliquez sur l'icône « Rendu » (représenté par un cube avec un sablier) comme indiqué en figure 2. Vous devez apercevoir au centre de la zone d'affichage un cube de très petite taille.

OpenSCAD fonctionne avec des unités absolues, mais généralement les différentes formes sont interprétées en millimètres, de sorte que le cube paraît vraiment petit. Pour agrandir le cube, vous devez effectuer un zoom.

Placez le pointeur de la souris dans la zone d'affichage, appuyez et maintenez la touche « Ctrl » enfoncée et à l'aide de la molette de la souris vous pouvez

effectuer un zoom avant et arrière. Vous pouvez aussi utiliser l'icône « Voir tout » situé en dessous de la zone d'affichage comme visible en figure 2A

Maintenant, maintenez le bouton gauche de la souris enfoncé et bougez le pointeur dans la zone d'affichage, vous devez voir pivoter la vue de l'objet.

De même, en maintenant le bouton droit de la souris enfoncé, vous pouvez déplacer verticalement et horizontalement la vue de l'objet (voir la figure 3).

Après avoir admiré votre cube de tous les côtés, nous allons maintenant examiner un peu plus en détail les primitives géométriques d'OpenSCAD. Tout d'abord, nous devons dire que l'instruction « cube() » dans le code

Intersections

En plus de l'union et de la différence de solides, une opération courante est leur intersection. Par exemple, une intersection entre un cube et une sphère est une méthode permettant d'obtenir un cube aux angles arrondis. Tapez le code ci-après et cliquez sur l'icône « Rendu », vous obtenez le résultat de la figure « Intersection_1 ».

```
intersection() {
  cube(16, center=true);
  sphere(r=12);
}
```

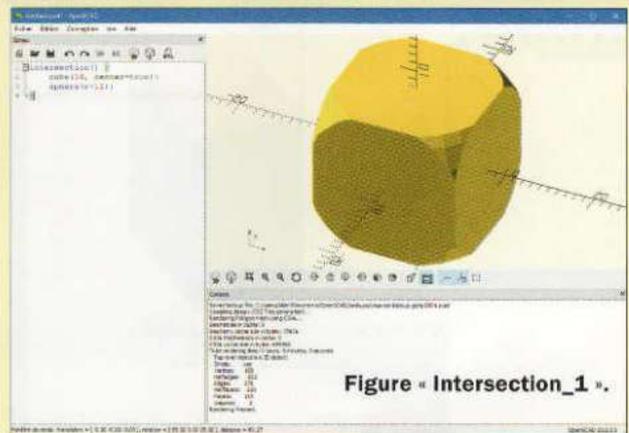


Figure « Intersection_1 ».

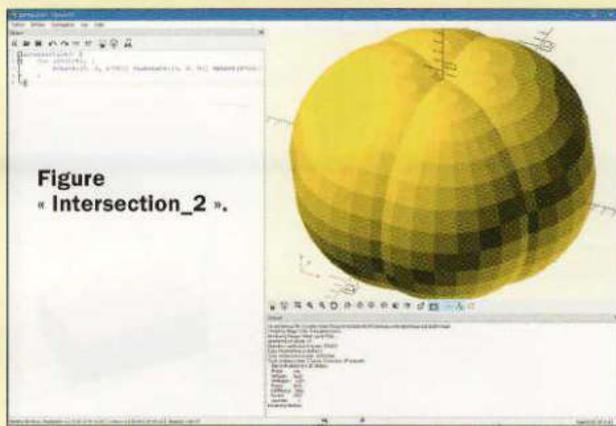


Figure « Intersection_2 ».

Toutefois, il convient de prendre des précautions lors de l'intersection de plusieurs objets à l'aide d'une boucle « for ». En effet, OpenSCAD crée une union implicite des objets contenus dans la boucle, il en résulte que cette union sera utilisée comme un seul objet dans l'intersection.

Voici un exemple de code ci-après, le résultat obtenu est visible en figure « Intersection_2 ». Attention, l'affichage n'a pas lieu immédiatement, sur notre configuration relativement puissante cela a pris 47 secondes, il se peut que selon votre ordinateur cela prenne plusieurs minutes pour le rendu.

```
intersection() {
  for (i=[0:5]) {
    rotate([0, 0, i*60]) translate([4, 0, 0]) sphere(r=10);
  }
}
```

Pour éviter ce problème, il existe une instruction spéciale « intersection_for » qui exécute implicitement une boucle « for » en n'effectuant pas l'union mais l'intersection, de manière à être compatible avec l'opération d'intersection.

Voici un exemple de code ci-après, le résultat obtenu est visible en figure « Intersection_3 ».

```
intersection() {
  intersection_for (i=[0:5]) {
    rotate([0, 0, i*60]) translate([4, 0, 0]) sphere(r=10);
  }
}
```

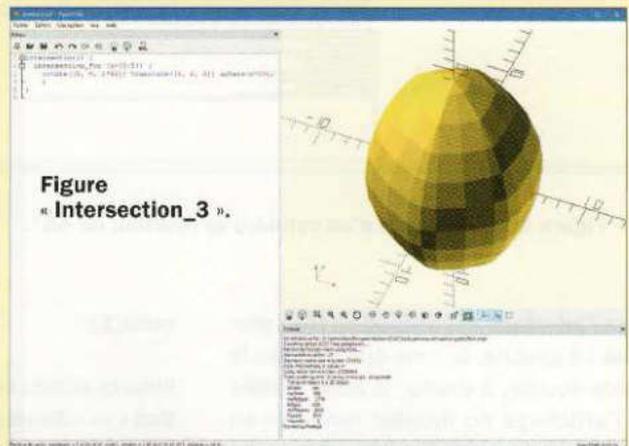


Figure « Intersection_3 ».

(fenêtre de gauche) permet de tracer des rectangles parallélépipédiques.

Les mesures sur les trois axes sont indiquées sous la forme de vecteur, c'est-à-dire un triplet de nombres entre crochets (voir la figure 4) :

```
cube([3,4,5]);
```

Les deux autres primitives solides les plus courantes sont l'instruction « sphere() », dont le seul paramètre est

le rayon, et l'instruction « cylinder() », qui peut également tracer des cônes tronqués en spécifiant les rayons des bases inférieure (r) et supérieure (r2).

De nombreuses primitives acceptent le paramètre « center », qui spécifie de dessiner le modèle centré sur les 3 axes et non à partir du plan d'origine.

Par exemple, si vous tapez le code suivant vous obtenez le résultat de la figure 5 :

```
sphere(3);
translate([-8, 0, 0]) cylinder(r=3, h=4, center=true);
translate([8, 0, 0]) cylinder(r=3, r2=1, h=4, center=true);
```

Dans le code ci-dessus, vous remarquez la présence de l'instruction « translate() » qui permet d'effectuer une transformation géométrique qui s'applique à l'élément suivant (dans ce cas, c'est le cylindre) et le déplace conformément au vecteur spécifié en tant que paramètre.

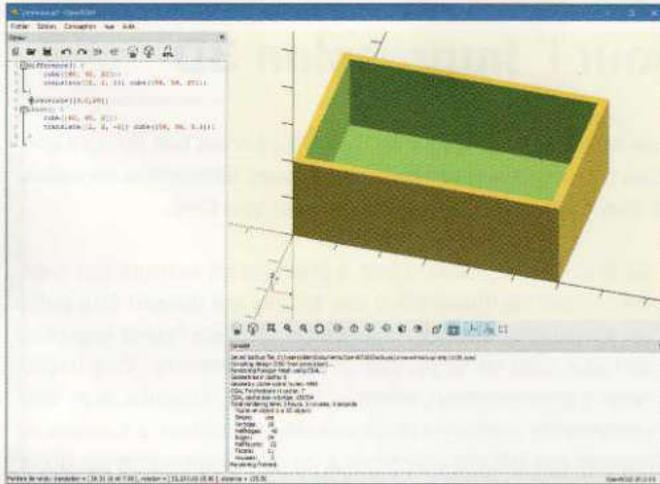


Figure 8 : en modifiant la ligne « `%translate([0,0,20])` », le couvercle n'est plus visible.

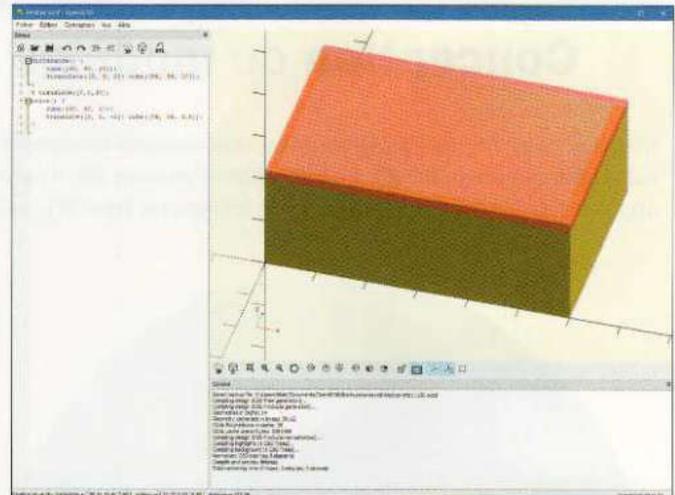


Figure 9 : ici les composants soustraits de l'objet.

Une transformation peut également être appliquée à plusieurs éléments, en les plaçant entre des accolades.

Par exemple, appliquons une translation à un cylindre puis une rotation de 45° au cylindre déplacé et à un parallélépipède (voir la figure 6). Le code est le suivant :

```
rotate([0, 45, 0]) {
  cube([6,6,12]);
  translate([3, 3, 12]) cylinder(r=3,
h=6);
}
```

NB : en géométrie, une translation est une transformation géométrique qui correspond à l'idée intuitive de « glissement » d'un objet, sans rotation, retournement ni déformation de cet objet.

La rotation est décrite par la transformation (instruction) « `rotate()` », qui prend comme paramètre un vecteur dont les angles correspondent aux trois rotations à effectuer par rapport aux trois axes.

Il est important de se rappeler que **le centre de la rotation se trouve toujours à l'origine des axes**. Il est donc pratique dans de nombreux cas de préparer d'abord la composante avec la bonne inclinaison, puis de la traduire dans sa position définitive.

Les autres transformations simples sont :

- l'échelle dont l'instruction est « `scale()` ». Elle multiplie la taille

de l'objet par la valeur donnée en tant que paramètre ;

- le redimensionnement dont l'instruction est « `resize()` », et qui adapte l'objet jusqu'à atteindre les valeurs désirées.

Par exemple, le code suivant :

```
scale([0.5,1,1]) sphere(r=10);
resize([5,0,0]) sphere(r=10);
```

dessinera une sphère déformée par 5 le long de l'axe des X et par 10 le long des autres axes.

Enfin, l'instruction « `mirror` » permet d'obtenir une vue en miroir d'un objet et prend comme paramètre le vecteur perpendiculaire au plan par rapport auquel le modèle est mis en miroir. Comme dans le cas de la rotation, ce plan passera nécessairement par l'origine.

Le premier modèle

Maintenant, comme premier modèle ou objet, nous allons construire un boîtier rectangulaire simple, dont la base sera composée de la différence entre deux parallélépipèdes, dont l'un est plus petit et est déplacé des dimensions des côtés. Voici le code ci-après :

```
difference() {
  cube([60, 40, 20]);
  translate([2, 2, 2]) cube([56, 36,
20]);
}
```

Il est important de se rappeler que **lorsque vous faites des différences et des unions de solides**, ces derniers **ne doivent pas avoir des faces** ou des parties de faces **en commun**, mais doivent **se croiser**.

Par exemple, si le deuxième parallélépipède du cube ci-dessus avait été un cube ([56, 36, 18]), OpenSCAD ne serait pas en mesure de déterminer la façon de gérer la face supérieure et donc ne générerait pas de modèle valide.

Pour éviter ces problèmes, dans le cas où vous ne pouvez pas utiliser les mesures de cette manière, ajoutez ou supprimez 0,01 des dimensions appropriées. C'est une différence assez petite pour ne pas impacter le modèle fini, mais suffisante pour qu'OpenSCAD distingue les parties internes et externes.

Le couvercle, par contre, sera formé par l'union de deux parallélépipèdes. L'un aussi grand que le boîtier et l'autre plus étroit, de manière à s'emboîter. Voici comment il est construit :

```
translate([0,0,20])
union() {
  cube([60, 40, 2]);
  translate([2, 2, -2]) cube([56, 36,
3.9]);
}
```

La compilation du code représentera un parallélépipède complet, comme visible en figure 7.

Conception de modèles pour l'impression 3D

Même si notre modèle a été correctement compilé et exporté sous la forme d'un fichier de type STL, il n'est pas dit qu'il soit idéal pour l'impression 3D. En effet, dans l'univers 3D, il existe des contraintes spécifiques et souvent différentes de celles imposées par d'autres utilisations de fichiers de type STL, telles que le rendu ou la fabrication avec une CNC.



Un exemple de rupture : ici deux côtés, tous les deux de dimensions 20 mm x 20 mm x 2 mm, celui imprimé horizontalement reste entier, alors que celui imprimé verticalement se rompt avec une coupure nette le long de l'une des couches.

La première considération à prendre en compte est bien évidemment les dimensions des détails qui doivent être suffisamment grandes pour apparaître une fois l'objet imprimé, compte tenu de la précision de l'imprimante. Une imprimante pour amateur, comme la 3DRAG, travaille avec des couches de quelques dixièmes de millimètres d'épaisseur. Insérer des détails inférieurs à un demi-millimètre est donc inutile et contre-productif, car ils ne seront pas reproduits.

Une autre vérification évidente, mais importante, concerne le fait que l'objet soit suffisamment robuste et qu'il ne comporte pas de points faibles qui pourraient engendrer une cassure de l'objet. L'aspect permettant de vérifier l'épaisseur des parois, qui avec la matière utilisée par la 3DRAG devrait être d'au moins un millimètre, est qu'il n'y ait pas de pièces trop fines soumises à un effort particulier.

À cet égard, il est important de rappeler que les imprimantes travaillent selon des couches horizontales. Un objet long et étroit placé horizontalement est constitué de filaments parallèles qui lui confèrent une bonne résistance. La même pièce imprimée verticalement sera divisée en couches plus fines et l'objet sera donc moins résistant.

Un exemple de cassure est visible sur la photo ci-dessous. Les deux côtés étant de taille égale, celui imprimé horizontalement reste entier, tandis que l'autre imprimé verticalement se casse de façon nette le long de l'une des couches.

Les imprimantes qui travaillent en déposant des filaments de matériau ont évidemment besoin d'une surface de support sur laquelle les déposer. Si d'une part les imprimantes peuvent remédier au problème en ajoutant des structures de support, d'autre part, il faut garder à l'esprit la contrainte liée à la phase de conception permettant d'obtenir des objets mieux finis et avec moins de gaspillage de matériau.

De ce point de vue, l'un des critères à prendre en compte est le porte-à-faux d'une partie de l'impression 3D par dépôt de filament fondu ne reposant pas ou peu sur une base solide. Des angles supérieurs à 45 degrés conviennent généralement bien, alors qu'en dessous de 45°, les filaments tombent dans le vide en fonction du matériau utilisé.

Pour résoudre tous ces problèmes, il peut être utile de prêter attention à l'orientation de la pièce à imprimer. Une simple rotation permet parfois d'éliminer les effets de porte-à-faux (comme nous l'avons fait pour le couvercle du boîtier).

En guise d'alternative au support d'un filament complet, une ouverture rectangulaire peut être prévue sur une paroi verticale qui permet ainsi de placer les filaments sur les parois latérales. Cette technique, qui se nomme technique du pont ou bridge, peut fonctionner sur de petites distances, environ un centimètre, en fonction de la vitesse. L'impression à des vitesses plus élevées permet une plus grande distance dans certaines limites.

Cependant, au lieu d'utiliser des ponts, il peut être intéressant de concevoir l'objet de telle sorte qu'il n'existe pas de telles situations, comme dans notre boîtier où les passages des connecteurs se trouvent sur le bord.

Un dernier aspect concerne l'économie de matériau utilisé. Il est vrai que l'imprimante peut influencer cet aspect, notamment via les paramètres de remplissage des modèles solides. Mais dans certains cas, le modèle lui-même peut être modifié pour réduire le volume imprimé. Un exemple limite peut être l'utilisation de maillages (en tenant compte des contraintes sur les angles des crêtes) au lieu de parois pleines.

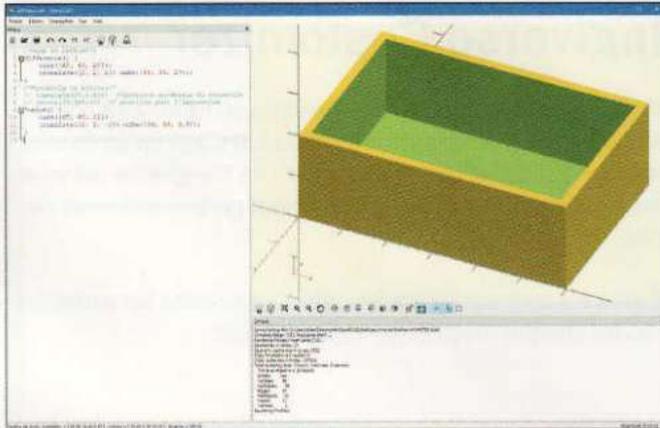


Figure 10 : rendu du fichier « Boitier.stl » dans OpenSCAD.

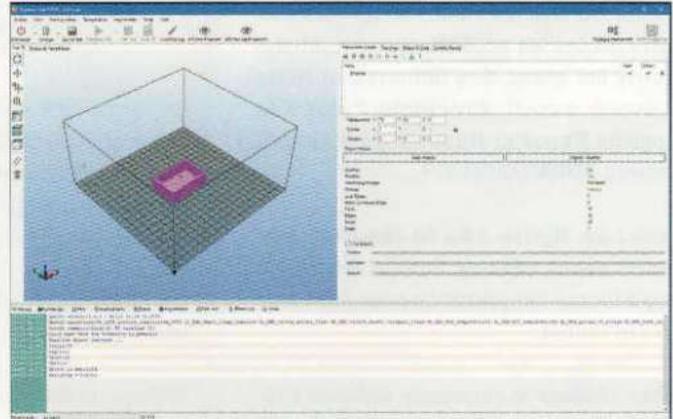


Figure 10A : ouverture du fichier « Boitier.stl » dans Repetier-Host.

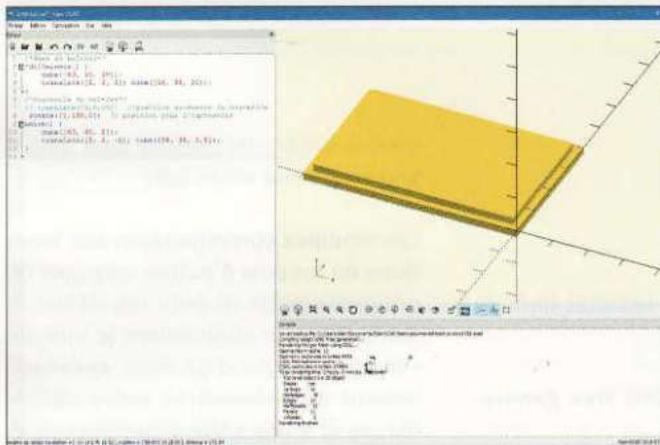


Figure 11 : rendu du fichier « Couvercle.stl ».

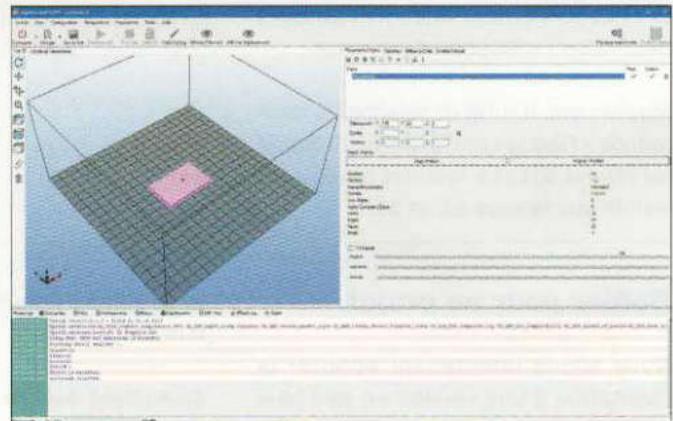


Figure 11A : ouverture du fichier « Couvercle.stl » dans Repetier-Host.

Cependant, nous pouvons exploiter les **modificateurs** d'OpenSCAD pour mieux visualiser la structure de l'objet.

En plaçant en début de ligne le symbole « % » (%translate([0,0,20])), la partie supérieure du modèle (ici du boîtier) est **affichée** de manière **transparente**, comme illustré en figure 8.

Les autres modificateurs pris en charge sont « * », utilisé pour **exclure totalement un composant** du modèle, « I » qui, au contraire, permet uniquement le **rendu du composant** spécifié et « # » qui rend normalement le composant, mais en le dessinant en **rose transparente**. Ce dernier, par exemple, est utile pour vérifier le positionnement correct des composants soustraits de l'objet avec l'opération de différence (voir la figure 9).

Le code est le suivant, appuyez sur l'icône « **Aperçu** » à gauche de l'icône « **Rendu** » :

```

difference() {
  cube([60, 40, 20]);
  %translate([2, 2, 2]) cube([56, 36,
20]);
}
# translate([0,0,20])
union() {
  cube([60, 40, 2]);
  translate([2, 2, -2]) cube([56, 36,
3.9]);
}

```

Comme tout langage de programmation, il est possible d'ajouter des commentaires. OpenSCAD utilise les **commentaires** semblables à ceux du langage C++. À l'aide de « // », il est possible d'ajouter une ligne de commentaire et avec « /* » et « */ » il est possible de délimiter plusieurs lignes de commentaires.

Complétons notre premier modèle avec quelques commentaires descriptifs et sauvegardons le fichier avec l'extension « .scad ».

Avant de pouvoir l'exporter au format STL pour l'imprimer en 3D, il nous reste une dernière étape.

```

/*Base du boîtier*/
difference() {
  cube([60, 40, 20]);
  translate([2, 2, 2]) cube([56, 36, 20]);
}
/*Couvercle du boîtier*/
// translate([0,0,20]) //position
au-dessus du couvercle
// rotate([0,180,0]) // position pour
l'impression
*union() {
  cube([60, 40, 2]);
  translate([2, 2, -2]) cube([56, 36,
3.9]);
}

```

À l'aide de la commande « **Rendu** », le modèle est affiché après quelques secondes (voir la figure 10). Vous pouvez utiliser la commande dans le menu « **Fichier** » → « **Exporter** » → « **Exporter comme STL** ».

Nous avons appelé notre fichier « **Boîtier.stl** » (évitons les accents et ponctuations dans les noms des fichiers), et nous l'avons ouvert directement dans le logiciel **Repetier-Host** qui gère l'imprimante 3DRAG/3DRAG+.

Voici **en figure 10a le résultat que nous avons obtenu**. Il nous reste plus qu'à lancer l'imprimante pour fabriquer notre boîtier.

Pour réaliser le couvercle, **enlevez** simplement le modificateur « ***** » devant l'instruction « **union** » puis **placez-le** devant l'instruction « **difference** » et **décommentez** la ligne « **rotate...** », cela permet d'imprimer le couvercle sans nécessiter l'utilisation de support.

Maintenant, il suffit de relancer la commande « **Rendu** » et d'exporter le fichier au format STL. Le résultat obtenu est visible aux **figures 11 et 11A**.

Boîtier pour un projet

Nous allons maintenant aborder la réalisation d'une version un peu plus complexe du boîtier, décrite de manière paramétrique et en exploitant au mieux le potentiel d'OpenSCAD. Le code source complet de ce modèle est disponible dans le listing 1.

Le fichier se nomme « **scatola_arduino.scad** ». Examinons les différentes fonctionnalités utilisées et la manière dont elles le sont.

Tout d'abord, nous devons définir les dimensions du boîtier, comme nous voulons les rendre paramétriques, nous allons enregistrer ces dimensions sous la forme de variables.

Une chose importante à considérer est que les variables dans OpenSCAD se comportent de manière particulière et ressemblent davantage à des constantes paramétriques qu'à des variables réelles d'autres langages. En particulier, leur valeur est calculée lors de la compilation et ne changera pas lors du rendu.

Par exemple, en utilisant la commande « **echo** » pour imprimer du texte dans le journal des logs, le code suivant :

Thingiverse Customizer

En parcourant les modèles pour imprimante 3D disponibles en ligne, vous pouvez voir à quel point OpenSCAD est un programme très utilisé. Il s'agit en particulier du langage pris en charge par l'application « **Customizer** » de Thingiverse qui vous permet de charger des modèles paramétriques facilement personnalisables par les utilisateurs du site.

La meilleure chose à faire est de créer un fichier « **.scad** » qui utilise les variables déclarées au début, avant chaque module, au format suivant :

```
//Description
variable = valeur_par_defaut;
```

L'application « **Customizer** » permet beaucoup plus. Pour en savoir plus sur ce sujet, vous pouvez consulter la documentation en ligne à l'adresse <https://customizer.makebot.com/docs>.

```
a = 1;
echo(a);
a = 2;
echo(a);
```

nous donne comme résultat dans la fenêtre des logs :

```
Compiling design (CSG Tree generation)...
ECHO: 2
ECHO: 2
```

Revenons à notre boîtier, nous n'aurons aucun problème à définir les mesures de notre carte.

Par exemple, les dimensions d'une carte Arduino UNO peuvent être définies à l'aide de variables dont la valeur est déclarée au début du fichier.

Après avoir inséré les dimensions de la carte selon les trois axes, la position des trous des supports (selon les coordonnées X et Y, par rapport à la carte) et les ouvertures à effectuer dans le bord du boîtier (dans notre cas, d'un seul côté, pour les connecteurs d'alimentation et la prise USB), nous pouvons définir certains paramètres tels que la largeur des parois, la marge de tolérance à laisser autour de la carte et la hauteur des supports.

À ce stade, nous avons besoin d'une nouvelle fonctionnalité d'OpenSCAD. Puisque nous voulons placer des supports en différentes positions, nous

devons utiliser un module pour définir leur forme une seule fois.

Les **modules correspondent aux fonctions ou macros** d'autres langages de programmation et pour les définir, il suffit d'utiliser simplement le mot-clé « **module** » suivi d'un nom, éventuellement de paramètres entre parenthèses et d'une série d'instructions et de composants entre accolades.

Lors de la construction du modèle, il est alors possible de le rappeler en utilisant simplement le nom, avec les valeurs des paramètres entre parenthèses.

Dans notre cas, nous définissons un **module « pin »** auquel nous passons en paramètres les positions (x, y) sur lesquelles les connecteurs doivent être positionnés par rapport à la carte.

Le code du module « **pin** » correspond simplement à l'union de deux cylindres, translattée dans la position souhaitée.

Passons maintenant à la définition du boîtier réel. Comme précédemment, la structure principale est composée de la différence entre deux cubes, cependant les mesures sont calculées à partir des paramètres définis précédemment.

Dans ce cas, nous souhaitons ajouter les supports de la carte et supprimer les trous des connecteurs qui dépassent.

Listing 1 : scatola_arduino.scad

```

/* ***** Dimensions de la carte ***** */

board_x = 69;
board_y = 53;
board_z = 13;

pins = [
  [14, 3],
  [15, 51],
  [66, 7],
  [66, 36]
];

// Cela suppose que tous les connecteurs soient d'un côté.

connectors = [
  [3, 11],
  [31, 14]
];

/* ***** Paramètres de construction ***** */

wall = 2;
ease = 1;
v_ease = 4;
pin_height = 2;

/* ***** Modules ***** */

module pin(x,y) {
  translate([x + wall + ease, y + wall + ease, wall - 0.01]) union() {
    cylinder(h=v_ease + pin_height, r=1);
    cylinder(h=v_ease, r=2.5);
  }
};

/* ***** Boîtier ***** */

d = (wall + ease) * 2;

union() {
  difference () {
    cube([board_x + d, board_y + d, board_z + wall + v_ease]);
    translate([wall, wall, wall])
    cube([board_x + ease*2, board_y + ease * 2, board_z + v_ease + 1]);
    for ( c = connectors ) {
      translate([-1, c[0] + wall + ease, wall + v_ease])
      cube([wall + 2, c[1], board_z + 1]);
    }
  }
  for ( p = pins ) {
    pin(p[0],p[1]);
  }
}

```

laquelle aurait affiché :

0
1
2

Dans le cas des ouvertures des connecteurs, nous nous sommes limités à un seul côté.

Cependant, avec une carte ayant des connecteurs dépassant de plusieurs côtés, cela peut être un bon exercice pour effectuer un cycle sur deux niveaux de vecteurs pour les connecteurs.

Dans ce cas également, une fois le modèle terminé et enregistré, la commande « **Rendu** » suffira. Elle sera suivie de l'exportation au format STL en un fichier qui pourra être utilisé avec les programmes d'impression 3D les plus courants.

Si vous souhaitez partager votre modèle, même sur un site ne prenant pas directement en charge les formats « .scad » ou « .stl », vous pouvez enregistrer votre modèle au format image.

Utilisez la commande : « **Fichier** » → « **Exporter** » → « **Exporter comme Image...** ».

Conclusion

Dans cette première partie de ce tutoriel, nous avons seulement abordé l'énorme potentiel d'OpenSCAD, en nous concentrant principalement sur les caractéristiques utilisables immédiatement pour la réalisation de modèles simples mais pratiques.

Dans la prochaine partie de ce didacticiel, nous aborderons des fonctionnalités plus complexes, qui font d'OpenSCAD une excellente solution pour les besoins de modélisation à tous les niveaux.

Rappelons qu'OpenSCAD est un programme gratuit (sous licence GNU GPL) et multiplateforme, disponible pour la plupart des distributions Linux et téléchargeable gratuitement pour Windows et Mac OS X sur le site officiel :

<http://www.openscad.org/>. ■

Dans les deux cas, nous utilisons une boucle « **for** ». Dans OpenSCAD, les boucles « **for** » peuvent travailler **sur un vecteur ou sur une plage numérique**, dans notre cas, nous travaillons sur les vecteurs « pins » et « connectors ».

Si nous avons voulu travailler sur une plage numérique, nous aurions pu utiliser une syntaxe comme celle-ci :

```

for (i=[0:2]) {
  echo(i);
}

```



Apprenez à maîtriser

KiCad EDA

**Sixième
partie**

de *Francesco Ficili et Vincenzo Germano*

Nous poursuivons l'étude de Pcbnew, l'éditeur utilisé pour la mise en œuvre des circuits imprimés dans KiCad, en illustrant l'utilisation des bibliothèques et de l'éditeur d'empreintes. Nous continuons ainsi le développement de notre projet pratique. Sixième partie.

Dans le numéro précédent, nous avons poursuivi l'analyse de l'outil Pcbnew, que nous avons déjà commencé à aborder dans la quatrième partie de ce cours.

En effet, nous avons abordé l'éditeur de « layout », les remplissages des zones ainsi que la gestion des bibliothèques. Toujours dans le numéro précédent, nous avons avancé dans la réalisation du projet pratique, en complétant la phase de positionnement des composants. Nous allons maintenant conclure le cours consacré à KiCad.

Nous compléterons notre aperçu de l'outil Pcbnew et nous terminerons la réalisation de notre projet pratique. Enfin, nous aborderons la création des fichiers gerber nécessaires à la réalisation de nos circuits imprimés.

Demoboard PIC18 : le routage

Nous ne nous attarderons pas sur le routage du projet, le sujet ayant déjà été traité dans les précédentes leçons et

une analyse détaillée du routage d'une carte de cette taille serait excessivement lourde. Nous nous limiterons donc à mettre en évidence certaines lignes directrices valables pour tout circuit à router.

Avant de router la carte, tous les composants doivent être positionnés de la manière la plus pratique en limitant au maximum la longueur des pistes entre les interconnexions des composants.

Ensuite, vous devez définir les règles de conception du projet (Design Rules). Pour cela, allez dans le menu « Règles de Conception ». Vous devez notamment tenir compte de :

- La **largeur des pistes** ;
- l'**isolation** entre les pistes ;
- du **diamètre** des vias ;
- du **diamètre de perçage** des vias (il ne faut pas que le diamètre de perçage d'un via soit supérieur au diamètre du via lui-même).

Lors du routage, évitez de créer des pistes avec des angles droits, car ceux-ci provoquent des perturbations à haute fréquence et des échauffements en cas de passage de forts courants dans le cuivre.

Évitez des pistes très longues ainsi que des boucles (effets inductifs). Réalisez toujours au moins un plan de masse sur l'une des faces du circuit imprimé (dans le cas d'un double face). Faites attention à la position de la sérigraphie, en évitant, autant que possible, que celles-ci chevauchent les emplacements des composants, des pistes et des vias.

La figure 1 montre le circuit imprimé du projet une fois le routage terminé, tandis que la figure 2 montre la représentation en 3 dimensions à l'aide du visualisateur 3D de KiCad (menu « Affichage → 3D Visualisateur »).

À l'aide des modèles 3D associés aux empreintes, Pcbnew générera le rendu 3D de la carte. Pour le rendu 3D, nous avons utilisé les bibliothèques 3D créées par Walter Lain.

Elles peuvent être téléchargées à l'adresse :

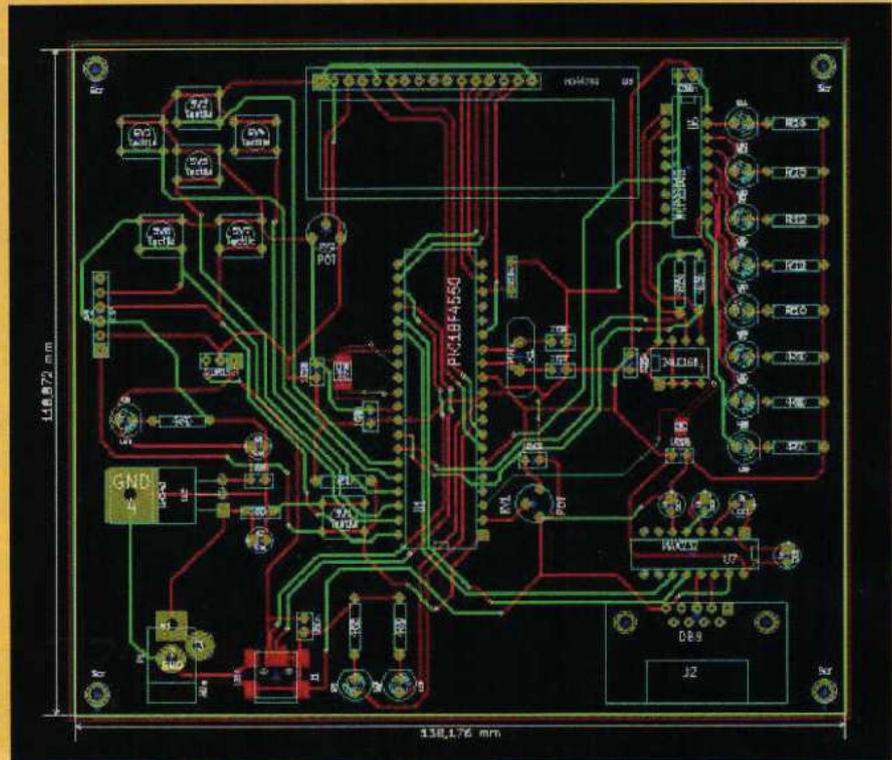


Figure 1 : circuit imprimé de la Demoboard PIC18F4550 une fois le routage terminé (sans le plan de masse).

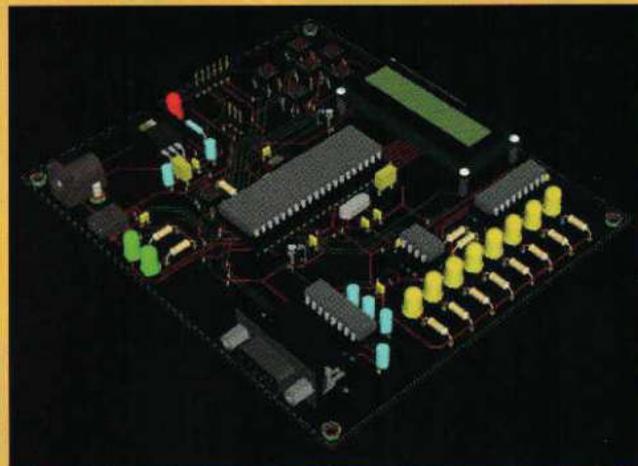


Figure 2 : représentation en 3D de la Demoboard PIC18F4550.

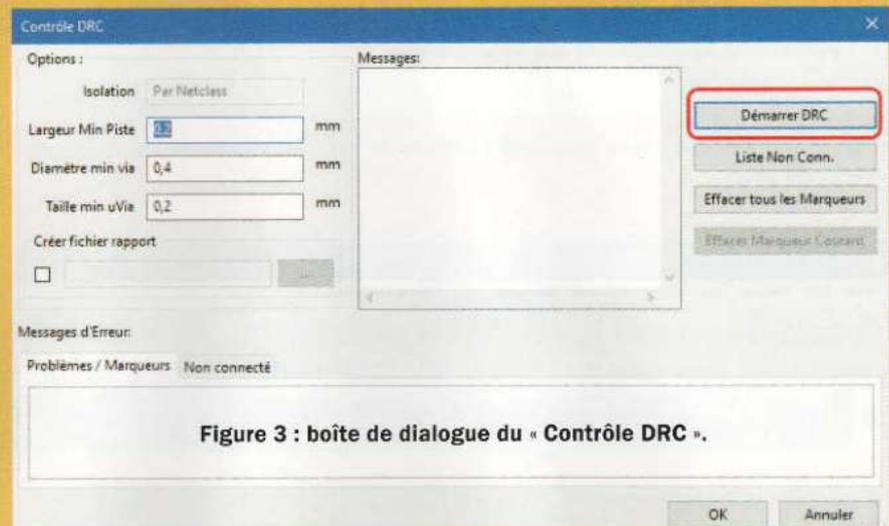


Figure 3 : boîte de dialogue du « Contrôle DRC ».

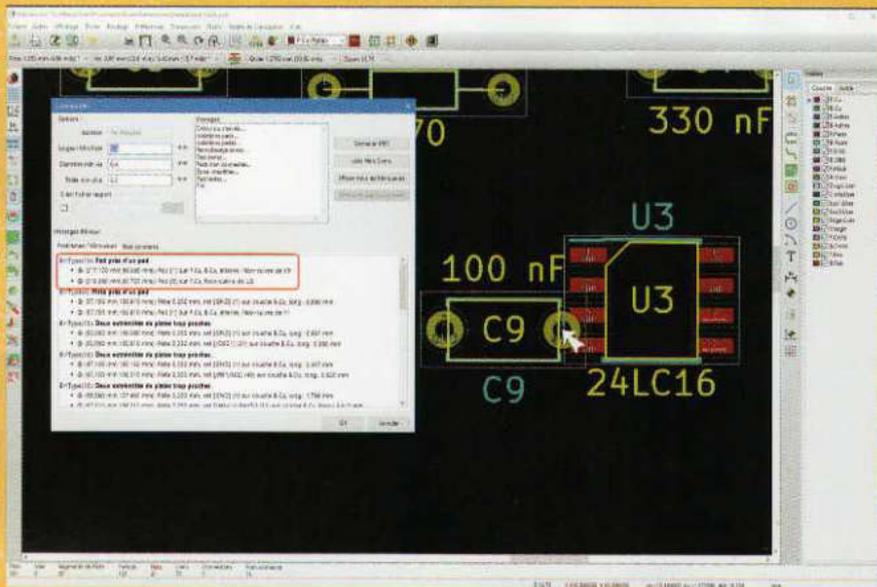


Figure 4 : exemple d'erreur concernant la distance minimale entre les pastilles de 2 composants.

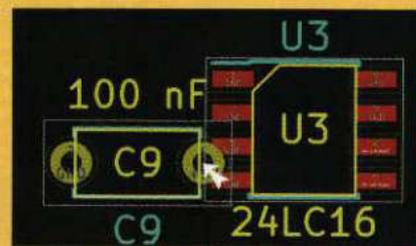


Figure 5 : la flèche blanche indique où l'erreur se situe.

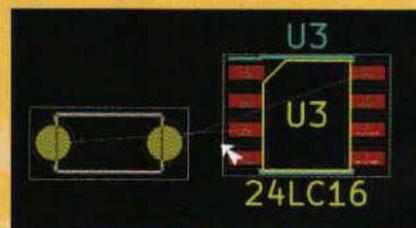


Figure 6 : déplacement de C9 pour corriger l'erreur.

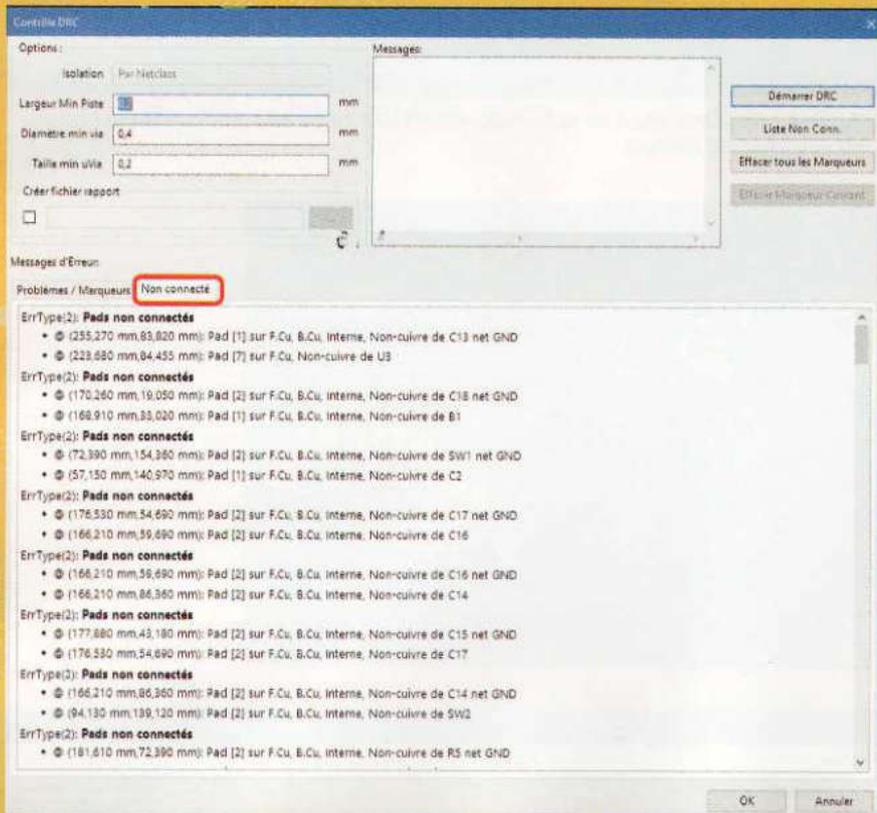


Figure 6A : l'onglet « Non connecté » permet de vérifier si des pads ne sont pas connectés.



Figure 7A : génération des fichiers de sortie à partir du menu principal.



Figure 7 : génération des fichiers de sortie à partir de l'icône « Tracer en format... ».

<http://smisioto.no-ip.org/electronica/kicad/kicad-en.htm>

NB : pour les figures 1 et 2, le plan de masse a été supprimé, afin de permettre une vision plus claire du routage.

Vérification du PCB : contrôle des règles du projet

Une fois le routage terminé, Pcbnew offre la possibilité d'effectuer une

vérification automatique du projet. Il vérifie :

- le respect des règles définies lors de la phase de création du projet ;
- la présence de nœuds non connectés.

Pcbnew offre également la possibilité de remplir automatiquement des zones. Pour vérifier les règles de conception, vous devez cliquer sur l'**icône avec le symbole d'une coccinelle** qui se trouve en haut de la barre de projet, à côté de l'icône « Netlist ».

Sinon, vous pouvez aller dans le menu « Outils → DRC ». Une fois cette opération effectuée, la fenêtre illustrée en figure 3 s'ouvre.

Pour démarrer le contrôle des règles de conception du projet. Vous devez cliquer sur le bouton « Démarrer DRC » (cercle en rouge), comme illustré en figure 3.

Les options du contrôle sont affichées dans la zone en haut à gauche. Il est également possible de générer et d'enregistrer un fichier contenant le rapport.

Les contrôles suivants sont effectués :

- isolation (à partir des préférences définies lors de la phase de conception et des propriétés des composants) ;
- largeur minimale des pistes ;
- diamètre minimal des vias ;
- taille minimale des vias.

La figure 4 montre un exemple de contrôle effectué qui renvoie une erreur concernant la distance minimale entre les pastilles de 2 composants. L'erreur et sa correction sont illustrées (voir la flèche blanche entre C9 et U3 en figure 5). Comme vous pouvez le voir sur l'image, la pastille de C9 vient toucher les pads 3 et 4 de U3. La règle concernant la distance minimale entre 2 pads imposée dans le projet a donc été violée.

Pour résoudre le problème, il suffit de déplacer C9 afin que la distance minimale soit respectée, comme illustré en figure 6.

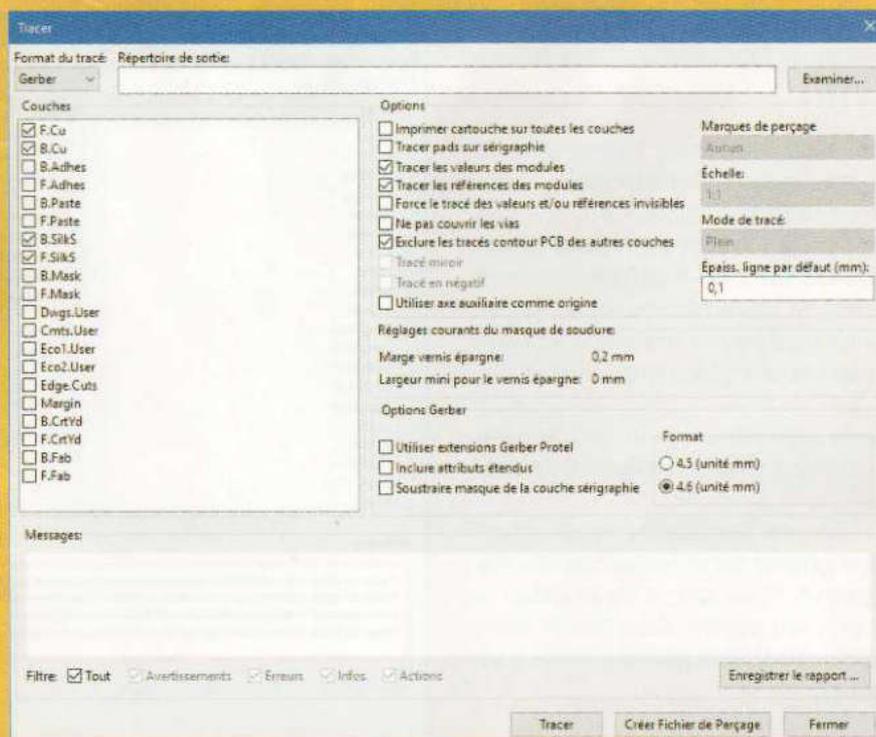


Figure 8 : la boîte de dialogue « Tracer ».

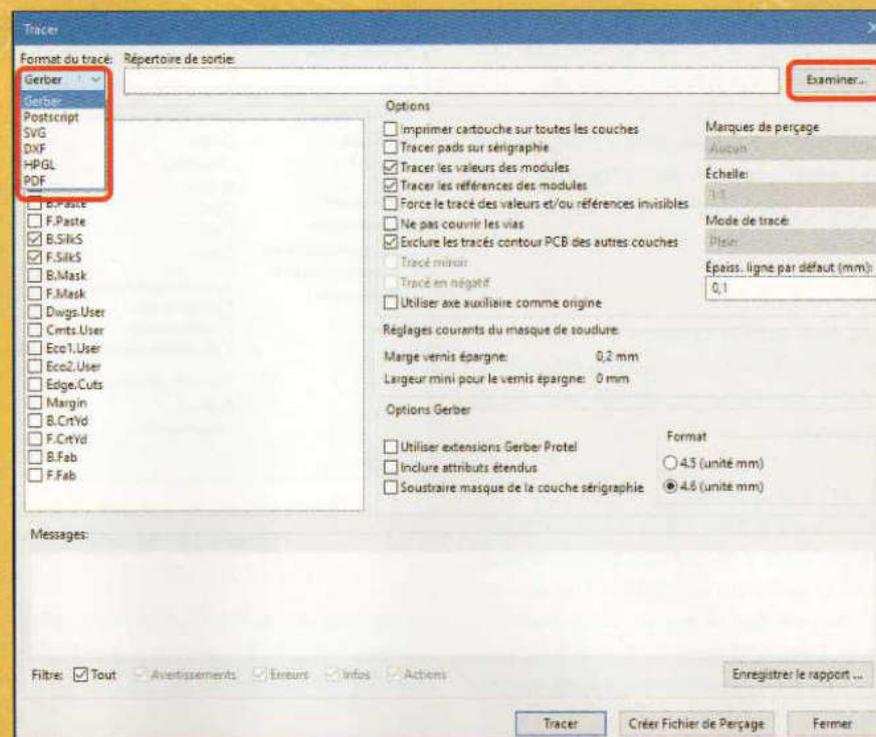


Figure 8A : menu déroulant permettant de sélectionner le type de fichier de sortie.

Il est également possible de vérifier uniquement la présence de nœuds déconnectés, en évitant de vérifier toutes les règles. Pour cela, cliquez sur l'onglet « Non connecté » (entouré en rouge) dans la boîte de dialogue « Contrôle DRC » (voir la figure 6A).

Génération des fichiers de sortie

Une fois que vous êtes certain qu'aucune règle de conception n'a été violée et qu'il n'y a pas de nœud déconnecté dans le circuit imprimé, il est possible

de passer à la phase de création des fichiers de fabrication appelés fichiers Gerber.

Pour cela, cliquez sur l'icône située à droite de celle de l'imprimante, elle ressemble à la lettre « π ». Il s'agit de l'icône « **Tracer en format HPGL, POSTSCRIPT ou GERBER** » comme visible en figure 7 (entourée en rouge). Sinon vous pouvez aller dans le menu « Fichiers → Tracer » (voir la figure 7A).

Après avoir cliqué sur l'icône, la boîte de dialogue illustrée en figure 8 s'ouvre.

Examinons les étapes nécessaires pour générer correctement les fichiers de sortie. Vous devez d'abord définir le chemin du dossier dans lequel enregistrer les fichiers générés (fichiers de sortie).

Pour cela, cliquez sur le bouton « Examiner » (en haut à droite, voir la figure 8A), une nouvelle boîte de dialogue s'ouvre dans laquelle vous pouvez sélectionner ou créer un dossier qui contiendra vos fichiers de sortie.

Vous pouvez également sélectionner un chemin relatif, de sorte que vous n'aurez pas à modifier ce paramètre si vous devez ouvrir le projet sur un autre ordinateur.

Une fois que le chemin d'enregistrement des fichiers de sortie a été défini, il est nécessaire de choisir le format de sortie.

KiCad est capable de générer des fichiers de sortie dans différents formats. Vous pouvez les sélectionner dans la boîte de dialogue « Tracer » en faisant défiler le menu déroulant en haut à gauche (voir la figure 8A). Les formats de sortie suivants sont disponibles :

- Gerber ;
- Postscript ;
- SVG ;
- DXF ;
- HPGL ;
- PDF.

Une fois que vous avez sélectionné le format désiré, généralement « Gerber » pour la fabrication du circuit imprimé,

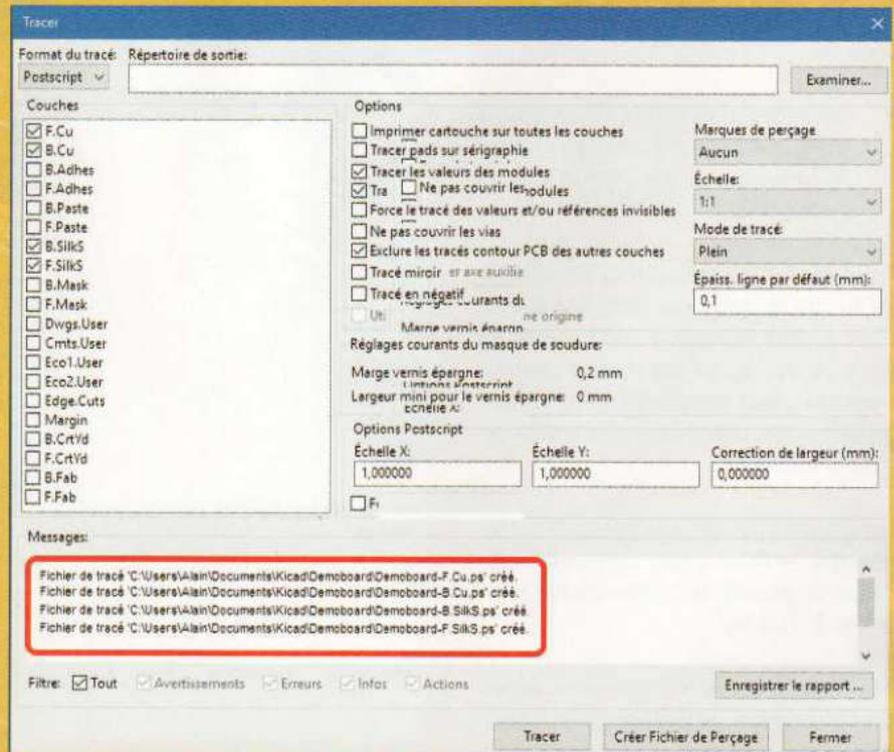


Figure 9 : rapport sur la génération des fichiers Gerber (entouré en rouge).

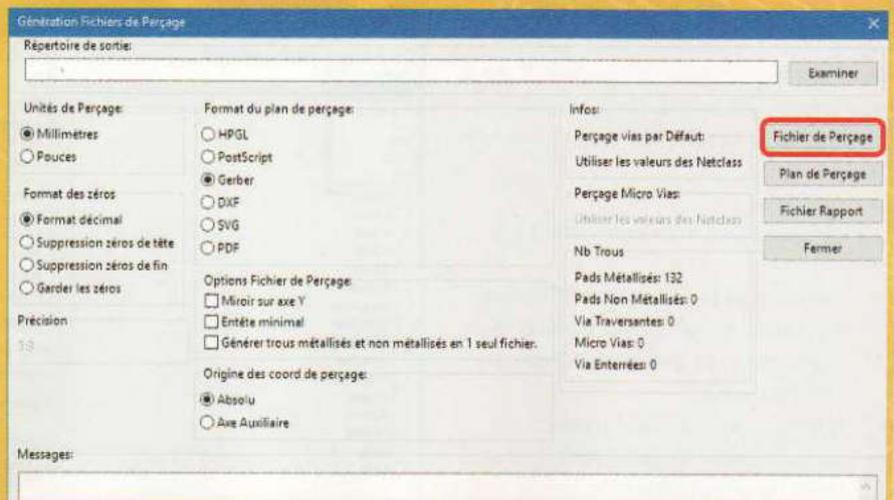


Figure 10 : la boîte de dialogue du fichier de perçage.

vous pouvez sélectionner les différentes couches que vous souhaitez ajouter à votre fichier de sortie.

La liste des couches se trouve dans la partie gauche de la fenêtre « Tracer » (il suffit de cocher ou de décocher la case correspondante).

À ce stade, vous pouvez modifier certaines options présentes dans la partie droite de la fenêtre « Tracer ».

Une fois tous les réglages effectués, cliquez sur le bouton « Tracer » pour

générer les fichiers de sortie. L'outil fournit un rapport sur la génération des fichiers en bas dans la fenêtre « Tracer », comme illustré en figure 9 (entouré en rouge).

En plus des fichiers liés aux différentes couches, des fichiers de perçage doivent également être générés.

Ce type de fichier est généré dans un processus séparé, qui peut être sélectionné en cliquant sur le bouton « Créer Fichier de Perçage », en bas de la fenêtre.

Dans ce cas également, une fenêtre spécifique s'ouvre (voir la figure 10), à partir de laquelle il est possible de définir une série d'options pour la génération des fichiers de perçage, ainsi que le chemin du dossier où les enregistrer.

N'oubliez pas que les fichiers de perçage générés doivent être cohérents avec le format choisi pour les fichiers des couches.

Par conséquent, si vous générez des fichiers gerber pour les couches, vous devez sélectionner l'option « Gerber » pour le format du plan de perçage et les « mm » pour les unités.

Une fois les réglages terminés, cliquez sur le bouton « Fichier de Perçage » (entouré en rouge sur la figure 10) pour commencer à générer les fichiers. Dans ce cas également, le résultat de l'opération est résumé dans la boîte de messages.

Dans notre cas c'est le fichier « Demoboard.drl » qui a été généré, il correspond au fichier de perçage (voir la figure 11).

Création du fichier de position des empreintes

Pcbnew permet également de créer des fichiers de position des empreintes des composants pour un montage en technologie CMS (montage en surface). Pour accéder à cette option, sélectionnez le menu « Fichiers → Fichiers de Fabrication → Fichier Position (.pos) des Empreintes » (voir la figure 12A).

Après avoir effectué un clic sur l'élément du menu, la fenêtre de la figure 12 s'affiche.

À partir de cette fenêtre, vous pouvez choisir le dossier ainsi que le chemin dans lequel enregistrer les fichiers générés par ce menu en cliquant sur le bouton « Examiner ».

Les autres options permettent de sélectionner l'unité de mesure (mm), ainsi que la possibilité de générer un fichier de position des empreintes des composants pour chaque couche (cochez la case « Un fichier par face » de la carte).

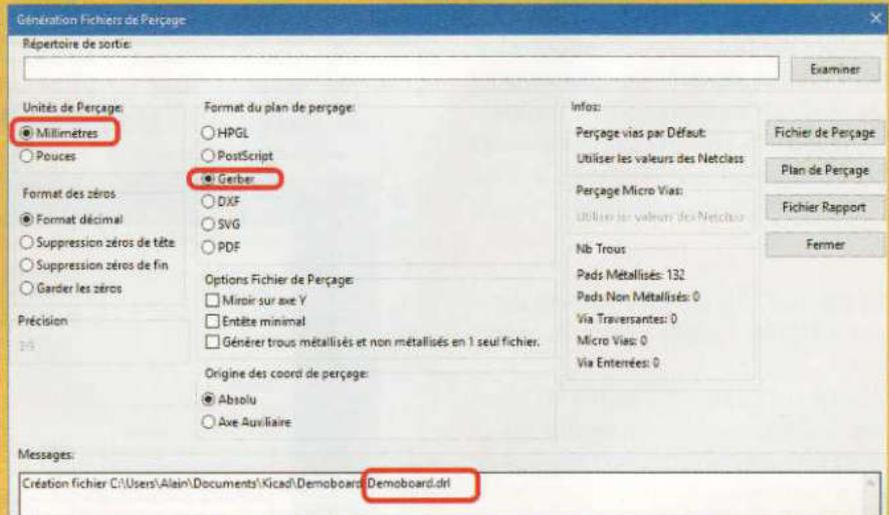


Figure 11 : le fichier « Demoboard.drl » a été généré, il correspond au perçage des composants, des vias et des fixations.

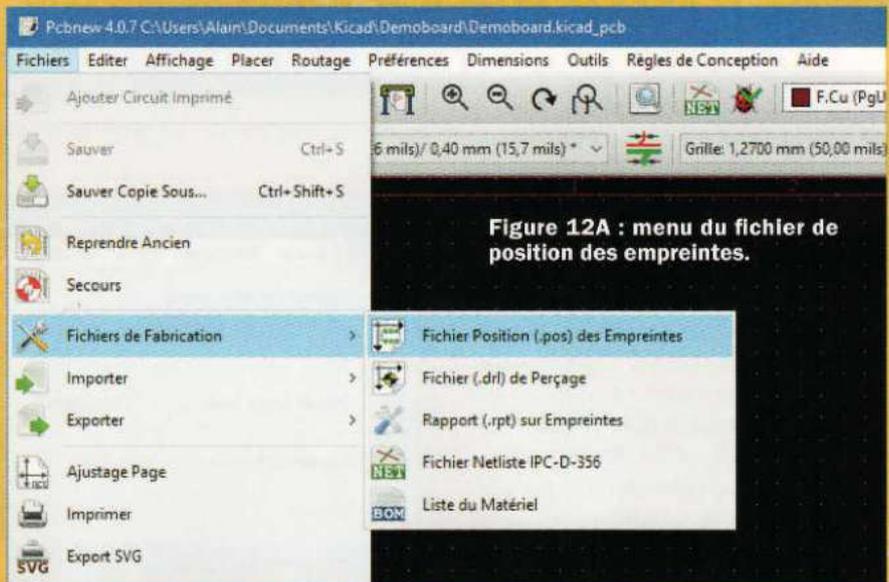


Figure 12A : menu du fichier de position des empreintes.

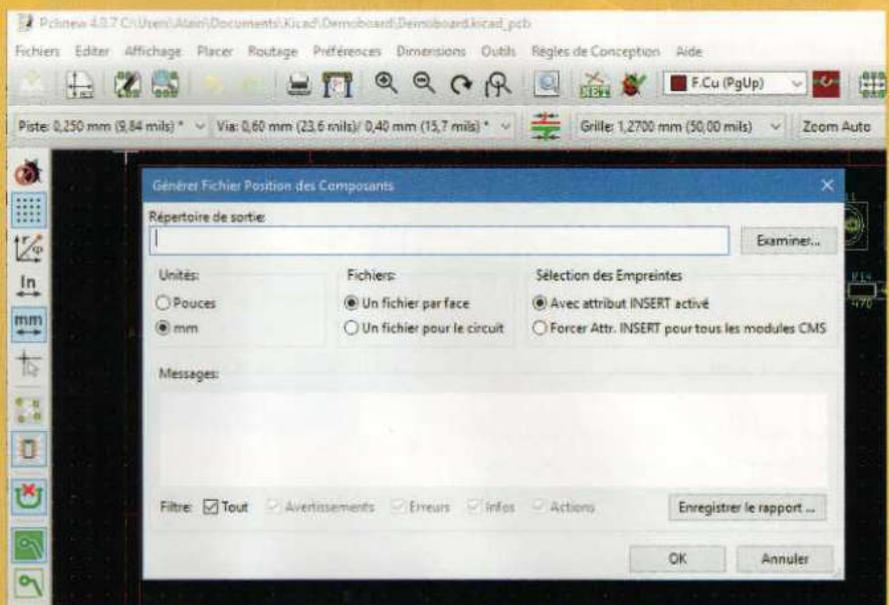


Figure 12 : la boîte de dialogue du fichier de position des empreintes.

Une fois les réglages terminés, les fichiers de position des empreintes peuvent être générés en cliquant simplement sur le bouton « OK ». La figure 13 montre le résultat obtenu dans la fenêtre « Messages » en bas de la boîte de dialogue (partie entourée en rouge).

Création de la nomenclature ou « BOM » (Bill of materials)

Une fois la réalisation de notre circuit imprimé terminée, nous devons générer la liste des composants nécessaires à sa fabrication afin de les acheter auprès d'un fournisseur de composants électroniques. KiCad nous offre la possibilité de générer la nomenclature des composants ou « BOM » (acronyme de Bill of materials) du projet.

Sa génération est très simple, il suffit de sélectionner le menu « Fichiers → Fichiers de Fabrication → Liste du Matériel » comme indiqué en figure 14.

Après avoir sélectionné cet élément du menu, une boîte de dialogue apparaît, vous permettant de sélectionner le dossier et le chemin dans lequel enregistrer la nomenclature sous la forme d'un fichier Excel (.csv). Le format d'exportation est donc un fichier « .csv » contenant les références des composants, la quantité, le type de boîtier et la désignation (valeur) attribuée lors de la conception schématique.

Visualisation des fichiers Gerber avec Gerberview

Analysons maintenant le dernier outil de la suite KiCad (sans considérer que le calculateur de PCB et le convertisseur Bitmap/Sikscreen fassent partie des outils, mais sont plutôt considérés comme des utilitaires), c'est-à-dire le visualisateur de fichiers gerber dénommé « Gerberview ».

Comme son nom l'indique, il s'agit d'un simple visualisateur qui nous permet de vérifier que les fichiers gerber générés sont corrects et, par conséquent, d'éviter d'envoyer des fichiers erronés qui produiraient des PCB incorrects, avec une perte de temps et d'argent.

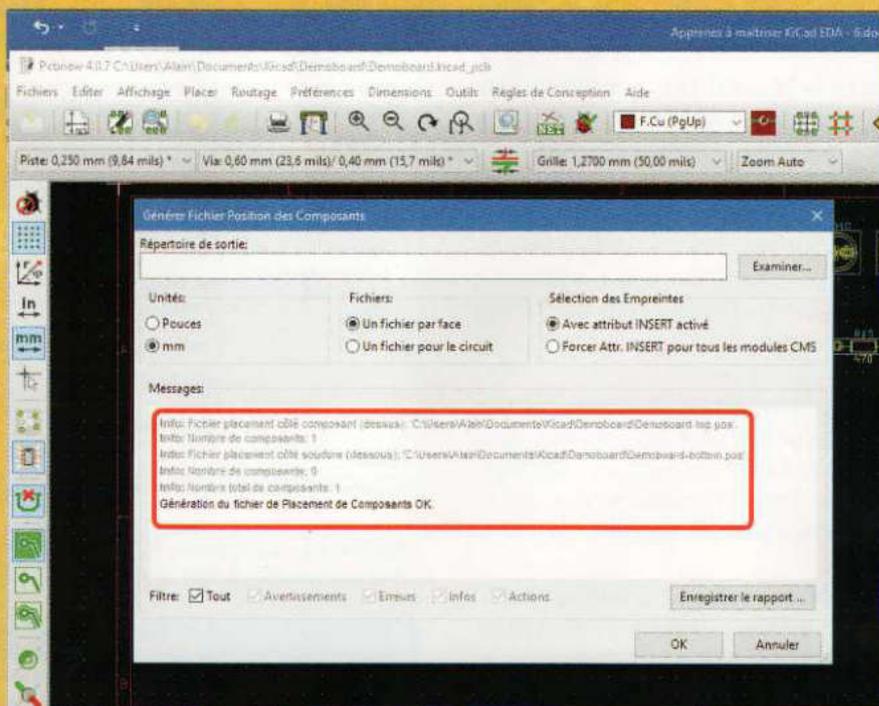


Figure 13 : génération du fichier de position des empreintes.

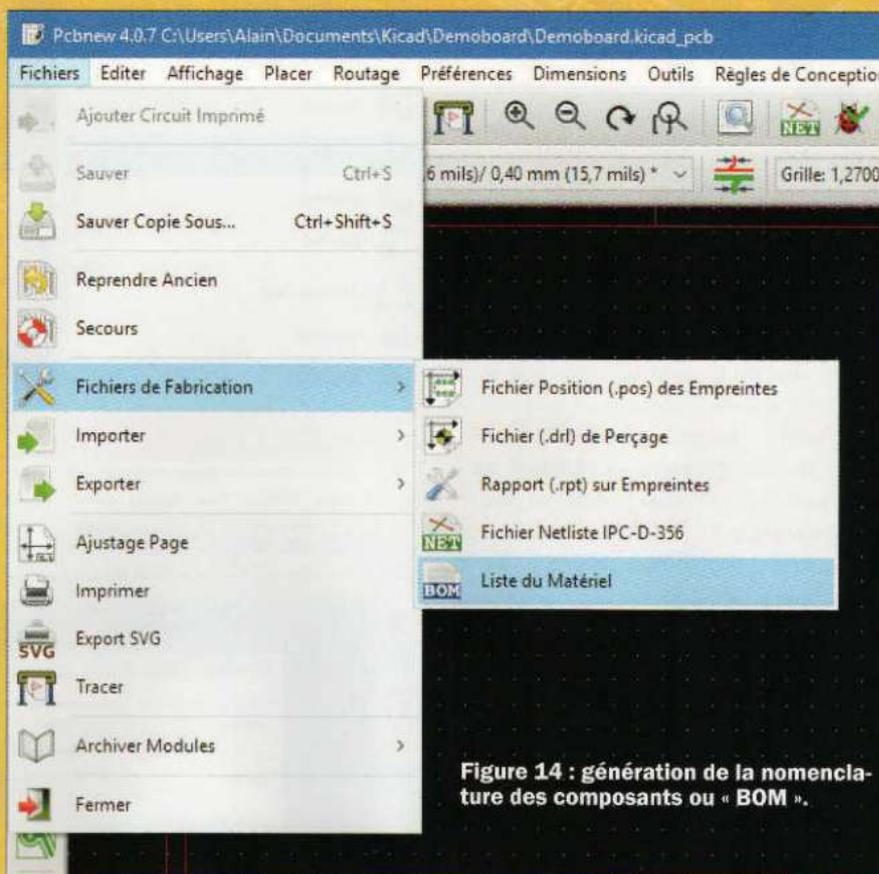
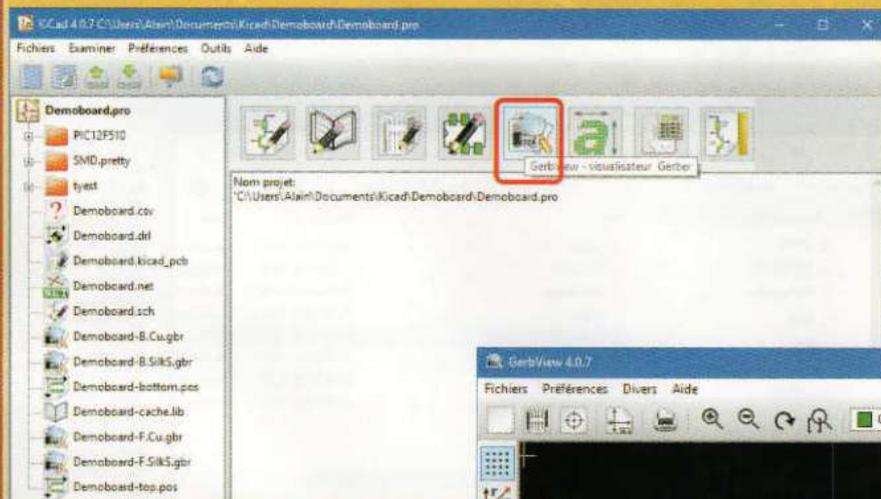


Figure 14 : génération de la nomenclature des composants ou « BOM ».

Pour lancer « Gerberview », il suffit de cliquer sur l'icône comportant une loupe (la 5^{ème} en partant de la gauche) dans la barre d'outils de KiCad, comme indiqué en figure 15. Une fenêtre s'ouvre, ressemblant à celle de la figure

16, avec un menu permettant de charger des fichiers gerber et des fichiers de perçage. Vous avez la possibilité de voir les différentes couches, d'agrandir, d'imprimer, etc. Sur la gauche de la fenêtre se trouve une barre d'outils

Figure 15 : l'icône « Gerberview » du visualisateur de fichiers Gerber.



contenant des options d'affichage, sur la droite vous pouvez sélectionner les calques à afficher.

Pour charger un fichier Gerber, sélectionnez le menu « Fichiers → Charger Fichier Gerber » (voir la figure 16A).

Une boîte de dialogue s'ouvre alors et vous demande de sélectionner le fichier à visualiser (voir la figure 17).

Un exemple de visualisation d'un fichier Gerber est illustré en figure 18, il s'agit de la face en cuivre de notre projet Demoboard. Comme nous l'avons noté au début de ce chapitre, la tâche principale de « Gerberview » est l'inspection des fichiers gerber générés.

Par exemple, il est possible de vérifier que le trou de perçage d'un pad d'un composant se trouve bien au centre et à l'intérieur du pad lui-même, c'est-à-dire que le trou de perçage se situe bien à l'intérieur de la pastille en cuivre et que ce dernier n'est pas plus grand (diamètre supérieur) que la partie de la pastille en cuivre.

Pour cela, une fois qu'un fichier Gerber a été chargé (comme nous venons de le voir), il est possible de superposer un fichier de perçage, en le chargeant à l'aide du menu « Fichiers → Charger Fichier de Perçage EXCELLON ».

Le résultat de ce type d'opération est présenté dans les détails en figure 19.

Une fois la vérification des fichiers Gerber effectuée, vous pouvez envoyer le

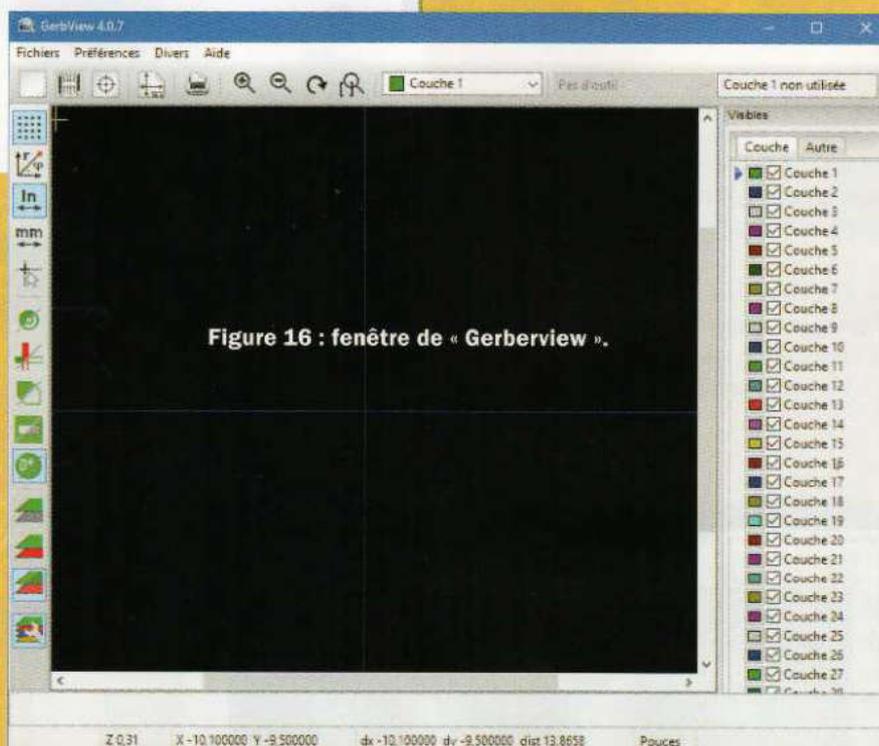


Figure 16 : fenêtre de « Gerberview ».

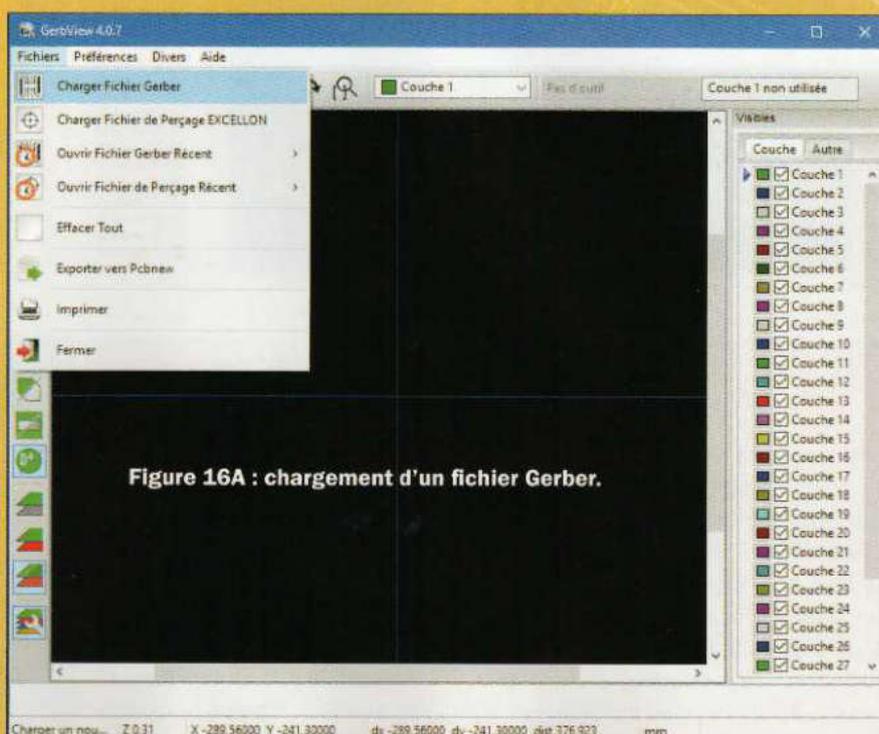


Figure 16A : chargement d'un fichier Gerber.

circuit imprimé en fabrication à l'entreprise de votre choix.

Comme vous pouvez le constater, les trous de perçage sont parfaitement alignés sur les pastilles des composants, et nous sommes donc sûrs que la fabrication du circuit imprimé se déroulera correctement.

Conclusion

Avec cette dernière leçon, nous achevons notre voyage à la découverte de KiCad, en complétant ainsi l'aperçu général de Pcbnew et en présentant les derniers outils de la suite.

Le projet KiCad, qui nous a permis d'avoir un logiciel de CAO précieux pour la conception des circuits imprimés, couvre toute la chaîne de production.

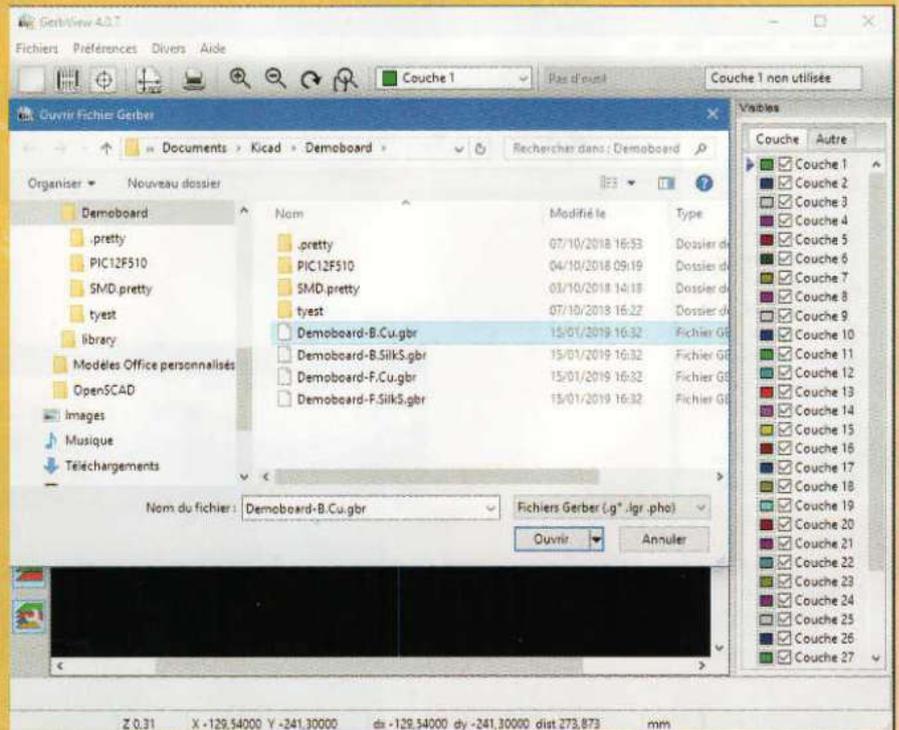


Figure 17 : sélection d'un fichier Gerber à charger dans « Gerberview ».

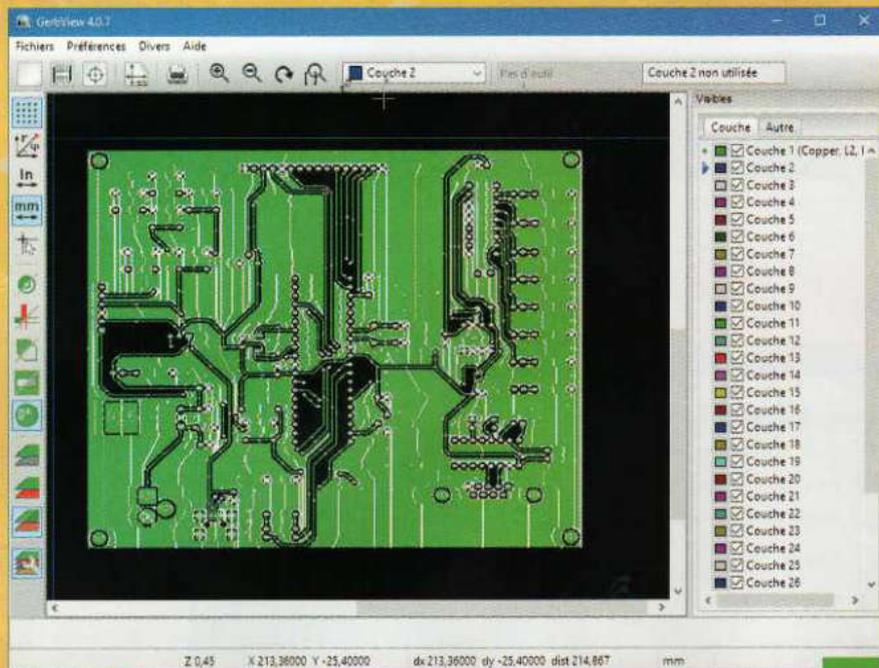
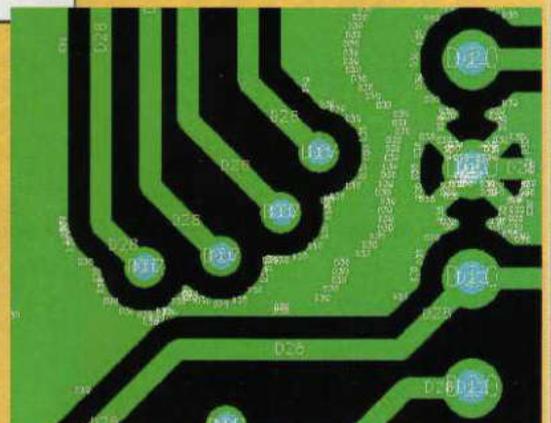


Figure 18 : visualisation du fichier Gerber de notre projet dans « Gerberview ».

Figure 19 : superposition du fichier de perçage avec le fichier Gerber.

Une nouvelle version (5), en cours de développement, est disponible actuellement apportant encore des améliorations. Nous espérons et souhaitons à l'équipe de développement, qui a conçu cet excellent logiciel Open Source de CAO électronique, de continuer dans cette direction en nous proposant à l'avenir des mises à jour et des futures versions.

N'oublions pas de citer **Jean-Pierre Charras** qui a développé en 1992 la première version de KiCad, ainsi que le **CERN** (<https://home.cern/fr>) qui a largement contribué au développement de KiCad en perfectionnant la partie graphique et en introduisant des capacités de routage automatique.



Calculateur de PCB

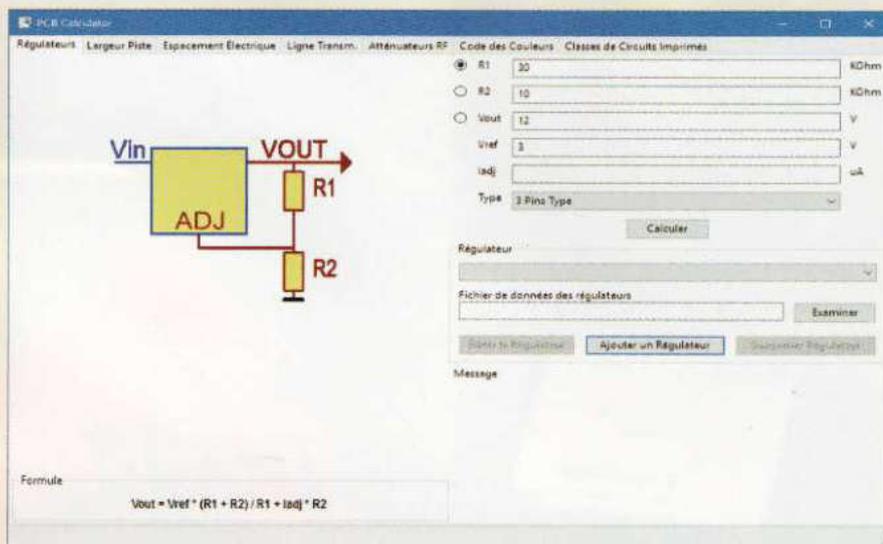


Figure 20 : calcul du diviseur de tension d'un régulateur linéaire.

Le calculateur de PCB est un utilitaire disponible dans la suite KiCad.

Ce dernier n'est rien de plus qu'un ensemble d'outils de calcul pour circuits imprimés, tous regroupés dans un seul et même utilitaire. Il permet de :

- calculer les valeurs du diviseur de tension pour des alimentations à régulateurs linéaires (représenté en figure 20) ;
- calculer la largeur des pistes de cuivre en fonction du courant les traversant ;
- calculer les valeurs de l'isolation électrique en fonction de la tension ;
- calculer les paramètres des lignes de transmission ;
- calculer les valeurs des résistances d'un atténuateur RF (représenté en figure 21) ;
- déterminer la valeur d'une résistance en fonction du code des couleurs (représenté en figure 22).

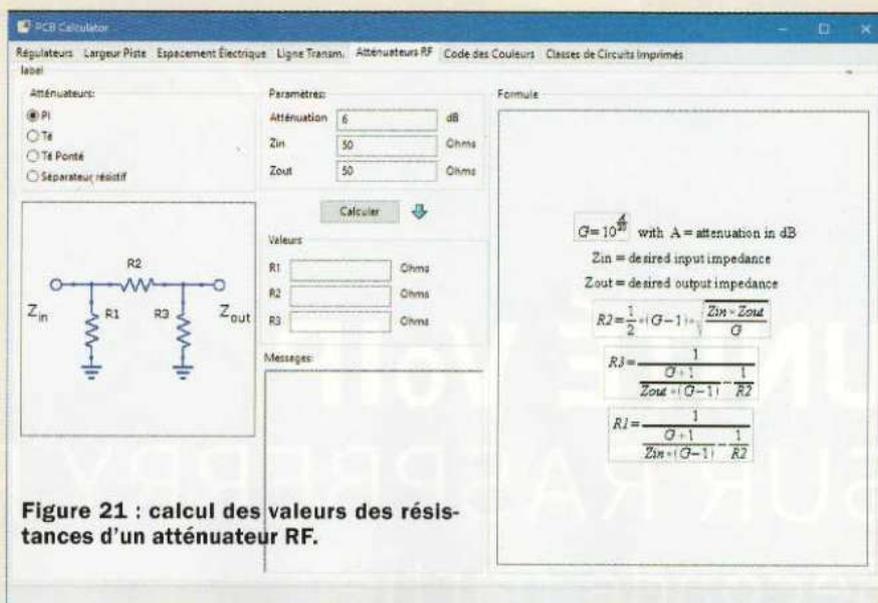


Figure 21 : calcul des valeurs des résistances d'un atténuateur RF.



Figure 22 : code des couleurs pour les résistances.

Nous mettons en pratique les notions de téléphonie sur « IP » que nous avons abordées dans le précédent numéro 145, en créant notre propre standard téléphonique complet doté de toutes les fonctionnalités les plus avancées. Seconde et dernière partie.



UNITÉ VoIP SUR RASPBERRY Pi Seconde partie

de Giuseppe Mazzucato

Le RaspberryPi est un mini-ordinateur que nous avons déjà utilisé dans diverses applications. Profitant de sa prédisposition à fonctionner comme un petit serveur Linux, nous avons créé un central téléphonique VoIP, que nous avons commencé à décrire dans le précédent numéro. Nous avons abordé les bases de la téléphonie sur « IP » pour ceux qui ne possèdent pas les connaissances suffisantes sur le sujet. Ensuite, nous avons expliqué quel matériel et logiciel utiliser pour réaliser le central téléphonique, en précisant que nous utilisons le logiciel Asterisk qui est Open Source.

Jusqu'à présent, nous avons étudié la configuration de base de notre central téléphonique, qui permettait de commencer à l'utiliser dans la pratique. Dans cette seconde et dernière partie, nous allons étudier la configuration étendue de manière

à obtenir des fonctions avancées avec lesquelles, par exemple, nous allons implémenter un service de sélection des appels entrants.

L'un des services les plus utiles pour le central téléphonique est celui qui permet l'acheminement des appels entrants vers le poste interne correspondant. Ce service vise à mettre en contact rapidement ceux qui appellent avec l'interlocuteur le plus approprié au sein d'une entreprise par exemple.

Service de réception et de tri des appels

Nous illustrerons les différentes fonctions à l'aide d'un exemple, décrit par l'organigramme visible en figure 1.

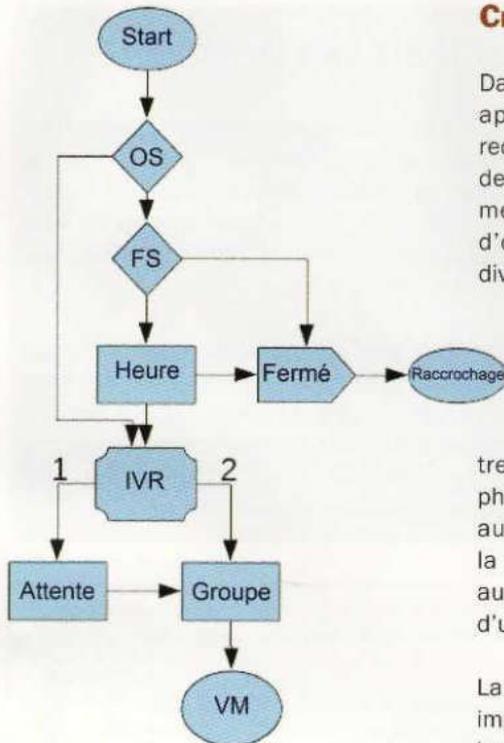


Figure 1 : accueil du central téléphonique.

Ce dernier peut être facilement adapté aux besoins de n'importe quelle petite ou moyenne entreprise.

La partie centrale de l'exemple est constituée d'un « IVR » (Réponse Vocale Interactive) qui achemine les appels vers différentes zones en fonction du choix de l'appelant, selon le classique « Appuyez sur la touche 1 pour l'assistance » ou « Appuyez sur la touche 2 pour le service commercial », etc.

L'accès à l'IVR est limité par le contrôle qui, en détectant l'heure actuelle, comprend qu'il se trouve dans la zone horaire de travail. Si cela n'est pas le cas, il lit un message de courtoisie à l'appelant.

Lors de la création d'un service de tri d'appels, nous vous recommandons de toujours dessiner un organigramme qui servira de guide pour la configuration du central téléphonique.

Il est également pratique de créer la configuration à partir de la fin. En effet, chaque fois que nous configurons un bloc, le bloc suivant doit figurer dans l'organigramme.

Création d'un fichier audio

Dans chaque service de réception des appels, des messages vocaux sont requis, lesquels sont diffusés pour guider l'appelant dans la navigation des menus, ou pour indiquer les heures d'ouverture ou encore pour fournir divers types d'informations de support.

Les messages sont créés à partir du menu « **Administration** → **Enregistrements système** », à partir duquel vous pouvez enregistrer des messages à l'aide d'un téléphone interne ou charger des fichiers audio préalablement préparés (voir la figure 2). Par exemple, des fichiers audio enregistrés avec un PC équipé d'un microphone.

La qualité de l'enregistrement est très importante pour que l'appelant ait une impression positive. Nous vous recommandons par conséquent d'utiliser une voix claire, voire de vous adresser à des intervenants professionnels pour enregistrer les messages.

Une alternative gratuite est d'utiliser la synthèse vocale de l'un des nombreux services disponibles sur Internet. Par exemple, le service de traduction et de prononciation de Google qui permet de mettre en œuvre des messages.

La synthèse vocale a atteint, ces dernières années, une qualité qui, sans être comparable à celle d'un orateur professionnel, peut-être de meilleure qualité que celle de nombreuses personnes.

Groupes d'appels

Depuis le module accessible à partir du menu « **Applications** → **Groupes d'appels** », il est possible de définir des groupes de postes internes auxquels des appels entrants doivent être adressés (voir la figure 3).

En plus de la liste des postes internes faisant partie du groupe, il est possible de définir la stratégie permettant de les faire sonner. Il existe différentes stratégies, mais la plus utilisée est la stratégie « ringall », qui fait sonner tous les postes internes appartenant à un groupe.

Il existe aussi la stratégie de « line hunting », c'est-à-dire la recherche de ligne (ou groupe de recherche). Il s'agit de la méthode de distribution des appels téléphoniques d'un seul numéro entrant vers un groupe de plusieurs lignes téléphoniques.

Pour résumer, la stratégie consiste à faire sonner les postes internes disponibles les uns après les autres.

Il est également possible de définir une durée maximale de sonnerie, de sorte que, si personne ne répond dans la limite de temps définie, la gestion de l'appel évolue vers l'étape suivante. Nous avons utilisé cette fonction pour rediriger l'appel vers un répondeur.

Files d'attente

Les files d'attente sont configurables à partir du menu « **Applications** → **Files** ».

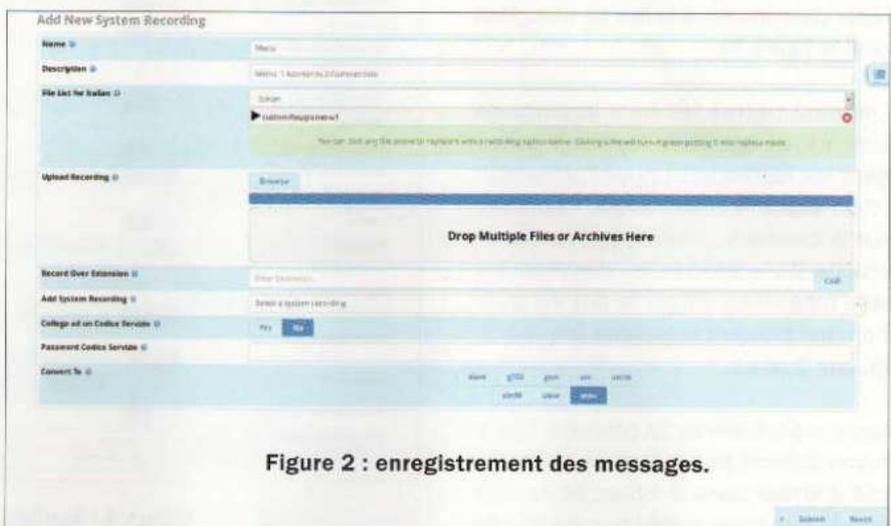


Figure 2 : enregistrement des messages.

Elles peuvent être considérées comme des extensions des groupes d'appels, auxquelles elles ajoutent de nombreuses fonctions que l'on trouve habituellement dans les centres d'appels professionnels (voir la figure 4).

La différence fondamentale par rapport au groupe est que la file d'attente conserve l'ordre des appels entrants, garantissant ainsi que le premier appelant dans l'ordre chronologique est aiguillé vers le premier opérateur à disposition.

En outre, il est également possible d'avoir des agents dynamiques vers lesquels les appels sont redirigés uniquement lorsqu'ils sont connectés à la file d'attente.

Ensuite, de nombreux autres paramètres de configuration vous permettent de choisir le comportement souhaité en cas de surcharge de la file d'attente. Par exemple, en n'acceptant plus d'appels ou en les transférant vers d'autres services.

Parmi les fonctions les plus sophistiquées présentes dans le module, nous mentionnons la possibilité de communiquer la position dans la file d'attente et le temps d'attente estimé pour les utilisateurs en attente.

IVR : Réponse Vocale Interactive (Interactive Voice Response)

Ce bloc fonctionnel, doté de sorties multiples, permet le choix entre différents chemins dans le flux de réception (voir la figure 5).

Les paramètres les plus importants sont le message audio reproduit, contenant les instructions pour l'utilisateur (c'est-à-dire le menu vocal, « appuyez sur la touche 1... » ou « appuyez sur la touche 2... », etc.) et les destinations vers lesquelles diriger le flux en fonction des touches appuyées (touche 1, touche 2, etc.).

Les comportements du central téléphonique doivent être prévus, à la fois en cas d'erreur dans le choix, et dans le cas où rien n'est entré (aucune touche

GNR et sélection de passage

Les standards téléphoniques sont souvent connectés à des lignes appelées GNR (groupe de numérotation réduit), qui sont identifiées par un numéro court suivi de chiffres supplémentaires pour identifier des postes individuels. Il existe 10 numéros GNR pour lesquels l'extension comporte un seul chiffre, 100 numéros GNR pour deux chiffres et 1000 numéros GNR pour une extension à 3 chiffres.

Dans ces cas, il existe une fonction appelée « sélection de passage » pour laquelle le numéro de poste interne peut être atteint de l'extérieur en entrant le numéro de l'extension suivi du numéro de poste. Cette sélection est très utile car elle accélère l'accès aux personnes dont le numéro interne est connu et peut être simulée même sans GNR, à l'aide d'un IVR, en vous obligeant à saisir le numéro de poste si vous le connaissez.

Figure 3 : configuration des groupes d'appels.

Figure 4 : configuration de la file d'attente.

Figure 5 : Réponse Vocale Interactive ou IVR.

IVR General Options

IVR Name: Menu

IVR Description: Menu principale

IVR DTMF Options

Annuncio: Menu

Attiva Chiamata Diretta: Disattivato

Timeout: 10

Alert info: Double-Click to see options or help screens

Invalid Retries: 3

Invalid Retry Recording: Default

Append Announcement to invalid: Yes No

Return on invalid: Yes No

Invalid Recording: Default

Invalid Destination: Code
401 Assistenza

Timeout Retries: 3

Timeout retry Recording: Default

Append Announcement on Timeout: Yes No

Return on Timeout: Yes No

Timeout Recording: Default

Timeout Destination: Code
401 Assistenza

Return to IVR after VM: Yes No

IVR Entries

Digits	Destination	Return	Elimina
1	Code 401 Assistenza	Yes No	
2	Gruppi di chiamata 400 Gruppo Commerciale	Yes No	
digit pressed	no choice one no	Yes No	

Submit Reset Elimina

appuyée pendant le menu vocal) pendant une période de temps spécifiée.

Une fonction intéressante est celle qui permet d'activer l'appel direct vers des postes internes, c'est-à-dire celle qui simule la « sélection de passage » des centraux téléphoniques traditionnels (cela revient à définir un poste

interne vers lequel un appel doit arriver directement en ajoutant le numéro du poste).

Les annonces

Le module d'annonce est l'un des plus simples et vous permet de lire un

fichier audio avant de passer à l'étape suivante (voir la figure 6).

Pour sa configuration, il suffit de choisir le fichier audio à lire et ensuite vers quel bloc aller à la fin de la lecture.

Dans notre exemple, il est utilisé en cas de fermeture de l'entreprise pour

Figure 6 : configuration des annonces.

Announcement: Edit

Descrizione: Chiusura

Registrazione: Chiusura

Ripeti: Disattivato

Permetti Salta: Yes No

Ritorna all'IVR: Yes No

Non Rispondere al Canale: Yes No

Destination after Playback: Termina Chiamata
Hangup

informer les appelants des heures d'ouverture du secrétariat avant de raccrocher.

Conditions horaires

Les conditions horaires sont des blocs avec deux sorties qui permettent au flux d'appels d'évoluer différemment selon l'heure et le jour.

L'application la plus répandue est celle qui permet de paramétrer la fermeture et l'ouverture du secrétariat, mais elle peut également être utile dans de nombreux autres cas (voir la figure 7).

Nous devons d'abord définir la période à l'aide du menu « **Applications** → **Groupes temporels** », en définissant la période comme étant l'union d'un ensemble de tranches horaires et journalières. Le système permet de définir, en plus des horaires, les jours de la semaine et du mois.

Il est ainsi possible de paramétrer non seulement les heures d'ouverture, mais également les périodes de fermeture pour les week-ends et/ou pour les vacances.

Après avoir défini les tranches horaires, vous pouvez créer le bloc fonction en accédant au menu « **Applications** → **Conditions Temporelles** » (voir la figure 8).

À ce bloc doit être attribué un groupe temporel et la configuration des destinations des deux sorties : une si l'heure est dans la période et l'autre si elle se situe en dehors de la période.

Les contrôles horaires sont très pratiques car ils modifient automatiquement le chemin de l'appel entrant sans aucune intervention humaine. Cependant, dans certains cas, il sera nécessaire de gérer des exceptions pour lesquelles il sera utile de modifier manuellement le flux via un module que nous verrons ci-après.

Contrôles de flux

Nous pouvons imaginer les contrôles de flux comme étant des aiguillages

Figure 7 : configuration des groupes temporels.

Figure 8 : configuration des conditions horaires.

d'une installation de chemin de fer, qui ont une entrée, deux sorties et une commande permettant de connecter l'entrée à l'une des deux sorties.

Dans le cas du central téléphonique, la commande est donnée par un code de fonction qui est entré dans l'un des téléphones internes.

Dans notre exemple, nous avons utilisé deux contrôles de flux que nous avons appelés « **Ouverture Spéciale** » (OS) et « **Fermeture Spéciale** » (FS) et nous devons forcer l'état d'ouverture et de fermeture en contournant le contrôle horaire (voir la figure 9).

Ces contrôles sont utiles pour gérer les exceptions pendant les heures d'ouverture et de fermeture.

Les contrôles de flux peuvent être créés et modifiés à partir du menu « **Applications** → **Call Flow Control** ».

Il faut alors définir les destinations des deux sorties et la valeur du code de fonction ainsi que l'état initial.

Le code de fonction peut également être accompagné d'un mot de passe pour empêcher l'activation du contrôle par du personnel non autorisé.

Pour compléter l'exemple, n'oublions pas de modifier l'itinéraire d'entrée créé comme expliqué dans la première partie (à laquelle nous nous référons).

Cela permet de gérer les appels entrants provenant du service de réception nouvellement créé.

L'ARCEP ou l'autorité de régulation des communications électroniques et des postes

L'ARCEP est une autorité administrative indépendante chargée de réguler les communications électroniques et les postes en France. Elle est chargée d'accompagner l'ouverture à la concurrence du secteur des télécommunications, et de réguler les marchés correspondants. Dans ce secteur d'activité, le rôle essentiel de l'Autorité est de veiller à l'exercice d'une concurrence effective et loyale au bénéfice des consommateurs sur le marché des communications électroniques.

L'Autorité peut :

- fixer, dans le cadre légal, des obligations générales s'appliquant à tous les opérateurs, sous réserve qu'elles soient homologuées par le ministre chargé des télécommunications. C'est le principe de la régulation dite « symétrique », qui s'impose de la même manière à chaque opérateur sur le marché, à l'exemple de la « portabilité mobile » (conservation du numéro lors du passage d'un opérateur mobile à un autre) ;
- sanctionner des opérateurs ne remplissant pas leurs obligations, et intervenir pour régler les différends entre opérateurs en matière d'accès au réseau (conditions techniques et tarifaires) ;
- attribuer des ressources en fréquences et en numérotation ; les opérateurs ont besoin de ces ressources dites « rares » car naturellement limitées, pour mener à bien leurs activités.

Les autres modules utiles de FreePBX

Dans la construction de notre exemple, nous avons inséré les modules FreePBX les plus utilisés. Cependant, il en existe de nombreux autres, gratuits et payants, qui pourraient être intéressants. Voici ci-après une liste partielle des modules que nous considérons comme pertinents.

Le module « **Répertoire** » permet de mémoriser les numéros de téléphone fréquemment utilisés et d'attribuer des numéros courts (speed dial).

Le module « **Codes de service** » permet de modifier les codes permettant d'obtenir certaines fonctions avancées du central téléphonique.

À partir du module « **Admin** », vous pouvez mettre à jour, installer et supprimer des modules FreePBX en les téléchargeant, attention certains sont payants.

Le module « **Conférence** » permet la création de salles de conférence audio virtuelles utiles pour les réunions avec des participants répartis dans plusieurs endroits.

Avec le module « **Follow Me** » (sui-vez-moi), vous pouvez paramétrer la gestion des appels lorsque vous ne pouvez pas répondre, par exemple en transférant l'appel sur le téléphone mobile.

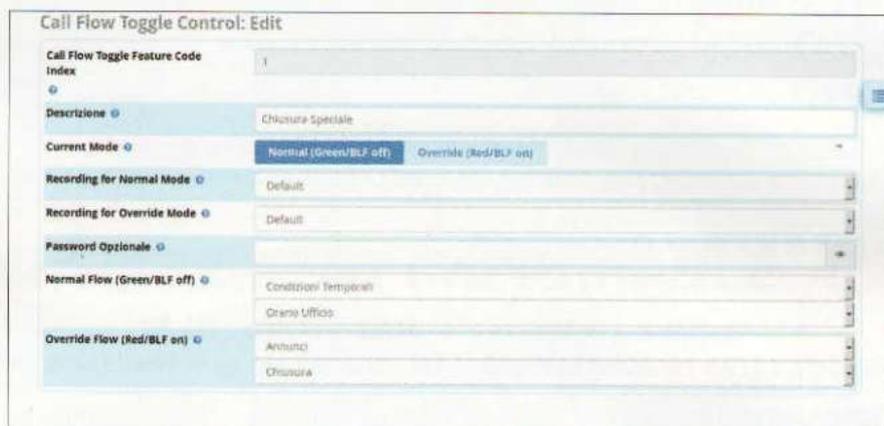


Figure 9 : les contrôles de flux.

Le module « **Dashboard** » (Tableau de bord), qui apparaît par défaut lors de la connexion de l'administrateur, contient les informations sur l'état du central et les notifications éventuelles.

Avec le module « **CDR Reports** », il est possible d'analyser la base de données des appels entrants et sortants.

Conclusion

Dans cette série de deux articles, nous avons illustré la mise en œuvre d'un central téléphonique en utilisant des outils gratuits à la disposition de tous ceux qui souhaitent s'essayer à cette activité.

Le sujet est complexe et, dans l'espace dont nous disposons, nous n'avons pu illustrer que les principaux aspects,

mais ce que nous avons expliqué devrait suffire à gérer les situations les plus courantes.

Jusqu'à il y a quelques années, la création d'un central téléphonique était réservée aux spécialistes qui utilisaient des équipements dédiés et donc très coûteux. La situation a complètement changé avec la disponibilité de centraux téléphoniques sous forme de logiciels libres qui permettent aujourd'hui de les implémenter avec du matériel courant et donc de faible coût.

Le principal avantage est celui de l'ouverture du projet (logiciels Open Source), cela nous libère des contraintes souvent imposées par les fabricants de produits propriétaires (et donc coûteux) et nous permet de développer des projets qui correspondent le mieux à nos besoins. ■

ABONNEZ-VOUS

OUI, Je m'abonne à

ELECTRONIQUE
ET LOISIRS
LE MENUEL DE L'ELECTRONIQUE POUR TOUS

A PARTIR DU N° 147 ou supérieur



N°

E0146

Ci-joint mon règlement de € correspondant à un abonnement de 4 revues Annuel

Règlement CB directement sur le site www.electroniquemagazine.com rubrique Abonnement

Adresser mon abonnement à :

Nom _____ Prénom _____

Adresse _____

Code postal _____ Ville _____

Tél. _____ e-mail _____

Date, le _____

Signature obligatoire ▷

TARIFS FRANCE

4 numéros 30€,00

TARIFS CEE/EUROPE

4 numéros 34€,00

DOM-TOM/HORS CEE OU EUROPE:

CONSULTEZ le site
www.electroniquemagazine.com
rubrique Abonnement

POUR TOUT CHANGEMENT D'ADRESSE,
N'OUBLIEZ PAS DE NOUS INDIQUER
VOTRE NUMÉRO D'ABONNÉ (INSCRIT
SUR L'EMBALLAGE)

L'ASSURANCE de ne manquer aucun numéro en recevant votre revue directement dans votre boîte aux lettres près d'une semaine avant sa sortie en kiosques.

BÉNÉFICIER de 50 % de remise sur les CD-ROM.

Bulletin à retourner à: JMJ - Abo. ELM

B.P. 20025 - 13720 LA BOUILLADISSE - Tél. +334 427 063 96 - Fax +334 427 063 95

Directeur de Publication

Rédacteur en chef

Jean Marc MOSCATI
CD908
13720 Belcodène

Direction - Administration

JMJ éditions
B.P. 20025
13720 LA BOUILLADISSE
Tél.: +334 427 063 96

Secrétariat - Abonnements

Petites-annonces - Ventes

À la revue

Vente au numéro

À la revue

Publicité

À la revue

Maquette - Illustration Composition - Photogravure

JMJ Editions SARL

Impression

Rotimpres
C/ Pla de l'Estany sn
17181 Aiguaviva (Girona)
Espagne

Distribution

MLP

55 Boulevard de la Noirée
38070 Saint-Quentin-Fallavier

Hot Line Technique

+334 427 063 96 non surtaxé
du lundi au vendredi de 15 h à 17 h

Web

www.electroniquemagazine.com

E-mail

support@electroniquemagazine.com

JMJ éditions

Sarl au capital social de 7800 €
RCS MARSEILLE : 421 860 925
APE 221E
Commission paritaire: 1221 K 79056
ISSN: 1295-9693
Dépôt légal à parution

EST RÉALISÉ
EN COLLABORATION AVEC :

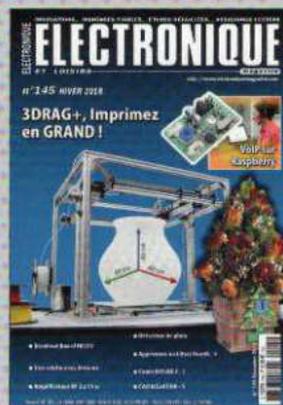
ELECTRONICA
Electronica In

I M P O R T A N T

Reproduction, totale ou partielle, par tous moyens et sur tous supports, y compris l'internet, interdite sans accord écrit de l'Editeur. Toute utilisation des articles de ce magazine à des fins de notice ou à des fins commerciales est soumise à autorisation écrite de l'Editeur. Toute utilisation non autorisée fera l'objet de poursuites. Les opinions exprimées ainsi que les articles n'engagent que la responsabilité de leurs auteurs et ne reflètent pas obligatoirement l'opinion de la rédaction. L'Editeur décline toute responsabilité quant à la teneur des annonces de publicités insérées dans le magazine et des transactions qui en découlent. L'Editeur se réserve le droit de refuser les annonces et publicités sans avoir à justifier ce refus. Les noms, prénoms et adresses de nos abonnés ne sont communiqués qu'aux services internes de la société, ainsi qu'aux organismes liés contractuellement pour le routage. Les informations peuvent faire l'objet d'un droit d'accès et de rectification dans le cadre légal.

13,58 €* la revue frais de port inclus pour la France Métropolitaine

* Frais d'expédition (CEE, les DOM-TOM et autres pays), contactez-nous pour un devis ou bien à calculer directement sur notre site



Au sommaire : Reconnaissance faciale avec RaspberryPi, nous allons utiliser les fonctionnalités de la librairie SimpleCV afin de reconnaître les visages et les autres détails du corps humain dans des images - Programmeur hebdomadaire, un temporisateur hebdomadaire avec une sortie relais, un clavier et un afficheur LCD pour les réglages des différentes fonctions - Radio FM & lecteur MP3 USB - Enregistreur vocal avec PIC & EEPROM - 1. Apprenez à maîtriser KiCad EDA-1 - Effet trémolo pour guitare, faire vibrer le son des cordes à travers un système optique original de modulation d'amplitude, Cours MPLAB X IDE-1. Étoile de Noël - Boule de Noël à changement de couleur

Au sommaire : Un amplificateur HI-FI 200 W à MOSFET, il pourra sonoriser correctement des pièces volumineuses sans la moindre distorsion - Chambre d'écho digitale - RandA : connectons RaspberryPi avec Arduino avec une carte de développement qui permet de créer une passerelle entre le monde du RaspberryPi et le monde d'Arduino - Enregistreur vocal avec PIC & EEPROM - 2. Cours MPLAB X IDE - 2 pour découvrir MPLABX IDE, le nouvel environnement de développement intégré de Microchip - Apprenez à maîtriser KiCad EDA - 2 - Comment ponter, un amplificateur - Contrôleur pour moteur DC - Télécommande infrarouge 4 canaux.

Au sommaire : Apprenons à utiliser RandA, programmation et l'utilisation de la carte - STOP au BLACKOUT !, système de mesure et de limitation automatique de la consommation électrique jusqu'à 6 kW sous 220 VAC - Chauffage par induction 1 kW, La technologie ZVS permet une régulation de la tension à l'aide d'une commutation douce - Détecteur de flamme - Enregistreur vocal avec PIC & EEPROM - 3, Cours MPLAB X IDE - 3, le nouvel environnement de développement intégré produit et distribué par Microchip Technology - Récepteur 1 & 2 canaux 433.92 MHz compatible MMS3200, UM86409 et UM3759, Chaque récepteur peut être combiné jusqu'à 10 télécommandes

Au sommaire : Jeu vidéo d'arcade avec RaspberryPi Demoboard MP3, testons l'énorme potentiel du module lecteur audio DFRO299 - Barrière laser, ce montage sert à détecter la présence - Pédale multi-effets, ce montage avec une pédale multi-effets proposant deux types de distorsions fuzz - STOP au BLACKOUT ! (II), cet électronique permet en cas de dépassement de la puissance maximale, de désactiver un ou plusieurs appareils - Générateur haute tension, modulaire - Apprenez à maîtriser KiCad EDA - 4, commençons par l'analyse de Pcbnew - Apprenons à utiliser RandA - 3 - Cours MPLAB X IDE - 4 - Indicateur d'état de la batterie

Au sommaire : Unité VoIP sur RaspberryPi - 1, nous vous proposons de réaliser un standard téléphonique doté de toutes les fonctionnalités les plus avancées, La 3DRAG+, une version améliorée capable d'imprimer en 3D des pièces de dimensions 400 mm x 400 mm x 400 mm - Cartes de prototypage NE555 - Cours MPLAB X IDE - 5 - Une crèche avec Arduino - Maîtriser KiCad EDA-5, Amplificateur BF 2 x 15 W à TDA7297 - Détecteur de pluie - Thermostat à microcontrôleur, testez ses fonctionnalités Apprenons à utiliser RandA - 4 qui sert de passerelle entre les deux mondes d'Arduino et du RaspberryPi - Un sapin de Noël électronique.

CD-ROM ENTIÈREMENT IMPRIMABLE

Nouveauté

79 à 84, 85 à 90, 91 à 96, 97 à 102, 103 à 108, 109 à 114, 115 à 120, 121 à 126, 127 à 132, 133 à 138, 139 à 144

Nouveauté

1 à 12, 13 à 24, 25 à 36, 37 à 48, 49 à 60, 61 à 72, 73 à 84, 85 à 96, 97 à 108, 109 à 120, 121 à 132, 133 à 144

CD 6 Numéros 25,76 €* / CD 12 Numéros 45,76 €*
50% de remise pour nos abonnés sur tous nos CD

* frais de port inclus pour la France Métropolitaine

JMJ EDITIONS - CD 908 - 13720 BELCODÈNE. Règlement par chèque à l'ordre de MJM ÉDITIONS et/ou règlement par Paypal ou CB sur notre site : www.electroniquemagazine.com - Tél. : 04 42 70 63 96

Analyseurs de semi-conducteurs Peak Atlas

DCA55 : L'analyseur de semi-conducteurs DCA55 permet d'identifier les transistors, les thyristors, les triacs, les diodes et les LED. Identification automatique des broches. Mesure du gain et du courant de fuite pour les transistors bipolaires. Mesure de la tension de seuil pour les MOSFET à enrichissement. Mesure de la tension directe pour les diodes, les LED et les jonctions base-émetteur des transistors. Alimentation : 1 pile 12 V GP23A (incluse). Dimensions : 103 x 70 x 20 mm

Transistor :

- plage du gain (hFE): 4 à 65000 - précision du gain: $\pm 5\%$ hfe
- tension maximale collecteur/émetteur: 2,0 V à 3,0 V
- tension base/émetteur VBE pour les Darlington : 0,95 V à 1,80 V

MOSFET :

- plage de la tension grille/source : 0,1 V à 5,0 V
- courant de drain : 2,45 mA à 255 mA
- courant de drain d'appauvrissement : 4,5 mA
- courant drain/source JFET: 0.5 mA à 5, 5mA

Thyristor/Triac :

- courant de gâchette : 4,5 mA - courant de maintien : 5,0 mA

Diode :

- courant de test : 5,0 mA
- tension directe pour identification des LED : 1,50 V à 4,00 V

Réf.: DCA55 Prix : 69,00€



DCA75 Pro : En plus des fonctionnalités du DCA55, le DCA75 pro permet d'identifier une vaste gamme de semi-conducteurs avec leurs brochages et mesurer leurs paramètres. L'instrument peut être utilisé de manière autonome ou branché à un PC pour identifier le type de composant, identifier le brochage et mesurer plusieurs paramètres des composants tels que le gain d'un transistor, les tensions de seuil des MOSFET et IGBT, les caractéristiques des jonctions, etc. Possibilité de mises à jour gratuites en ligne.

Fonctionnalités :

- Identification automatique du type de composant (par exemple, transistor, JFET, IGBT, régulateur, diodes zener, etc.)
- Identification automatique du brochage (connexion dans n'importe quel sens)
- Mesure du gain d'un transistor / Mesure du seuil de la grille d'un MOSFET
- Mesure de la transconductance des MOSFET et JFET / Mesures des caractéristiques des jonctions PN
- Mesure du courant de fuite
- Effectue une analyse du composant plus détaillée avec le logiciel PC inclus sur la clé USB Peak (convient de Windows XP à Windows 10)
- Mise sous tension et hors tension automatique / Conception compacte et ultramince
- Complet avec sondes pinces plaquées or premium / Possibilité de mise à jour en ligne

**Réf.: DCA75
Prix : 138,00€**

ZEN 50 : Analyseur de diodes ZEN 50

- analyse les diodes zener, VDR, TVS, LED (max 50 V) ;
- mesure la tension de claquage (de 0,00 V à 50,00 V) avec une résolution de 20 mV ;
- mesure la résistance équivalente (max 1 k Ω à 15 mA) ;
- courants de mesure sélectionnables : 2 mA, 5 mA, 10 mA, 15 mA ;
- augmentation progressive de la tension pour supporter des mesures jusqu'à 50 V à 15 mA ;
- fonction « HOLD » permettant de figer la valeur d'une mesure, même si le composant est retiré / pinces crocodiles à contacts dorés ;
- 1 x pile AAA alcaline.

**Réf.: ZEN50
Prix : 58,00€**

Impédancemètre LCR45 : Le LCR45 de Peak Electronics est un impédancemètre compact. Cet appareil permet de mesurer aussi bien les inductances, les condensateurs et les résistances. Le résultat s'affiche sur un écran alphanumérique. Lorsqu'un composant est connecté et que vous appuyez sur le bouton de test, l'appareil de mesure indique le type de composant concerné ainsi que ses caractéristiques.

- Fréquences de test : continue, 1 kHz, 15 kHz ou 200 kHz
- Plage de mesure de la résistance : 0 - 2 M Ω
- Précision de la résistance : $\pm 1,0\%$
- Plage de mesure de la capacité : à 10 000 μ F
- Précision de la capacité : $\pm 1,5\%$
- Plage de mesure de l'inductance : 0 à 10 H
- Précision de l'inductance : $\pm 1,5\%$
- Dimensions : 103 x 70 x 20 mm

Réf.: LCR45 Prix : 139,00€