

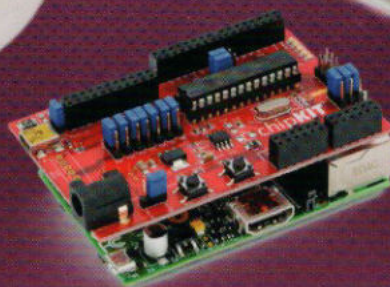
n°140 Automne 2017

Sustentation magnétique

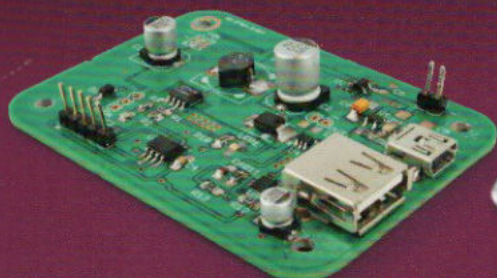
Générateur de bruit blanc et rose

Compteur d'énergie 220 V

- ChipKIT Pi
- Cours Arduino X
- Imprimante 3D bicolore VI
- Veilleuse LED 220 V et 12/24 V
- Programmez sous iPhone, iPad VI
- Vision numérique pour RaspberryPi



Chargeur smartphone intelligent



N° 140 Septembre 2017

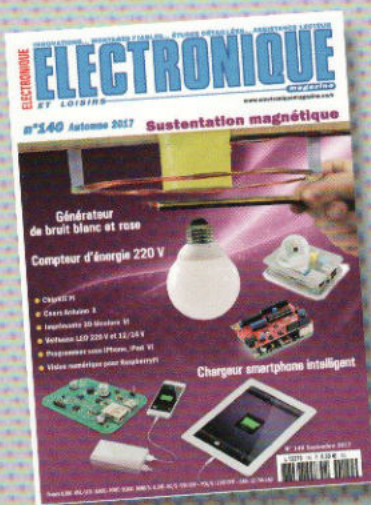
L 13270 - 140 - F: 8,30 € - RD



Sommaire

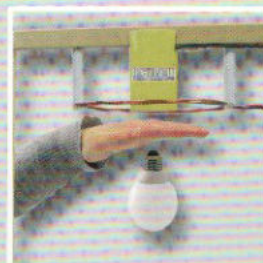
ARTICLES

Numéro 140
Automne 2017
100 pages



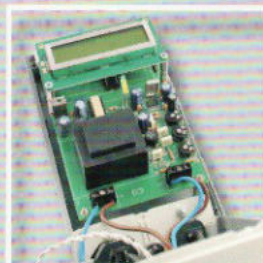
04 DIDACTIQUE **SUSTENTATION MAGNETIQUE**

Dans cet article, nous vous proposons une expérience didactique hors du commun. Il n'y a pas de truc, ni de tromperie ... juste une bonne combinaison de technologie, de maîtrise et d'imagination. Suspendre une ampoule en l'air sans la présence de fil pour la maintenir et de plus l'allumer sans aucune pile ou câble, mais comment est-ce possible ?



20 MAISON **COMPTEUR D'ÉNERGIE ÉLECTRIQUE 220 VAC**

En reliant en série ce compteur d'énergie à la ligne d'alimentation 220 VAC de votre maison ou de votre appartement, il devient possible de mesurer la puissance active, c'est-à-dire celle réellement consommée par les appareils. Ce compteur est doté d'un afficheur LCD qui visualise la consommation instantanée et la consommation globale.



35 LABORATOIRE **GÉNÉRATEUR DE BRUIT BLANC ET ROSE**

Conçu pour être utilisé lors des mesures avec l'analyseur de spectre décrit dans les précédents numéros 138 et 139, ce montage génère un bruit blanc et rose dont la fréquence de base peut atteindre 1 MHz. Il est capable de générer un bruit blanc ou un bruit rose (1/f) jusqu'à une fréquence de 1 MHz, ou encore un bruit blanc avec une bande passante de 20 kHz.



47 INFORMATIQUE **VISION NUMÉRIQUE AVEC RASPBERRYPI ET CAMÉRA PI**

Après avoir fait connaissance avec les différentes fonctionnalités du module Caméra Pi dans le numéro 137, qui est un module vidéo de qualité spécialement conçu pour le RaspberryPi, nous allons l'étudier pour acquérir les premières connaissances de la vision numérique afin de reconnaître des formes et des couleurs.



59 ALIMENTATION **CHARGEUR POUR SMARTPHONE INTELLIGENT**

Ce chargeur universel intelligent est facilement transportable, il est idéal pour faire fonctionner et recharger des smartphones ou d'autres dispositifs mobiles. Doté de connexions mini-USB et micro-USB, il s'adapte automatiquement au standard de recharge du dispositif auquel il est connecté.



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

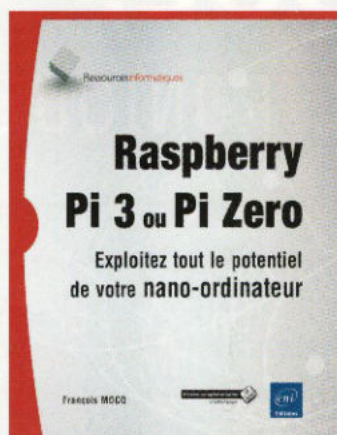
Chères lectrices, chers lecteurs, en cette période de rentrée, nous vous proposons de faire des économies avec un compteur d'énergie qui vous indiquera précisément la consommation de vos appareils. Les plus curieux pourront faire des expériences de lévitation magnétique grâce à l'article consacré à la sustentation magnétique avec une application concrète.

Dans la continuité de l'analyseur de spectre décrit dans les deux derniers numéros, nous vous proposons un appareil complémentaire qui est un générateur de bruit blanc et rose, et qui nous l'espérons complètera votre laboratoire.

Pour les fans du RaspberryPi, nous consacrons deux articles, un traitant d'une application concrète de vision numérique à l'aide de Caméra Pi et l'autre de la carte d'adaptation « chipKIT Pi » permettant de créer, compiler et programmer des applications Arduino sur la plate-forme RaspberryPi.

Dans ce numéro, nous arrivons à la fin des cours Arduino et iPhone, mais nous vous réservons une surprise dans le prochain numéro avec deux nouveaux cours, un consacré à la programmation en C des microcontrôleurs PIC32 et l'autre à l'utilisation d'un logiciel de CAO Open Source performant pour réaliser vos circuits imprimés.

La Rédaction



www.framboise314.fr

66 IMPRIMANTE 3D BICOLORE

3DVERTEX : LE LOGICIEL REPETIER-HOST

Dans le précédent numéro, nous avons abordé l'utilisation de l'imprimante 3DVERTEX en analysant les différents menus de son firmware. Nous avons appris à sélectionner le nombre d'extrudeuses, à reconnaître les différentes indications de l'afficheur LCD et enfin à charger et décharger le ou les filament(s). Nous allons maintenant étudier le logiciel de tranchage (Slicer) et d'interpréteur G-code qui se nomme Repetier-Host.



77 ARDUINO

COURS ARDUINO - X

Dans cette dixième partie du cours Arduino, nous allons apprendre à utiliser la carte « Ethernet Shield » qui vous permettra de connecter une carte Arduino sur un réseau intranet ou sur Internet. Nous allons développer un sketch orienté web à travers l'utilisation de la carte « Ethernet Shield ». Connecter une carte Arduino au web signifie l'accès à un grand nombre d'applications intéressantes.



82 INFORMATIQUE

CHIPKIT PI : COMMENT RENDRE LE RASPBERRYPI PLUS POLYVALENT

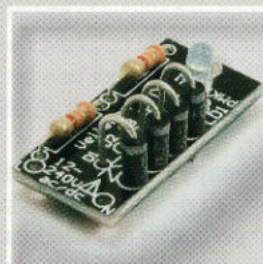
Voici une nouvelle carte de prototypage qui arrive dans le monde des produits embarqués afin de mettre en valeur les qualités du RaspberryPi et d'Arduino. Microchip a développé une carte de prototypage compatible avec le monde du RaspberryPi et d'Arduino. Découvrons-la !



86 GADGET

VEILLEUSE A LED UNIVERSELLE

Cette veilleuse à LED peut être utilisée comme un point lumineux pour remplacer une ampoule à incandescence classique fonctionnant sous 230 VAC ou 12/24VDC. Elle est alimentée directement à partir du secteur sans transformateur. Vous pouvez utiliser des LED de n'importe quelle couleur, cependant si vous devez vous en servir pour éclairer une zone, vous devrez utiliser des LED blanches.



89 IPHONE

COURS DE PROGRAMMATION SUR IPHONE : SIXIÈME PARTIE

Dans cette dernière leçon du cours consacré à l'iPhone, nous allons découvrir et apprendre à utiliser l'accéléromètre et le récepteur GPS qui équipent l'iPhone, au moyen de deux exemples d'applications. Avec cette vue d'ensemble des éléments clés d'iOS, nous concluons cette dernière partie de notre cours de programmation sur iPhone.



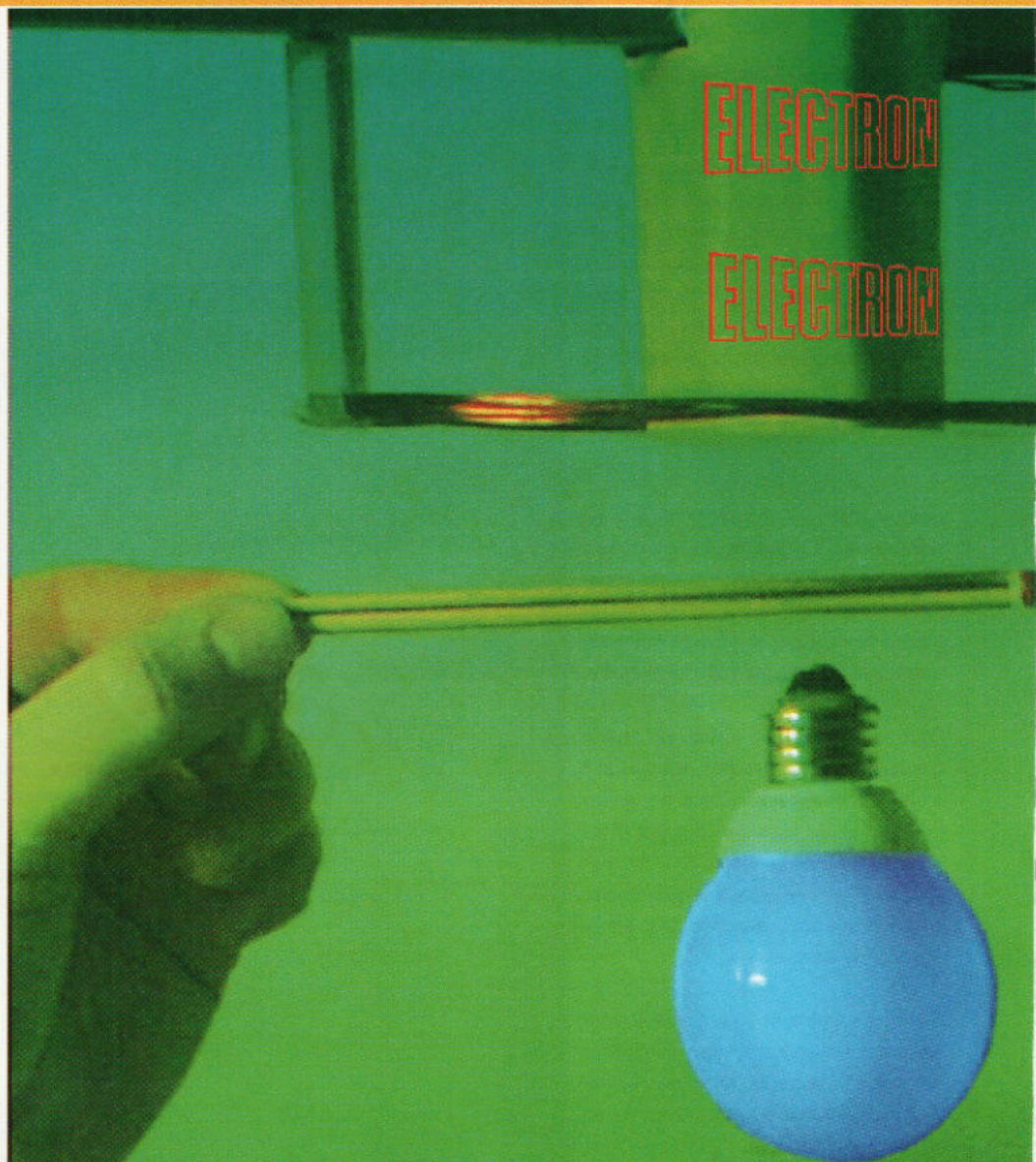
Vous êtes certainement resté dubitatif devant les performances étonnantes d'un magicien ou une scène d'un film de science-fiction dans lequel vous apercevez un objet en suspension dans l'air. Il ne s'agit pas d'un phénomène paranormal mais tout simplement l'effet de sustentation, à savoir la suspension d'objets de natures diverses à une certaine hauteur du sol. Nous pouvons résumer le phénomène par le fait qu'il s'agit de maintenir un objet à une faible distance au-dessus d'une surface sans qu'il soit en contact avec elle.

Lorsque nous voyons certains effets spéciaux, nous nous demandons comment ils sont réalisés. Étant modestement du milieu scientifique, nous sommes arrivés à la conclusion qu'à part les phénomènes paranormaux et les forces obscures, ces effets sont obtenus avec de la technologie et donc à l'aide d'une force qui n'est pas sombre mais plutôt naturelle.

Puisque la technique permet de suspendre un objet dans l'air, nous nous sommes dit : pourquoi ne pas le faire ? Pourquoi ne pas construire quelque chose et le proposer à nos lecteurs ? Après quelques semaines de tests, nous n'avons pas trouvé le secret de la lévitation, mais nous avons fabriqué un système qui permet d'obtenir la sustentation d'un objet.

Le résultat de notre travail est décrit en détail dans ces pages pour permettre à nos lecteurs qui le désirent de mettre en œuvre le système et l'utiliser pour impressionner des amis, en improvisant un numéro de prestidigitation ou tout simplement d'utiliser le gadget à la maison qui attirera l'attention de ceux qui le verront.

Ce que nous décrivons est la sustentation magnétique d'une ampoule à incandescence, qui, en plus de rester en suspension dans l'air sans fils, s'allume.

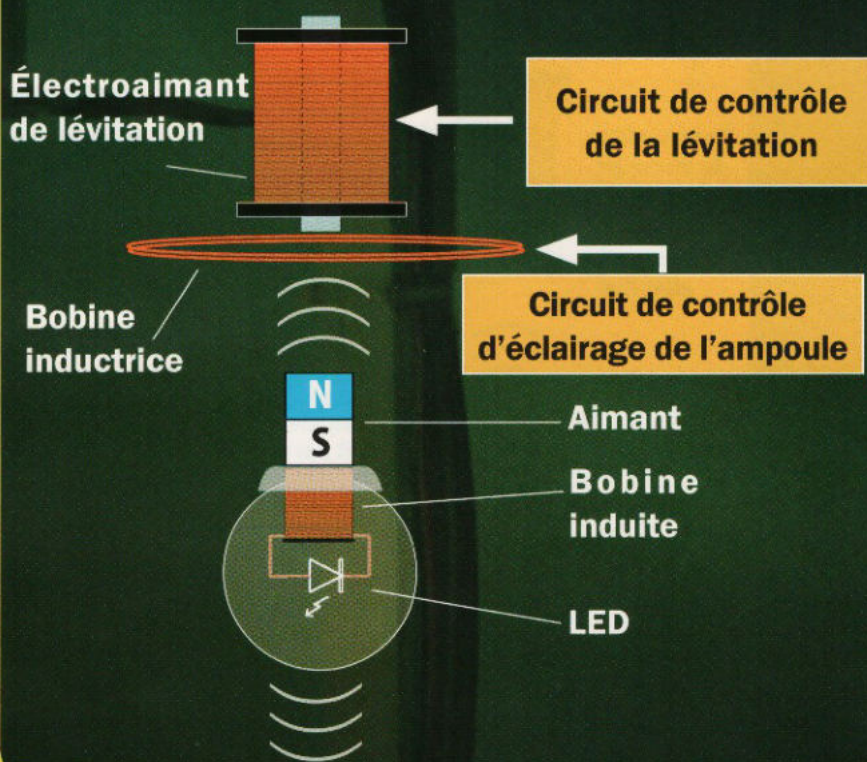


SUSTENTATION MAGNETIQUE, LA MAGIE DE L'ÉLECTRONIQUE

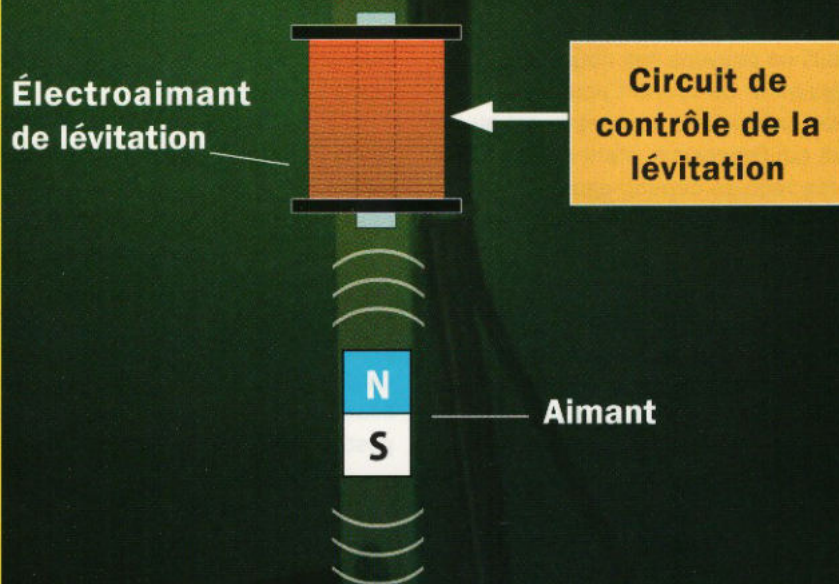
..... de BORIS LANDONI

Dans cet article, nous vous proposons une expérience didactique hors du commun. Il n'y a pas de truc, ni de tromperie ... juste une bonne combinaison de technologie, de maîtrise et d'imagination. Suspendre une ampoule en l'air sans la présence de fil pour la maintenir et de plus l'allumer sans aucune pile ou câble, mais comment est-ce possible ?

LÉVITATION ET ÉCLAIRAGE PAR INDUCTION



LÉVITATION



Rien ne vous empêche de personnaliser le système et de faire léviter une balle de tennis ou de ping-pong, ou d'autres objets, à condition qu'ils ne soient pas métalliques.

Nous avons construit notre système de sustentation magnétique en modifiant de manière appropriée une ampoule blanche, qui a été démontée, vidée, et équipée d'un aimant, d'un petit circuit électronique et d'une LED pour l'éclairage.

Attention, pour réaliser cette opération nous vous conseillons d'utiliser des gants épais pour travaux, ne le faites pas à mains nues car il y a des risques de coupures !!!

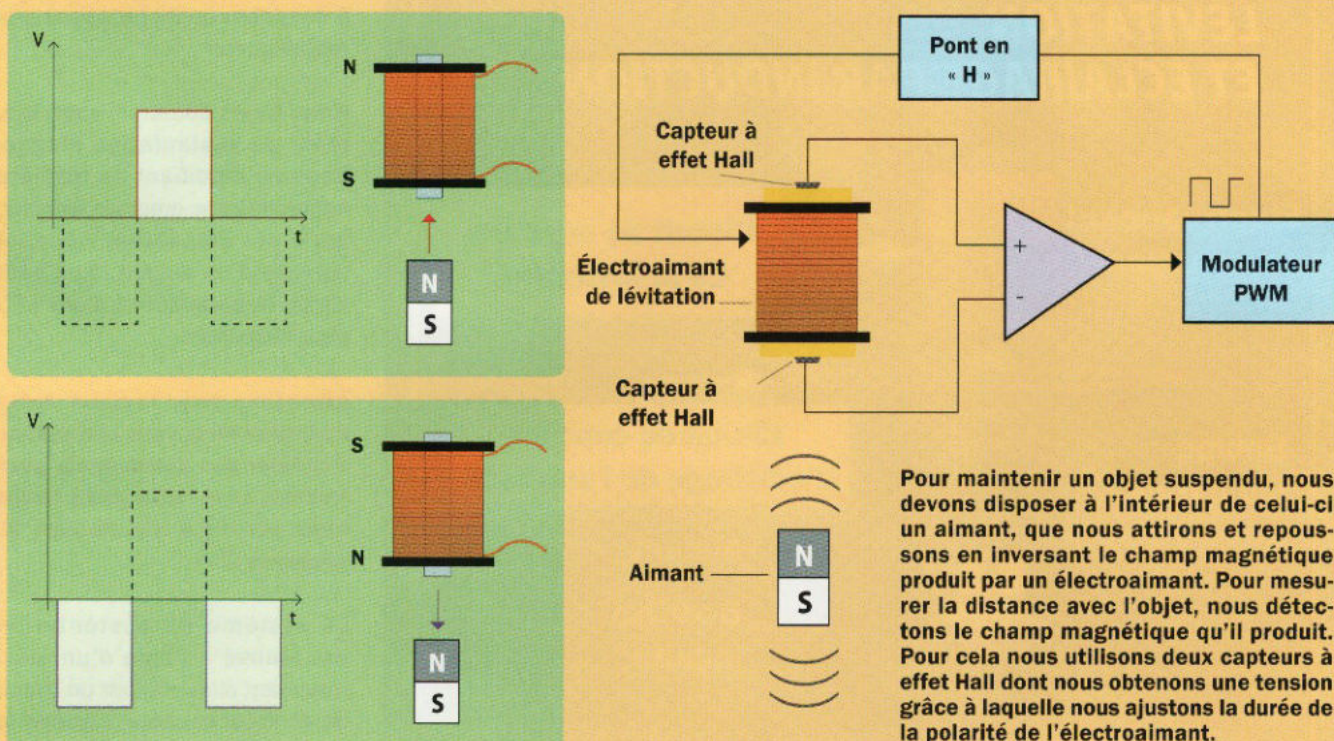
Le système de sustentation est réalisé à l'aide d'un électroaimant alimenté par un circuit de contrôle capable d'ajuster la polarité et l'intensité du champ magnétique en fonction de la distance et de la masse de l'ampoule. Pour obtenir l'éclairage de l'ampoule, nous avons produit un second champ magnétique d'intensité plus faible, de sorte qu'il n'interfère pas avec celui qui produit la sustentation de l'objet. Le champ rayonné par une seconde bobine induit, dans le circuit interne de l'ampoule, un courant suffisant pour alimenter 4 LED à haute luminosité.

Comment ça marche ?

Mais tout d'abord entrons dans le détail, en abordant les problèmes auxquels nous devons faire face et les solutions que nous devons apporter pour réaliser notre projet.

Notre défi le plus important est de maintenir un objet suspendu dans le vide en utilisant les propriétés du magnétisme, plus précisément l'attraction magnétique. Pour attirer un objet et le garder en suspension, il est nécessaire qu'il contienne du

Principe de fonctionnement de la sustentation magnétique ou lévitation



fer, donc nous avons pensé à introduire une sphère en métal ou un écrou.

Par conséquent, avec un électroaimant, nous pouvons attirer plus ou moins fortement le morceau de fer « doux » (aimant) introduit dans l'objet afin que celui-ci lévite.

Pour conserver cette distance et donc l'objet en suspension dans l'air, le champ électromagnétique produit par l'électroaimant doit être suffisamment important pour vaincre la gravité (attraction terrestre). Si la force électromagnétique est beaucoup plus grande que celle de gravitation, l'objet ne lévitera pas mais

restera collé à l'électroaimant, inversement si la force électromagnétique est insuffisante, l'objet tombera sur le sol.

Ici se pose le premier vrai problème, comment faire en sorte que le champ électromagnétique ait la force voulue, c'est-à-dire ni trop grande ni trop petite ?

Nous ne pouvons pas nous limiter à un régulateur de courant pour déterminer avec précision le courant fourni au circuit qui l'engendre, étant donné que la force requise dépend également de la distance entre l'objet en lévitation et l'électroaimant.

Il faut donc que quelque chose soit en mesure d'adapter dynamiquement le système aux conditions de fonctionnement, c'est-à-dire qu'il soit capable d'attirer et positionner l'objet à la bonne distance. Cela revient à maintenir l'objet dans une zone déterminée par une distance minimum et maximum.

Nous avons donc besoin d'un système de rétroaction ou contre réaction, qui soit capable d'augmenter l'intensité du champ électromagnétique si l'objet

a tendance à tomber vers le sol et le réduire si l'objet se rapproche trop de l'électroaimant.

Pour atteindre cet objectif, nous avons de nombreuses possibilités parmi lesquelles la mesure de la hauteur de l'objet à l'aide d'un télémètre ultrasonique ou infrarouge, ou au moyen d'une barrière infrarouge ou encore au moyen de capteurs magnétiques.

Après plusieurs tests, nous avons rejeté les deux premières solutions, car les capteurs utilisés étaient trop lents, en ce sens que lorsqu'ils envoyaient le signal de retour il était trop tard pour régler l'intensité du champ électromagnétique. Soit l'objet tombait, soit il se collait à l'électroaimant. En outre, l'utilisation de tels capteurs impliquait leur gestion par un microcontrôleur.

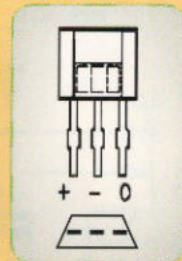
La barrière infrarouge est certainement plus facile à mettre en oeuvre, cependant il est indispensable de positionner l'objet avec précision avant de le relâcher pour qu'il soit attiré par l'électroaimant, de façon à couper le faisceau. De plus, la barrière infrarouge risque d'être



Principe de fonctionnement de la sustentation magnétique ou lévitation

Pour détecter la position de l'objet en lévitation, nous utilisons deux capteurs à effet Hall **SS495A** fabriqués par la société Honeywell. Ces capteurs disposent de 3 broches et ont un boîtier de type TO-92 semblable à un transistor. Deux broches sont réservées à l'alimentation (+5 V et le masse) et la 3^{ème} correspond à la sortie. Sa consommation en courant est de 8,7 mA sous 5 VDC.

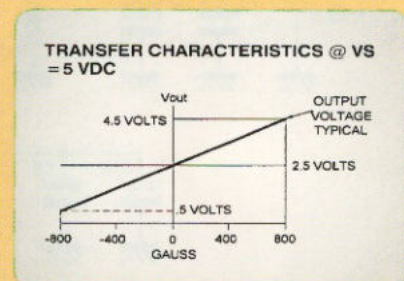
Ces capteurs à effet Hall détectent les champs magnétiques en utilisant bien évidemment l'effet Hall, qui consiste en la déviation de la trajectoire des électrons dans un conducteur parcouru par courant électrique et soumis à un champ électromagnétique. Typiquement un capteur à effet Hall est constitué par une plaque de silicium aux extrémités de laquelle se trouvent des électrodes. En appliquant un champ électrique E à la surface de silicium (c'est-à-dire en alimentant le capteur), lorsque celle-ci est soumise à un champ magnétique B perpendiculaire, les électrons le long du champ E sont déviés et se réunissent parallèlement aux lignes du champ magnétique provoquant une accumulation de charges et donc l'apparition d'une force d'opposition au mouvement des charges, c'est-à-dire l'apparition d'une tension proportionnelle au champ magnétique B.



Brochage du SS495A vu de face.

Dans le cas du SS495A, la broche de sortie « 0 » génère une tension qui varie linéairement avec l'intensité de l'induction magnétique, pour des valeurs négatives allant de -600 gauss jusqu'à des valeurs positives allant jusqu'à 600 gauss. Pour la valeur minimale (-600 gauss), la tension de sortie est d'environ 0,5 V et atteint 4,5 V pour 600 gauss. Notez que les valeurs positives et négatives définissent la direction du champ magnétique par rapport à la surface du capteur. En d'autres termes « +600 » gauss correspondent à l'intensité maximale lorsque le pôle nord (N) de l'aimant se situe face au côté du capteur comportant les inscriptions (marquage sur le boîtier), tandis que « -600 » gauss est l'intensité maximale dans le sens opposé, pôle sud (S) de l'aimant vers le marquage du capteur.

Si le capteur n'est pas complètement perpendiculaire aux lignes de champ, la tension qui en résulte diminue (si le champ est positif) ou augmente légèrement (si le champ est négatif) en fonction de l'angle formé entre le capteur et les lignes de champ.



Caractéristiques de transfert du capteur SS495A.

perturbée par la lumière ambiante, il faut alors utiliser des filtres.

Pour toutes ces raisons, nous avons préféré adopter une solution un peu plus raffinée, mais moins facile à comprendre. Nous utilisons des capteurs à **effet Hall** pour détecter la position de l'objet. Un **capteur à effet Hall** permet de **détecter la déviation de la trajectoire des électrons dans un conducteur ou semi-conducteur soumis à un champ électromagnétique, à cause de l'effet de la force de Lorentz.**

Plus généralement, l'effet Hall a été découvert en 1879 par **Edwin Herbert Hall** qui l'a énoncé comme suit :

« Un courant électrique traversant un matériau baignant dans un champ magnétique, engendre une tension perpendiculaire à ce dernier ».

Pour résumer, le capteur Hall détecte la présence de champs magnétiques et génère en sortie (selon le modèle) soit un signal numérique ou une tension analogique d'amplitude proportionnelle à l'intensité du champ magnétique. Dans notre cas, nous utilisons des **capteurs** dont la **sortie** est **analogique**, ce qui permet une meilleure précision sur la position de l'objet et sont donc bien appropriés pour ajuster linéairement l'intensité du champ électromagnétique.

Pour détecter la position de l'objet au moyen du magnétisme, nous ne pouvons pas simplement introduire un morceau de fer, car celui-ci ne génère pas son propre champ magnétique.

En effet, à l'intérieur de l'objet que nous devons sustenter dans le vide, nous avons inséré un petit aimant permanent (fixe), disposé de telle sorte que ses pôles

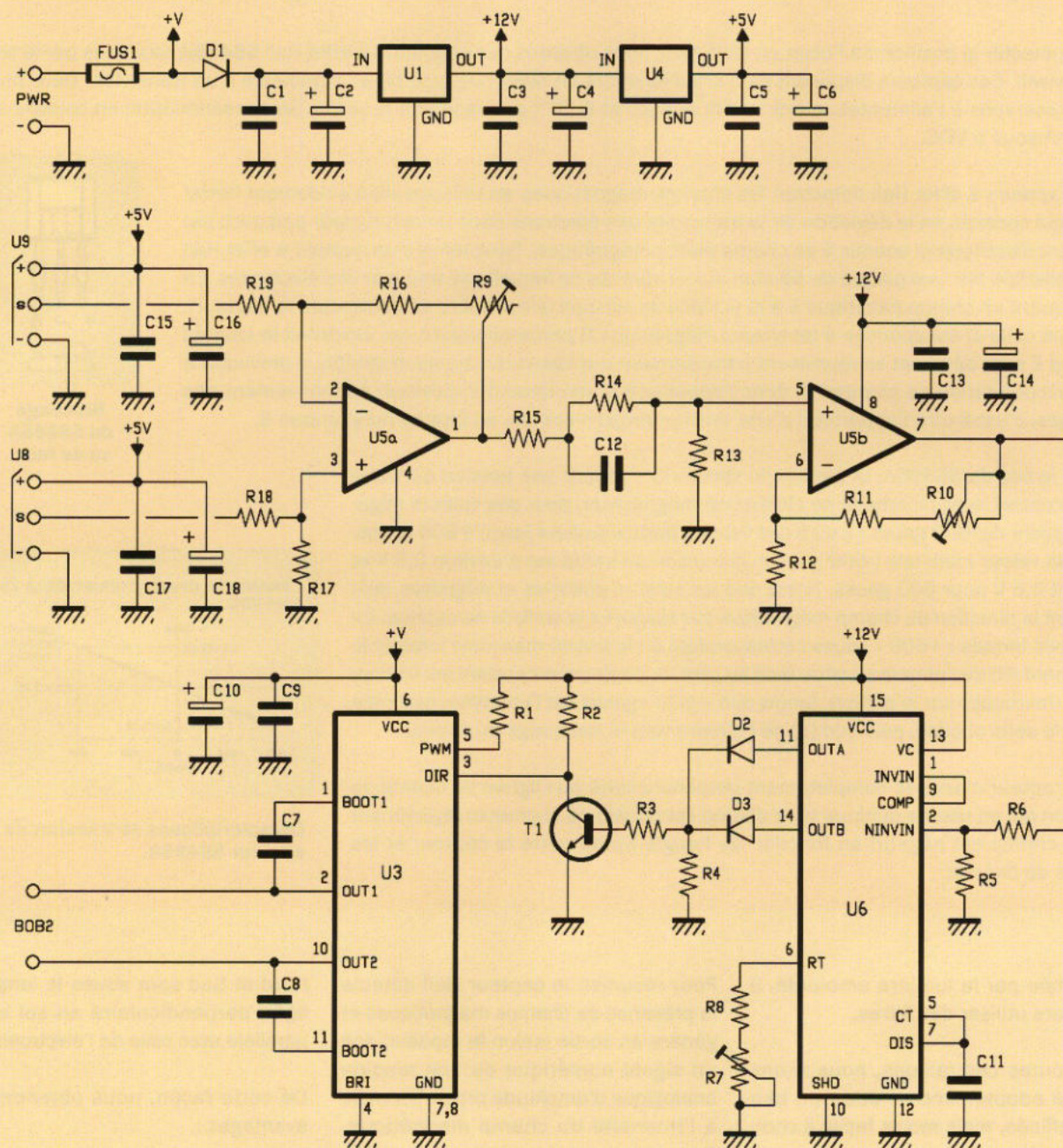
Nord et Sud sont situés le long d'une ligne perpendiculaire au sol et donc parallèle avec celle de l'électroaimant.

De cette façon, nous obtenons deux avantages :

- nous pouvons détecter la position de l'objet en utilisant un capteur à effet Hall ;
- nous pouvons plus facilement et rapidement intervenir sur la position de l'objet en lévitation, car nous ne nous limitons pas à modifier l'intensité du champ c'est-à-dire à attirer ou repousser l'objet s'il se trouve trop bas ou trop proche de l'électroaimant, mais nous intervenons aussi sur la direction du champ magnétique.

Cependant au moment de l'établissement du champ magnétique, nous

Figure 1 : schéma électrique du circuit de lévitation.



pouvons rencontrer un problème dû à l'inertie de l'objet, dans le sens où en alimentant l'électroaimant lorsque l'objet tombe, il n'est pas sûr que nous puissions l'arrêter, alors que lors de l'extinction il est possible que dans certains cas l'objet ne se colle pas sur l'électroaimant.

En effet, souvent l'objet continue sa chute même lorsque l'électroaimant tente de le faire remonter. Ainsi en insérant un aimant permanent dans l'objet, nous pouvons l'attirer ou le repousser.

Comment cela est-il possible ? Simple-ment en inversant le champ généré par l'électroaimant.

Dans la pratique, **lorsque l'objet tombe**, la bobine de l'électroaimant est parcourue par un courant de façon à générer une **polarité opposée à celle du pôle de l'aimant** situé dans l'objet pour **l'attirer** (polarité « Sud » si l'aimant présente la polarité « Nord » et vice versa), tandis que si l'objet a tendance à être **trop près de l'électroaimant**, la bobine de l'électroaimant est

parcourue par un courant de façon à générer une **polarité identique** à celle de l'aimant afin de **repousser** l'objet.

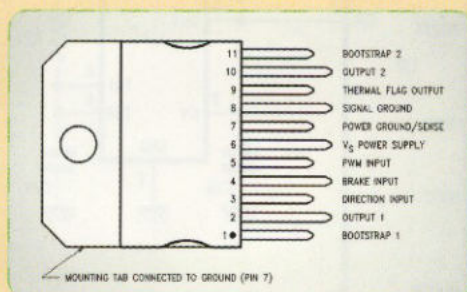
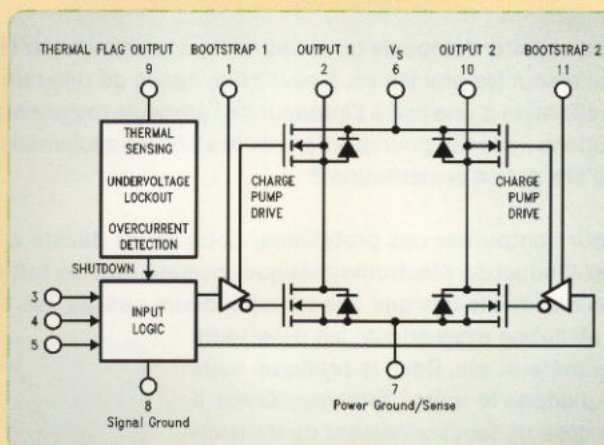
En insérant dans l'objet l'aimant permanent, il permet la lévitation de l'objet et résout ainsi de nombreux problèmes. Cependant pour détecter la position de l'objet avec une précision suffisante, il nous fallait quelque chose de plus.

Nous avons dit que nous utiliserions un capteur à effet Hall afin de détecter le champ magnétique produit par l'aimant.

Le circuit intégré LMD18201T

Pour piloter les bobines qui induisent les champs magnétiques de l'électroaimant et de l'ampoule, nous utilisons deux circuits intégrés qui sont des drivers ou contrôleurs de moteur demi pont LMD18201T produit par Texas Instruments. Ce circuit intégré est destiné à la commande de moteurs électriques en rotation continue et pas à pas, sa logique interne permet une commande PWM.

Pour une utilisation normale, il est nécessaire d'appliquer sur la broche 5 un signal PWM généré par un circuit externe qui détermine l'énergie à transférer au moteur. La broche 3 détermine le sens de rotation du moteur. Dans la pratique, l'état logique appliqué sur la broche 3 permet de déterminer la polarité des impulsions de courants présents sur les broches 2 (OUT1) et 10 (OUT2). En appliquant une tension de 5 V sur la broche 2, celle-ci devient positive par rapport à la masse, alors la sortie



« OUT1 » fournit un courant au moteur et la sortie « OUT2 » absorbe le courant provenant du moteur. Si la broche 2 est à 0, l'inverse se produit (« OUT1 » absorbe, « OUT2 » fournit le courant). La broche 4 (frein) permet d'arrêter le moteur, quand elle est placée à un niveau logique 1, les sorties « OUT1 » et « OUT2 » se trouvent à un niveau élevé.

Le circuit intégré dispose d'un système de protection thermique qui intervient lorsque la puce atteint 170 °C, en bloquant le fonctionnement des MOSFET. Cette condition est signalée par la broche 9, qui se porte à un niveau logique bas lorsque la température devient supérieure à 145 °C.

En fait, nous avons besoin de deux capteurs, car avec un seul capteur nous n'obtiendrons pas l'effet voulu.

En effet, un seul capteur serait perturbé par le champ électromagnétique produit par l'électroaimant qui attire ou repousse l'aimant permanent et donc l'objet.

Par conséquent, le signal issu du capteur à effet Hall serait la somme du champ électromagnétique produit par l'électroaimant et par l'aimant permanent situé dans l'objet.

Si à cela nous ajoutons le fait que le champ magnétique de l'électroaimant peut être inversé, nous voyons que le signal fourni par un seul capteur serait peu fiable, et aussi parce que la tension due à l'effet magnétique de l'aimant est faible par rapport à celle générée par l'électroaimant. Le problème consiste donc à éliminer l'effet produit par le champ magnétique de l'électroaimant.

La solution que nous avons trouvée était de rajouter un deuxième capteur à effet Hall.

Dans la pratique, nous avons placé un capteur en haut et l'autre en bas de l'électroaimant, c'est-à-dire chacun en correspondance avec une partie polaire.

Ainsi, la partie la plus proche détecte le magnétisme de l'aimant permanent. Il en résulte que pour les deux capteurs une proportion de la tension de sortie dépend de l'action du champ produit par l'électroaimant, tandis que pour le capteur le plus près de l'objet, sa tension de sortie tiendra compte de la distance avec l'aimant de l'objet suspendu.

En envoyant les deux signaux à un étage différentiel, il est possible d'éliminer la composante due au champ magnétique de l'électroaimant et d'extraire ainsi la tension due à l'aimant permanent de l'objet.

Le montage

Nous allons étudier maintenant le circuit électronique qui gère le système de sustentation, en tenant compte de toutes les contraintes. Il se compose de

3 parties, le circuit d'alimentation, le circuit de sustentation et enfin le circuit d'éclairage de la lampe.

Examinons la partie la plus complexe, celle de lévitation qui comporte 3 circuits intégrés U3, U5 et U6. Ces circuits génèrent une tension rectangulaire modulée en largeur d'impulsion en fonction de la distance avec l'objet en lévitation.

Plus précisément, nous faisons varier le rapport cyclique du signal modulé afin d'augmenter ou de diminuer la force qui attire ou repousse l'aimant inséré dans l'objet en sustentation.

En fait, le procédé par lequel nous faisons léviter l'objet consiste à soumettre le petit aimant à l'intérieur à un champ électromagnétique alternatif, qui l'attire et le repousse alternativement à une vitesse très élevée.

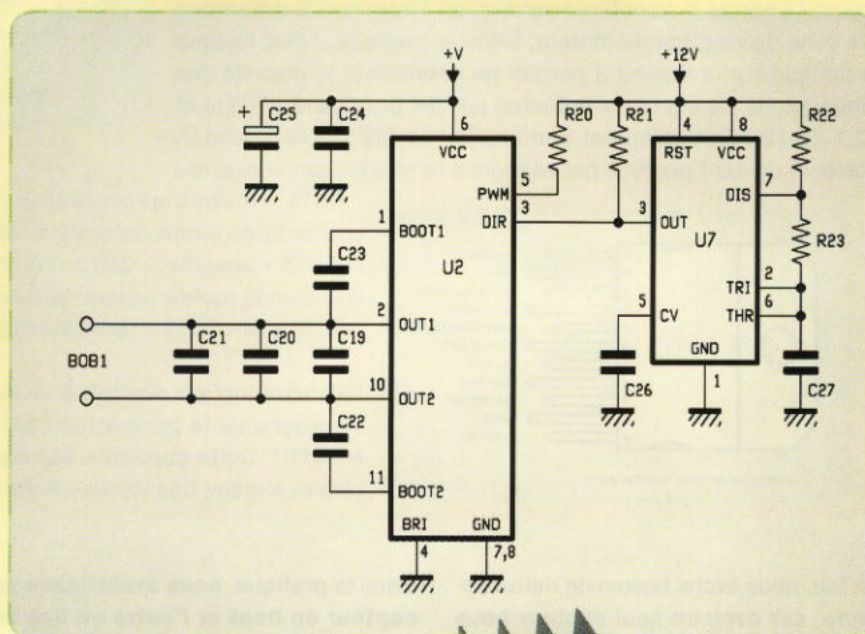
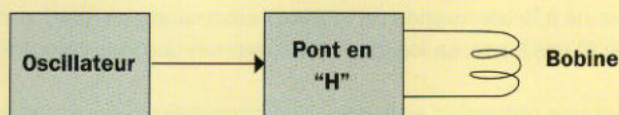
Lorsque nous devons rapprocher l'objet, nous augmentons la durée de la demi-période, le champ attire alors l'objet (il présente un pôle opposé à celui de l'aimant).

L'éclairage de l'ampoule

Pour éclairer l'ampoule ou un autre objet suspendu par lévitation magnétique, nous avons opté pour une solution élégante qui résout les problèmes, à savoir l'exclusion de relier un fil électrique à l'objet parce qu'il peut entraver la sustentation et l'utilisation d'une pile à l'intérieur de l'ampoule (augmentation du poids qui agit sur la lévitation). En plus de cela, comment aurions-nous fait pour que l'ampoule s'allume seulement lorsqu'elle est en sustentation ?

Pour contourner ces problèmes, nous avons décidé d'utiliser l'induction électromagnétique, comme cela se fait dans les dispositifs tels que des transpondeurs passifs, les trains à lévitation magnétique, les détecteurs de métaux, etc. Dans la pratique, nous exploitons le même principe qui est à la base du fonctionnement du transformateur. Au moyen d'un enroulement ou d'une bobine alimentée par un courant variable, nous créons un champ électromagnétique qui interagit avec le milieu ambiant. En disposant une bobine dans la zone où règne le champ électromagnétique, nous obtenons aux bornes de celle-ci une tension variable induite par le champ magnétique.

Dans notre cas particulier, le courant variable induit par le champ électromagnétique qui traverse la bobine est généré par un circuit dont le synoptique et le schéma électrique sont illustrés dans cet encadré.



Si nous devons éloigner l'objet, nous augmentons encore plus la durée de la demi-période afin que le champ électromagnétique présente une polarité identique à celle de l'aimant placé dans l'objet.

Le signal de modulation est généré par le circuit U6, un **SG3525A**, qui est un **générateur PWM**. Les 2 amplificateurs opérationnels **U5a** et **U5b** adaptent les signaux provenant des capteurs à effet Hall.

Examinons le montage par ordre, en commençant par la génération de la tension de commande.

Elle est obtenue en utilisant un générateur de signaux rectangulaires issus d'un générateur de signaux triangulaires couplé à un comparateur de tension dans

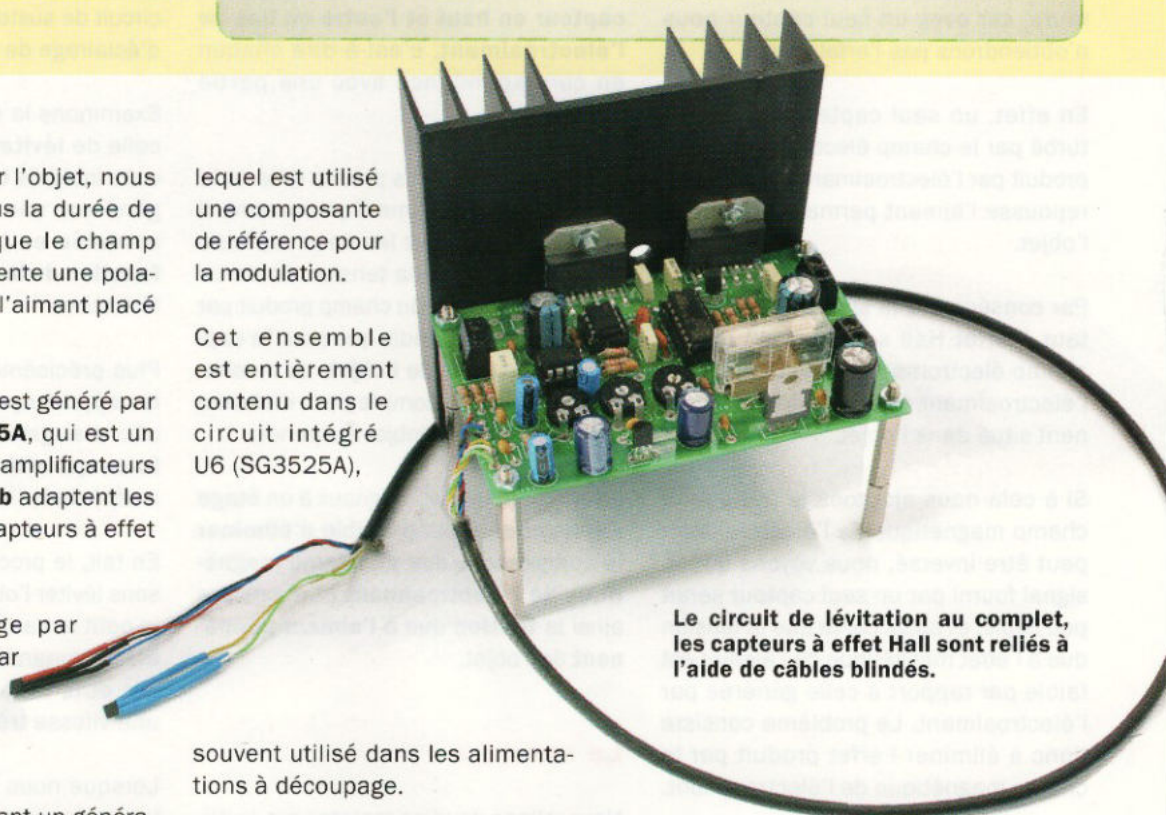
lequel est utilisé une composante de référence pour la modulation.

Cet ensemble est entièrement contenu dans le circuit intégré U6 (SG3525A),

souvent utilisé dans les alimentations à découpage.

En plus de disposer d'un modulateur PWM, il contient également un oscillateur et un étage driver pour

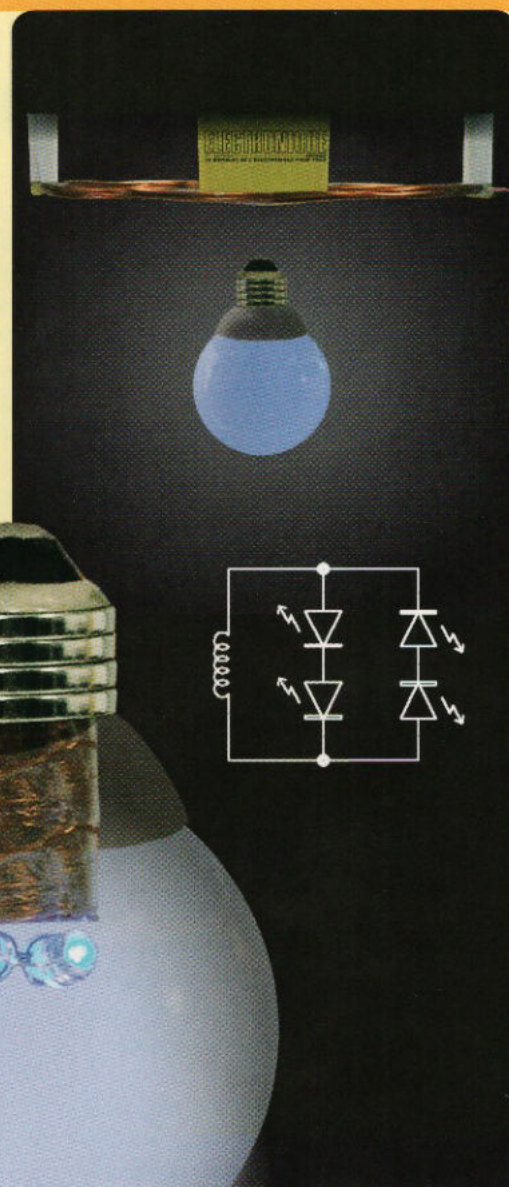
commander un étage de puissance constitué de transistors de commutation (généralement des MOSFET).



Le circuit, qui prend place sur l'unique circuit imprimé contenant également la section de contrôle de la sustentation magnétique, est composé d'un multivibrateur astable basé sur le circuit intégré U7 (NE555) fonctionnant à une fréquence d'environ 300 kHz. La sortie de U7 commande le circuit LMD18201 qui est doté d'un pont en « H », principe que nous avons déjà utilisé dans la section de lévitation. Le signal ainsi amplifié est envoyé dans la bobine « BOB1 » qui est montée de manière concentrique par rapport à l'électroaimant. Le courant produit par le générateur est de type rectangulaire, le champ électromagnétique est pulsé et alternatif.

Les impulsions électromagnétiques produites par la bobine « BOB1 », qui sont tantôt dans un sens tantôt dans l'autre, traversent la bobine disposée à l'intérieur de l'ampoule et dont les bornes sont reliées à deux blocs de 2 LED connectées en série. Les 2 blocs de LED sont disposés orthogonalement. Le champ électromagnétique induit aux bornes de la bobine une tension alternative qui alimente les blocs de LED. Lorsque la polarité est positive, le bloc de LED présentant l'anode s'allume, tandis que lorsque la polarité est inversée, c'est l'autre bloc de LED qui s'allume.

Le système d'allumage de l'ampoule est semblable à un transformateur, où le primaire correspond à la bobine inductrice (4 spires de fil émaillé dans l'air), et le secondaire est situé dans l'ampoule et est formé par la bobine induite. Étant donné la très faible perméabilité magnétique absolue de l'air, le flux magnétique entre le primaire et le secondaire est faible et donc le courant fourni par la bobine située dans la lampe est faible. Cependant, il est suffisant pour alimenter une paire de diodes à chaque demi-période et ainsi il est possible de voir l'ampoule s'allumer.



Dans notre application, nous déterminons la **fréquence de l'oscillateur** à l'aide de **C11**, **R7** et **R8**. Le signal provenant de la broche 7 de U5b est injecté dans la broche 2 de U6 et comparé avec la référence de tension interne.

Pour son fonctionnement, le SG3525 génère 2 tensions rectangulaires de phase opposée de manière à piloter un étage de puissance de type « push-pull ».

Dans notre cas, nous avons besoin de la totalité de l'onde et non de deux demi-ondes comme pour un étage « push-pull » classique, étant donné que nous devons piloter une seule bobine. Pour cette raison, les diodes D2 et D3 permettent d'obtenir la somme des deux demi-ondes afin d'avoir un signal entier qui attaque la base du transistor T1.

Le collecteur de **T1 pilote**, avec un signal unidirectionnel, la **broche 3** (DIR)

du circuit intégré **U3** (LMD18201T). Il s'agit d'un circuit permettant de piloter des moteurs, il dispose d'un circuit de puissance en pont en « H » utilisant des transistors MOSFET. Il est capable de driver des charges de 3 A sous une tension maximale de 55 V.

La tension présente entre les broches « **OUT1** » et « **OUT2** » de **U3** est **positive** lorsqu'un **niveau logique haut** est appliqué à la **broche 3**, et devient **négative** lorsque la broche « **DIR** » est à un **niveau logique 0** (T1 conducteur).

Notez que le circuit intégré **LMD18201T** dispose de plus d'une logique de commande gérant entièrement un moteur, à savoir qu'il permet de faire varier la vitesse du moteur mais aussi de le freiner, en fonction des signaux présents sur les broches 4 (frein) et 5 (PWM).

Dans la configuration typique, le LMD18201T reçoit le signal PWM sur la broche 5 afin de faire varier la vitesse, il détermine la polarité de la tension de sortie selon le niveau logique présent sur la broche 3 (DIR).

Dans notre cas, nous utilisons simplement la partie puissance correspondant au pont en « H ». C'est pour cela que nous fixons un niveau logique haut sur la broche 5, et faisons varier le niveau sur la broche 3.

En maintenant la broche 5 à un niveau logique haut, la sortie du pont fournit une tension légèrement inférieure à celle de l'alimentation.

En faisant **varier l'état logique de la broche 3**, nous **invertissons la polarité de la tension** entre les broches « **OUT1** » et « **OUT2** », afin d'obtenir une tension

Lévitaiton sur des rails

Ce type de lévitation magnétique est appliqué à un train qui se déplace dans l'air sur un rail grâce à l'attraction et à la répulsion magnétique. Le train n'est pas affecté par la friction des roues sur les rails, il subit seulement la traînée aérodynamique. Par conséquent, la sustentation magnétique permet de se déplacer à des vitesses très élevées (de l'ordre de 650 km/h et plus, vitesse comparable à celle d'un avion de ligne) avec une consommation énergétique et un niveau de bruit acceptables.

Bien que la grande vitesse obtenue par sustentation magnétique permet de rivaliser avec le transport aérien, sur de longues distances le coût est plus élevé à cause de la construction d'infrastructures spécifiques.

Le train à sustentation magnétique se limite à une utilisation sur de courtes distances et sur des lignes très fréquentées (comme pour le métro).

À l'heure actuelle au Japon, des tests sont effectués à Yamanashi. Il s'agit de la rame « Série LO » qui devient le premier train de l'histoire à sustentation magnétique à dépasser la vitesse de 600 km/h.

La première version commerciale doit relier Tokyo et Nagoya en 2027. D'autres projets ont été développés

dans d'autres pays sans grand succès, comme par exemple le train Allemand « Transrapid », dont la ligne d'essai a été fermée en 2012.

Le train à sustentation utilise deux champs magnétiques, un qui soutient et qui centre les wagons par rapport à la voie et l'autre qui fait avancer le convoi.

Il existe deux types de sustentation :

- la **sustentation électromagnétique** (EMS) qui utilise des électroaimants classiques. La traînée électromagnétique est ici très faible, voire nulle. Le rail « porteur » contient des aimants ou des électroaimants. L'interaction entre les aimants à bord du train et les aimants disposés le long de la voie crée une force magnétique induite qui compense la gravité et permet la lévitation. Ces aimants repoussent le train vers le haut et assurent l'existence d'une garde suffisante entre le « rail » et le train, de l'ordre du centimètre ;
- la **sustentation électrodynamique** (EDS) qui utilise des aimants supraconducteurs. Des bobines supraconductrices sont



placées dans le train et des électroaimants sont placés le long de la voie. Lorsque le train se déplace, un courant est induit dans la voie. La force de Laplace résultante fait léviter le train. Le déplacement du train engendre une traînée électromagnétique très importante, d'où une consommation énergétique élevée. Le projet le plus abouti est le modèle Japonais.

Les trains actuellement réalisés semblent donner de meilleurs résultats avec la technologie « EDS ».

Les électroaimants disposés dans la rame sont parcourus par courant électrique à 3 phases. Chaque groupe magnétique est composé de 3 bobines placées dans une rangée

qui reflète le signal PWM produit par le SG3525, mais qui est **unidirectionnelle** et **alternative**. En outre, U3 fournit le courant qui excite l'électroaimant afin de faire léviter l'ampoule.

Mais alors, comment vérifier la position et comment l'ampoule reste toujours stationnaire en un point ?

Comme nous l'avons mentionné précédemment, le contrôle de la distance est effectué par les 2 capteurs à effet Hall **U8** et **U9**. Chaque capteur **SS495A** dispose de 3 broches, dont 2 sont réservées à l'alimentation du capteur, et la

troisième broche (nommée « 0 » pour « output » ou sortie) fournit le signal de sortie.

Dans le circuit intégré **U6**, la tension rectangulaire de sortie a un rapport cyclique qui dépend directement du niveau du potentiel appliqué sur la broche 2. Dans la pratique, plus le potentiel est élevé, plus le rapport cyclique est grand et vice versa. Il en découle que pour un bon fonctionnement des capteurs à effet Hall, **U8 doit être positionné au-dessus de l'électroaimant** et **U9 du côté opposé**, c'est-à-dire entre l'électroaimant et l'objet en lévitation.

Les capteurs doivent être dans le même sens. Celui près de l'objet en lévitation (**U9**) doit avoir sa partie la plus grande orientée vers le pôle inférieur de l'électroaimant et pour **U8** c'est l'inverse, sa partie la plus petite doit être orientée vers le pôle supérieur de l'électroaimant (voir le synoptique dans l'encadré intitulé « Principe de fonctionnement de la sustentation magnétique ou lévitation »).

Cette disposition des capteurs, en ce qui concerne l'aimant de l'objet, a un effet magnétique opposé. En effet, un capteur montre un côté et l'autre capteur le

chacune traversée par un courant de 120° en retard de phase par rapport à la précédente (les courants sont égaux entre eux).

Comme pour un moteur électrique en rotation, le champ électromagnétique change constamment de polarité pendant l'avancement.

En maintenant constante la polarité des électroaimants situés dans la rame ou en utilisant des aimants permanents, le changement de direction du champ magnétique entraîne le train vers l'avant.

Pour obtenir le mouvement vers l'arrière (marche arrière), il suffit d'inverser l'ordre de deux des phases. Le même système peut à la fois faire avancer et freiner le train.

Comme pour les moteurs électriques tournants, la vitesse dépend de la fréquence du courant et de la quantité d'électroaimants par unité de longueur. Donc plus la fréquence est élevée et/ou plus le nombre d'électroaimants est grand, plus le train aura une vitesse élevée.

Le train à sustentation magnétique n'a pas de connexions électriques avec les rails. L'électricité à bord provient du phénomène d'induction électromagnétique, par l'intermédiaire des bobines, un courant induit est généré par le champ produit par le système en lévitation. L'électricité produite est utilisée pour les voyageurs à bord (lumière, climatisation, etc).

côté opposé, il va sans dire que si l'objet en lévitation se rapproche d'un capteur celui-ci génèrera une tension croissante et l'autre une tension décroissante.

La situation est différente pour le champ produit par l'électroaimant, car les capteurs, bien qu'ils ne soient pas orientés de la même manière sur les pôles (un capteur avec le grand côté et l'autre avec le petit côté) fournissent la même tension.

En fait, s'il est vrai que U8 et U9 sont orientés différemment (côtés différents par rapport aux pôles de l'électroaimant),

il ne faut pas oublier qu'un capteur est exposé au pôle Nord et l'autre au pôle Sud de l'électroaimant.

Donc, si U8 est placé au pôle Nord et positionné de manière à fournir une tension de 3 V par exemple, U9 qui est placé au pôle Sud et positionné à l'envers par rapport à U8 va subir des phénomènes magnétiques identiques et fournira aussi une tension de 3 V.

Compte tenu de cela, nous pouvons dire que, lorsque l'objet est en lévitation et que donc l'aimant qu'il contient se rapproche de l'électroaimant, la tension fournie par U9 sur sa broche « S » augmente plus que la tension « S » de U8.

Ainsi l'amplificateur opérationnel « U5a » diminue sa tension de sortie, ce qui provoque un abaissement de la tension de sortie de « U5b » (broche 7) et donc une diminution de la tension appliquée à la broche 2 de U6.

Le résultat est une diminution du rapport cyclique de la tension PWM, qui se traduit par une augmentation de la largeur des demi-ondes négatives à la sortie du pont.

En reliant de manière appropriée la bobine, cela provoque une diminution du champ magnétique et donc l'éloignement de l'objet suspendu.

Au contraire, si l'objet et donc l'aimant qu'il contient s'éloignent, le capteur U9 génère une tension plus basse, ce qui provoque une augmentation du potentiel de sortie de « U5a » et donc de celui de « U5b ». Il en découle une augmentation du rapport cyclique du signal PWM produit par le SG3525.

En conséquence, le signal rectangulaire présent entre les sorties « OUT1 »



et « OUT2 » du circuit U3 est constitué par des impulsions positives d'une plus grande durée que celles négatives et, en raccordant la bobine de manière appropriée, une augmentation du champ magnétique se produit ce qui attire l'objet suspendu.

Il est évident que si vous changez la position des capteurs, vous devez inverser les fils de la bobine (BOB2) de l'électroaimant (bobine de sustentation), sinon l'objet risque de tomber ou de rester collé à l'électroaimant.

Les trimmers insérés dans le circuit servent à modifier la position de l'objet suspendu et à optimiser le fonctionnement de l'ensemble.

Pour être exact, **R9 règle le gain de l'amplificateur différentiel** et **R10** celui du **deuxième étage** d'amplification.

Chacun des trimmers, dans une certaine mesure, influe sur l'amplification de la composante du capteur U9 et donc sur la position de l'objet en lévitation. Ainsi en augmentant le gain, on augmente théoriquement la distance entre l'objet et l'électroaimant et vice versa.

En ce qui concerne **R7**, elle règle la **fréquence de fonctionnement de l'oscillateur** interne de **U6** et donc la fréquence de l'onde rectangulaire qui alimente la bobine (BOB2) de l'électroaimant. Cette fréquence doit être réglable afin d'adapter le montage au type d'électroaimant utilisé.

L'ajustement permet d'obtenir une plus grande intensité du champ magnétique

dans le noyau de l'électroaimant pour un courant équivalent dans la bobine.

En résumé, **R7 détermine la distance à laquelle l'objet peut rester en lévitation par rapport à l'électroaimant.**

La fréquence de l'onde rectangulaire, qui commande la bobine et donc qui maintient en suspension la lampe, varie de 40 kHz (R7 ayant sa résistance maximale) à 500 kHz (R7 court-circuitée).

Nous venons d'étudier la partie du montage la plus importante, c'est celle qui est responsable de la lévitation magnétique.

Passons maintenant à l'étage qui éclaire la lampe par induction électromagnétique, il s'agit du même principe de fonctionnement qu'un transformateur.

L'**éclairage** est réalisé grâce à la bobine « **BOB1** » qui irradie le voisinage de la lampe par un champ électromagnétique à haute fréquence et d'intensité constante. La bobine « **BOB1** » agit comme l'enroulement primaire d'un transformateur.

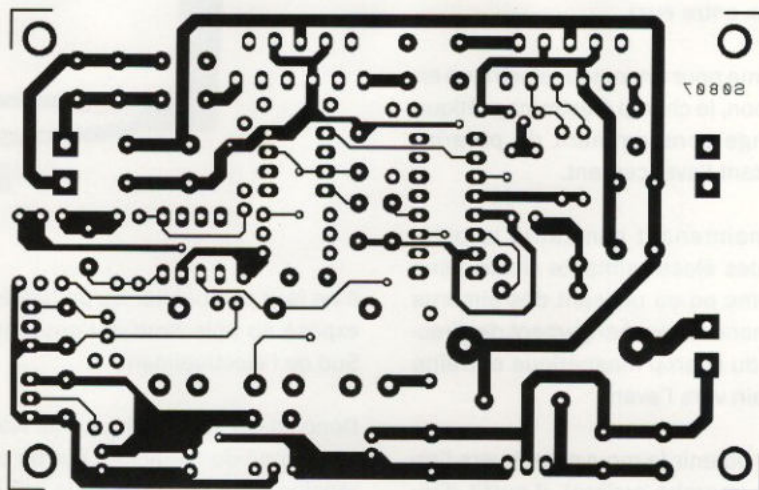
À l'intérieur de la lampe en lévitation se trouve une autre bobine qui correspond au secondaire d'un transformateur et à travers laquelle le champ électromagnétique présent induit une tension de fréquence variable égale à celle du champ. La tension est alors utilisée pour alimenter deux paires de diodes reliées entre elles de manière antiparallèle.

Ainsi il est possible d'exploiter les deux demi-ondes, lorsque la polarité est positive seulement 2 LED conduisent et pour la demi-onde négative c'est l'autre paire de LED qui devient passante.

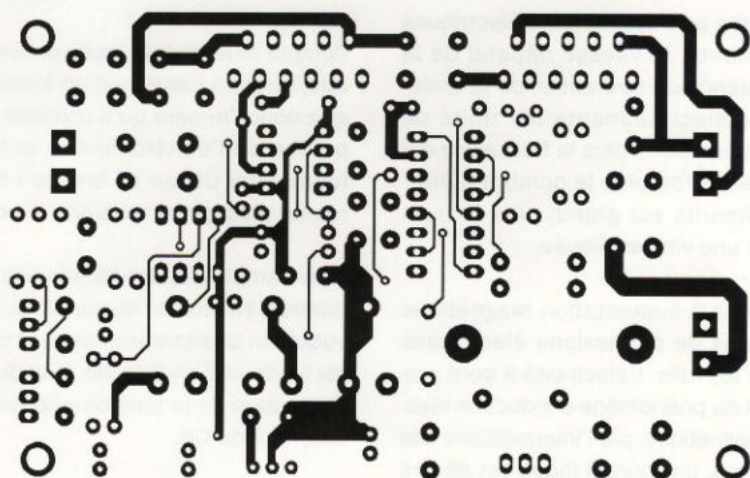
Notez que nous n'avons pas prévu de résistances de limitation en série avec les LED, étant donné que le courant fourni par la bobine est de l'ordre d'une dizaine de milliampères et donc inférieur au maximum tolérable par les LED.

La tension qui pilote la bobine d'éclairage de l'ampoule est fabriquée à partir du circuit **temporisateur U7** (NE555), configuré en **multivibrateur astable**. Les composants de temporisation **R22**, **R23** et **C27** sont fixes et dimensionnés

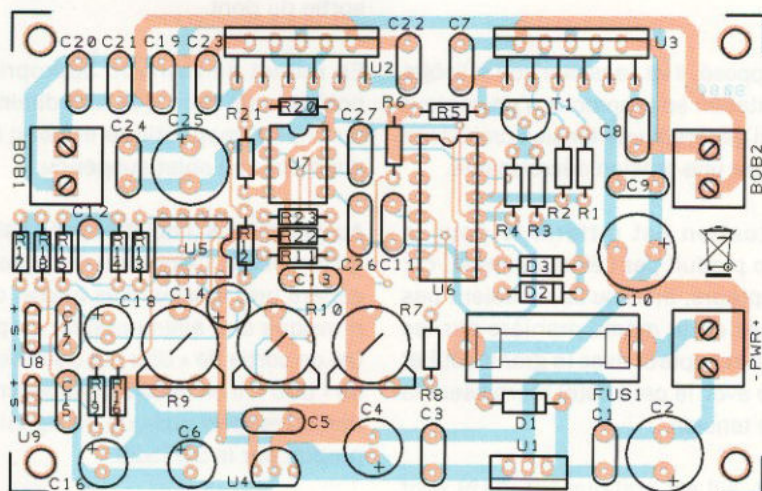
Plan de montage du système de sustentation magnétique



Circuit imprimé à l'échelle 1 : 1 côté soudures du système de sustentation magnétique.



Circuit imprimé à l'échelle 1 : 1 côté composants du système de sustentation magnétique.

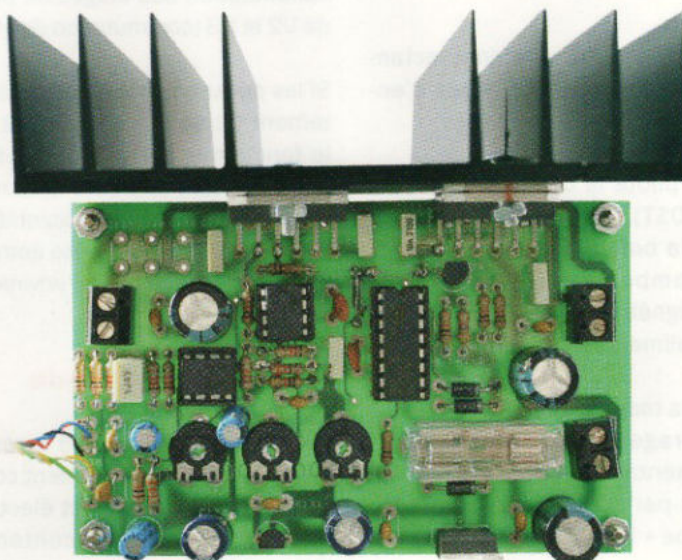


Liste des composants du système de sustentation magnétique

R1..... 10 k Ω
 R2..... 10 k Ω
 R3..... 4,7 k Ω
 R4..... 10 k Ω
 R5..... 100 k Ω
 R6..... 0 Ω (pont)
 R7..... trimmer 220 k Ω mono tour
 R8..... 10 k Ω
 R9..... trimmer 220 k Ω mono tour
 R10.... trimmer 4,7 k Ω mono tour
 R11.... 10 k Ω
 R12.... 12 k Ω
 R13.... 12 k Ω
 R14.... 12 k Ω
 R15.... 12 k Ω
 R16.... 33 k Ω
 R17.... 10 k Ω
 R18.... 10 k Ω
 R19.... 10 k Ω
 R20.... 10 k Ω
 R21.... 10 k Ω
 R22.... 330 Ω
 R23.... 12 k Ω

C1..... 100 nF multicouche
 C2..... 470 μ F/16 V électrolytique
 C3..... 100 nF multicouche
 C4..... 220 μ F/25 V électrolytique
 C5..... 100 nF multicouche
 C6..... 220 μ F/25 V électrolytique
 C7..... 10 nF/100 V polyester
 C8..... 10 nF/100 V polyester
 C9..... 100 nF multicouche
 C10.... 1000 μ F/16 V électrolytique
 C11.... 100 pF céramique
 C12.... 1 μ F/63 V polyester
 C13.... 100 nF multicouche
 C14.... 100 μ F/16 V électrolytique

Photo de l'un de nos prototypes du système de sustentation magnétique.



C15.... 100 nF multicouche
 C16.... 100 μ F/16 V électrolytique
 C17.... 100 nF multicouche
 C18.... 100 μ F/16 V électrolytique
 C19 à C21 non connectés
 C22.... 10 nF/100 V polyester
 C23.... 10 nF/100 V polyester
 C24.... 100 nF multicouche
 C25.... 1000 μ F/16 V électrolytique
 C26.... 10 nF/100 V polyester
 C27.... 100 pF céramique

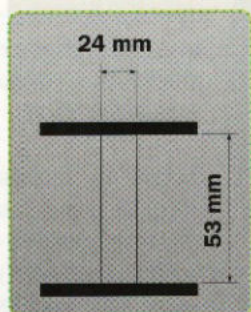
D1..... 1N4007
 D2..... 1N4007
 D3..... 1N4007

U1..... 7812
 U2..... LMD18201T
 U3..... LMD18201T
 U4..... 78L05

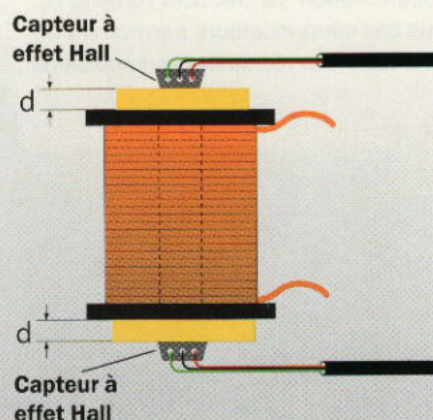
U5..... LM358N
 U6..... SG3525A
 U7..... NE555N
 U8..... SS495A
 U9..... SS495A
 T1..... BC547

Divers

Borniers 2 pôle (x3)
 Support circuit intégré 2 x 4 broches (x2)
 Support circuit intégré 2 x 8 broches
 Barrette mâle 3 pôles (x2)
 Barrette femelle 3 pôles (x2)
 Porte fusible pour circuit imprimé
 Fusible retardé 3 A
 Dissipateur
 Vis 20 mm 3 MA (x2)
 Écrou 3 MA (x2)



Construction de la bobine de l'électroaimant et positionnement des 2 capteurs à effet Hall. Les capteurs doivent être disposés suivant l'orientation illustrée sur la figure ci-contre, avec une distance identique par rapport au noyau.



pour obtenir une fréquence d'environ **300 kHz**.

Le circuit U7 produit un signal rectangulaire avec un rapport cyclique d'environ 50 %.

Ce signal pilote le driver en pont U2 (LMD18201T), dont les sorties sont reliées à la bobine « BOB1 ». Celle-ci irradie l'ampoule en induisant un champ magnétique à partir duquel les LED sont alimentées.

Les circuits magnétiques de lévitation et d'éclairage puisent leur énergie d'une alimentation constituée essentiellement par deux régulateurs intégrés de type « 78XX ».

La tension d'entrée (+V) traverse le fusible (FUS1) et alimente directement les circuits de puissance, à savoir les deux drivers en pont U2 et U3.

La même tension « +V » passe par la diode D1, qui protège le reste du circuit contre les inversions de polarité, et est filtrée convenablement par les condensateurs C1 et C2 pour enfin atteindre le premier régulateur U1. Ce dernier est un 7812 qui fournit les 12 VDC nécessaires au fonctionnement du NE555, du SG3525 et des 2 amplificateurs opérationnels (LM358N).

La tension de 12 VDC alimente aussi un deuxième régulateur U4 (78L05) qui fournit une tension de 5 VDC pour les deux capteurs à effet Hall.

Notez le nombre important de condensateurs de filtrage sur l'entrée et la sortie de U1, sur la sortie de U4 et sur la ligne d'alimentation du LM358N (broche 8). Tous ces condensateurs permettent de filtrer les bruits résiduels des impulsions

causés sur la ligne principale (+V) par la commutation des étages de puissance de U2 et U3 (commutation des bobines).

Si les perturbations ne sont pas correctement filtrées, elles peuvent affecter le fonctionnement des amplificateurs opérationnels, du SG3525 et même du NE555. L'ensemble est alimenté par une tension continue comprise entre 15 V et 18 V avec une courant d'environ 2,5 A.

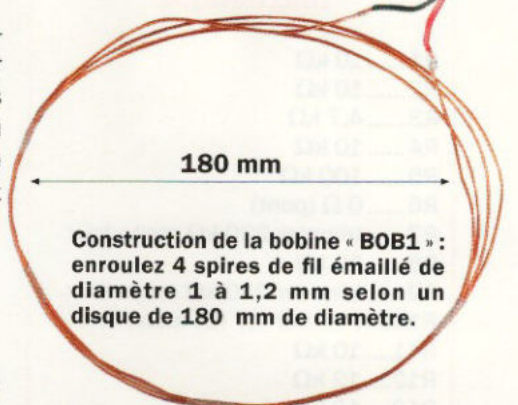
Réalisation pratique

Cela étant dit, passons à la réalisation pratique et voyons comment construire non seulement le circuit électronique, mais aussi le cadre contenant les bobines qui permettent la lévitation et l'éclairage de l'ampoule.

Commençons par le circuit imprimé double face qui contient l'ensemble du montage (la sustentation et l'éclairage de l'ampoule). Vous pouvez télécharger les typons sur notre site dans le sommaire détaillé de la revue à l'onglet « Télécharger ».

Une fois le circuit gravé et percé, commencez par souder les composants ayant un bas profil : les résistances, les diodes (attention à l'orientation) et les condensateurs non polarisés. Les condensateurs C19 à C21 ne sont pas montés sur le circuit imprimé.

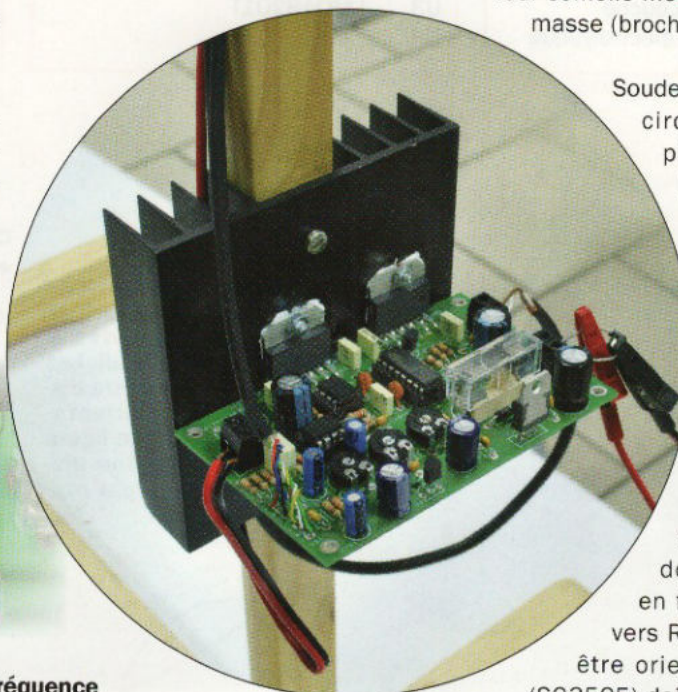
Ensuite, soudez les supports des circuits intégrés SG3525A, NE555 et LM358N.



Continuez avec les trimmers, les condensateurs électrolytiques (attention à la polarité, le « - » est indiqué sur le boîtier), les régulateurs (le 7812 doit avoir le côté métallique vers l'extérieur du circuit imprimé).

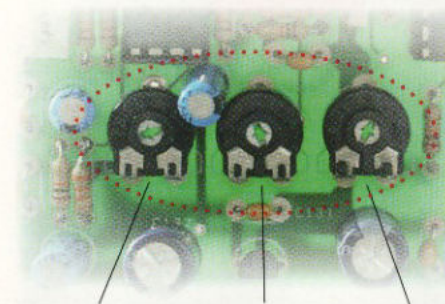
Enfin, soudez le porte fusible et les borniers au pas de 5 mm. Référez-vous au plan de câblage des composants.

Concernant les circuits LMD18201T, nous vous conseillons d'abord de les fixer sur le dissipateur en appliquant de la pâte thermique et en utilisant des vis et écrous M3, puis ensuite de les présenter sur les emplacements du circuit imprimé. Il n'est pas nécessaire d'intercaler une feuille de mica entre les LMD18201T et le dissipateur, car leur semelle métallique est reliée à la masse (broche 7).

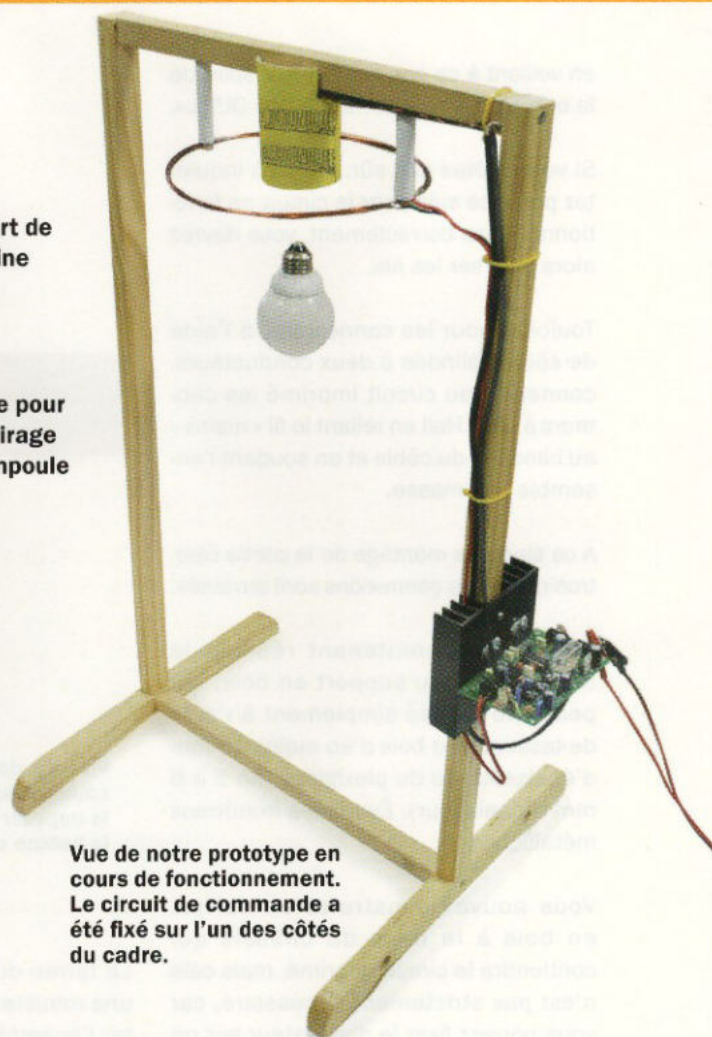
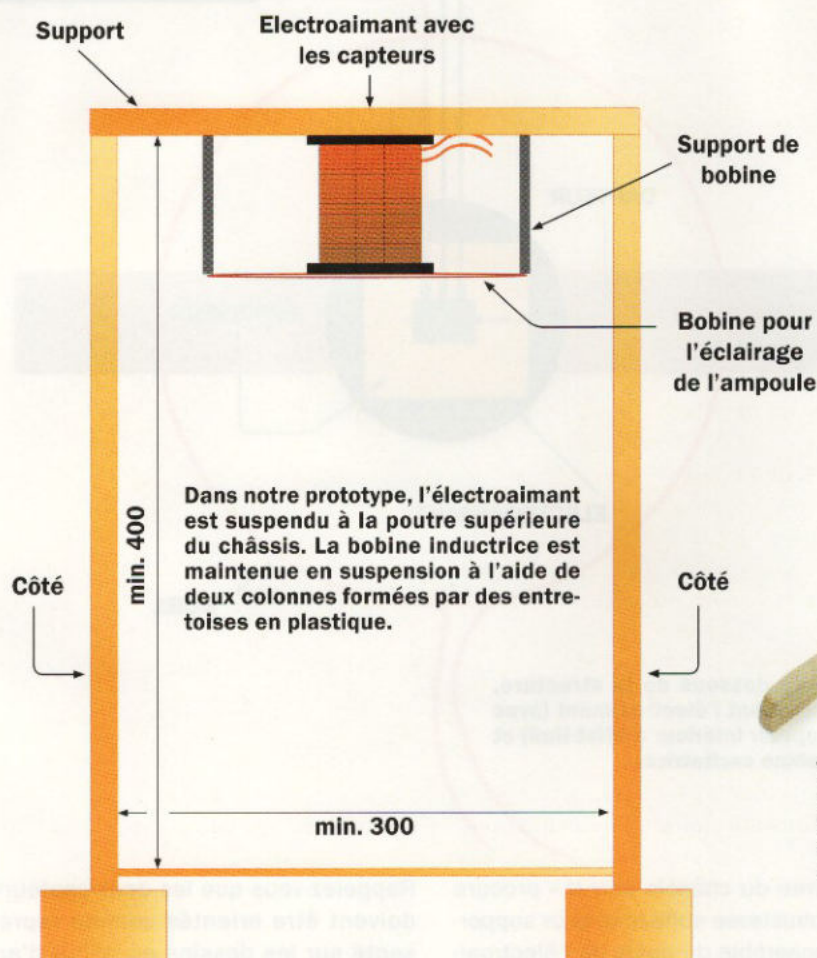


Soudez une patte de chaque circuit et vérifiez s'ils plaquent bien contre le radiateur, soudez ensuite toutes les autres pattes. Le dissipateur doit avoir une résistance thermique d'au moins 2,5 °C/W.

Insérez les circuits intégrés dans les supports respectifs, U5 (LM358N) doit avoir son repère en forme de « U » dirigé vers R12, U7 (NE555) doit être orienté vers R20 et U6 (SG3525) doit avoir son repère en



R9 – distance R10 – distance R7 – fréquence



forme de « U » dirigé vers R5. N'oubliez pas d'insérer un fusible dans le porte fusible.

Mettez de côté le circuit imprimé et passez à la réalisation de la bobine excitatrice « BOB1 », de l'électroaimant et du circuit à insérer dans l'ampoule pour l'allumer.

Procédez par ordre en commençant par la bobine. Elle est constituée par 4 spires de 18 cm de diamètre de fil de cuivre émaillé de diamètre 1 à 1,2 mm.

L'électroaimant doit être réalisé en bobinant, sur un support de 24 mm de diamètre contenant un noyau cylindrique de fer doux (par exemple un boulon de grande dimension), 868 spires de fil de cuivre émaillé de 0,85 mm de diamètre disposées en 14 couches de 62 spires jointives chacune, cela correspond à une hauteur de 53 mm.

La bobine de l'électroaimant que vous venez de fabriquer doit être connectée

au bornier « BOB2 » du circuit imprimé. Notez que le noyau de l'électroaimant doit avoir la même hauteur que les couches, c'est-à-dire 53 mm.

Dans le cas où le noyau n'a pas exactement la même hauteur, essayez de l'usiner et d'ajuster les spires afin que le noyau ait la même hauteur que les couches. Attention un noyau en fer doux se casse facilement, donc travaillez en douceur.

Pour fabriquer la bobine qui doit être insérée dans l'ampoule, il faut bobiner une self à air, sur un support de diamètre 9 mm (par exemple un tube en plastique ou le corps d'un stylo à bille), environ 800 spires de fil de cuivre émaillé de diamètre 0,28 mm, ce qui correspond à 11 couches d'une hauteur d'environ 20 mm de 72 spires chacune.

Cette bobine ne comporte pas de noyau, une fois le bobinage terminé, retirez délicatement le support.

Ensuite grattez l'émail des 2 extrémités de la bobine à l'aide d'un cutter, puis soudez sur les extrémités les 4 LED en formant un quadrilatère comme le montrent les photographies du prototype.

D'un point de vu électrique, montez les LED de façon antiparallèle. Les LED émettent une lumière concentrée, afin d'obtenir un éclairage uniforme vous devez utiliser des diodes ayant un grand angle d'émission.

Si vous utilisez des LED à haut rendement, elles ont un petit angle d'émission, vous ne verrez pas l'objet entier en lévitation. Il faut émettre une lumière diffuse qui éclaire de façon uniforme à l'intérieur de l'objet en lévitation.

Les connexions de la bobine « BOB1 » doivent être reliées à l'aide de 2 fils de 2 x 0,75 mm² au bornier du circuit imprimé, sans se soucier de la polarité.

Enfin, connectez la bobine « BOB2 » de l'électroaimant au bornier correspondant,

en veillant à ce que la première spire de la bobine corresponde au point « OUT1 ».

Si vous n'êtes pas sûr, ne vous inquiétez pas à ce sujet, car le circuit ne fonctionnera pas correctement, vous devrez alors inverser les fils.

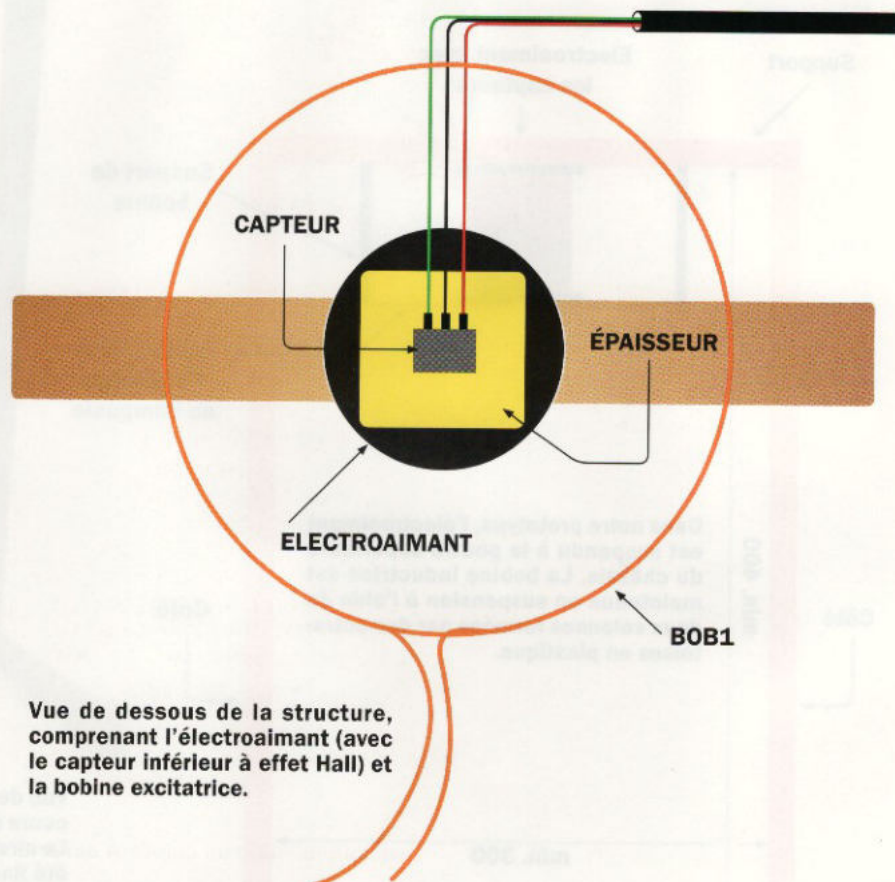
Toujours pour les connexions, à l'aide de câbles blindés à deux conducteurs, connectez au circuit imprimé les capteurs à effet Hall en reliant le fil « moins » au blindage du câble et en soudant l'ensemble à la masse.

À ce stade, le montage de la partie électronique et les connexions sont terminés.

Vous devez maintenant réaliser la construction du support en bois, qui peut être réalisé simplement à l'aide de tasseaux de bois d'au moins 10 mm d'épaisseur ou du plexiglass (de 5 à 6 mm d'épaisseur). Évitez les matériaux métalliques.

Vous pouvez construire un coffret en bois à la base du châssis qui contiendra le circuit imprimé, mais cela n'est pas strictement nécessaire, car vous pouvez fixer le dissipateur sur un côté (sur le tasseau de bois) et laisser le montage à l'air libre.

Attention toutefois de ne pas toucher avec les mains le circuit lorsque le montage est sous tension.



Vue de dessous de la structure, comprenant l'électroaimant (avec le capteur inférieur à effet Hall) et la bobine excitatrice.

La forme du châssis en « U » procure une robustesse suffisante pour supporter l'ensemble du poids de l'électroaimant et des bobines.

Vous pouvez masquer les câbles sur les côtés du châssis à l'aide de gaines et de colliers ryls, si vous devez faire une présentation.

Il est préférable de centrer l'ensemble électroaimant plus bobines par rapport au châssis, cela est plus élégant. Avant de fixer l'électroaimant, collez les capteurs à effet Hall sur chaque pôle en respectant l'orientation.

Le circuit intégré U9 est disposé vers le bas, c'est-à-dire vers l'objet qui doit léviter et U8 est positionné sur le pôle supérieur de l'électroaimant.

Rappelez-vous que les deux capteurs doivent être orientés comme représenté sur les dessins en début d'article et centrés par rapport aux pôles de l'électroaimant.

Prenez la bobine « BOB1 » et fixez-la avec de la colle silicone en utilisant des entretoises en plastique ou des tasseaux de bois, de sorte qu'elle soit concentrique par rapport à l'électroaimant. Le système de base est prêt.

Maintenant, il faut préparer l'objet qui doit léviter, dans notre cas il s'agit d'une ampoule. Procurez-vous un aimant cylindrique de dimensions sensiblement égales à celles d'un culot d'une lampe de type E27 ou sinon utilisez 4 aimants en forme de disque empilés les uns sur les autres (souvenez-vous que si vous mettez des mêmes pôles d'un côté, les aimants se repousseront).

Avec de la colle silicone, collez l'(es) aimant(s) sur la tête de la petite bobine (celle qui vient à l'intérieur de l'ampoule) de sorte que les LED se situent vers le bas.

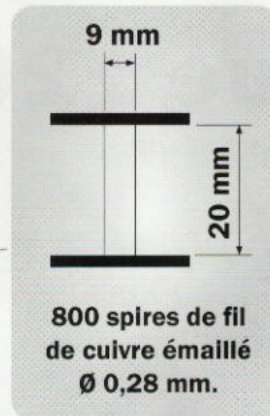
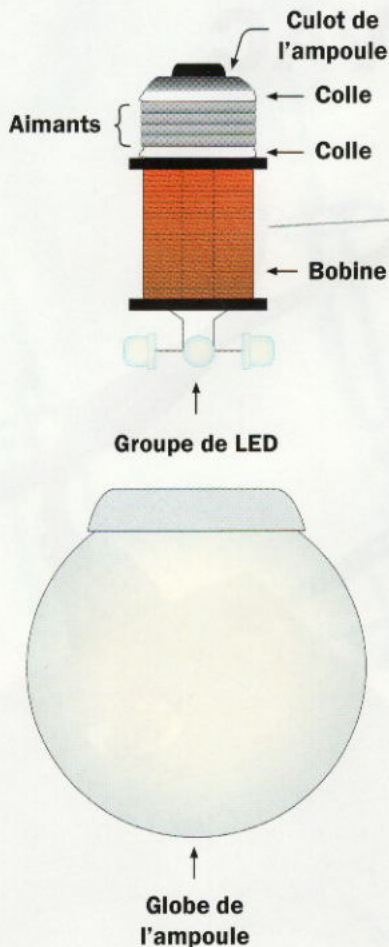
L'ampoule doit être de type opaque (blanche) de manière à



La lampe que nous avons utilisée est de type LED. À côté vous la voyez démontée. Nous avons récupéré le globe et la partie inférieure du culot.



ASSEMBLAGE DE LA LAMPE



L'ensemble, qui est inséré dans l'ampoule, contient l'enroulement de la bobine connecté aux LED. Dans la partie supérieure, vous apercevez les 4 aimants qui ressemblent à un culot E27.



obtenir une lumière diffuse et à masquer le contenu. Démontez l'ampoule sans la casser, **en utilisant des gants de travaux**. Ouvrez-la en tirant sur le filetage et mettez de côté le globe en verre de l'ampoule. Insérez les LED et la bobine, et collez l'aimant à la base du globe en verre. L'ampoule ainsi confectionnée est prête.

Si vous avez l'intention d'utiliser d'autres objets, rappelez-vous qu'ils ne doivent pas être en métal. Ce qui ne doit pas manquer dans l'objet c'est l'aimant. Vous devez bien le fixer à l'intérieur de l'objet choisi.

Dans le cas de l'ampoule, pour donner un rendu plus réaliste, nous avons utilisé 4 aimants en forme de disque, l'ensemble ressemble au pas de vis du culot de l'ampoule. Coupez la partie supérieure du filetage et récupérez seulement le fond du culot, positionnez par-dessus les 4 aimants en reliant les fils.

L'ensemble « fond du culot + 4 aimants » ressemble à un culot complet d'une lampe de type E27.

L'autre côté de la bobine comportant les 4 LED doit être inséré dans le globe en verre de la lampe, de telle sorte que les aimants et le fond du culot se trouvent à l'extérieur du globe. La fixation peut être effectuée à l'aide d'un joint de silicone.

Les essais

Une fois la structure mécanique assemblée, les capteurs connectés et les bobines reliées au circuit de commande, procurez-vous une alimentation fournissant une tension d'environ 16 VDC avec un courant d'au moins 2,5 A.

Reliez l'alimentation au bornier « PWR » sur le circuit (respectez la polarité).

Avant la mise sous tension, positionnez le potentiomètre **R7 à mi-course** et les potentiomètres **R9 et R10 au 3/4 de la course** du curseur dans le **sens antihoraire**.

Prenez l'ampoule modifiée, délicatement, en la tenant par le globe contre la paume de votre main, puis présentez-la à l'intérieur du cadre en bois en l'approchant

doucement de l'électroaimant jusqu'à ce que vous sentiez qu'elle soit attirée.

Dès que vous avez la sensation qu'elle peut léviter, lâchez-la doucement en gardant toujours votre paume de la main en dessous, dans le cas où elle viendrait à tomber. Si l'ampoule chute, remettez-la sous l'électroaimant comme précédemment, prenez un petit tournevis et tournez légèrement le potentiomètre R10 encore plus dans le sens antihoraire, jusqu'à ce que le système soit capable de sustenter l'ampoule.

Ne vous inquiétez pas si elle ondule légèrement, car parfois le circuit a besoin d'un moment pour stabiliser le réglage.

Si nécessaire, retouchez le potentiomètre R9 qui détermine la sensibilité du capteur magnétique qui détecte la distance de l'objet.

Conclusion

À travers cette expérience didactique, nous espérons vous avoir fait découvrir les facettes invisibles de la science, notamment certaines forces physiques telles que le magnétisme ou la gravité qui ne sont pas visibles mais qui pourtant régissent notre vie.

Compteur d'énergie électrique 220 VAC

de Arsenio Spadoni

En reliant en série ce compteur d'énergie à la ligne d'alimentation 220 VAC de votre maison ou de votre appartement, il devient possible de mesurer la puissance active, c'est-à-dire celle réellement consommée par les appareils (TV, électroménager, ordinateurs, etc.). Ce compteur est doté d'un afficheur LCD qui visualise à la fois la consommation instantanée et la consommation globale. Il utilise un composant spécifique pour ce type d'application, il s'agit du MCP3905 de Microchip.



L'avènement des compteurs électroniques pour l'électricité domestique et industrielle a apporté beaucoup d'avantages pour l'utilisateur, y compris la possibilité d'éviter le passage de la personne effectuant le relevé de consommation.

De plus, il permet à l'utilisateur de mesurer sa consommation instantanée, le pic maximum de consommation et surtout le relevé à distance de la consommation énergétique globale.

Cependant toutes ces fonctions utiles, telles que l'interrogation directe du compteur et la visualisation sur un écran de la consommation instantanée et des pics de courant, peuvent ne pas être accessibles selon l'emplacement du compteur. En effet, ces derniers peuvent se situer loin des appartements ou dans des endroits inaccessibles.

Nous avons donc pensé à tous ceux qui veulent contrôler leur propre consommation électrique dans le cas où le

compteur ne se trouve pas à proximité, par exemple pour vérifier la consommation de la machine à laver ou du fer à repasser.

Nous avons conçu et réalisé un véritable compteur électronique de consommation. En fait, notre appareil est capable de mesurer la puissance active c'est-à-dire la puissance réellement consommée sur la ligne, sans tenir compte de la composante réactive qui correspond à une infime partie de la consommation.

Comme pour le compteur électrique installé par EDF, notre compteur indique la consommation instantanée, mais aussi il permet de visualiser le nombre de kWh consommés pour une période définie.

À la différence du compteur installé par l'opérateur, notre compteur permet de réinitialiser à tout moment le comptage des kWh consommés. Cela est appréciable si vous avez l'intention de mesurer la consommation sur une certaine période, par exemple dans le cas où vous louez votre maison pour les vacances.

Notre appareil, comme pour un compteur électrique classique, mesure la **puissance apparente** qui est la somme vectorielle de la **puissance active** (la puissance réellement utilisée) et de la **puissance réactive** provenant des éléments réactifs.

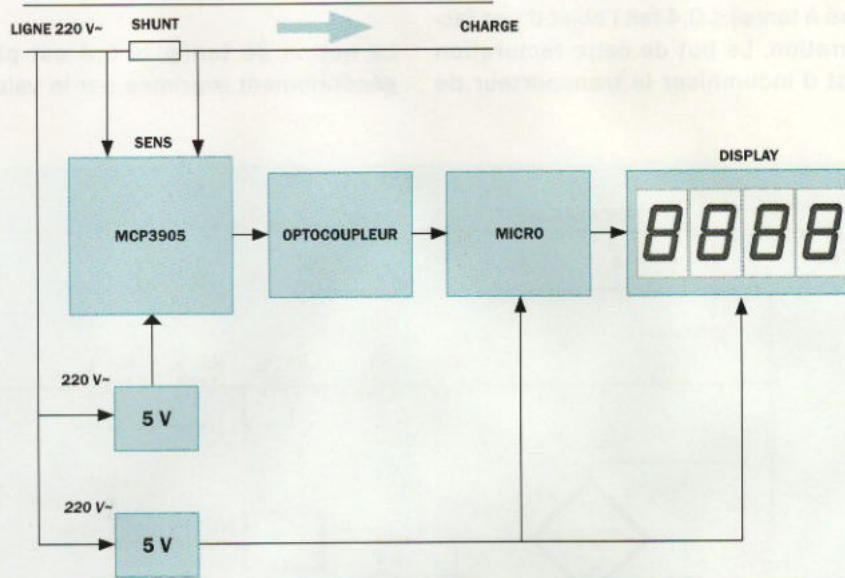
Toutefois, le **déphasage** entre le **courant** et la **tension** doit respecter certaines valeurs (il ne doit pas dépasser $21,5^\circ$). Le compteur est capable de déterminer la puissance active, dont la valeur est liée à la puissance apparente par un coefficient constant.

La **puissance réactive** apparaît dans tout système ayant des composants réactifs (c'est-à-dire capacitifs ou inductifs). Elle peut être soit « produite » (circuit capacitif) soit « consommée » (circuit inductif) par les différents éléments des circuits. Même si elle est imaginaire (nombres complexes), la puissance réactive a une signification physique réelle et sa valeur est essentielle au dimensionnement et à la stabilité des réseaux électriques.

La **puissance active** est utilisée pour générer un travail (par exemple celui d'un moteur) ou de la chaleur, la puissance réactive correspond, lorsqu'elle est « consommée », à l'établissement du champ magnétique dans les machines électriques (transformateurs, machine asynchrone, etc.).

Si la puissance réactive prélevée par les consommateurs est trop élevée par rapport à la puissance active, l'augmentation du courant dans le réseau électrique entraîne des **pertes thermiques**, des surcharges des transformateurs de distribution, l'échauffement des câbles d'alimentation et des chutes de tension, il est donc primordial d'y remédier.

Synoptique et principe de fonctionnement



L'afficheur LCD est monté au-dessus du circuit imprimé à l'aide d'entretoises de hauteur appropriée. L'ensemble du circuit est logé dans un boîtier en plastique référencé G422.

Il faudrait alors surdimensionner les installations, avec les conséquences économiques que cela entraîne.

Il est donc préférable de compenser cette puissance réactive en améliorant le **facteur de puissance**, par l'installation de systèmes « produisant » de l'énergie réactive. Ces systèmes peuvent être des condensateurs, des ensembles d'inductances et de condensateurs, des machines tournantes ou des compensateurs statiques.

En France, la tarification des clients pour lesquels la livraison de l'énergie est faite en haute tension (puissance de

branchement à partir de 250 kVA) tient compte de l'énergie réactive. L'énergie réactive consommée au-delà d'un seuil fixé à $\tan(\varphi) \leq 0,4$ fait l'objet d'une facturation. Le but de cette facturation est d'indemniser le transporteur de

l'énergie pour les pertes causées sur le réseau, et d'inciter les clients à améliorer leur installation.

La notion de $\tan(\varphi) \leq 0,4$ est plus généralement exprimée par la valeur

du **facteur de puissance $\cos(\varphi)$** correspondant à **0,93**.

L'unité de puissance active est le **watt**, la puissance apparente est quant à elle exprimée en voltampère, noté **VA**.

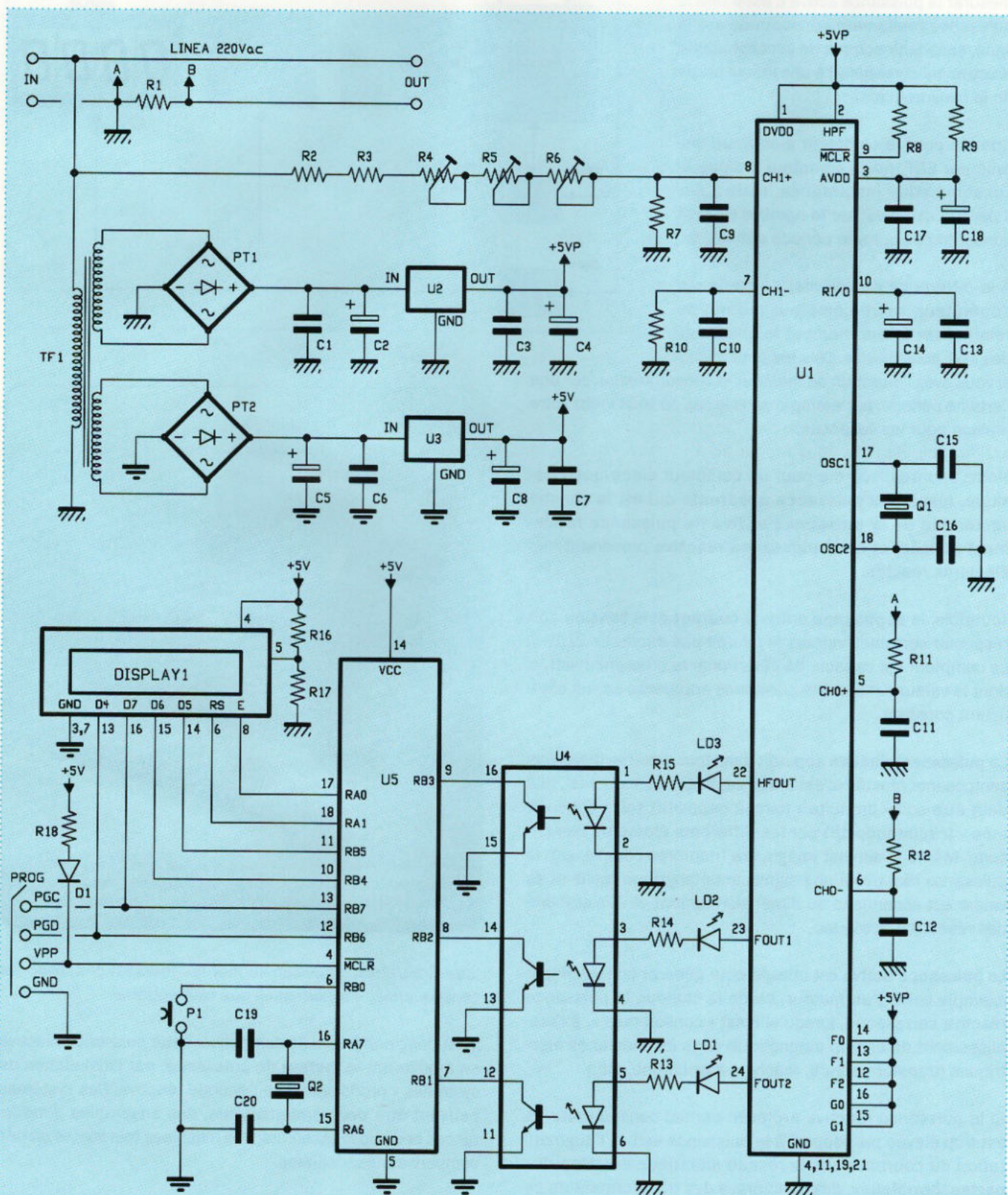


Schéma électrique de notre compteur d'énergie.

La puissance réactive s'exprime en voltampère réactif (VAR, var, ou VAR).

Le facteur de puissance (λ) est une caractéristique d'un récepteur électrique. Pour un dipôle électrique alimenté en régime de courant variable au cours du temps (sinusoïdal ou non), il est égal à la puissance active consommée par ce dipôle divisée par le produit des valeurs efficaces du courant et de la tension (puissance apparente). Il est toujours compris entre 0 et 1.

$$\lambda = P / (U \cdot I)$$

En particulier, si le courant et la tension sont des fonctions sinusoïdales du temps, le facteur de puissance est égal au cosinus du déphasage entre le courant et la tension :

$$\lambda = \cos(\varphi)$$

Le facteur de puissance est un paramètre qui rend compte de l'efficacité qu'a un dipôle pour consommer de la puissance lorsqu'il est traversé par un courant.

Pour bien comprendre cette notion, faisons une comparaison mécanique avec le « facteur d'embrayage » d'une boîte de vitesses :

- lorsque la pédale d'embrayage est enfoncée, le moteur tourne (le courant circule) mais ne transmet aucune puissance au véhicule, le facteur de puissance est nul ;
- lorsque la pédale d'embrayage est relevée, le moteur tourne et toute sa puissance produite est transmise au véhicule en une puissance motrice, le facteur de puissance vaut 1 ;
- lorsque l'embrayage patine, nous nous trouvons dans une situation intermédiaire, cela correspond au cas où le facteur de puissance est compris entre 0 et 1.

Rappelons que, lorsque le réseau alimente des utilisateurs qui relient des ampoules à incandescence, des cuisinières électriques, des fours électriques, ou des fers à repasser ou encore des machines à laver au réseau électrique, la puissance apparente est quasiment égale à la puissance active.



Examinons plus en détails notre montage. Il s'agit d'un dispositif muni d'un afficheur LCD sur lequel nous pouvons visualiser à chaque instant la consommation électrique (comme le fait un compteur électronique de l'opérateur historique).

C'est donc un instrument de mesure fiable qui est capable de donner des indications conformes à celles du compteur officiel.

Pour réaliser la mesure de la consommation électrique de la ligne, le montage doit être placé en série avec cette dernière. De plus, il doit être disposé en aval du compteur officiel et du disjoncteur principal de la maison ou de l'appartement.

Notre montage est prévu pour une mesure monophasée (une seule phase). Par conséquent, vous pouvez l'utiliser dans des réseaux électriques classiques jusqu'à environ 6 kW, étant donné que cela correspond à la plupart des abonnements électriques.

Le schéma électrique

Étudions maintenant la structure de notre circuit. Le cœur du montage est le circuit intégré U1, un **MCP3905** fabriqué par Microchip et spécialement conçu pour des applications concernant les compteurs d'énergie électrique.

Ce composant est capable de mesurer la consommation réelle d'électricité en se basant sur la lecture de la puissance active, indépendamment de l'angle de déphasage entre la tension et le courant et donc du « $\cos\varphi$ » (par contre les anciens compteurs électromécaniques indiquaient la puissance active pour un facteur de puissance défini à 0,93).

Non seulement notre montage permet de mesurer la consommation instantanée, mais en plus il est doté d'une sortie spécialement conçue pour « piloter » un compteur électromécanique de type odomètre mécanique.

Pour être exact, le MCP3905 dispose de **3 sorties**, dont chacune produit des **impulsions dont la fréquence dépend strictement de la consommation enregistrée** sur la ligne. Plus la consommation est grande, plus la fréquence des impulsions est grande et vice-versa.

Pour visualiser la consommation énergétique, nous devons utiliser un convertisseur qui nous permette de déterminer le courant absorbé en fonction des impulsions fournies par le MCP3905. Pour cela, nous faisons appel à un microcontrôleur de type PIC16F88, toujours de chez Microchip, qui permet d'interpréter les impulsions et d'en déduire les valeurs numériques correspondantes afin de les afficher sur l'écran LCD.

Pour fonctionner correctement, le MCP3905 doit être directement couplé à la ligne d'alimentation secteur, raison pour laquelle nous avons dû recourir aux circuits que vous apercevez dans le schéma. Autrement dit, nous devons mettre la masse du MCP3905 sur la ligne AC tout en évitant de relier le reste de la masse du montage à la ligne AC. Nous avons donc utilisé un couplage optique entre les sorties du circuit U1 et les entrées du microcontrôleur.

Cependant, cela nécessite l'utilisation de deux sources d'alimentation séparées galvaniquement. La première, qui alimente le circuit U1 (+5VP), est fournie par l'un des enroulements secondaires

Le circuit intégré MCP3905

Ce circuit intégré, produit par la firme Microchip, permet de mesurer la puissance active d'un appareil connecté à une ligne électrique monophasée. Il comporte deux convertisseurs A/D 16 bits de type Delta-Sigma, dont l'un lit la différence de potentiel aux bornes d'une résistance montée en série avec la ligne secteur. Cette configuration permet de calculer le courant consommé sur la ligne. La chute de tension introduite par la résistance permet donc de calculer la puissance apparente consommée par la charge. La détection au niveau de l'entrée de la ligne permet d'effectuer une comparaison de phase. En comparant le retard entre la tension et le courant du réseau, il est possible de mesurer le décalage de phase et donc la relation entre la puissance apparente mesurée et la puissance active, et donc le calcul de cette dernière.

Ce composant a été conçu pour réaliser des compteurs d'énergie électrique, il communique la consommation instantanée à l'aide de 3 sorties qui fournissent des impulsions de fréquences directement proportionnelles à la consommation. Ce circuit permet de piloter des compteurs électromécaniques comportant des roues codeuses, le circuit produit des impulsions périodiques en fonction de la consommation qu'il détecte.

L'entrée en courant est tellement sensible qu'il est possible d'utiliser une résistance en série (connectée entre les broches « CHO+ » et « CHO- ») de seulement 0,2 mΩ. Par conséquent, le circuit est capable de détecter des tensions aussi faibles que 0,2 mV pour chaque ampère consommé.

Tableau 1

G1	G0	Gain CH0	Tension maximale admissible sur CH0
0	0	1	± 470 mV
0	1	2	± 235 mV
1	0	8	± 60 mV
1	1	16	± 30 mV

La mesure est ainsi amplifiée par un amplificateur interne à gain programmable grâce à la combinaison des broches logiques G0 et G1. Le Tableau 1 montre les niveaux d'amplification en fonction du réglage des broches 15 et 16.

Le gain du capteur de courant doit être choisi en fonction de la puissance limite à détecter et de la résistance placée en série avec la ligne. Avec une résistance de 0,001 Ω et un courant de 30 A, correspondant à la puissance maximale d'un compteur électrique classique à usage domestique, cela nous donne une tension de 0,03 V, soit 30 mV. Ainsi, les broches de configuration de l'amplificateur à gain programmable sont mises à un niveau logique 1 (+ 5 V) afin d'obtenir une amplification maximale.

Les signaux des sorties sont au nombre de trois. Les sorties « FOUT1 » et « FOUT2 » génèrent des impulsions de niveau logique bas, chacune de 275 ms de largeur. La sortie « HFOUT » génère une fréquence plus élevée avec des impulsions de niveau haut, chacune de 90 ms de largeur.

Tableau 2

F1	F0	Fc (Hz)	Fc (Hz) avec un quartz de 3,58 MHz	Fout avec les entrées à la valeur maximale DC	Fout avec les entrées à la valeur maximale AC
0	0	$fclock/2^{21}$	1,71	0,74	0,37
0	1	$fclock/2^{20}$	3,41	1,48	0,74
1	0	$fclock/2^{19}$	6,83	2,96	1,48
1	1	$fclock/2^{18}$	13,66	5,93	2,96

du transformateur TF1, par le pont redresseur PT1 et par le régulateur U2 (7805). Il s'agit d'une tension stabilisée de 5 V, elle alimente aussi les LED internes de l'optocoupleur U4.

La seconde alimentation, qui est sensiblement identique à la précédente, est dérivée de l'autre secondaire du transformateur, du pont de diodes PT2 et du régulateur U3 (7805). Nous obtenons ainsi une tension stabilisée de 5 V nécessaire au fonctionnement du microcontrôleur, de l'afficheur LCD

et des sorties (transistors internes) de l'optocoupleur U4. Notez que cette tension est complètement isolée de la première.

L'afficheur LCD utilisé est de type standard, 16 caractères sur une ligne doté d'un contrôleur Hitachi HD44780 ou compatible. Sa gestion est un peu particulière, en effet il est géré uniquement sur 4 bits (D4 à D7).

Pour cela nous utilisons 4 lignes du microcontrôleur RB4, RB5, RB6, RB7

configurées en sortie pour l'envoi des données et deux lignes supplémentaires RA0 et RA1 configurées en sortie pour la gestion des signaux de commande.

Avant de continuer l'analyse du schéma électrique, arrêtons-nous quelques instants sur le fonctionnement de l'afficheur LCD, plus précisément sur la logique de communication entre l'afficheur et le microcontrôleur qui s'effectue sur 4 bits ou 8 bits pour les données (4 bits dans notre cas) plus 3 lignes de commandes R/W, RS, et E.

Tableau 3

F2	F1	F0	HFc (Hz)	HFc (Hz) avec un quartz de 3,58 MHz
0	0	0	$fclock/2^{15}$	109,25
0	0	1	$fclock/2^{15}$	109,25
0	1	0	$fclock/2^{15}$	109,25
0	1	1	$fclock/2^7$	27.968,75
1	0	0	$fclock/2^{16}$	219,51
1	0	1	$fclock/2^{16}$	219,51
1	1	0	$fclock/2^{16}$	219,51
1	1	1	$fclock/2^{16}$	219,51

Typiquement, les compteurs utilisent les 2 premières sorties, la troisième sert essentiellement à l'étalonnage du circuit d'entrée (CH1+ et CH1-), car elle fournit une fréquence plus élevée et donc plus facilement mesurable par des fréquencemètres ou des oscilloscopes.

Les sorties « FOUT1 » et « FOUT2 » génèrent une impulsion pour chaque kWh consommé et sont utilisées seulement pour piloter les compteurs électromécaniques. Par contre la sortie « HFOUT » fournit des impulsions qui dépendent de la consommation instantanée.

La fréquence exacte des sorties « FOUT1 » et « FOUT2 » est donnée par la formule suivante :

$$f_{out} (Hz) = (8,06 * V_o * V_1 * G * F_c) / V_{ref}^2$$

où « V_o » est la tension mesurée entre les broches 5 et 6 (CH0+ et CH0-) ; V_1 la tension mesurée entre les broches 8 et 7 (CH1+ et CH1-) ; G est le gain du canal 0 (capteur de courant au niveau de CH0+ et CH0-) et F_c la constante de fréquence sélectionnée. Cette dernière dépend du paramétrage des broches F0 et F1, et représente le facteur de division de la fréquence d'horloge

(Fclock) connectée entre les broches 17 et 18, comme indiqué dans le Tableau 2.

La fréquence de la sortie « HFOUT » de calibration est donnée par la formule suivante :

$$f_{outHF} (Hz) = (8,06 * V_o * V_1 * G * HF_c) / V_{ref}^2$$

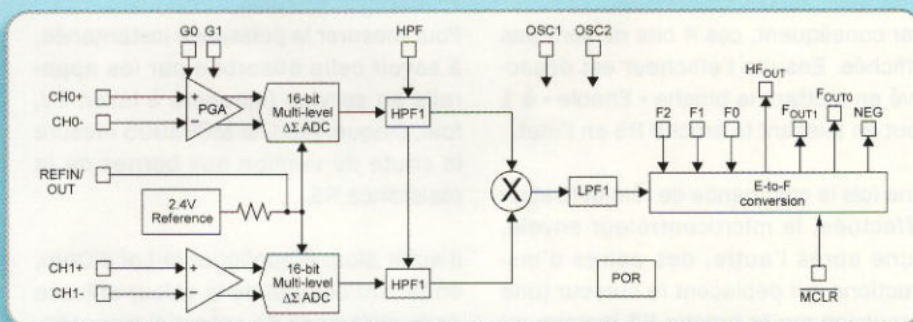
Dans ce cas, F_c dépend du paramétrage des broches F0, F1 et F2, comme indiqué dans le Tableau 3. Dans notre circuit, F1 et F0 sont au +5 V, le paramétrage correspond à « 11 » ($F_c = fclock / 2^{18}$).

Les formules et les tableaux montrent que la fréquence générée par le circuit intégré dépend directement de la tension mesurée sur les entrées, donc de l'amplitude de la tension du réseau et du courant consommé par la charge, soit le produit $V * I$. La fréquence des impulsions est fonction directement de la puissance, même si qu'une partie de celle-ci est prélevée par le MCP3905.

Les valeurs maximales des fréquences indiquées dans les tableaux sont celles obtenues lorsque se présente sur les entrées l'amplitude maximale prévue.



DVDD	1	24	FOUT0
HPF	2	23	FOUT1
AVDD	3	22	HFOUT
NC	4	21	DGND
CH0+	5	20	NEG
CH0-	6	19	NC
CH1-	7	18	OSC2
CH1+	8	17	OSC1
MCLR	9	16	G0
REFIN/OUT	10	15	G1
AGND	11	14	F0
F2	12	13	F1



La première R/W (Read/Write) détermine si nous souhaitons recevoir des données du LCD (read = lire) ou afficher des caractères, c'est-à-dire écrire (write) sur l'écran LCD.

Dans notre cas, nous devons simplement écrire les caractères (données) arrivant du microcontrôleur. D'après la documentation du constructeur de l'afficheur, la broche 7 (R/W) doit être à un niveau logique bas (0) pour que l'afficheur soit en mode écriture. Il suffit donc de relier la broche 7 à la masse du montage.

L'afficheur reçoit les caractères dans sa mémoire tampon (buffer) pour les afficher.

La commande de l'afficheur doit définir la position de chaque caractère et le caractère lui-même. Autrement dit, pour afficher un caractère, nous devons toujours envoyer une paire de données, la première paire correspond à la position du caractère et la seconde paire correspond au contenu du caractère (par exemple la lettre A ou le chiffre 1, etc.).

Cela est géré par la broche RS de l'afficheur qui permet une sélection du registre (commandes ou données), la broche RS permet donc de distinguer les données arrivant sur le bus, à savoir s'il s'agit de commandes à interpréter ou de caractères à visualiser.

Par exemple, une commande correspond à un octet qui déplace le curseur ou réinitialise l'écran LCD, tandis que les caractères à visualiser correspondent bien évidemment au texte s'affichant sur l'écran LCD.

Ainsi, le microcontrôleur met la broche RS à un niveau haut (1 logique) lorsqu'il envoie des commandes sur le bus de données (broches RB4 à RB7), c'est-à-dire les instructions que l'afficheur LCD doit effectuer (par exemple l'avancement du curseur ou la position à partir de laquelle le PIC doit commencer à écrire les caractères).

Lorsque la broche RS est mise à un niveau bas (0 logique), l'afficheur sait que les données provenant des broches RB4 à RB7 du microcontrôleur sont des caractères à afficher.

La broche E (Enable) correspond à l'entrée de validation et est utilisée pour activer l'afficheur si elle est placée à un niveau logique 0, ou le désactiver si elle est à un niveau logique 1. Cette broche E autorise donc l'écriture sur les registres de l'afficheur.

Avant d'écrire un ou plusieurs caractères sur l'afficheur, le microcontrôleur doit toujours réinitialiser le contrôleur de l'afficheur avec une instruction spéciale (0000) qui envoie ces 4 bits sur le bus de données D4 à D7 après avoir mis à 0 la broche E pour activer l'écran et à un niveau haut (1) la broche RS de manière à interpréter les données arrivant sur le bus comme une commande.

Par conséquent, ces 4 bits ne sont pas affichés. Ensuite l'afficheur est désactivé en mettant la broche « Enable » à 1 tout en laissant la broche RS en l'état.

Une fois la commande de réinitialisation effectuée, le microcontrôleur envoie, l'une après l'autre, des paires d'instructions qui déplacent le curseur (une impulsion sur la broche RS indique au driver HD44780 que les informations qu'il reçoit doivent être interprétées comme des commandes) et qui écrivent les caractères (RS est mise à 0, le driver sait ainsi que les données reçues doivent être affichées sur la matrice à cristaux liquides, ce sont des caractères et non des commandes).

Comme cela est requis, chaque instruction est cadencée par l'impulsion d'activation sur la broche E (Enable).

Outre les broches décrites précédemment, l'afficheur en comporte 5 autres.

Deux broches sont utilisées pour l'alimentation (la broche 4 ou V_{DD} doit être reliée au + 5 V et la broche 3 ou GND doit être connectée à la masse du montage).

La broche 5 ou V_0 doit être reliée au point milieu d'un potentiomètre, l'une des extrémités de ce dernier est reliée au + 5 V et l'autre extrémité à la masse. La broche V_0 permet ainsi d'ajuster le contraste du LCD.

Enfin, les deux dernières broches 1 et 2 respectivement « BL+ » et « BL- » doivent être connectées au + 5 V (broche 1) et à la masse (broche 2). Ces 2 broches permettent un rétroéclairage de l'écran afin de distinguer les caractères dans un environnement obscur.

Le fonctionnement du montage

Le circuit doit être connecté de telle sorte que les 2 points de l'entrée « IN » sont reliés à la ligne électrique provenant du disjoncteur (ou différentiel) et les 2 points de la sortie « OUT » doivent alimenter la charge, en l'occurrence il s'agit de l'installation électrique de la maison ou de l'appartement. Si vous inversez les connexions, vous pouvez rencontrer des problèmes.

Pour mesurer la puissance instantanée, à savoir celle absorbée par les appareils en service (machine à laver, TV, four, plaque, etc.), le MCP3905 mesure la chute de tension aux bornes de la résistance R1.

Il suffit alors d'appliquer la Loi d'Ohm, en tenant compte de la valeur efficace de la différence de potentiel mesurée.

Pour rappel la **Loi d'Ohm** est formulée de la manière suivante :

la différence de potentiel ou tension U (en volts) aux bornes d'une résistance R (en ohms) est proportionnelle à l'intensité du courant électrique I (en ampères) qui la traverse, soit :

$$U = R \cdot I$$

En courant alternatif et en régime sinusoïdal, la Loi d'Ohm s'exprime par la formule : $U = Z \cdot I$

où Z correspond à la somme des impédances des appareils connectés sur la ligne électrique.

Par exemple, étant donné que la résistance de mesure R1 vaut 0,001 Ω , si le MCP3905 détecte aux bornes de la résistance une chute de tension de crête de 0,01414 V, cela signifie que le courant dans la ligne atteint une valeur maximale de 14,14 A.

Étant donné que nous devons déterminer la puissance efficace, qui correspond à la valeur « RMS » du courant absorbé par la ligne électrique, nous avons besoin de connaître la valeur efficace de la tension. Pour une grandeur sinusoïdale, la valeur efficace de la tension équivaut à environ 0,707 fois celle de la tension de crête.

Un calcul rapide nous indique qu'une tension de crête de 0,01414 V correspond à une tension efficace de 0,01 V_{RMS} .

De même, le courant maximal étant de 14,14 A, le courant efficace vaut 10 A ($14,14 \cdot 0,707$).

Par conséquent, si le MCP3905 détecte aux bornes de R1 (entre les points A et B, c'est-à-dire les broches 5 et 6 de U1) une chute de tension crête de 0,01414 V, cela signifie que la ligne électrique absorbe un courant efficace de 10 A.

Pour une tension secteur de 220 V_{eff} environ, la consommation électrique correspondante sur la ligne est de 2200 W.

Jusqu'à présent, tout est simple et clair, cependant le MCP3905 ne s'arrête pas là. En fait, nous avons dit que ce circuit intégré pouvait calculer la puissance réelle consommée, c'est-à-dire la puissance active.

Mais comment fait-il dans la pratique ? Lorsqu'il mesure la chute de tension aux bornes de R1, il détecte en fait la puissance apparente, car la mesure est basée sur l'absorption qui résulte de la somme vectorielle de la puissance active et de la puissance réactive et du courant circulant.

Pour calculer la puissance active, qui résulte de la dissipation dans la partie résistive des appareils, le circuit intégré

Le compteur Linky, la révolution technologique aux multiples avantages

Linky est le compteur électrique intelligent qui est en cours de déploiement par ENEDIS, le gestionnaire du réseau électrique en France. Il s'agit avant tout d'un compteur communicant, il peut donc transmettre votre consommation de manière autonome à votre fournisseur d'électricité. Par conséquent, vous êtes exempté des fastidieux rendez-vous de relève du compteur pour lesquels vous devez être présent.

Qui plus est, c'est un compteur intelligent capable de s'auto-paramétrer. Ainsi, les techniciens peuvent effectuer la mise en service de votre compteur à distance ainsi que certaines réparations sans se déplacer. Il facilite les démarches aussi bien pour vous que pour votre fournisseur.

Un autre avantage de ce compteur est qu'il permet d'adapter les heures creuses et les heures pleines en fonction des besoins du client. Il permet aussi une facturation sur une consommation réelle et non plus estimée.

Linky fait partie de la catégorie des compteurs communicants (AMM), c'est-à-dire pouvant être interrogés et actionnés à distance. Pour cela, il utilise la technologie CPL sur le réseau basse tension, comme premier niveau de communication. Un deuxième niveau de communication est assuré via le réseau de téléphonie mobile GPRS entre le concentrateur (installé dans un poste de transformation et de distribution) et le système central. À terme, lorsque les 35 millions de compteurs seront remplacés, il y aura environ 700 000 concentrateurs installés. Cette génération de compteurs communicants est présentée comme une des étapes vers les réseaux intelligents.

Le compteur Linky, pour calculer l'énergie consommée par le client, mesure les courants et les tensions pour chaque phase de l'installation. Une seule phase est concernée en monophasé et les 3 phases sont mesurées en triphasé. Ces mesures sont effectuées par calcul de la valeur efficace vraie du courant et de la tension.

La facturation est calculée en kilowatts par heure (kWh). Il s'agit du produit de la puissance active par le temps.

Le compteur incorpore un dispositif électronique de coupure qui contrôle le dépassement de l'intensité du courant (apparent à un disjoncteur électronique) dont le seuil de détection est fixé à la valeur du courant correspondant à la puissance souscrite par l'abonnement du client. La mesure est effectuée par comparaison de l'intensité efficace instantanée avec l'intensité de référence déduite de la puissance de référence souscrite.

Cependant, le compteur Linky fait l'objet de diverses critiques formulées par des associations : mise en cause possible de la vie privée, mise en cause possible de la santé publique due aux ondes électromagnétiques, pollution environnementale car il faut se débarrasser de millions de compteurs actuels électromécaniques en parfait état de marche.



doit mesurer le déphasage entre la tension et le courant. Les broches « CH0+ » et « CH0- » du MCP3905 mesurent la chute de tension aux bornes de la résistance R1 et donc le module du courant absorbé et la puissance apparente.

Avec les broches « CH1+ » et « CH1- » (respectivement les broches 8 et 7), il mesure le déphasage. Le mécanisme de lecture est relativement simple.

Le circuit intégré mesure la chute de tension aux bornes de R1, nous obtenons une composante sinusoïdale correspondant au courant absorbé sur la ligne, étant donné que la chute de tension est due au courant circulant.

La lecture de la tension à l'entrée, c'est-à-dire avant la résistance R1, détermine la phase de la sinusoïde du réseau.

Étant donné que le circuit se trouve en amont de la ligne domestique et donc de l'installation électrique où doit s'effectuer la mesure de la puissance, la tension prélevée par le MCP3905 au moyen des résistances R2, R3, R4, R5, R6 est en phase avec celle de la ligne, nous pouvons la prendre comme référence.

Ainsi, la différence de potentiel aux bornes de R1, qui dépend du courant, peut être déphasée d'un angle qui varie en fonction de la présence ou de l'absence de composants réactifs sur la ligne.

Par conséquent, le circuit U1 lit la différence de phase entre la tension d'entrée et la chute de tension dans R1. En mesurant l'écart temporel entre les deux tensions et connaissant la fréquence du réseau, nous pouvons déterminer le déphasage.

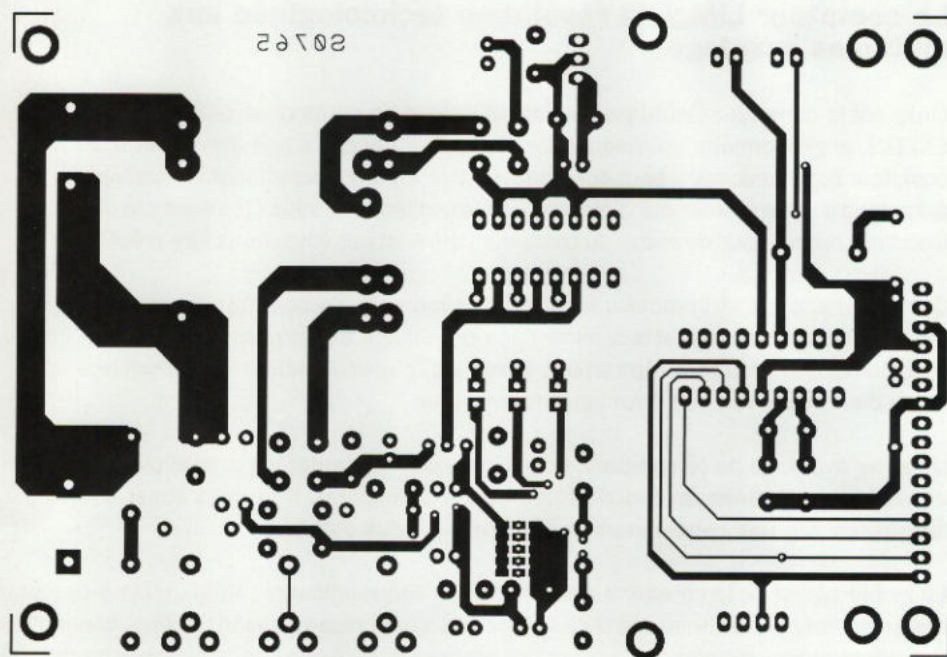
Par exemple, lorsque la fréquence est de 50 Hz, la période (360°) dure 20 millisecondes. Supposons que nous ayons un retard de 1 ms, cela correspond à un déphasage de 18°. En connaissant ainsi l'angle de déphasage, le MCP3905 est capable de déterminer, à partir de la composante réactive dont il connaît le module, soit la partie réactive ou la partie active.

Plan de montage du compteur d'énergie

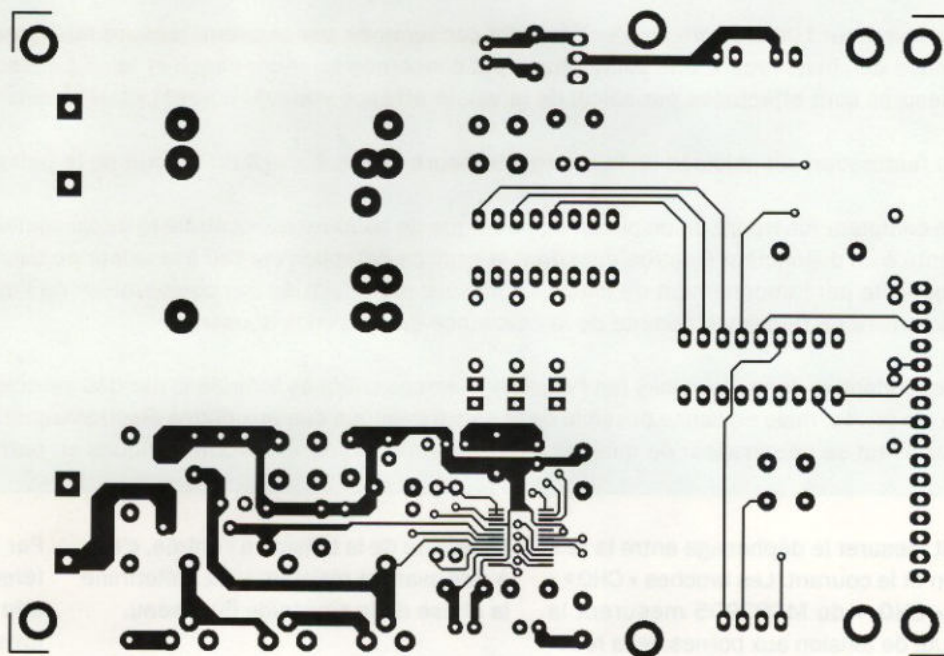
Tous les composants, y compris le transformateur d'alimentation, sont montés sur un seul circuit imprimé.

L'afficheur LCD est disposé du côté droit du circuit imprimé, il est fixé au moyen de deux entretoises et relié électriquement au circuit imprimé principal au moyen d'une barrette mâle à 16 pôles.

Le circuit intégré MCP3905 est en boîtier CMS, il est soudé du côté composant du circuit imprimé.



Circuit imprimé du compteur d'énergie à l'échelle 1 : 1 côté soudures.



Circuit imprimé du compteur d'énergie à l'échelle 1 : 1 côté composants.

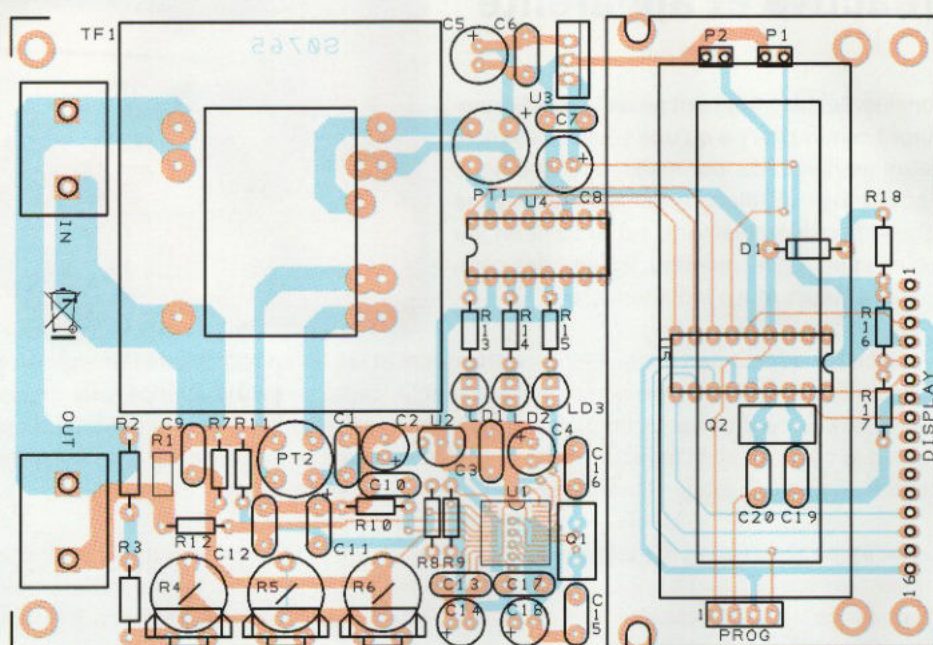
Liste des composants du compteur d'énergie

R1.....shunt 0,001 Ω 1 W
R2.....330 k Ω
R3.....330 k Ω
R4.....trimmer monotour 300 k Ω
R5.....trimmer monotour 100 k Ω
R6.....trimmer monotour 1 k Ω
R7.....1 k Ω

R8.....1 k Ω
R9.....10 Ω
R10.....1 k Ω
R11.....1 k Ω
R12.....1 k Ω
R13.....820 Ω
R14.....820 Ω
R15.....820 Ω
R16.....4,7 k Ω
R17.....560 Ω

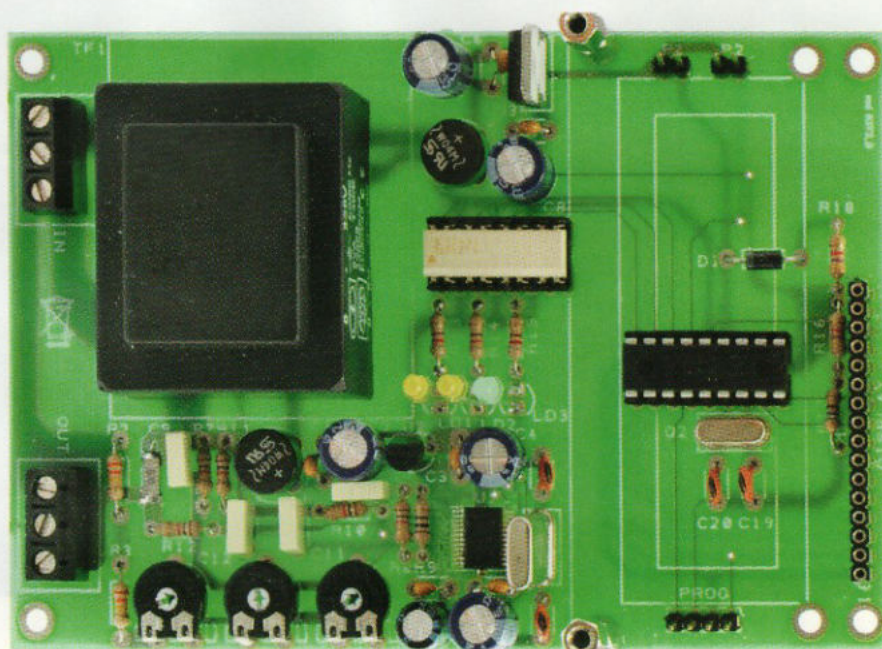
R18.....4,7 k Ω

C1.....100 nF multicouche
C2.....220 μ F/35 V électrolytique
C3.....100 nF multicouche
C4.....220 μ F/35 V électrolytique
C5.....220 μ F/35 V électrolytique
C6.....100 nF multicouche
C7.....100 nF multicouche
C8.....220 μ F/35 V électrolytique



Plan de câblage des composants du compteur d'énergie.

Photo de l'un de nos prototypes du compteur d'énergie.



C933 nF/100 V polyester
C1033 nF/100 V polyester
C1133 nF/100 V polyester
C1233 nF/100 V polyester
C13100 nF multicouche
C1410 µF/10 V électrolytique
C1522 pF céramique
C1622 pF céramique
C17100 nF multicouche
C18220 µF/35 V électrolytique
C1910 pF céramique
C2010 pF céramique

D11N4007
PT1pont de diodes W04M

PT2pont de diodes W04M
Q1quartz 3,579545 MHz
Q2quartz 20 MHz
U1MCP3905
U278L05
U37805
U4TLP627-4
U5PIC16F88 (MF765)
LD1LED 3mm jaune
LD2LED 3mm jaune
LD3LED 3mm verte
TF1transformateur 220 AC/2 x 9 VAC 3VA
DISPLAY1..CDL4162
P1bouton poussoir

P2 bouton poussoir

Divers :

Bornier 2 pôles au pas de 10 mm (x 2)
Support circuit intégré 2 x 8 broches
Support circuit intégré 2 x 9 broches
Barrette mâle 2 pôles (x2)
Barrette mâle 4 pôles
Barrette femelle 16 pôles
Barrette mâle 16 pôles
Embase mâle pour châssis
Embase femelle pour châssis
Porte fusible pour châssis
Fusible 10 A rapide

Puissance active, réactive et apparente

La puissance électrique peut être considérée différemment selon le fonctionnement du circuit. Dans un circuit à courant continu il n'y a qu'une seule puissance, elle est consommée par les utilisateurs en raison de leur résistance électrique ou de l'absorption des inductances et des moteurs électriques. Dans un régime variable ou alterné, les choses changent considérablement, car entrent en jeu certains composants qui n'ont aucun effet en régime continu. Il s'agit des composants réactifs, typiquement les condensateurs et les inductances.



En régime variable, il se produit quelque chose de très particulier. Les condensateurs et les bobines stockent l'énergie électrique, pour ensuite la restituer au circuit. En fonction du type de composant réactif, capacité ou inductance, une grande quantité d'énergie peut être prélevée de la source d'alimentation pour ensuite être restituée. Plus précisément, l'énergie stockée est directement proportionnelle à la capacité et à l'inductance, mais est inversement proportionnelle à la résistance parasite des composants, c'est-à-dire à la résistance totale vue en série par la source d'alimentation.

Dans le cas du régime sinusoïdal, sur les lignes électriques domestiques et industrielles, nous trouvons des appareils inductifs et capacitifs qui provoquent des phénomènes pouvant parfois créer des problèmes sur la ligne électrique. Par exemple, des charges très inductives ou très capacitives peuvent provoquer des pics d'absorption, ou au contraire injecter sur la ligne une grande quantité d'énergie stockée. Dans les deux cas, il y a une surcharge de la ligne électrique.

Les appareils capacitifs absorbent une grande quantité de courant dès l'instant où ils sont alimentés, tandis que des appareils inductifs déchargent le courant accumulé lorsque la tension de la ligne diminue. En régime sinusoïdal stable, nous pouvons ainsi définir trois puissances électriques : la puissance active, la puissance réactive et la puissance apparente. La première correspond à la puissance consommée (dissipée) par le circuit, généralement due à la résistance des câbles et de tous les éléments résistifs (ampoules à incandescence, résistances des moteurs électriques, etc.). La puissance réactive provient des composants réactifs, à savoir les inductances et les condensateurs. Ils ne consomment pas d'énergie, ils la stockent simplement pour la restituer ensuite à la ligne électrique. Enfin, la puissance apparente est la somme vectorielle des deux modules précédents (voir le graphe plus loin).

Les compteurs des gestionnaires d'électricité détectent cette dernière, cependant ils doivent mesurer et afficher la puissance active. Pour cela, ils se basent sur le fait que le déphasage entre la tension et le courant est constant, et que pour des composants actifs et réactifs, il existe un rapport constant entre la puissance mesurée, la puissance apparente et la puissance réelle (active).

Le déphasage est l'angle formé par la tension et le courant, il dépend de la présence de composants réactifs. Une inductance produit un déphasage du courant de 90° en retard par rapport à la tension, tandis qu'un condensateur produit un déphasage de la tension de 90° en retard par rapport au courant.



Ci-contre, notre prototype assemblé. Le trimmer sur la droite permet d'effectuer l'étalonnage du circuit qui est nécessaire pour compenser la tolérance de la résistance de shunt de 0,001 Ω. Tout le courant absorbé sur la ligne circule à travers cette résistance.

En fait, dans la représentation vectorielle de la figure 1, le module de la puissance active est représenté par le segment A, celui de la puissance réactive par le segment B et celui de la puissance apparente par le segment C.

La trigonométrie nous enseigne que le rapport A/C correspond au cosinus de l'angle formé entre l'axe des abscisses (axe des X) et le segment C (angle φ), tandis que le rapport B/C correspond au sinus.

En connaissant le module de la puissance apparente et le déphasage φ , nous pouvons déterminer la puissance active selon la formule suivante :

$$P_a = A = C * \cos(\varphi)$$

Dans l'exemple précédent, un déphasage de 18° correspond à une puissance active de : $P_a = P_{app} * 0,951$

où P_a est la puissance active et P_{app} la puissance apparente.

À ce stade, nous avons étudié la manière dont le circuit intégré Microchip détermine la partie de la puissance active comprise dans la puissance apparente qu'il mesure.

Examinons maintenant le reste du montage, à savoir comment le MCP3905 communique sa mesure.

L'amplitude du déphasage (ϕ) entre la tension et le courant dépend du rapport entre la réactance capacitive ($1/2\pi fC$) et la réactance inductive ($2\pi fL$). Si les deux sont égales, le déphasage est nul et le circuit se comporte comme s'il était seulement constitué par une seule résistance. Autrement dit, la puissance absorbée est toujours dissipée.

La quantité de puissance réactive, qui ne pose pas de problème au niveau de la consommation car elle est en fait stockée mais pas dissipée, peut cependant créer de nombreuses difficultés pour la gestion des lignes.

En effet, si le déphasage entre la tension et le courant est trop grand, il ne permet pas de déterminer la consommation réelle d'électricité, mais seulement la consommation instantanée. Il en résulte que le gestionnaire d'électricité peut se retrouver avec des pics de tension sur deux des phases d'une ligne triphasée, car sur la 3^{ème} phase il y a des prélèvements et des réinjections de grandes quantités d'électricité dues aux composants réactifs. L'opérateur est alors incapable de dimensionner correctement les lignes ou de distribuer de manière uniforme les phases, car il ne dispose pas d'un paramètre de référence pour le calcul de la consommation.

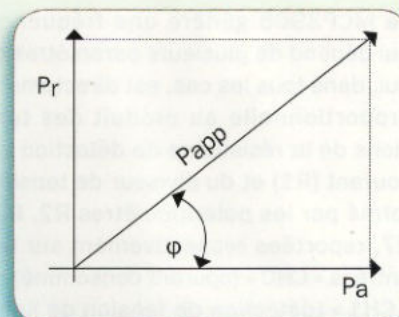
En outre, puisque les compteurs indiquent la consommation de la puissance active en considérant un certain déphasage déterminé par rapport à la puissance mesurée (apparente) et la puissance consommée (active), si l'angle ϕ est plus grand que celui admis, le compteur mesurera une puissance inférieure à la puissance réelle consommée.

C'est pourquoi l'une des techniques pour fausser la mesure de la puissance, en particulier avec les anciens compteurs à disque, consistait à enlever le condensateur de correction du facteur de puissance des moteurs électriques. En effet, cela permettait d'augmenter le déphasage entre le courant et la tension et donc les compteurs mesuraient des puissances inférieures à celles réellement consommées.

Aujourd'hui, avec les nouveaux compteurs électroniques, cela n'est plus possible, car ceux-ci contiennent des circuits capables de détecter le déphasage et enregistrer la puissance réactive absorbée, de manière à communiquer aux opérateurs la télémesure et démasquer les fraudeurs.

Pour ces raisons, tous les utilisateurs connectés aux lignes à usage public doivent présenter un certain angle de déphasage maximum entre la tension et le courant. En France, il est prévu que le cosinus de cet angle ($\cos\phi$ ou facteur de puissance) soit supérieur ou égal à 0,93. Les compteurs électriques sont paramétrés pour mesurer une puissance apparente avec un angle de déphasage entre la tension et le courant ayant un cosinus de 0,93.

En reliant des moteurs électriques de grande puissance sur la ligne, l'inductance globale introduite a tendance à déphaser le courant en retard par rapport à la tension, ce qui augmente l'angle et donc le facteur de puissance a tendance à diminuer. Pour contourner ce problème dans la pratique, une correction du facteur de puissance est effectuée en neutralisant la réactance inductive par l'ajout d'une capacité sur la ligne. Les deux se compensent mutuellement, même partiellement, car elles ont un effet inverse. Cela réduit l'angle de déphasage et donc le $\cos\phi$ est proche de 1.



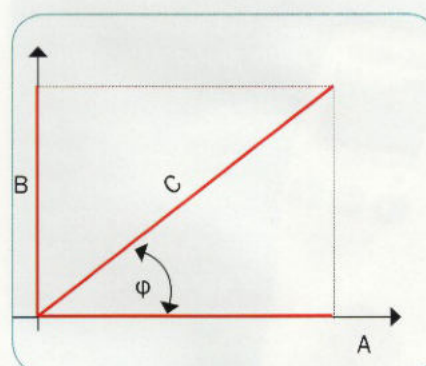
Comme nous l'avons mentionné précédemment, le circuit intégré comporte 3 sorties qui fournissent des impulsions dont la fréquence dépend de la puissance active absorbée.

Étant donné que le circuit de mesure est isolé électriquement du microcontrôleur, pour transmettre les impulsions nous utilisons 3 photocoupleurs, dont les transistors de sortie sont configurés en « collecteur ouvert », les résistances internes de pull-up du microcontrôleur servant alors de résistances de collecteur. Chaque fois que l'entrée reçoit un niveau logique haut, chaque optocoupleur transfère l'état sous forme de niveau bas à la ligne correspondante du PIC.

Un niveau logique 0 laisse l'optocoupleur dans un état bloqué et donc le collecteur de la sortie du phototransistor reste à un niveau haut.

Le programme du microcontrôleur analyse les impulsions et en déduit la quantité de puissance sur la base de la fréquence des impulsions.

Il prépare ensuite les données et les convertit en mots de 4 bits pour les envoyer à l'afficheur LCD afin que ce dernier visualise la valeur de la puissance consommée. Plus précisément, le microcontrôleur comptabilise le délai entre deux impulsions consécutives sur la sortie HFOUT.



Une fois l'intervalle de temps déterminé, il calcule et affiche la valeur de la puissance consommée. Il en résulte que plus la puissance consommée est grande, plus l'affichage est rafraîchi et inversement.

Le MCP3905 génère une fréquence qui dépend de plusieurs paramètres et qui, dans tous les cas, est directement proportionnelle au produit des tensions de la résistance de détection en courant (R1) et du diviseur de tension formé par les potentiomètres R2, R3, R7, reportées respectivement sur les entrées « CHO » (courant consommé) et « CH1 » (détection de tension de ligne secteur) puis vers les convertisseurs A/D respectifs.

En plus de ce produit, la fréquence dépend de la configuration des broches G0 et G1 qui déterminent le gain de l'entrée de l'amplificateur « CHO » (capteur de courant) et des broches F0, F1, F2 qui configurent le diviseur de fréquence.

En effet, la manière dont le MCP3905 est connecté dans notre montage permet de générer une fréquence qui correspond à 1000 impulsions par kWh.

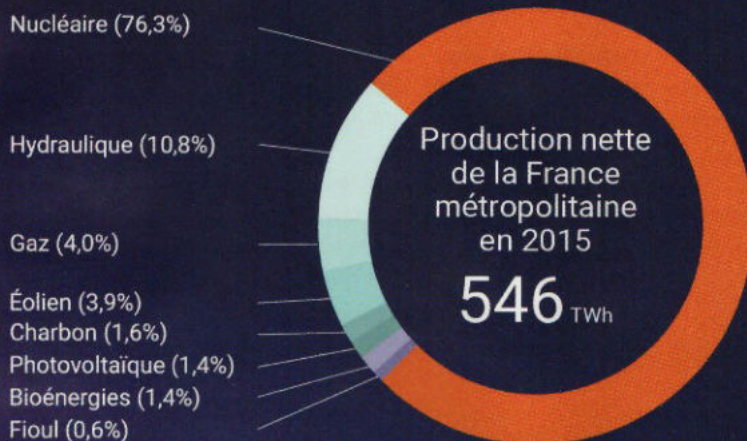
Dans le commerce

Dans le commerce, il existe plusieurs appareils à usage domestique capables d'indiquer la puissance consommée par une charge. Bien que ce soient des appareils très complets pouvant calculer le coût total de la consommation d'énergie, ils ne coûtent pas très cher et sont faciles à utiliser.

Le modèle E305EM5, commercialisé par la société Comelec, est parfaitement adapté pour un usage domestique. Cet appareil dispose d'un grand afficheur à cristaux liquides, il dispose d'une batterie de secours pour conserver les données. Il est capable de mémoriser jusqu'à 9999 heures d'enregistrement avec une consommation maximale de 999,9 kW/h. Il est adapté pour des charges allant jusqu'à 3,6 kW. Pour plus d'information visitez le site www.comelec.fr.



La production et la consommation d'énergie en France



Nous donnons ci-après les principaux chiffres à retenir, relatifs à la production, à la consommation ou encore aux exportations d'électricité pour l'année 2015 en France métropolitaine.

La production électrique repose à 76,3 % sur le nucléaire. La production d'électricité en France métropolitaine a atteint 546 TWh en 2015. Plus des 3/4 de cette production proviennent toujours du parc nucléaire de 58 réacteurs.

L'hydroélectricité reste la deuxième source d'électricité malgré une forte baisse de sa production par rapport à 2014 en raison d'un niveau de pluviosité bien plus faible.

La contribution des autres énergies renouvelables a fortement augmenté en 2015. Le parc éolien, dont la puissance raccordée au réseau a dépassé les 10 000 MW durant l'année, compte pour 3,9 % de la production électrique française. La production photovoltaïque a aussi connu en 2015 une croissance, mais sa part dans le mix électrique reste limitée.

C'est la production électrique des centrales à gaz qui a connu la plus forte croissance en 2015. Elle s'est élevée à 22,1 TWh, soit près de 4 % du mix national et à un niveau légèrement plus élevé que la production éolienne. Les centrales à gaz ont en effet été davantage sollicitées comme moyens

En considérant qu'un kWh correspond à une consommation électrique de 1000 W pendant une heure et qu'une heure vaut 3600 secondes, cela signifie qu'un kW de puissance absorbée génère des impulsions en sortie de 0,36 Hz.

Si la puissance consommée est de deux kW, la fréquence vaut alors 0,72 Hz, et ainsi de suite. Bien sûr, nous parlons des sorties « FOUT1 » et « FOUT2 ».

La sortie « HFOUT », en fonction du réglage, génère une fréquence 4 fois supérieure. Le microcontrôleur ne lit que les sorties « FOUT1 » et « HFOUT » sur les 3 disponibles.

La sortie « HFOUT », comme déjà évoqué, représente la puissance instantanée.

Sur la sortie « FOUT1 », le microcontrôleur lit les impulsions et à chaque impulsion incrémente le comptage des kWh consommés.

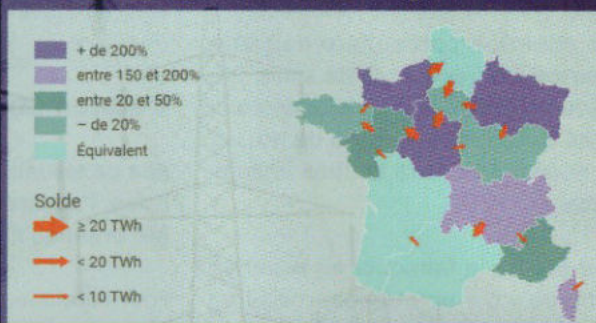
Ceux-ci sont stockés et conservés dans la mémoire EEPROM (de manière à retrouver les informations en cas de coupure de courant).

À tout moment, en appuyant sur le bouton P1, le comptage des kWh peut être remis à zéro, le comptage reprend alors à partir de 0.

Consommation brute de la France métropolitaine en 2015



Rapport production / consommation par région en 2015



d'appoint lors des périodes de forte demande. La production des centrales à charbon et à fioul a quant à elle légèrement augmenté malgré la fermeture des 6 dernières centrales à charbon de plus de 250 MW en France.

La consommation d'électricité en France métropolitaine s'est élevée à 475,4 TWh en 2015, soit 2,2 % de plus qu'en 2014. Cette hausse est principalement liée aux températures en moyenne plus fraîches en début d'année (entraînant un recours accru au chauffage) et plus chaudes en été (augmentant les besoins de froid).

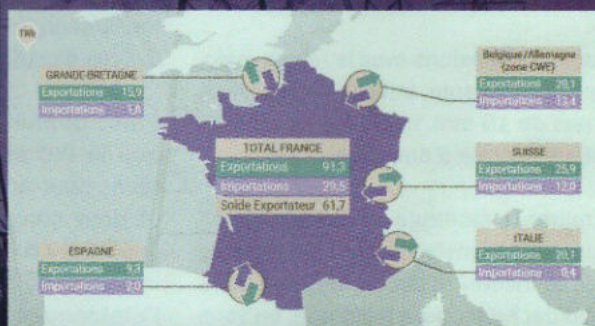
La croissance de la demande d'électricité s'est fortement ralentie depuis plusieurs décennies, notamment en raison de la tertiarisation de l'activité économique et de l'impact des actions de maîtrise de la demande d'énergie.

Elle reste toutefois très « thermosensible » : chaque degré Celsius en moins en hiver entraîne, selon RTE, un appel de puissance supplémentaire sur le réseau de 2 400 MW (soit l'équivalent de 2 réacteurs nucléaires).

Les échanges transfrontaliers d'électricité ont très légèrement augmenté en 2015, la France restant largement exportatrice nette avec 91,3 TWh exportés vers ses pays voisins et 29,6 TWh importés.

Cette augmentation des échanges est notamment due à une optimisation des flux dans la zone « CWE » (France,

Solde des échanges d'électricité avec les pays voisins



Allemagne, Autriche, Belgique, Pays-Bas, Luxembourg) et au renforcement de l'interconnexion entre la France et l'Espagne.

Réalisation pratique

Une fois les aspects théoriques du montage analysés, nous devons passer à la réalisation pratique.

Tous les composants tiennent sur un seul circuit imprimé y compris l'afficheur LCD et le transformateur.

Vous trouverez sur notre site www.electroniquemagazine.com dans le sommaire détaillé de la revue les typons des circuits imprimés, les fichiers Gerber (fabrication professionnelle), le programme « ET765.hex » pour le microcontrôleur ainsi que le code source en téléchargement gratuit.

Une fois le circuit imprimé fabriqué et percé, par photogravure pour les amateurs (typons) ou à l'aide des Gerber pour les professionnels, commencez par souder comme d'habitude les composants ayant un profil bas.

Pour cela aidez-vous du plan de câblage des composants. Commencez par souder les résistances, si vous n'êtes pas habitué avec le code des couleurs, munissez-vous d'un multimètre et mesurez la valeur de chaque résistance avant de la souder.

Cela permet d'éviter des erreurs dont parfois la recherche de panne peut prendre plusieurs heures, car nous

pensons toujours à un composant actif défectueux ou au microcontrôleur mal programmé ou tout simplement au programme qui ne fonctionne pas.

Continuez en soudant la diode D1, cette fois vérifiez la position de la bague (vers la gauche, vue de dessus).

Ensuite insérez et soudez les supports des circuits intégrés, les condensateurs non polarisés puis ceux polarisés.

Pour ces derniers vérifiez la polarité, le négatif (patte la plus courte) est marqué sur le boîtier et la patte la plus longue correspond toujours au plus d'un condensateur polarisé.

L'afficheur LCD est monté à l'aide de barrettes mâles/femelles à 16 pôles au pas de 2,54 mm, au-dessus du microcontrôleur U5 et du quartz Q2.

Une attention particulière doit être apportée à la résistance de shunt R1, elle doit être fabriquée à la main étant donné qu'il n'est pas facile de trouver un modèle de 0,001 Ω dans le commerce.

Vous pouvez la fabriquer en pliant en forme de « U » un morceau de fil de cuivre nu de 4 cm de longueur et de section 0,6 mm².

Si la valeur de la résistance de shunt n'est pas tout à fait correcte, cela ne pose pas de problème car les trimmers permettent d'étalonner le circuit afin de compenser les différences.

Pour les connexions avec le réseau électrique et la charge, utilisez des borniers au pas de 15 mm, capables de résister à une vingtaine d'ampères.

Terminez le montage de l'ensemble en soudant le transformateur. Vous devez ensuite insérer le circuit imprimé dans un boîtier en matière plastique, en vous assurant que les composants reliés directement à la tension du secteur soient bien isolés du reste du montage.

Vérifiez à l'aide d'un ohmmètre, sans mettre sous tension, **si les 2 masses sont bien isolées entre elles**, il doit y avoir une résistance infinie.

Il ne faut jamais oublier qu'une partie du montage est reliée directement au circuit 220 VAC. Cela nécessite donc un minimum d'attention, en particulier dans la recherche de pannes éventuelles.

Vous ne devez jamais toucher le montage avec vos mains lorsqu'il est sous tension !

En ce qui concerne l'installation, l'appareil doit être connecté au neutre (N) et à la phase (F ou L) de la ligne électrique.

La phase doit aller vers le bornier correspondant au point « A » de R1. Pour identifier la phase, utilisez un tournevis testeur de phase.

Normalement vous n'avez pas besoin de relier le montage à la terre, à moins que vous utilisiez un boîtier métallique. Dans ce cas, reliez le fil de terre (jaune/vert) au boîtier.

L'étalonnage

Afin de visualiser des indications réalistes, l'appareil doit être réglé et calibré avant qu'il puisse être utilisé.

Pour l'étalonnage, il est nécessaire de se procurer une charge de puissance connue et de regarder directement les indications de l'afficheur LCD.

Vous pouvez aussi, toujours avec la même charge connue, mesurer la fréquence sur la sortie « HFOUT » du MCP3905. Dans cette phase, le compteur d'énergie doit être seulement relié à la ligne électrique et à la charge de test.

Étant donné que nous considérons que le PIC est programmé correctement pour afficher la consommation et pour convertir les indications de la sortie « HFOUT » en une puissance instantanée absorbée, pour effectuer l'étalonnage vous devez insérer une charge connue en sortie et ajuster les trimmers jusqu'à ce que vous obteniez sur l'afficheur la puissance consommée de ladite charge.

Vous pouvez effectuer l'étalonnage d'une autre manière en connectant un fréquencemètre ou un oscilloscope

sur la sortie « HFOUT » (broche 22 de U1) et en alimentant le circuit. Sachant que le MCP3905 génère sur « FOUT1 » et « FOUT2 » 1000 impulsions pour chaque kWh consommé, si la charge est de 1000 W vous devez lire une fréquence de 0,36 Hz.

Dans tous les cas, la fréquence est donnée par la formule :

$$f = 0,36 \text{ Hz} * P_u / 1000$$

où P_u est la puissance de l'utilisateur (en W).

Par exemple, en utilisant une charge de test de 1,2 kW, la fréquence doit être de $0,36 \times 1,2 = 0,432$ Hz.

Étant donné que les valeurs sont faibles et donc difficilement mesurables par un fréquencemètre ou un oscilloscope, pour l'étalonnage il est plus commode d'utiliser la sortie « HFOUT » car la fréquence est 4 fois plus grande.

Pour une charge de test de 1 kW, vous devez mesurer sur « HFOUT » une fréquence d'environ 1,44 Hz, et pour une charge de test de 1,2 kW une fréquence d'environ 1,728 Hz.

Si la valeur mesurée ne correspond pas, il est alors nécessaire de retoucher le réglage des trimmers R4, R5 et R6, en accordant beaucoup d'attention car ils sont situés dans une zone du montage directement soumise à la tension du secteur.



Nous avons utilisé, pour la prise secteur et celle de la charge, des embases mâles/femelles pour châssis. Vous pouvez utiliser d'autres types de prises.

Conçu pour être utilisé lors des mesures avec l'analyseur de spectre décrit dans les précédents numéros 138 et 139, ce montage génère un bruit blanc et rose dont la fréquence de base peut atteindre 1 MHz.



Générateur de bruit blanc et rose

..... de Marco LANDONI

Lorsque nous vous avons présenté, dans les numéros 138 et 139 d'Electronique et Loisirs Magazine, le projet de transformation d'un oscilloscope en un analyseur de spectre, nous avons pensé en toute logique à décrire un générateur de bruit qui pourra être utilisé conjointement avec l'analyseur afin de vérifier la bande passante d'appareils audio.

Cet appareil complètera donc votre laboratoire de mesure, il s'agit d'un générateur de bruit (blanc et rose) comportant des gammes sélectionnables. Nous vous décrirons plus loin dans l'article la manière d'utiliser ce générateur de bruit conjointement avec l'analyseur de spectre.

Étant donné que l'analyseur de spectre dispose d'une bande passante de 5 MHz, ce qui est plus que suffisant pour travailler en basse fréquence (BF), nous avons pensé vous proposer un générateur de bruit qui couvre cette gamme de fréquences, afin d'effectuer diverses mesures en utilisant simultanément les deux instruments.

Vous pourrez ainsi contrôler visuellement la bande passante d'un filtre et le caractériser graphiquement. Vous pourrez aussi déterminer la fréquence de résonance d'un circuit « L - C » en quelques secondes, ou encore, à l'aide d'une extension optionnelle pour le générateur de bruit, vous pourrez

déterminer l'acoustique d'une salle équipée d'un système de sonorisation.

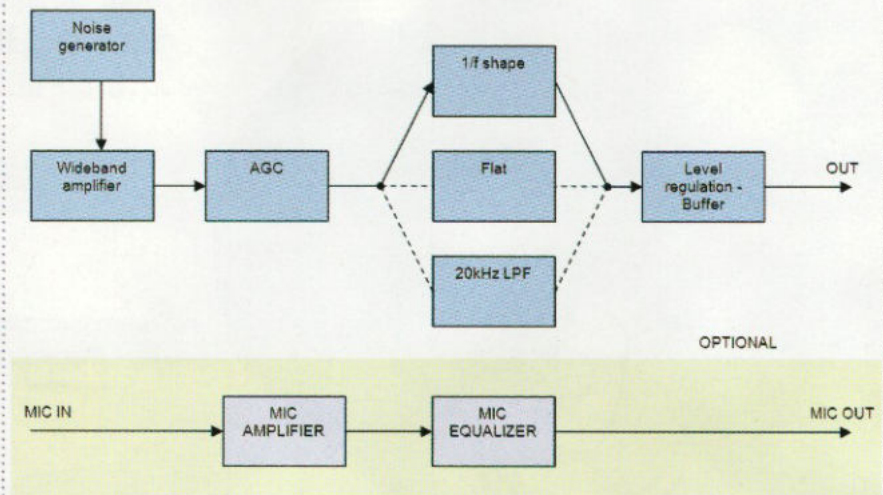
Le générateur de bruit, que nous vous présentons, est capable de **générer un bruit blanc ou un bruit rose** ($1/f$) jusqu'à une fréquence de **1 MHz**, ou encore un bruit blanc avec une **bande passante de 20 kHz**, ce qui est approprié pour les applications audio. Ces trois fonctions sont sélectionnables par l'intermédiaire d'un sélecteur. Il est possible de régler le niveau de sortie selon 3 niveaux sélectionnables à l'aide d'un commutateur.

Ceux d'entre vous qui doivent ajuster l'acoustique d'une pièce ou d'une salle, ou déterminer la réponse en fréquence d'un système Hi-Fi, devront fabriquer une petite carte supplémentaire (préamplificateur d'égalisation) qui leur permettra d'utiliser un microphone afin de capter le spectre sonore à analyser. La consommation du générateur est d'environ 20 mA, il est donc alimenté à l'aide d'une pile de 9 V.

Schéma électrique

Le générateur de bruit est constitué essentiellement par le transistor T2, utilisé dans une configuration inhabituelle. En effet, sa jonction base-émetteur est polarisée en inverse de telle manière que le **transistor travaille dans la zone de claquage** (breakdown). En d'autres termes, nous utilisons la jonction du transistor comme une **diode Zener**.

Schéma synoptique du générateur de bruit



Structure du générateur de bruit, en pointillé vous pouvez voir le circuit d'égalisation en fréquence du microphone qui permet le test acoustique d'une pièce ou d'une salle.

Dans ces conditions, pour un transistor « petits signaux », la tension de claquage (breakdown), c'est-à-dire celle à laquelle les jonctions sont polarisées en inverse, a une valeur d'environ 5 à 6 V. Le phénomène de conduction dépend donc de la combinaison de l'**effet Zener** et de l'**effet d'avalanche**.

À courant constant, dans une jonction, l'effet Zener et encore plus l'effet d'avalanche représentent un mécanisme physique générant un bruit électrique par rapport à la conduction normale lorsque la jonction est polarisée dans le sens direct.

Nous pouvons donc profiter de ce phénomène afin de générer la plus grande

quantité possible de bruit électrique. Le bruit électrique est constitué par un ensemble de signaux émis simultanément, chaque signal est caractérisé par une fréquence et une amplitude.

Dans notre circuit, le bruit généré par le phénomène d'avalanche (breakdown) dans la jonction base-émetteur se nomme « **bruit blanc** ».

Le bruit est ainsi prélevé de la base de T2 et, à l'aide de T4, est immédiatement amplifié de β fois, où β est le **gain en courant** du transistor T4.

Ainsi le transistor T4 avec le transistor T6 forment un « **étage cascode** » (un transistor configuré en émetteur

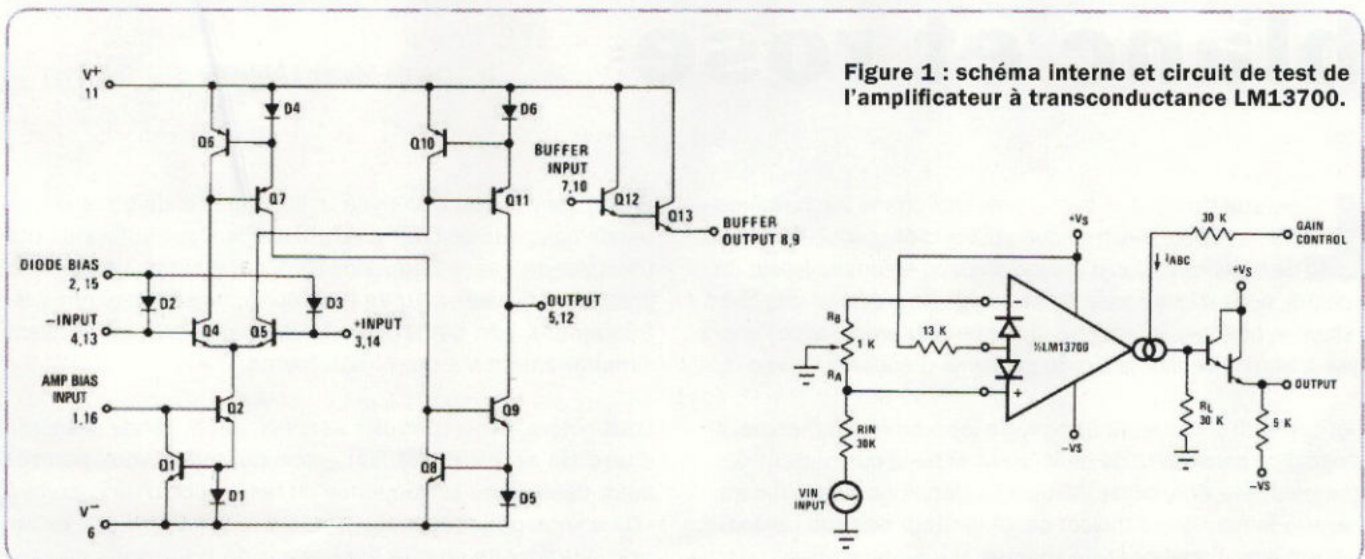
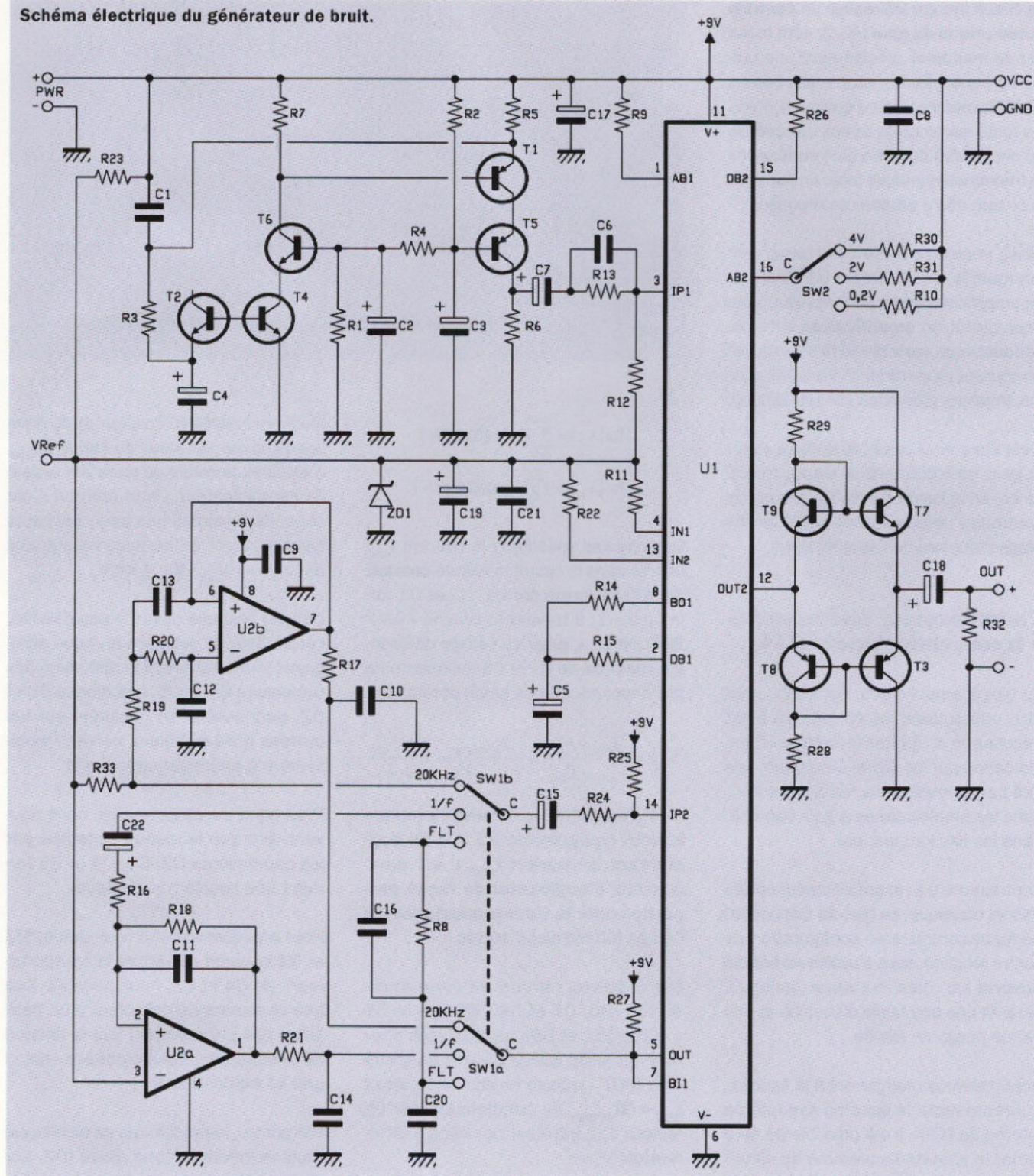


Figure 1 : schéma interne et circuit de test de l'amplificateur à transconductance LM13700.

Schéma électrique du générateur de bruit.



commun est relié en série avec un transistor monté en base commune), nécessaire à l'obtention d'un gain élevé et d'une large bande passante.

De même, les transistors T1 et T5 forment un amplificateur en « cascode », à la différence de T4 et T6 qui sont des NPN, cet étage est constitué par deux transistors de type PNP.

Ainsi, au point commun entre le collecteur de T5 et à la résistance R6, est présent un signal correspondant au bruit avec une amplitude suffisante pour piloter un étage de contrôle automatique du gain.

Cette partie du montage est nécessaire, car **il est impossible de garantir que le processus de génération du**

bruit (effet breakdown) **soit toujours homogène** en terme d'amplitude, en raison des tolérances de fabrication des transistors T2 et T4 et de la température ambiante.

En fait, sur différents prototypes que nous avons testés, la tension crête à crête dans R6 pouvait varier d'environ 2 Vcc à 4 Vcc.

Le circuit intégré **U1** réalise un **contrôle automatique du gain (AGC)**, dont le but est de maintenir précisément une tension crête à crête constante aux bornes de R27, valeur qui est d'environ 4,5 Vcc. De toute évidence, le temps de réponse du circuit AGC doit être bien supérieur à la fréquence minimale mise en jeu dans le circuit, afin d'adapter sa réponse.

Ainsi, vous comprenez certainement pourquoi le circuit intégré U1 n'est pas un amplificateur opérationnel classique, mais plutôt un **amplificateur à transconductance variable (OTA)** qui fournit un courant proportionnel à la différence des tensions présentes sur ses entrées.

Il est donc ainsi possible de faire varier le gain (précisément la transconductance) en agissant sur les deux tensions continues utilisées pour polariser les étages internes de l'amplificateur.

L'amplificateur opérationnel à transconductance (OTA)

Ce type d'amplificateur est conçu pour être utilisé dans les circuits où il est nécessaire d'ajuster le facteur d'amplification par un signal électrique, tels que par exemple dans les circuits AGC, dans les amplificateurs à gain variable, dans les oscillateurs, etc.

Contrairement à un amplificateur opérationnel classique, ce type de composant ne fonctionne pas en configuration de contre réaction, mais **s'utilise en boucle ouverte** car, dans certaines limites, il garantit une très faible distorsion et une bande passante élevée.

Reportez-vous maintenant à la figure 1, qui représente le schéma synoptique interne de l'OTA. Il est possible de faire varier le gain de l'ensemble du circuit en agissant sur la transconductance de l'étage différentiel d'entrée constitué par Q4 et Q5.

Rappelons-nous que la **transconductance** n'est rien de plus que le gain exprimé sous la forme d'un **rapport entre un courant de sortie et une tension d'entrée**. Cette grandeur est mesurée en **S** (siemens) ou **ampères par volts (A/V)**. Pour information, 1 siemens = 1 ampère par volt.

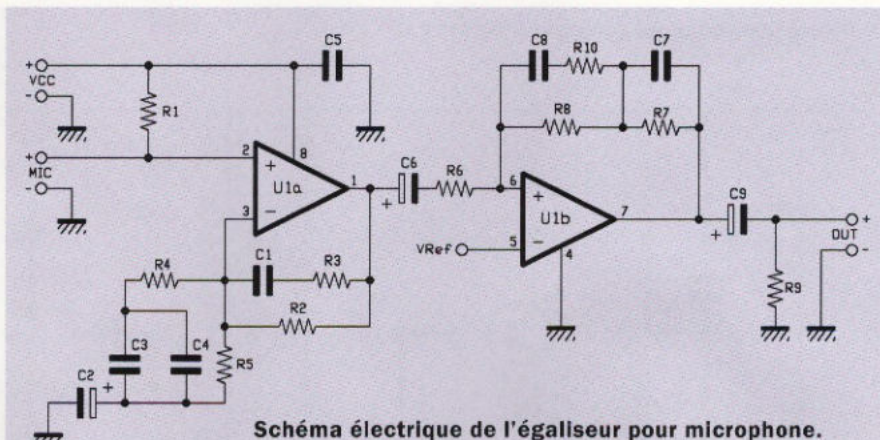


Schéma électrique de l'égaliseur pour microphone.

$$\begin{cases} \operatorname{Re}\{X_k\} = \sum_{n=0}^{N-1} x_n \cos(2\pi kn/N) \\ \operatorname{Im}\{X_k\} = j \sum_{n=0}^{N-1} x_n \sin(2\pi kn/N) \end{cases}$$

Dans ce cas spécifique le courant I_{ABC} injecté dans le circuit **miroir de courant de Wilson** formé par Q1, Q2 et D1 (de la figure 1) à travers la broche « AMP BIAS INPUT », polarise l'étage différentiel composé de Q4 et Q5 qui détermine la transconductance selon la relation :

$$G_m = \frac{I_{C5} - I_{C4}}{V_{IN+} - V_{IN-}} = \frac{I_{OUTDIFF}}{V_{IN+} - V_{IN-}} = \frac{1}{2} \frac{I_{ABC}}{V_T}$$

où V_T représente la tension thermique et vaut typiquement 25 mV. En augmentant le courant I_{ABC} , il est donc possible d'**augmenter de façon proportionnelle la transconductance** de l'étage différentiel d'entrée.

Les 3 autres miroirs de courant de Wilson (Q6, Q7 et D4 - Q8, Q9 et D5 - Q10, Q11 et D6), sont utilisés pour faire en sorte que la broche de sortie « OUTPUT » puisse fournir un courant $I_{OUT} = 2I_{OUTDIFF}$, en compensant par un facteur 1/2 introduit par l'étage différentiel.

Normalement, sur la sortie « OUTPUT » est connectée une résistance R_L dont l'un des côtés se situe à la masse, elle introduit une résistance de charge. Nous avons donc la formule suivante :

$$G = \frac{V_{OUT}}{V_{IN+} - V_{IN-}} = \frac{R_L I_{ABC}}{V_T}$$

où V_{OUT} est précisément la tension présente aux bornes de R_L .

Ainsi, en modifiant la valeur de R_L nous agissons sur la valeur du courant I_{ABC} , il est alors possible de contrôler le gain de l'amplificateur. Cette relation n'est cependant valable que pour des petits signaux, c'est à dire pour les signaux tels que : $|V_{IN+} - V_{IN-}| \ll V_T$

Dans la pratique cela n'a pas d'utilité, car au-delà de cette limite, nous atteignons immédiatement la distorsion des transistors Q4 et Q5. Les diodes D3 et D2, connectées en parallèle sur les entrées différentielles, servent précisément à prolonger cette limite.

D'un point de vu technique, nous pouvons dire que le réseau constitué par les composants D2, D3, Q4 et Q5 forment une boucle translinéaire.

Pour expliquer comment les diodes D2 et D3 peuvent linéariser le comportement de Q4 et Q5, nous pouvons dire que le courant de collecteur d'un transistor (Q4 par exemple) suit la tension de commande « base-émetteur » selon une loi exponentielle.

Par contre, la tension qui se développe dans la jonction d'une diode (D2, par exemple) suit une loi logarithmique. Cela signifie qu'en injectant un courant I_s à travers la résistance R_{IN} dans la jonction D2, aux bornes de celle-ci apparaît une **tension fortement déformée** (ou distordue) selon une **loi logarithmique**.

Cependant, cette même tension est appliquée à la jonction « base-émetteur » de Q4, ce qui **distordra cette tension selon une loi exponentielle**.

Et si le bruit dans l'habitacle était nul ?

Du Japon, arrive une nouvelle fonctionnalité intéressante qui nous montre que l'électronique peut beaucoup améliorer le confort dans une voiture et pas seulement en matière de sécurité, de tenue de route ou de climatisation.

Infiniti, qui est la marque de luxe de Nissan, pour sa nouvelle série a mis au point le système « ANC » pour « Active Noise Control » ou contrôle actif du bruit.

Ce système permet de supprimer les bruits à basse fréquence produits par le moteur. Il génère, à travers l'installation audio du véhicule, des ondes sonores de phases opposées à celles provenant du bruit du moteur. Ces ondes sont superposées au son de l'ensemble audio et sont émises par les haut-parleurs de l'habitacle, ainsi



elles ont pour effet d'annuler le bruit indésirable dans l'habitacle.

De cette façon, même ceux qui détestent les claquements des moteurs diesel sont satisfaits. Une solution intéressante, car ces véhicules sont équipés de moteurs V6 diesel de 3 L ou V6 essence de 3,7 L, sinon à haute vitesse il faudrait élever la voix pour s'entendre.

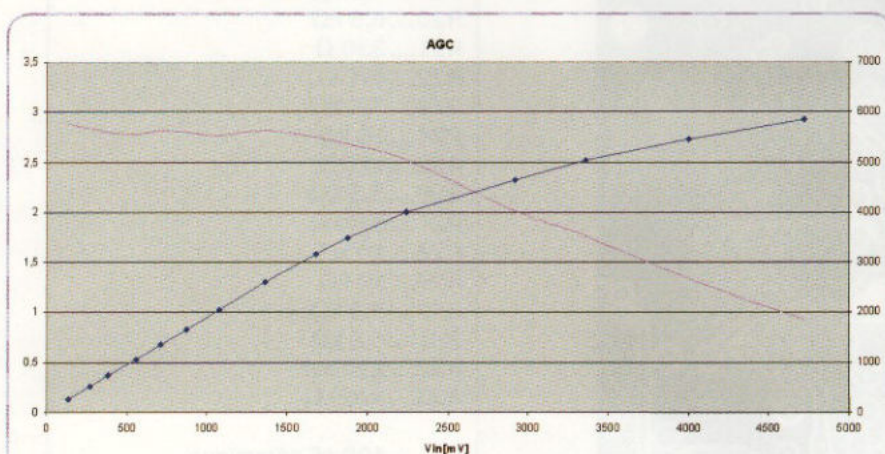


Figure 2 : fonction de transfert du circuit U1 (AGC).
Plus le signal d'entrée est grand, plus le signal de sortie est petit. Le gain du circuit varie d'un maximum d'environ 3 dB à un minimum de 1 dB.

Le résultat est que le **courant de collecteur de Q4 varie selon une relation linéaire avec le courant I_s** , car **les deux effets de distorsion se compensent** et s'éliminent l'un par rapport à l'autre. En d'autres termes, D2 génère une prédistorsion pour le pilotage de Q4. En tenant compte également des diodes D2 et D3 de linéarisation, le gain de l'étage vaut :

$$G = \frac{V_{OUT}}{V_{IN}} = 2 \frac{I_{ABC}}{I_D} \frac{R_L}{R_{IN}}$$

où I_D représente le courant de polarisation des diodes D2 et D3.

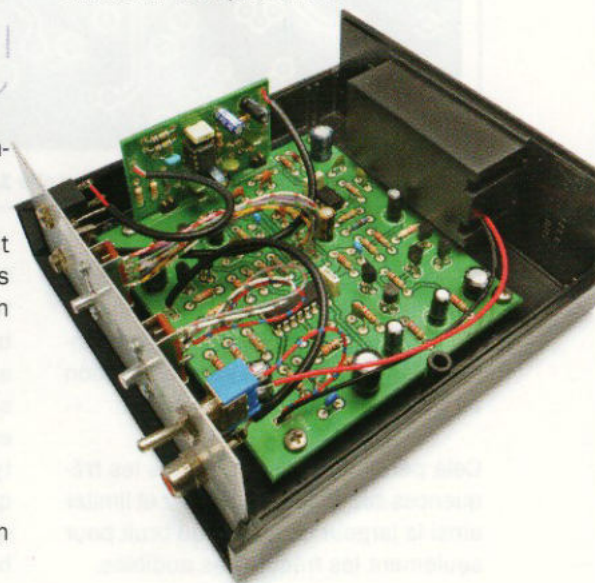
Cette dernière relation est également valable pour des signaux d'entrée plus grands que V_T . Si nous considérons un courant $I_D \approx 1 \text{ mA}$, nous obtenons :

$$\frac{V_{INMAX}}{R_{IN}} < \frac{I_D}{2}$$

Ces valeurs permettent une distorsion inférieure à 0,5 %.

Revenons au schéma électrique, le signal ainsi généré est redressé à l'intérieur de U1 et est appliqué au condensateur C5 à travers les broches 2 et 8. Le condensateur C5 se charge alors à la valeur maximale (crête) du signal, comme dans un circuit de type détecteur d'enveloppe.

L'enveloppe est utilisée pour faire varier le gain de l'étage en contrôlant la tension sur la broche 2. Plus le signal de sortie aux bornes de R27 est grand, plus le signal aux bornes de C5 est grand et donc le gain de l'étage qui en résulte sera plus faible.



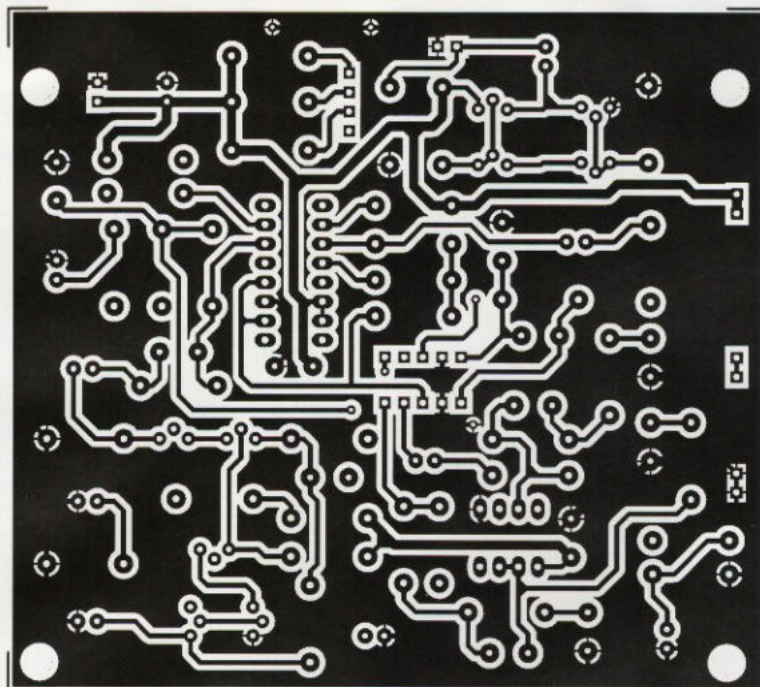


Figure A : circuit imprimé à l'échelle 1 : 1 côté soudures.

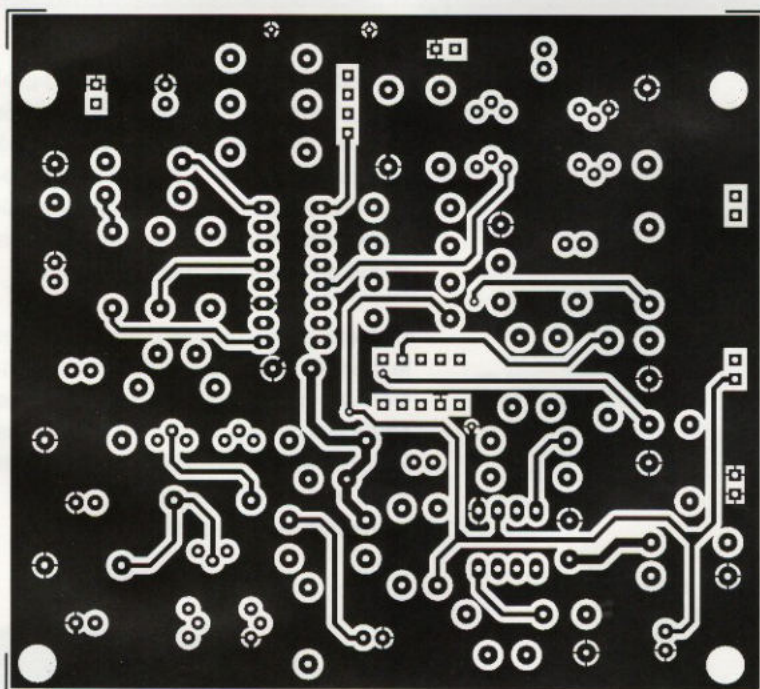


Figure B : circuit imprimé à l'échelle 1 : 1 côté composants.

Plan de montage du générateur de bruit

Liste des composants du générateur de bruit.

R1.....	3,3 k Ω
R2.....	4,7 k Ω
R3.....	22 k Ω
R4.....	10 k Ω
R5.....	1,2 k Ω
R6.....	2,7 k Ω
R7.....	2,2 k Ω
R8.....	180 k Ω
R9.....	5,6 k Ω
R10.....	68 k Ω
R11.....	330 Ω
R12.....	470 Ω
R13.....	22 k Ω
R14.....	100 Ω
R15.....	2,7 k Ω
R16.....	56 k Ω
R17.....	330 Ω
R18.....	4,7 M Ω
R19.....	6,8 k Ω
R20.....	6,8 k Ω
R21.....	330 Ω
R22.....	470 Ω
R23.....	3,3 k Ω
R24.....	10 k Ω
R25.....	470 Ω
R26.....	3,3 k Ω
R27.....	5,6 k Ω
R28.....	10 k Ω
R29.....	10 k Ω
R30.....	3,3 k Ω
R31.....	6,8 k Ω
R32.....	220 k Ω
R33.....	470 k Ω

C1.....	100 pF céramique
C2.....	10 μ F/16 V électrolytique
C3.....	10 μ F/16 V électrolytique

Si nous positionnons le commutateur **SW1** sur la position **20 kHz**, le signal est dirigé vers l'amplificateur opérationnel **U2b**, qui est utilisé en configuration de **filtre passe-bas de 2^{ème} ordre**.

Cela permet de réduire toutes les fréquences supérieures à 20 kHz et limiter ainsi la largeur de bande du bruit pour seulement les fréquences audibles.

Ceci est particulièrement utile pour tester des systèmes audio avec un bruit blanc, car en envoyant à des appareils audio un signal ayant un spectre très étendu, comme par exemple dans le cas d'un signal de type bruit blanc non limité en fréquence, il est possible que les fréquences supérieures à celles de la bande passante de l'amplificateur

causent des distorsions d'intermodulation dans le signal de sortie.

Si nous positionnons le commutateur **SW1** sur la position « **1/f** », le signal est dirigé vers l'amplificateur opérationnel **U2a**, qui est utilisé en tant que circuit intégrateur. Cela donne un signal de sortie caractérisé par une **pente constante de -20 dB** par décade, ce type de signal

C4..... 220 μ F/16 V électrolytique
 C5..... 470 μ F/16 V électrolytique
 C6..... 15 pF céramique
 C7..... 10 μ F/16 V électrolytique
 C8..... 100 nF/63 V polyester
 C9..... 100 nF/63 V polyester
 C10..... 470 pF céramique
 C11..... 22 nF/63 V polyester
 C12..... 1 nF/63 V polyester
 C13..... 1 nF/63 V polyester
 C14..... 470 pF céramique
 C15..... 10 μ F/16 V électrolytique
 C16..... 15 pF céramique
 C17..... 10 μ F/16 V électrolytique
 C18..... 470 μ F/16 V électrolytique
 C19..... 100 μ F/16 V électrolytique
 C20..... 100 pF céramique
 C21..... 100 nF/63 V polyester
 C22..... 10 μ F/16 V électrolytique

T1 BC557
 T2 BC546
 T3 BC557
 T4 BC546
 T5 BC557
 T6 BC546
 T7 BC546
 T8 BC557
 T9 BC546
 DZ1..... zener 4,7 V/500 mW
 SW1... double commutateur
 tripolaire à glissière
 SW2... double commutateur
 tripolaire à glissière
 U1..... LM13700N
 U2..... RC4558

Divers :

Barrette mâle 2 pôles (x 5)
 Support circuit intégré 2 x 4 broches
 Support circuit intégré 2 x 8 broches

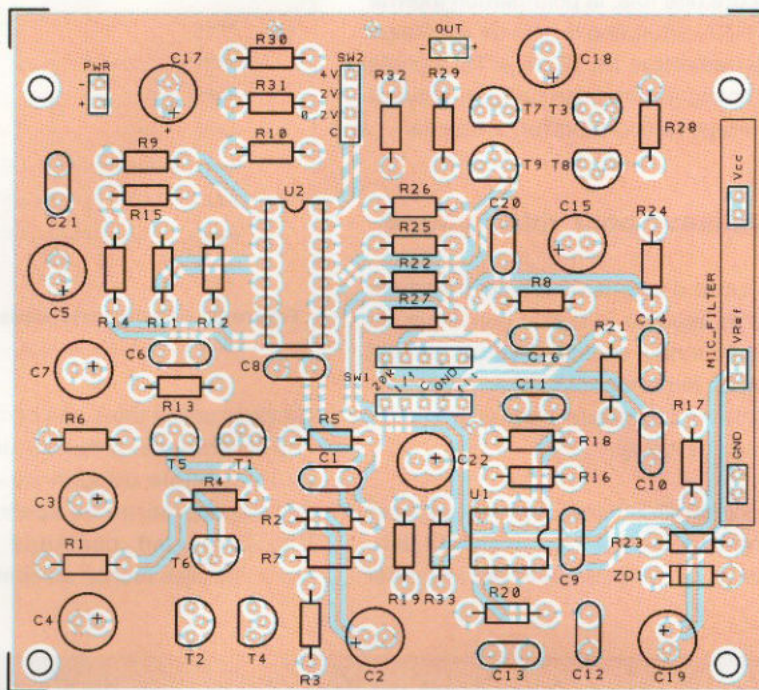


Figure C : plan de câblage des composants du générateur de bruit.

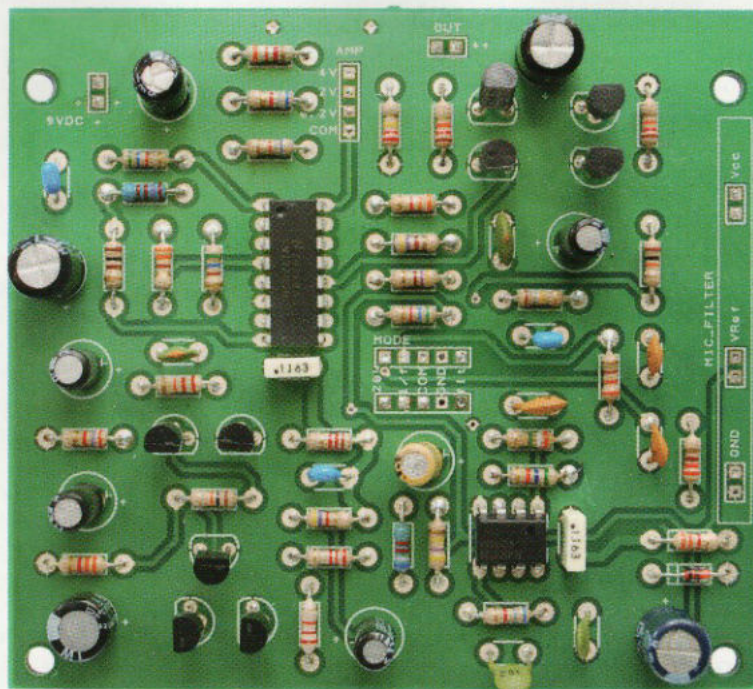


Figure D : photo de l'un de nos prototypes du générateur de bruit.

a un bruit caractéristique de « $1/f$ » ou encore il est appelé **bruit rose**.

La dernière position du commutateur **SW1** est « **FLT** » pour « **FLAT** », c'est à dire plat. Le signal passe en totalité et ne subit **aucune altération en fréquence**. Les gains en amplitude des trois positions possibles du commutateur SW1 ont été choisis de manière à

maintenir un **niveau constant « RMS »** en sortie, indépendamment de la position sélectionnée.

La **valeur efficace**, dite aussi valeur RMS (de l'anglais Root Mean Square, moyenne quadratique) d'un signal périodique est la racine carrée de la moyenne de cette grandeur au carré dans un intervalle de temps donné.

En continuant la description du schéma, nous nous apercevons qu'à travers le condensateur C15, le signal du bruit prélevé sur S1b est envoyé vers la broche 14 (IP2) qui constitue un étage composé de U1 et des 4 transistors T3, T7, T8, T9. Cet étage est un amplificateur à **gain sélectionnable**, qui est fixé par l'intermédiaire du sélecteur SW2.

Encore une fois, la polarisation interne de l'OTA détermine le gain de cet étage. Les transistors T3, T7, T8 et T9, montés en configuration push-pull, constituent un étage tampon (buffer) en sortie.

Égaliseur pour microphone

La carte d'extension qui sert à amplifier et égaliser le signal capté par un microphone à capsule électret est très simple. Reportez-vous au schéma électrique de l'égaliseur pour microphone (voir plus haut).

Le signal du microphone est appliqué à l'entrée non-inverseuse de U1a (broche 2), cet amplificateur opérationnel

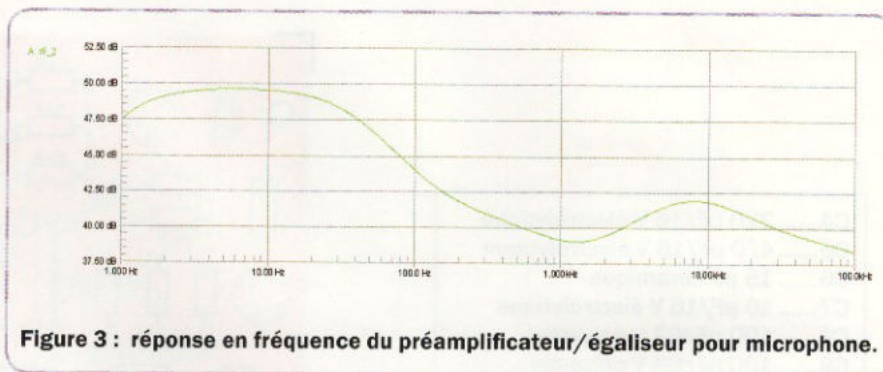


Figure 3 : réponse en fréquence du préamplificateur/égaliseur pour microphone.

est donc utilisé dans une configuration non-inverseuse. Il fournit une partie de l'ensemble du gain du circuit et égalise une partie du spectre audio. L'étage suivant, constitué par U1b, est à l'inverse monté en configuration

d'amplificateur inverseur. Il fournit la partie restante du gain et égalise la courbe de réponse en fréquence, afin de rendre la caractéristique de transfert en fréquence du microphone la plus plate possible.

Fast Fourier Transform ou Transformation de Fourier Rapide (FFT)

Le terme « Fast Fourier Transform » représente un algorithme de calcul qui permet d'effectuer efficacement une Transformée de Fourier classique. Cet algorithme est couramment utilisé en traitement numérique du signal pour transformer des données discrètes du domaine temporel vers le domaine fréquentiel, en particulier dans les analyseurs de spectre.

Ces deux formes de représentation du même signal sont parfaitement équivalentes, dans le sens où il est possible de décrire complètement dans les deux cas, le signal original en utilisant l'une ou l'autre description formelle. Malgré l'équivalence formelle, il est souvent utile d'analyser le spectre d'un signal plutôt que sa représentation temporelle, car cela permet de détecter la présence de composantes difficiles à identifier en observant seulement l'évolution temporelle.

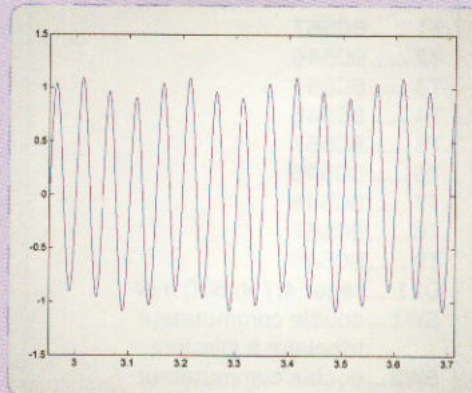
Malheureusement, la transition mathématique entre les deux représentations n'est pas simple, parce qu'elle est définie par des relations appelées « intégrales » entre des nombres complexes. Bien qu'en 1822, Fourier publia pour la première fois sa théorie, il a fallu attendre 1967 pour disposer d'un système capable de traiter numériquement un signal et d'effectuer la transformation en un temps raisonnable.

À l'époque il fallait environ 1 s pour traiter une séquence de 1024 points. Le système était disposé dans un rack dont la hauteur était celle d'un homme. De nos jours, les technologies ont évolué, mais l'idée de base du résultat obtenu en 1967 reste la même.

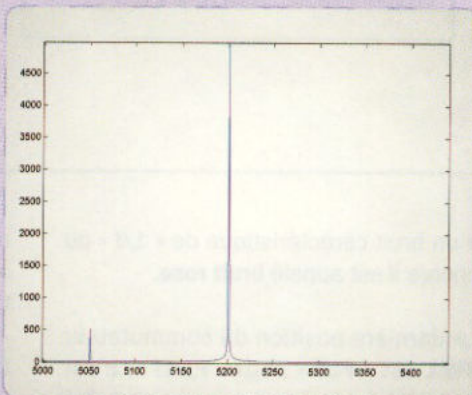
En laissant de côté les aspects purement mathématiques, nous allons maintenant expliquer comment nous obtenons le spectre discret d'un signal discret. Supposons que X_n représente la séquence de N échantillons du signal à analyser (voir la figure de la représentation temporelle et

fréquentielle d'un même signal), la relation de Fourier permet de calculer N autres valeurs désignées par X_k qui représentent parfaitement le contenu harmonique de la même séquence.

En réalité, X_k est un nombre complexe formé d'une partie réelle et d'une partie imaginaire. C'est pour cela que généralement, le spectre complet d'un signal correspond au module du nombre complexe en phase, même si dans la pratique la mesure de la phase a un aspect secondaire.



Représentation temporelle et fréquentielle d'un même signal. En analysant le contenu en fréquence, il est possible de déterminer facilement la fréquence fondamentale et la présence d'une composante de plus petite amplitude à une fréquence plus basse.



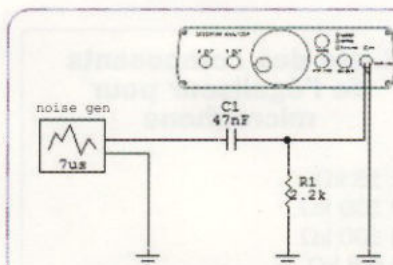
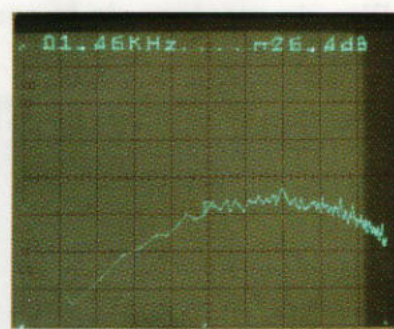
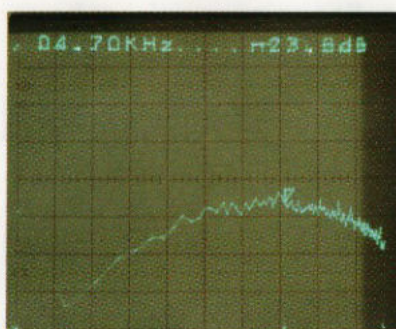


Figure 4 : schéma permettant de vérifier la courbe de réponse du filtre passe-bas (C1 = 47 nF, R1 = 2,2 kΩ).



En résumé, le montage se comporte comme un réseau de contre-réaction sélectif, il atténue les hautes fréquences et amplifie les basses fréquences, ce qui compense les carences du microphone utilisé.

Réalisation pratique

Le générateur de bruit et l'égaliseur pour microphone sont entièrement réalisés avec des composants traditionnels, par conséquent, nous ne nous

attarderons pas sur la technique de montage qui est relativement simple. Comme d'habitude, vous pouvez télécharger les typons des circuits dans le sommaire détaillé de la revue sur notre site www.electroniquemagazine.com.

Pour calculer le coefficient « k-ième » du spectre de X_n , appliquons la formule suivante :

$$\begin{cases} \operatorname{Re}\{X_k\} = \sum_{n=0}^{N-1} x_n \cos(2\pi kn/N) \\ \operatorname{Im}\{X_k\} = j \sum_{n=0}^{N-1} x_n \sin(2\pi kn/N) \end{cases}$$

Dans la pratique, le coefficient X_k est obtenu en multipliant le signal d'origine X_n point par point avec une fonction cosinus de fréquence différente pour chaque « k » (pour la partie imaginaire de X_k) et en additionnant les résultats intermédiaires. Cela signifie que, plus le signal X_n « ressemble » à une sinusoïde de même fréquence, plus la valeur finale de la somme X_k sera élevée.

Fondamentalement, ce que nous faisons mathématiquement à travers ces deux formules peut-être expliqué de la manière suivante. Nous allons chercher dans le signal X_n , un par un le contenu de chaque sinusoïde à une fréquence différente. Pour chaque coefficient complexe, il faut donc effectuer deux sommes de N éléments. Supposons que nous ayons $N = 1024$, par exemple, il est donc nécessaire d'effectuer au moins $2 * N * N = 2 * 1024 * 1024 = 2\,097\,152$ opérations !

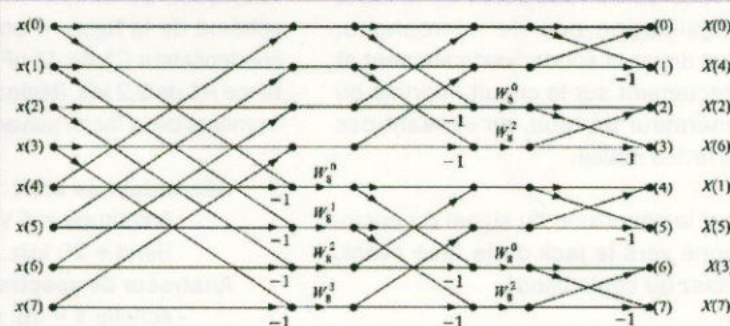
Heureusement en 1965, grâce à James Cooley et John Tukey, une nouvelle méthode de calcul de la transformée de Fourier discrète a été introduite. Cette méthode est beaucoup plus efficace et donc beaucoup plus rapide, elle se base simplement sur le regroupement avantageux des échantillons en les subdivisant en index pairs et impairs et en observant que les coefficients de chacun des deux groupes d'échantillons avec des index pairs ou impairs peuvent être obtenus, à leur tour, en les subdivisant en index paires et impaires, et en répétant ainsi de suite le processus. Cette méthode permet de réduire le nombre d'opérations à : $N * \log_2(N) = 10240$. En effet, $\log_2$ en base 2 de 1024 ($\log_2(1024)$) est égal à 10, en multipliant par N, nous obtenons 10240.

Pour résumer, nous pouvons dire que cette méthode subdivise une transformation de Fourier discrète d'une taille composite $n = n_1 n_2$ en plusieurs transformées de Fourier discrètes de tailles inférieures n_1 et n_2 .

Une autre optimisation de la vitesse de calcul est possible si nous observons que les valeurs des expressions $\cos(2\pi kn/N)$ et $\sin(2\pi kn/N)$ sont des variables périodiques de k et de n. Il suffit donc de stocker dans un tableau dans la RAM une séquence unique de valeurs et d'aller de temps en temps prélever uniquement le coefficient approprié, sans calculer réellement la valeur de la fonction trigonométrique. Ces coefficients sont appelés « Twiddle Factors » (W_N^k). Dans ce cas, les deux expressions, pour le calcul des coefficients de Fourier, prennent la forme suivante :

$$\operatorname{Re}\{X_k\} = x_0 W_0^k + x_1 W_1^k + \dots + x_n W_n^k \text{ et } \operatorname{Im}\{X_k\} = x_0 W_{0+N/4}^k + x_1 W_{1+N/4}^k + \dots + x_n W_{n+N/4}^k$$

Il s'agit d'une succession de multiplications et de sommes. Nous avons déjà évoqué dans d'autres articles que ces opérations étaient prises en charge par un DSP, ce type de processeur est spécialisé dans des applications de traitement numérique du signal.



Exemple de FFT réalisée sur N = 8 points. Notez la combinaison algébrique des échantillons et la multiplication par les « Twiddle Factors » relatifs.

Plan de montage de la carte égaliseur pour microphone (optionnel)

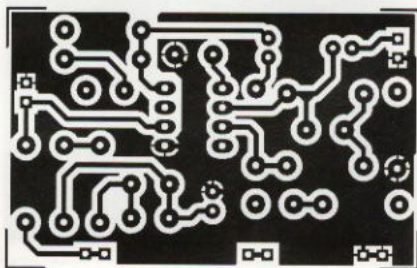


Figure E : circuit imprimé à l'échelle 1 : 1 côté soudures.

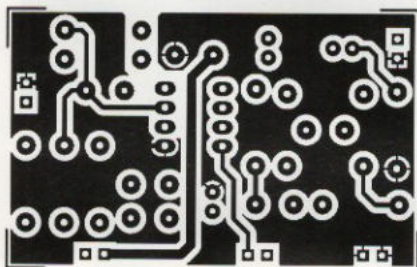


Figure F : circuit imprimé à l'échelle 1 : 1 côté composants.

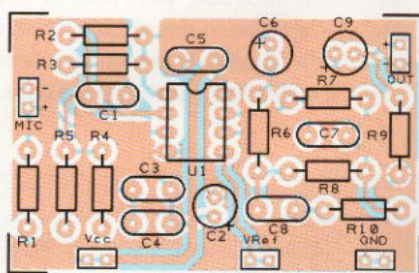


Figure G : plan de câblage des composants de l'égaliseur pour microphone.

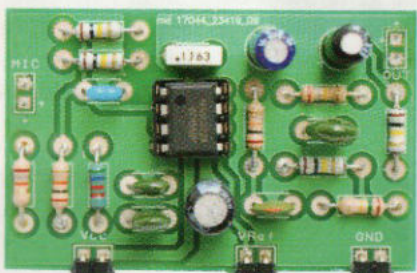


Figure D : photo de l'un de nos prototypes de l'égaliseur pour microphone.

Liste des composants de l'égaliseur pour microphone

R1	18 k Ω
R2	100 k Ω
R3	100 k Ω
R4	5,6 k Ω
R5	10 k Ω
R6	10 k Ω
R7	180 k Ω
R8	100 k Ω
R9	100 k Ω
R10	150 k Ω
C1	1,2 nF polyester
C2	47 μ F/16 V électrolytique
C3	3,3 nF polyester
C4	3,3 nF polyester
C5	100 nF polyester
C6	22 μ F/16 V électrolytique
C7	22 nF polyester
C8	47 pF céramique
C9	47 μ F/16 V électrolytique
U1	RC4558

Divers

- Barrette mâle 2 pôles (x 4)
- Support circuit intégré 2 x 4 broches

Les règles habituelles pour le montage sont applicables ici. Commencez par souder les composants ayant un profil bas puis continuez avec les composants de plus grande taille.

Le point le plus important pour le montage des composants est de vérifier l'orientation des condensateurs polarisés, de la diode, des transistors et des circuits intégrés. Tous les composants se trouvent facilement dans le commerce, le LM13700N et le RC4558 sont disponibles chez Farnell.

Si vous devez vous servir de la carte d'égalisation pour le microphone, vous devez la souder verticalement et directement sur le circuit imprimé du générateur de bruit, en utilisant des barrettes mâles.

Pour la connexion du signal du microphone vers le jack de la face avant, utilisez du câble blindé.

Nous vous recommandons d'effectuer des **connexions les plus courtes possibles entre le commutateur SW1 et le circuit imprimé.**

Exemples d'utilisation

Maintenant, nous allons illustrer brièvement quelques mesures que vous pouvez effectuer en utilisant le générateur de bruit ainsi que l'analyseur de spectre décrit dans les précédents numéros 138 et 139.

Caractéristiques de transfert d'un filtre

Prenons l'exemple d'un filtre passe-haut du premier ordre, comme illustré en figure 4.

Fabriquez un simple filtre selon le schéma de la figure 4 en utilisant un condensateur C1 de 47 nF et une résistance R1 de 2,2 k Ω . Réglez les deux instruments de la façon suivante :

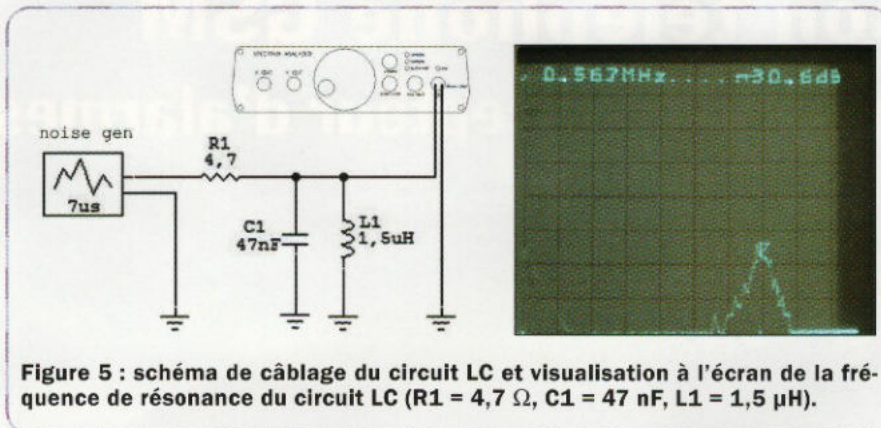
- Générateur de bruit :**
 - Amplitude = 4 Vp/p ;
 - Band = 20 kHz.
- Analyseur de spectre :**
 - échelle Y = dB, x1 ;
 - échelle X = LOG ;
 - f sampling : 78,125 ks/s ;
 - zoom = x 2 ;
 - max hold = ON.

Avec ces paramètres, nous obtenons sur l'écran de l'oscilloscope le spectre visible sur les deux écrans de la figure 4. Le déplacement du curseur peut alors déterminer la **fréquence de coupure du filtre**, en rappelant que celle-ci est définie comme étant la **fréquence pour laquelle l'amplitude est inférieure de 3 dB** (1,414 fois) par rapport à celle de la fréquence centrale.

Dans la bande, nous lisons environ -23,8 dB, et tout en déplaçant le curseur pour atteindre environ -23,8 dB - 3dB = -26,8 dB, nous lisons une fréquence de 1,46 kHz qui correspond à notre fréquence de coupure du filtre passe-haut.

Fréquence de résonance d'un circuit LC

Il est possible de déterminer la **fréquence de résonance d'un circuit L-C** (ou encore connaissant la fréquence de résonance, déterminer la valeur de L ou de C) comme illustré dans l'exemple de la figure 5. Les valeurs des composants sont : R1 = 4,7 Ω , C1 = 47 nF et L1 = 1,5 μ H.



Les instruments doivent être réglés de la façon suivante :

- **Générateur de bruit :**
 - Amplitude = 4 Vp/p ;
 - Band = FLAT.
- **Analyseur de spectre :**
 - échelle Y = dB, x1 ;
 - échelle X = LOG ;
 - f sampling : 10 Ms/s ;
 - zoom = x 2 ;
 - max hold = ON.

En déplaçant le curseur sur l'écran, nous pouvons localiser où se situe la fréquence de résonance du circuit, dans notre cas nous voyons qu'elle se situe à environ 567 kHz.

Notez que ces mesures doivent être effectuées avec la fonction **MAX HOLD** activée (ON), en appuyant sur le bouton approprié situé sur la face avant de l'analyseur de spectre. Ainsi, nous pouvons mémoriser et visualiser les valeurs maximales contenues dans le spectre du signal.

Mesure de la bande passante d'un système audio

Dans cet exemple, nous allons vous montrer comment utiliser la carte optionnelle pour le microphone. Le générateur de bruit doit être connecté à la ligne d'entrée d'un système audio (cela peut être un amplificateur Hi-Fi, ou encore une table de mixage dans le cas d'une sonorisation d'une salle) de manière à reproduire un bruit blanc à travers les haut-parleurs.

Le microphone doit être placé au centre de la pièce ou dans un point d'écoute particulier de manière à détecter la

pression sonore à un endroit précis. La sortie du préamplificateur microphonique (« MIC OUT »), c'est à dire la sortie de la carte optionnelle située à l'intérieur du générateur de bruit, doit être reliée à l'entrée « IN » de l'analyseur de spectre. Le microphone doit être connecté à l'entrée « MIC IN » du générateur de bruit.

Sur l'écran de l'oscilloscope nous obtenons, en réglant les instruments de mesure comme indiqué ci-après, la réponse en fréquence (bande passante) d'un système audio (préamplificateur, amplificateur, haut-parleurs, etc.). Cela permet d'identifier la présence de zones de résonances du milieu ambiant (salle, pièce, etc.).

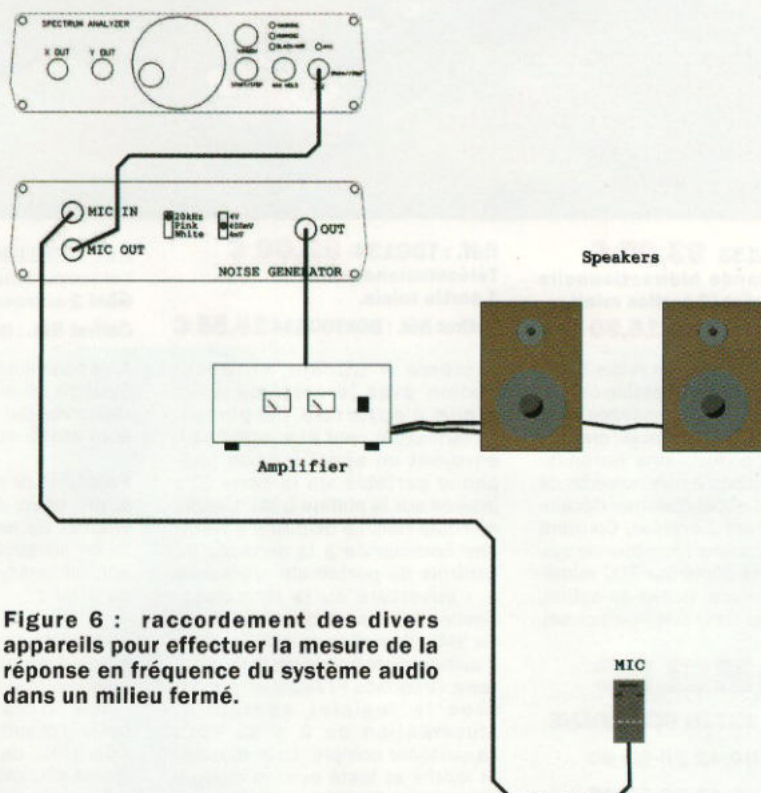


Figure 6 : raccordement des divers appareils pour effectuer la mesure de la réponse en fréquence du système audio dans un milieu fermé.

Le générateur de bruit présenté ici a été spécialement conçu pour être utilisé avec l'analyseur de spectre, dont

le projet a été présenté dans les numéros 138 et 139 d'Électronique et Loisirs Magazine. Ce dernier fonctionne en conjonction avec un oscilloscope traditionnel à double trace (numérique ou analogique), à condition qu'il soit doté du mode X/Y. La fréquence maximale d'échantillonnage de l'analyseur de spectre est de 10 Ms/s pour une bande passante utile de 5 MHz.

Pour ce dernier test, les paramètres des instruments sont les suivants :

- **Générateur de bruit :**
 - Amplitude = 400 mVp/p ;
 - Band = 20 kHz.
- **Analyseur de spectre :**
 - échelle Y = dB, x1 ;
 - échelle X = LOG ;
 - f sampling : 78,125 ks/s ;
 - zoom = x 2 ;
 - max hold = ON.

Contrôle à distance Utiliser son téléphone GSM comme télécommande et récepteur d'alarmes

Installations antivol pour les bâtiments civils et industriels, pour les voitures, contrôle des systèmes de climatisation, chauffage, contrôle de pompes et de systèmes d'irrigation, contrôle d'installations industrielles etc.



Réf. : TDG133 **93,00 €**
Télécommande bidirectionnelle
GSM 2 entrées/2 sorties relais.
Coffret Réf. : BOXTDG **15,90 €**

Permet de piloter deux relais à distance (en mode monostable ou bistable) grâce à des messages SMS (sécurisés par mot de passe) envoyés depuis votre téléphone portable. Mémoirisez jusqu'à huit numéros de téléphone d'appel d'alarme déclenchables via les 2 entrées. Convient également comme récepteur de système d'accès piloté par 200 numéros de téléphone. Boîtier en option. Requiert une carte SIM (non incluse).

COMELEC

CD 908 - 13720 BELCODÈNE

Tél. : 04 42 70 63 90

Fax : 04 42 70 63 95

www.comelec.fr

Réf. : TDG134 **92,00 €**
Télécommande GSM
1 sortie relais.
Coffret Réf. : BOXTDG134 **15,55 €**

Système à utiliser en association avec le système électrique d'ouverture du portail. La fermeture peut être activée en envoyant un appel avec le téléphone portable via la carte Sim insérée sur la platine GSM. L'appel ne coûte rien. Le dispositif enverra une commande à la centrale de contrôle du portail qui procédera à l'ouverture ou la fermeture. Gestion à distance des utilisateurs via SMS (demande de mot de passe d'authentification) ou via le PC local (avec l'interface FT782M en option) avec le logiciel approprié. Alimentation de 9 à 32 VDC. Le système comprend: le dispositif monté et testé avec le module GSM/GPRS SIM900 et l'antenne. Boîtier en option. Requiert une carte SIM (non incluse).

Réf. : TDG140 **99,00 €**
Télécommande bidirectionnelle
GSM 2 entrées/2 sorties relais.
Coffret Réf. : BOXTDG **15,90 €**

Avec commandes par DTMF ou SMS. Système de contrôle à distance bidirectionnel qui utilise le réseau GSM pour activer et contrôler.

Possibilité de mémoriser 8 numéros pour l'envoi des alarmes déclenchables via les 2 entrées. Alimentation comprise entre 9 et 32 Vdc. Applications typiques en mode SMS ou DTMF :

- Installations antivol pour les bâtiments civils et industriels.
- Installations antivol pour voitures
- Contrôle des systèmes de climatisation / chauffage.
- Contrôle de pompes et de systèmes d'irrigation.
- Contrôle d'installations industrielles.
- Carte SIM (non incluse).

Réf. : TDG135 **124,00 €**
Télécommande bidirectionnelle
GSM 2 entrées/2 sorties relais.
Coffret Réf. : BOXTDG135 **15,55 €**

Avec composeur téléphonique. Utilisé dans le secteur de la sécurité et de l'automatisme à distance, il dispose de 2 entrées optoisolées et de 2 sorties relais contrôlables par DTMF, appel téléphonique (menu vocal guidé) et commande SMS. Les deux sorties peuvent être utilisées pour gérer à distance des dispositifs divers tels que des lumières, moteurs, etc. Le dispositif offre la possibilité de mémoriser 8 numéros de téléphone pour l'envoi de message d'alarme par appel téléphonique ou sms. La gestion des paramètres peut être faite à distance par SMS ou en local via un PC avec le logiciel adéquat (nécessite une interface USB). L'utilisateur peut personnaliser les messages vocaux et les écouter à l'aide du petit HP incorporé. Boîtier en option. Carte SIM (non incluse).

Vision numérique avec **RaspberryPi** et **caméra Pi**



de Marco Magagnin



Après avoir fait connaissance avec les différentes fonctionnalités du module Caméra Pi dans le numéro 137, qui est un module vidéo de qualité spécialement conçu pour le RaspberryPi, nous allons l'étudier pour acquérir les premières connaissances de la vision numérique (ou vision par ordinateur) afin de reconnaître des formes et des couleurs.

Caméra Pi est un excellent module complémentaire (add-on) pour le RaspberryPi. Il permet de prendre des photos et enregistrer des vidéos de qualité, avec la possibilité d'appliquer un nombre important d'effets ainsi qu'une configuration étendue.

Dans cet article, nous allons décrire la manière de l'utiliser afin de reconnaître un contenu spécifique dans une image capturée. La question qui se pose immédiatement est la suivante : pourquoi faire de la vision numérique ? La réponse est simple. Parce qu'il s'agit d'une discipline désormais de plus en plus répandue dans les applications du monde réel, et maintenant, elle est à la portée des amateurs.

Nous pouvons dire aussi que cela est dû à la curiosité humaine de vouloir reproduire par des machines les capacités sensorielles et d'analyse de notre propre espèce. En résumé, cela revient à rendre les machines plus humaines. Cependant entre le rêve et la réalité, le chemin se révèle particulièrement long et complexe. En fait, la vision est une capacité que l'espèce humaine gère extrêmement bien, pratiquement sans aucun effort, ni besoin de formation. D'un simple coup d'œil, les humains acquièrent une énorme quantité d'informations qui sont transmises au cerveau pour être traitées en fonction des besoins à un instant donné. Le tout presque inconsciemment, alors que l'humain exécute d'autres activités.

Tableau 1

Facile	Difficile
Quelle est la taille d'un plat ? Est-il sale ?	Identifier une cuisine et localiser tous les plats sales.
Est-ce que quelque chose a changé dans deux images successives ?	Recherchez et suivre une personne ou un objet et le poursuivre dans un environnement encombré.
Mesurer le diamètre d'une roue. Vérifier s'il n'est pas circulaire ?	Identifier chaque pièce d'un ensemble mécanique.
De quelle couleur est une feuille ?	De quel type de feuille s'agit-il ?

La conduite en est un exemple concret, un conducteur lors du situation critique analyse la route avec ses yeux, traite les informations et exécute une manœuvre (un coup de volant ou un freinage pour éviter un obstacle), le tout de manière instantanée et naturelle.

Facile non ? Mais pour les machines, les choses ne se passent pas de la même façon. Il faudrait reproduire le comportement du système œil/cerveau en une logique numérique. Cependant, les choses sont un peu plus complexes que ce qui n'y paraît à première vue.

Prenons, par exemple, l'action liée à la lecture d'un livre. Les yeux acquièrent l'image dans le champ de vision et le cerveau traite les images reçues sur la rétine pour en extraire les informations. Tout d'abord, le cerveau doit identifier quelle partie du champ de vision est occupée par le livre, en faisant abstraction du reste. La partie « mécanique » de l'œil nécessite la mise au point en continu de l'image (ici du texte lu car le livre est généralement incliné), la vision binoculaire permet d'évaluer la distance et d'ajuster les cristallins en conséquence. Étant donné que l'objet est proche, la mise au point doit être effectuée en continu.

Ensuite, il faut interpréter les symboles et les signes contenus dans les pages du livre. Les symboles qui représentent des lettres doivent être identifiés, en les séparant d'autres contenus. Ensuite les lettres regroupées doivent être identifiées comme des mots. Et enfin, les mots doivent être compris pour donner un sens au contenu du livre.

Pour toutes ces opérations, il faut tenir compte d'un grand nombre de variables à considérer et à traiter correctement. Par exemple, les différentes polices de caractères d'un livre à l'autre doivent être reconnues, mais aussi les fautes

d'orthographe dans un texte, un angle de vision différent, le suivi du texte autour d'une image et d'une page à l'autre.

Et si, après une pause, nous ouvrons le livre à la mauvaise page, ou nous ouvrons un autre livre ?

Un autre problème est le suivi d'objets en mouvement, comme une balle de tennis lors d'un match. Tout cela pose des problèmes considérables pour réaliser des algorithmes pouvant reproduire le comportement des yeux et du cerveau, il s'agit de tâches très complexes.

Pourtant, des spécialistes de différentes disciplines tentent de mettre au point la vision numérique. Quels sont les avantages apportés par ces recherches ? Un avantage important est donné par certaines propriétés informatiques qui sont meilleures que les capacités humaines correspondantes.

Les ordinateurs peuvent « voir » la même scène pendant des jours, voire des années sans jamais se lasser. S'ils sont correctement programmés, ils peuvent effectuer des mesures précises (longueurs, angles) et analyser tous les détails présents dans une image. Ils peuvent comparer des images successives pour vérifier s'il y

a eu des changements, même dans de petits intervalles de temps ou sur le long terme. Ils peuvent reconnaître des zones de couleurs différentes, ou des nuances de gris, et en extraire des contours et des formes, puis les comparer avec d'autres formes précédemment mémorisées.

À ce stade, nous pouvons donner une définition de la vision numérique. Il s'agit de la recherche et du développement d'applications où les capacités visuelles des ordinateurs peuvent être utilisées dans des applications difficiles pour l'œil humain, telles que l'analyse d'images dans le domaine médical, dans la sécurité, dans la gestion de véhicules autonomes.

La vision numérique est utilisée dans un domaine particulier qui consiste à développer des applications pour le contrôle de qualité, la gestion des processus de production sur des chaînes de montage (usines de voitures, etc.).

Mais alors pourquoi traiter de la vision numérique au niveau amateur ? Eh bien pour plusieurs raisons. La première est, comme cela s'est produit dans d'autres secteurs, la réduction des coûts des dispositifs technologiques. Les caméras ainsi que les dispositifs de traitement de l'image étaient autrefois

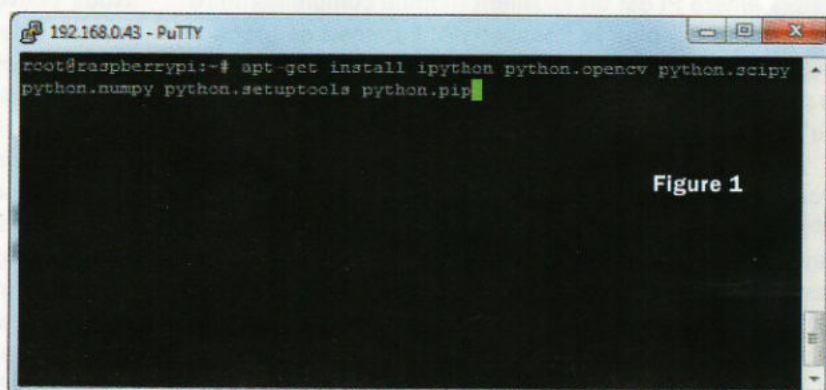


Figure 1

inaccessibles pour les amateurs, ces produits étaient réservés au domaine professionnel. De plus, les outils logiciels ont considérablement évolué, tout en étant moins coûteux. De nombreux logiciels Open Source fonctionnant sous Linux ont des caractéristiques proches de ceux utilisés dans les réalisations professionnelles.

La deuxième raison est que la vision numérique se développe très rapidement, en proposant des fonctionnalités supplémentaires dans différents domaines d'applications. Cela va de la superposition de lignes virtuelles lors de rencontres sportives au suivi d'objets ou de véhicules en mouvement (l'avion, le football, la course automobile, ...), ou encore dans l'agriculture pour la gestion des champs.

La vision numérique a aussi apporté un développement considérable des effets spéciaux pour le cinéma, mais aussi dans la médecine, la sécurité, etc.

Nous allons décrire les bases de la vision numérique avec des outils disponibles et abordables pour l'amateur. Pour cela, nous utiliserons un RaspberryPi, une Caméra Pi pour l'acquisition de l'image et des outils Open Source professionnels de traitement de l'image tels que **OpenCV**, **SimpleCV** et **Python**.

Nous vous conseillons de vous reporter à l'article paru dans la **revue 137 d'Electronique et Loisirs Magazine** qui traite de l'utilisation et de la configuration de Caméra Pi avec le RaspberryPi.

Nous allons nous consacrer dans cet article à la présentation de deux outils, « OpenCV » pour le traitement de l'image et le « framework » « SimpleCV » qui permet une utilisation facile avec le langage Python.

« OpenCV » est une librairie spécifique pour la vision numérique, développée à l'origine par Intel et sous licence Open Source BSD. Cette librairie est multiplateforme et peut être utilisée sous Linux, Mac OS X et Windows. Cette librairie est conçue principalement pour traiter les images en temps réel et supporte les principales fonctions suivantes :

- capture d'images en temps réel ;
- importation de fichiers vidéo ;
- traitement de l'image (luminosité, contraste, seuils, etc.) ;
- reconnaissance d'objets (faces, corps, etc.) ;
- reconnaissance de formes ayant des caractéristiques homogènes.

L'ampleur et la polyvalence des fonctionnalités d'OpenCV rendent assez difficile l'utilisation par un usager « normal » du logiciel, notamment dans le choix et

l'organisation des applications disponibles. Heureusement, nous utilisons une seconde librairie qui limite en partie certaines fonctionnalités d'OpenCV, afin de faciliter grandement son usage.

Parlons de « SimpleCV », qui est un « framework » écrit et utilisable en Python. Il est simple à utiliser, il inclut les fonctionnalités de la librairie « OpenCV », ainsi que les algorithmes de traitement de l'image sous formes de « briques ». Cela permet aux développeurs de créer des applications de vision artificielle, sans avoir nécessairement une connaissance approfondie de la vision numérique acquise dans les instituts de recherche les plus prestigieux dans le monde.

Avant de commencer à « travailler », nous donnons quelques indications supplémentaires concernant la complexité des problèmes de la vision numérique.

```

192.168.0.43 - PuTTY
root@raspberrypi:~# pip install https://github.com/sightmachine/SimpleCV/zipball/master
Downloading/unpacking https://github.com/sightmachine/SimpleCV/zipball/master
  Downloading master (unknown size): 53.7Mb downloaded
  Running setup.py egg_info for package from https://github.com/sightmachine/SimpleCV/zipball/master

Installing collected packages: SimpleCV
  Running setup.py install for SimpleCV

  Installing simplecv script to /usr/local/bin
Successfully installed SimpleCV
Cleaning up...
root@raspberrypi:~#
  
```

Figure 2

```

192.168.0.43 - PuTTY
root@raspberrypi:~# pip install cvwrite
Downloading/unpacking cvwrite
  Downloading cvwrite-1.1.3.tar.gz (174Kb): 174Kb downloaded
  Running setup.py egg_info for package cvwrite

Installing collected packages: cvwrite
  Running setup.py install for cvwrite

Successfully installed cvwrite
Cleaning up...
root@raspberrypi:~#
  
```

Figure 3

```

192.168.0.43 - PuTTY
root@raspberrypi:~# simplecv
SimpleCV 1.3.8 [interactive shell] - http://simplecv.org

Commands:
  "exit()" or press "Ctrl+D" to exit the shell
  "clear()" to clear the shell screen
  "tutorial()" to begin the SimpleCV interactive tutorial
  "example()" gives a list of examples you can run
  "forums()" will launch a web browser for the help forums
  "walkthrough()" will launch a web browser with a walkthrough

Usage:
  dot complete works to show library
  for example: image().save("/tmp/test.jpg") will dot complete
  just by touching TAB after typing image().

Documentation:
  help(image), ?image, image?, or image()? all do the same
  "docs()" will launch webbrowser showing documentation

SimpleCV:1>
  
```

Figure 4

Tableau 2

Couleur	Valeur
Rouge	(255, 0, 0)
Vert	(0, 255, 0)
Bleu	(0, 0, 255)
Jaune	(255, 255, 0)
Marron	(165, 42, 42)
Orange	(255, 165, 0)
Noir	(0, 0, 0)
Blanc	(255, 255, 255)

Vous trouverez dans le Tableau 1 quelques notions de bases sur la complexité de la mise en œuvre de la vision numérique.

Préparation du matériel

Notre système de vision est composé d'un RaspberryPi et du module Caméra Pi que nous avons présenté dans le numéro 137. Reportez-vous à ce numéro pour la configuration et l'utilisation basique de Caméra Pi.

Les bibliothèques développées pour la vision numérique sont conçues pour identifier et gérer les périphériques de vision connectés sur un port USB.

Le module Caméra Pi utilise un driver dédié qui gère l'interface directement à partir du connecteur CSI-2.

Pour remédier à ce problème, dans le programme d'exemple en Python que nous allons présenter, nous allons utiliser l'idée d'acquérir une image au moyen des commandes de gestion de Caméra Pi (celles que nous avons décrites dans le numéro 137), puis mémoriser l'image relative afin de la traiter avec « SimpleCV ».

Si vous ne désirez pas utiliser le module Caméra Pi, nous décrirons par la suite l'utilisation d'une webcam connectée à un port USB. Vous pouvez consulter la liste des webcams compatibles à l'adresse suivante :

http://elinux.org/RPi_USB_Webcams

Nous pouvons commencer. Ce que nous devons réaliser est une architecture applicative avec tous les modules nécessaires au fonctionnement correct des bibliothèques « OpenCV » et « SimpleCV » sous le système d'exploitation du RaspberryPi. Partons de la **configuration que nous avons développée dans l'article consacré à la Caméra Pi** (numéro 137), avec la Caméra Pi installée, le RaspberryPi connecté au réseau et sur lequel nous avons relié un moniteur connecté à la sortie HDMI.

Le clavier et la souris sont optionnels si nous nous connectons à distance au RaspberryPi à l'aide de PuTTY et WINScp.

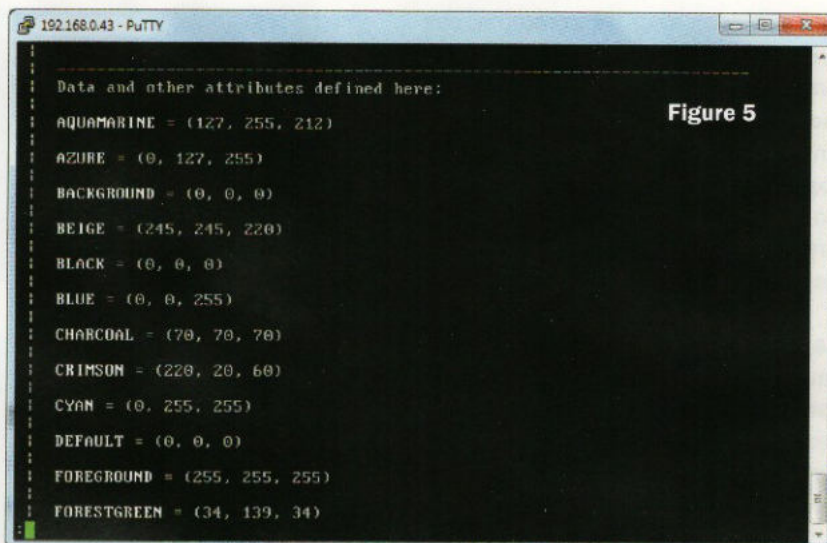


Figure 5

Comme toujours, nous prenons pour acquis les connaissances de base pour la gestion du système d'exploitation Raspbian du RaspberryPi, décrit dans les articles précédents (à partir du numéro 123).

Avant de commencer, nous devons mettre à jour le système d'exploitation à l'aide des commandes suivantes dans une fenêtre de terminal sur laquelle vous êtes connecté en tant qu'utilisateur « root » :

```
apt-get update
apt-get upgrade
```

Testons le fonctionnement de Caméra Pi avec la commande suivante :

```
raspistill -t 5000
```

Après avoir vérifié que tout fonctionne correctement, nous pouvons commencer à installer les composants logiciels pour notre système de vision numérique.

Commençons par installer la librairie « OpenCV », ainsi que ses packages prérequis et ses dépendances. En particulier « IPython » qui est un interpréteur du langage Python « amélioré ». Il fournit une interface (shell) interactive particulièrement riche en fonctionnalités, et capable de s'intégrer avec d'autres langages et environnements de développement, tels que, précisément « SimpleCV ».

Installons l'ensemble avec la commande suivante (figure 1) :

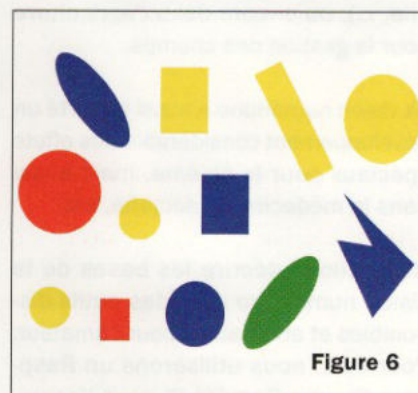


Figure 6

```
apt-get install ipython python-opencv
python-scipy python-numpy python-se-
tuptools python-pip
```

Ensuite, nous installons « SimpleCV » comme un paquet de Python. Notez que la commande « **pip** » (Python Package Index), que nous venons d'exécuter a installé un package qui nous permet de gérer les modules supplémentaires du langage Python (voir la figure 2) :

```
pip install https://github.com/sight-
machine/SimpleCV/zipball/master
```

Maintenant, nous installons la dépendance nécessaire au fonctionnement de « SimpleCV », à savoir « svgwrite ». « Svgwrite » s'occupe de la gestion du format vectoriel des images (SVG signifie Scalable Vector Graphics).

Comme il s'agit d'une extension du langage Python, nous l'installons avec la commande « **pip** » (figure 3) :

```
pip install svgwrite
```


Une fois les installations terminées, nous pouvons effectuer une première vérification en utilisant l'interface interactive de « SimpleCV » (une fonction équivalente à celle disponible pour le langage Python). Utilisons le shell pour vérifier le bon fonctionnement de la librairie « OpenCV ».

Tapez la commande suivante (voir la figure 4) : **simplecv**

Le shell s'ouvre en nous informant de la version et affiche un résumé des commandes. Pour sortir, tapons : **exit()**

Avant de commencer à expérimenter la vision numérique, faisons une pause pour présenter très brièvement les méthodes de numérisation et de stockage des images.

Au-delà des nombreux formats qui existent, et qui sont différents pour les couleurs, le noir et blanc ou encore la qualité de restitution (nombre de pixels et le nombre de couleurs), les images sont fondamentalement représentées par un ou plusieurs tableaux de pixels. Ces derniers sont des petits points très nombreux qui font varier la dimension et la qualité d'une image.

Le **pixel** (souvent abrégé en px) est l'unité de base permettant de mesurer la définition d'une image numérique matricielle. C'est l'unité minimale adressable par le contrôleur vidéo. C'est aussi l'unité utilisée pour spécifier les définitions d'affichage (largeur × hauteur).

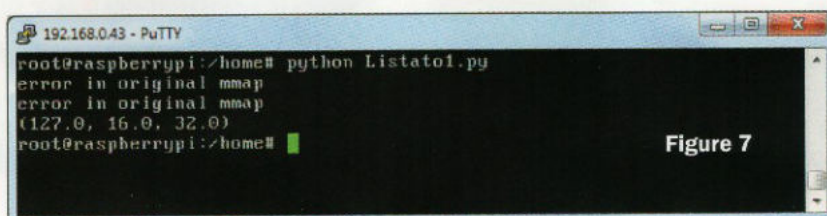
Une image en noir et blanc de 640 x 480 pixels est composée de 307 200 pixels (soit environ 0,3 mégapixels), chacun ayant une taille d'un ou plusieurs bits contenant les différentes nuances de couleurs présentes dans chaque point de l'image (chaque point est représenté par le pixel lui-même).

Une image représentée par des pixels ayant un seul bit sera une image en blanc (1) ou en noir (0). Le bit peut prendre la valeur 0 ou 1. L'image n'a pas de nuances de gris, il s'agit d'une image « **binarisée** ».

Les images en couleur nécessitent une matrice de pixels pour chaque couleur, généralement trois dans la

Listing 1

```
import subprocess
from SimpleCV import Image
import time
subprocess.call("raspistill -n -w %s -h %s -o Listing1_1.bmp" % (640, 480),
shell=True)
img = Image("Listing1_1.bmp")
img.show()
pixel = img[120, 200]
print pixel
img[120, 200] = (0,0,0)
pixel = img[120, 200]
img.save("Listing1_2.bmp")
print pixel
```



représentation RGB (Red, Green, Blue ou Rouge, Vert, Bleu). Ici aussi, la taille en bits de chaque pixel caractérise le nombre de couleurs qui peut être géré.

Dans le Tableau 2, vous apercevez les combinaisons des valeurs des pixels qui représentent les principales couleurs au format 8 bits. Dans cette représentation, les triades de valeurs identiques représentent toutes les nuances de gris.

Heureusement, lorsque nous travaillons sous « SimpleCV », il n'est pas nécessaire de connaître toutes les représentations numériques des couleurs, car elles peuvent être appelées par leur nom, comme par exemple : VERT, ROUGE, JAUNE, etc. Tapons la commande suivante (voir la figure 5) : **help(Color)**

Dans l'interface de « SimpleCV », la liste des couleurs reconnues s'affiche. À propos, la commande « **help(quelque chose)** » est le moyen d'obtenir de l'aide dans « SimpleCV ». Le menu d'aide s'obtient avec la commande « **help()** ».

Une fois que vous avez compris le mode de représentation d'une image, il devient plus facile de comprendre le mode de traitement.

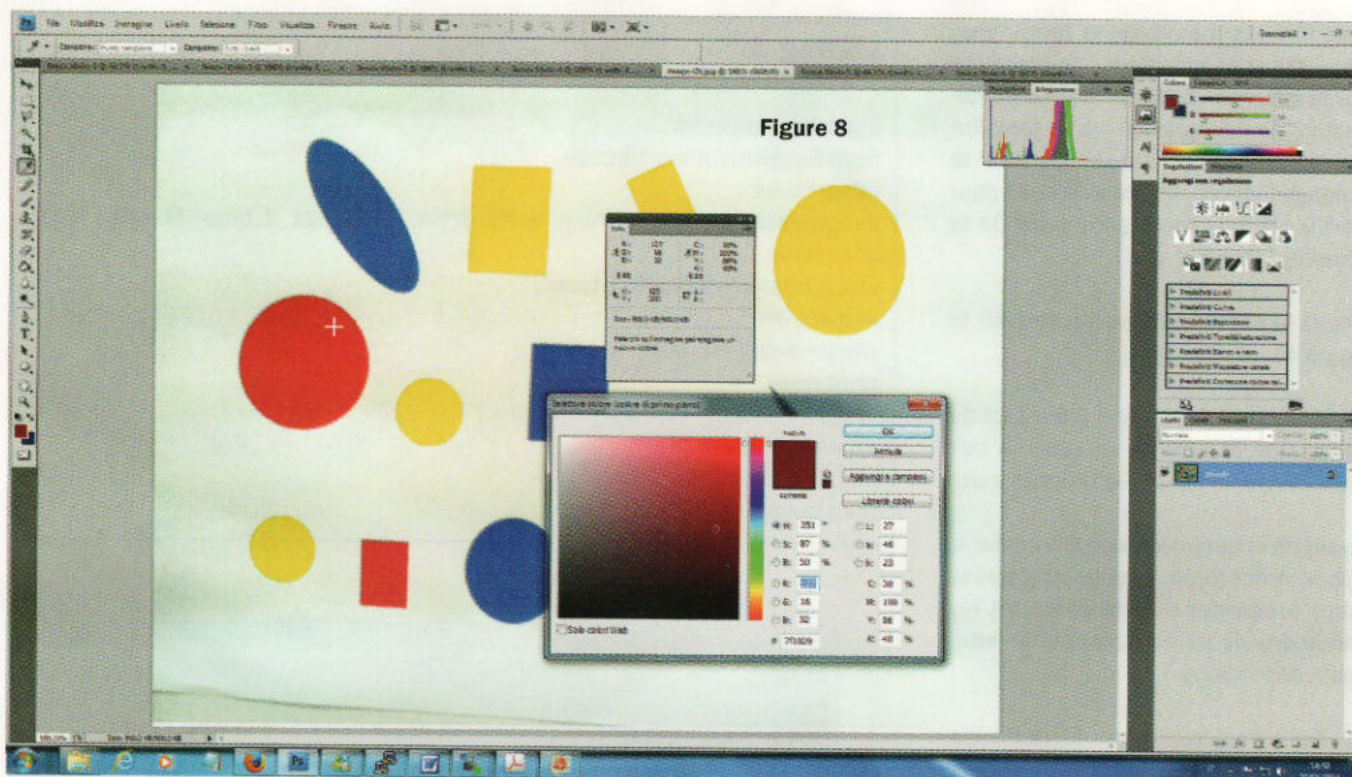
Par exemple, l'inversion horizontale d'une image se résout en inversant horizontalement chaque ligne de la matrice de pixels. Le premier pixel de chaque ligne devient le dernier, le dernier devient le premier, et ainsi de suite jusqu'aux pixels intermédiaires.

La transformation d'une image en couleur en une image en noir et blanc consiste à créer une matrice unique à partir des 3 originales où chaque pixel contient la moyenne des 3 valeurs des pixels correspondants aux matrices originales.

En substituant des pixels d'une image par d'autres de couleurs contrastées, il est possible de créer une image contenant du texte et des formes géométriques.

En effectuant des opérations algébriques sur des matrices de pixels, il est possible d'ajouter des images ou d'appliquer différents types d'effets.

Ainsi, si nous appliquons des opérations mathématiques plus complexes, il est possible de rechercher des formes dans des images et/ou de rechercher des similitudes entre différentes images.



Maintenant, nous allons étudier quelques exemples simples pour une première prise en main, bien que les problèmes liés au traitement de l'image soient complexes.

Comme base de référence, nous utilisons une feuille sur laquelle nous avons imprimé quelques formes géométriques simples, de tailles et de couleurs différentes (voir la figure 6).

Vous pouvez télécharger sur notre site l'image de la figure 6 nommée « **doc1.jpg** ». Utilisez-la pour faire des expériences, en modifiant la feuille et les paramètres de personnalisation des fonctions.

Vous pourrez mieux comprendre les problèmes liés à l'acquisition et au traitement de l'image. L'éclairage joue un rôle primordial, ainsi que les distances, l'angle de vision, la taille, le contraste et l'uniformité des couleurs.

Pour commencer, dans le listing 1, vous pouvez voir le code traitant de la reconnaissance de la couleur d'un pixel spécifique dans une image. Pour cela, nous prenons en compte que l'indexation des éléments (pixels) commence dans la partie supérieure gauche de l'image.

L'index du premier pixel dans le coin supérieur gauche de l'image a la valeur (0,0). En particulier, dans le listing 1, nous extrayons la valeur du pixel (120, 200). Elle est obtenue en comptant 120 pixels vers le bas et 200 pixels vers la droite, plus ou moins au centre du premier cercle rouge (environ) en haut à droite.

Dans le Listing 1, la fonction « **subprocess.call()** » permet de lancer un processus à l'extérieur du programme en cours d'exécution et d'attendre qu'il soit terminé avant de poursuivre le traitement.

Le processus externe qui est lancé est le programme « **raspistill** » avec ses paramètres, ce qui nous permet d'acquérir une image et de l'enregistrer sur une mémoire de masse avec le nom de notre choix.

La fonction « **Image()** » charge dans une variable le fichier image spécifié.

La fonction « **show()** » affiche l'image à l'écran. Enfin, la fonction « **img()** » permet d'accéder en lecture/écriture au contenu de l'image.

Copions le programme du listing 1 dans un fichier que nous nommons

« **Listing1.py** » dans le répertoire « **/home** », et exécutons-le avec la commande :

python Listing1.py

Dans ce listing et dans chacun des suivants, nous avons inclus l'enregistrement de l'image intermédiaire obtenue pendant le processus, de façon à pouvoir l'analyser ultérieurement.

Les images sont enregistrées dans le dossier « **/home** », auquel nous pouvons accéder (par exemple en utilisant WinSCP) afin de les ouvrir avec un programme de visualisation et/ou d'édition graphique.

Nous obtenons le résultat visible en figure 7, qui indique la valeur du pixel (127, 16, 32). Nous pouvons vérifier l'exactitude du résultat en ouvrant l'image avec Photoshop et en affichant la valeur du pixel comme visible en figure 8. Ils correspondent parfaitement.

Ceux d'entre vous ne possédant pas Photoshop peuvent utiliser sur le RaspberryPi le programme graphique Open Source nommé « **Gimp** ».

Le paquet « **Gimp** » doit être installé par l'utilisateur en utilisant le système

des gestionnaires de paquets. Comme il occupe beaucoup de place, vous devez vérifier votre espace disponible sur la carte SD. Pour installer « Gimp », ouvrez une fenêtre de terminal et tapez la commande suivante :

sudo apt-get install gimp

Gimp fait appel à 3 fenêtres et non une seule, son utilisation peut être un peu déconcertante au début.

Par défaut la fenêtre de gauche contient la boîte à outils.

Dans la fenêtre de droite, se trouvent les options des calques, des canaux et des dégradés.

Dans la fenêtre du milieu s'affiche l'image en cours d'édition. Il y a autant de fenêtres centrales que d'images ouvertes simultanément.

En revanche, les fenêtres latérales boîte à outils, options des calques, canaux et gradients, restent uniques. Il est possible de simplifier l'interface en optant pour le Mode fenêtre unique qui sera plus familier pour les utilisateurs de Photoshop. Pour activer ce mode, cliquez sur le menu « **Fenêtre** » puis sélectionnez « **Mode fenêtre unique** ».

Par défaut, le manuel de l'utilisateur de Gimp n'est pas installé. Gimp est un outil puissant et son mode d'emploi prend beaucoup de place sur la carte SD. Il est possible d'obtenir le manuel en appuyant sur la touche F1 ou allant dans le menu « Aide ».

Gimp est un outil très puissant qui nécessite beaucoup de mémoire. Son fonctionnement sur le RaspberryPi est donc assez lent, mais tout à fait acceptable.

Avec des photos de grande taille toutefois, il est possible que davantage de patience soit nécessaire. Augmentez la quantité de mémoire disponible pour le GPU pour améliorer les performances.

L'enregistrement des fichiers peut se faire sous un grand nombre de formats. Si le fichier doit être édité à nouveau, le format propriétaire Gimp XCF, utilisé par défaut, convient.

Il conserve les métadonnées intactes, utilise un format de compression sans perte pour une qualité d'image optimale, et prend en charge les images à plusieurs niveaux de calques.

Si l'image doit être mise en ligne ou partagée avec des tiers, il est recommandé d'utiliser un format plus répandu comme « **jpg** » ou le « **png** ».

Cet exemple simple nous permet de vérifier concrètement l'aide que nous apporte « SimpleCV » dans le traitement de l'image. La fonction « **img (120,200)** » fait tout le travail pour nous.

Pour définir un pixel particulier avec une couleur donnée, utilisons l'instruction suivante :

img[120, 200] = (0,0,0)

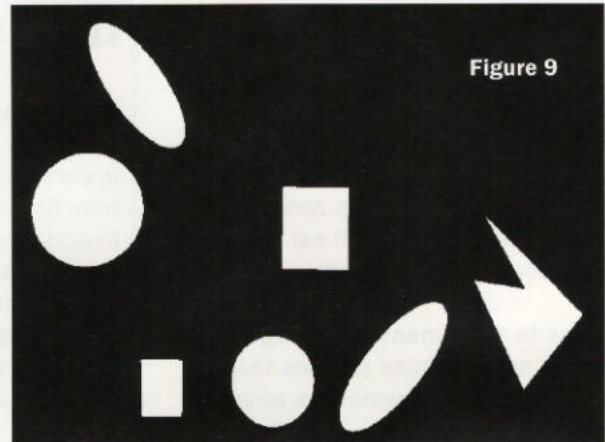


Figure 9

Evidemment, nous remplaçons les paramètres (0, 0, 0) par les valeurs R, V, B de la couleur que nous désirons. Cela nous permet d'opérer sur les images que nous voulons. Dans la librairie « SimpleCV », il existe un grand nombre de fonctions permettant de générer des lignes, des figures géométriques, des zones de remplissage, d'appliquer la transparence, de remplacer du texte, etc.

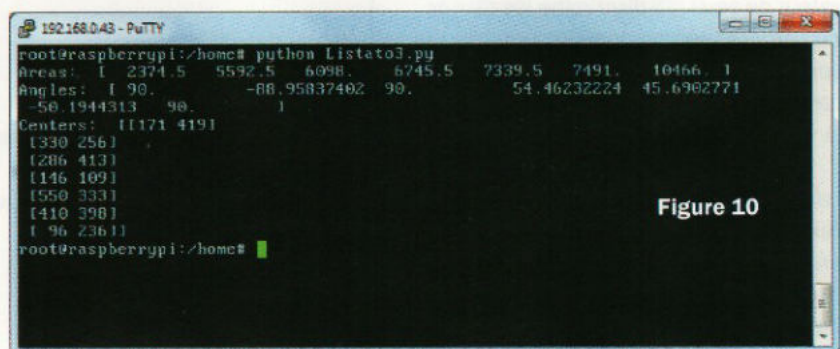


Figure 10

Listing 2

```

#!/usr/bin/python

import subprocess
from SimpleCV import Image
import time

subprocess.call("raspistill -n -w %s -h %s -o listing2_1.png" % (640, 480),
shell=True)

img = Image("listing2_1.png")
img.show()
time.sleep(5)
img = img.binarize()
img.show()
time.sleep(5)
macchie = img.findBlobs()
img.save("listing2_2.png")
img.show()
time.sleep(20)

```


Faisons un bond en avant et commençons à décrire la manière d'identifier une ou plusieurs formes dans une image. Débutons avec un exemple simple qui consiste à reconnaître des zones colorées dans une image ayant un fond blanc. Pour les zones ayant une couleur uniforme, il est possible de reconnaître un contour.

Dans le traitement de l'image, ces zones sont définies par des tâches. La fonction qui reconnaît les zones de couleur contrastant avec le fond est appelée « **findBlobs()** ». Dans le listing 2, vous pouvez apercevoir une application de cette fonction.

La fonction « **binarize()** » transforme une image originale en une image en noir et blanc sans nuance de gris, afin d'accentuer les tâches. Cependant les zones jaunes ne sont pas traitées par cette fonction, car elles ressemblent trop à l'arrière-plan blanc de l'image.

Vous pouvez essayer de « corriger » la valeur de seuil qui discrimine les tâches de l'arrière-plan en donnant une valeur comprise entre 0 et 255 au paramètre de la fonction « **binarize()** », par exemple la valeur 130 : « **binarize(130)** ».

La fonction « **findBlobs()** », en plus de reconnaître les tâches, restitue un ensemble d'informations pour chaque zone identifiée, telles que la dimension, l'angle (direction) par rapport à l'axe des X et les coordonnées du centre de la zone (voir la figure 10).

Ces informations sont très utiles dans des applications comme la robotique.

Dans le listing 3, nous trouvons un exemple de visualisation de ces informations. Elles sont contenues dans 3 tableaux. L'ordre de classement du tableau va dans le sens décroissant de la dimension des zones reconnues (de la plus grande à la plus petite).

En premier est indiqué la tâche ayant la plus grande taille, puis en décroissant, toutes les autres.

Nous pouvons également appliquer un filtre de sélection pour restreindre la recherche d'éventuelles zones potentielles de couleurs contrastées.

Listing 3

```
#!/usr/bin/python

import subprocess
from SimpleCV import Image
import time
subprocess.call("raspistill -n -w %s -h %s -o listing3_1.png" % (640, 480),
shell=True)
img = Image("listing3_1.png")
img.show()
time.sleep(5)
img = img.binarize()
img.show()
time.sleep(5)
tache = img.findBlobs()
img.show()
time.sleep(5)
print "Zones: ", tache.area()
print "Angles: ", tache.angle()
print "Centre: ", tache.coordinates()
```

Listing 4

```
#!/usr/bin/python

import subprocess
from SimpleCV import Color, Image
import time
subprocess.call("raspistill -n -w %s -h %s -o listing4_1.png" % (640, 480),
shell=True)
img = Image("listing4_1.png")
img.show()
time.sleep(5)
couleur = (22,36,75)
blue_distance = img.colorDistance(couleur).invert()
blobs = blue_distance.findBlobs()
blobs.draw(color=Color.BLACK, width=4)
blue_distance.save("listing4_2.png")
blue_distance.show()
time.sleep(5)
img.addDrawingLayer(blue_distance.dl())
img.save("listing4_3.png")
img.show()
time.sleep(5)
```

Un critère est, bien évidemment, la couleur. Nous pouvons décider de reconnaître uniquement les zones de couleur bleue, et mettre en évidence les bordures de ces dernières à l'aide de lignes dont les couleurs sont contrastées. Nous faisons cela avec le code du listing 4.

La couleur de référence (22, 36, 75) est toujours identifiée avec la méthode « **Photoshop** » ou « **Gimp** ».

La fonction « **colorDistance()** » soustrait de l'image des pixels qui représentent des couleurs « assez » différentes de la couleur de référence, tout

en préservant les pixels représentant des couleurs « assez » similaires.

Une fois les tâches souhaitées reconnues, nous les mettons en évidence avec un contour noir d'une épaisseur de 3 pixels à l'aide de la fonction « **draw()** ».

Après le lancement du programme, très probablement vous vous apercevrez qu'une ellipse verte foncée a été également reconnue, en effet c'est une couleur « assez » proche du bleu (voir les figures 11 et 12).

Les conditions d'éclairage peuvent affecter grandement les résultats de la reconnaissance. Vous devez faire plusieurs tests.

L'étape suivante consiste à reconnaître un contour ayant une forme particulière, tel qu'une ligne, un cercle ou une arête. Considérons une ligne comme une séquence de points qui se situent sur un segment de droite, ils constituent généralement une partie d'un contour d'une forme ou d'une figure.

La méthode utilisée par la fonction « **findLine()** » de « SimpleCV » est basée sur la **transformée de Hough**. La transformée de « Hough » est une technique de reconnaissance de formes inventée en 1962 par Paul Hough, et utilisée dans le traitement d'images numériques. L'application la plus simple permet de détecter les lignes présentes dans une image.

Cette méthode suppose que si des lignes ou des segments de droites sont présents dans une image, ils feront partie des contours présents dans l'image. On commence donc dans un premier temps par identifier tous les points de contours de cette image, par exemple à l'aide de techniques de mesures de gradients entre les valeurs des pixels autour de chaque point de l'image.

Les points de l'image présentant les gradients les plus élevés dans leur voisinage sont les plus susceptibles d'appartenir à des contours de cette image.

Chacun des points des contours ainsi identifiés (x, y) permet alors une projection dans un plan des coordonnées

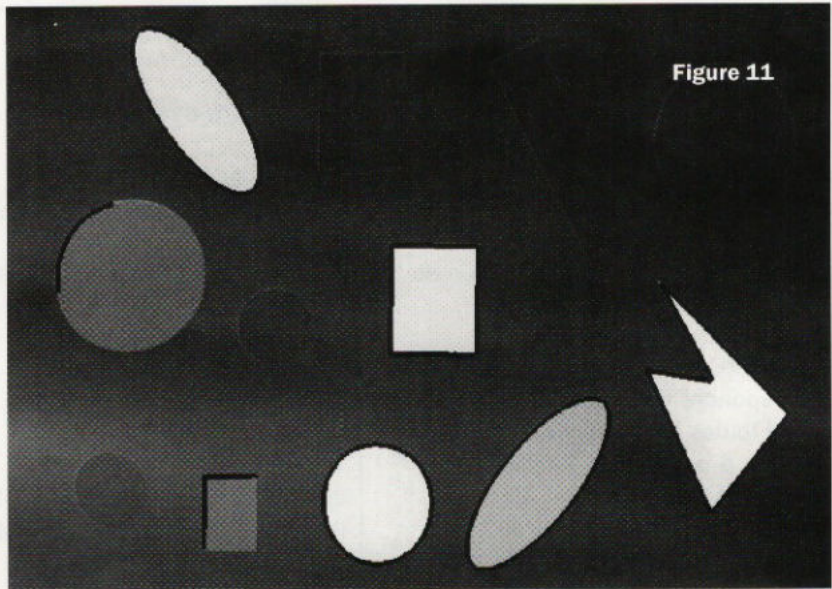


Figure 11

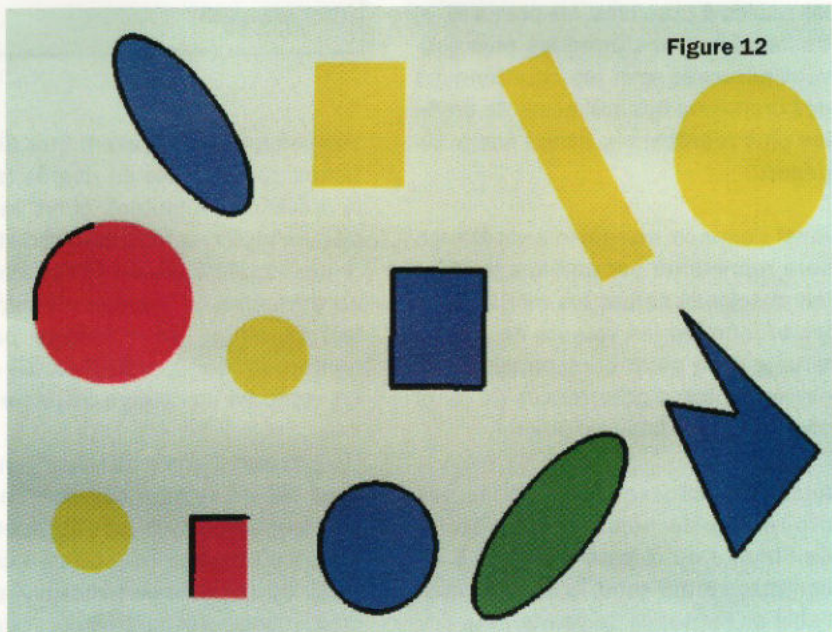


Figure 12

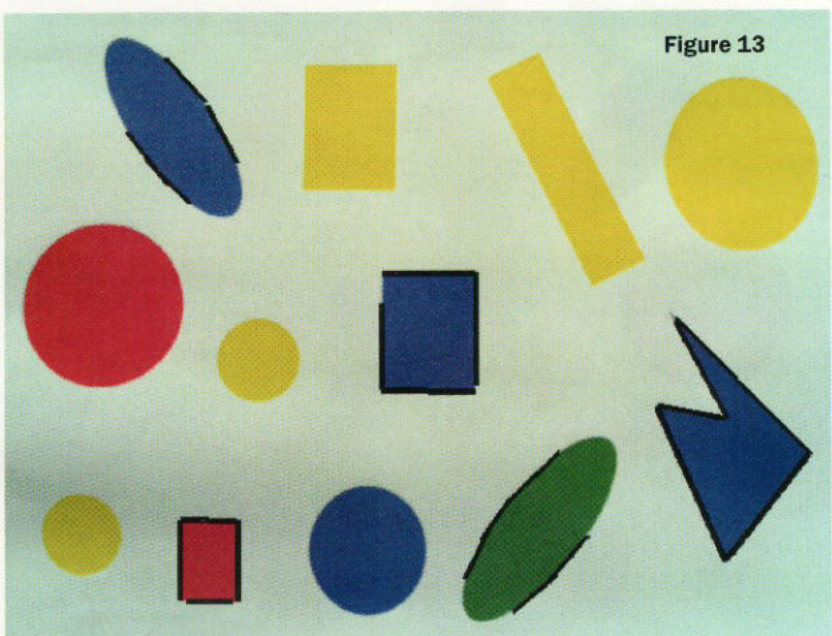


Figure 13

polaires de toutes les droites passant par ce point. Les équations des droites passant en chacun de ces points (x, y) sont représentées par l'équation :

$$\rho = y * \cos(\theta) + x * \sin(\theta)$$

où θ est l'angle de la droite et ρ la distance de la droite à l'origine (au lieu de $c = a x + b y$).

Au point (x, y) du contour, nous faisons correspondre une courbe $[\theta, \rho]$, où θ prend toutes les valeurs possibles de 0 à 2π , ρ prenant la valeur $\rho = y * \cos(\theta) + x * \sin(\theta)$.

Une fois appliquée à tous les points des contours, et, pour chacun, à tous les angles θ possibles, les points (θ, ρ) de l'espace transformé les plus souvent adressés sont les coordonnées des droites ou des segments de droite les plus représentés dans l'image de départ.

Ainsi l'espace transformé de Hough sera représenté par une image, dont les abscisses seront les angles θ , et les ordonnées les valeurs de ρ . L'intensité d'un point quelconque (θ, ρ) est le nombre d'occurrences de (θ, ρ) provenant de l'image d'origine.

Aucune hypothèse de continuité des droites ou de segments de droite de l'image de départ n'est prise en compte, ce qui rend la transformée fiable en l'absence de points.

Listing 5

```
#!/usr/bin/python

import subprocess
from SimpleCV import Color, Image
import time
subprocess.call("raspistill -n -w %s -h %s -o listing5_1.png" % (640, 480), shell=True)
img = Image("listing5_1.png")
img.show()
time.sleep(5)

# Règle la valeur du seuil
lines = img.findLines(threshold=30)
lines.draw(color=Color.BLACK, width=4)
img.save("listing5_2.png")
img.show()
time.sleep(15)
```

Les valeurs de θ peuvent être discrétisées par exemple en degrés (selon la précision souhaitée), et les valeurs de ρ en pixels représentant la distance (toujours inférieure en valeur absolue au diamètre de l'image de départ), la fréquence étant bornée par le nombre de points sélectionnés dans les contours de l'image de départ.

Dans la transformée de Hough, chaque ligne est un vecteur de coordonnées paramétriques où θ est l'angle et ρ la norme du vecteur (c'est-à-dire la longueur du segment perpendiculaire à la droite d'angle θ et passant par l'origine).

En transformant toutes les lignes possibles qui passent par un point, c'est-à-dire en calculant la valeur de ρ pour chaque θ , nous obtenons une **sinusoïde unique** appelée « **Espace de Hough** ».

Analysons toutes les lignes possibles qui passent par les points du plan à deux dimensions (notre image).

À chacun des points de l'image, sont affectées des notes croissantes lorsque ceux-ci se situent sur un segment qui relie deux points, tels que par exemple le sommet d'une figure géométrique.

Lançons le programme, nous trouverons probablement que certaines lignes droites ne sont pas reconnues (la feuille n'est pas parfaitement plate), alors qu'elles pourraient être reconnues comme des lignes « assez » rectilignes comme faisant partie d'un « côté » long d'une ellipse (voir la figure 13).

Utilisons la fonction « **find.Circle()** » pour reconnaître des formes circulaires dans l'image. Cette fonction dispose de **3 paramètres**. « **Canny** » représente le niveau de précision adopté par l'algorithme d'identification des formes circulaires.

La valeur intermédiaire est 100, avec une valeur plus élevée, l'algorithme

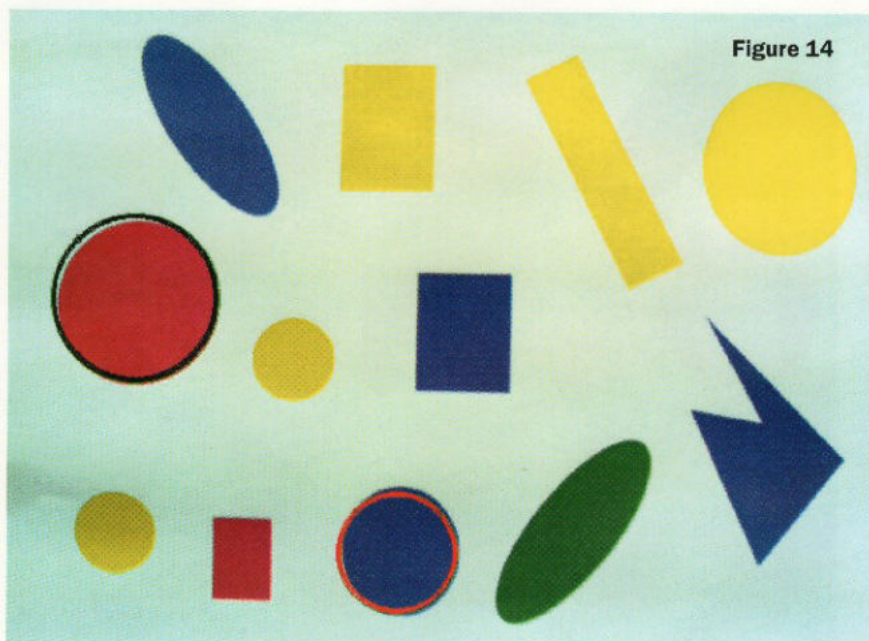


Figure 14

devient de plus en plus précis et reconnaît moins les cercles.

Avec une valeur plus faible, l'algorithme devient plus approximatif, il reconnaît des formes qui ne sont pas parfaitement circulaires.

« **Thresh** » correspond au niveau du seuil, comme dans le cas de la reconnaissance des lignes droites.

« **Distance** » représente la distance minimale selon laquelle la forme reconnue correspond à un seul cercle ou plusieurs cercles concentriques.

Le reste des instructions classent les cercles reconnus par ordre croissant en fonction de l'aire de chacun.

Les petits cercles [0] sont mis en évidence avec un contour rouge, tandis que tous les autres sont mis en évidence avec un contour noir (voir la figure 14).

Lancez le programme et ... amusez-vous. Essayez de changer les paramètres.

Prenez une feuille de papier sur laquelle vous imprimez des cercles non remplis, avec des couleurs différentes pour les circonférences. Faites des tests avec différentes intensités lumineuses, etc.

Nous espérons vous avoir convaincu avec la vision numérique, il ne s'agit pas d'une science exacte. Elle demande parfois, en fonction des problèmes rencontrés, une analyse approfondie et beaucoup d'expérimentations avant d'adopter des solutions définitives.

D'autre part, elle stimule l'imagination et procure un certain sens du divertissement.

Pour conclure cette première partie de la vision numérique, nous présentons une fonction de « SimpleCV » qui permet de lire les codes à barres.

Cette fonction s'appelle « findBarcode() » et est capable de reconnaître automatiquement le type et le contenu du code à barres dans une image.

Listing 6

```
#!/usr/bin/python

import subprocess
from SimpleCV import Color, Image
import time

subprocess.call("raspistill -n -w %s -h %s -o listing6_1.png" % (640, 480),
shell=True)
img = Image("listing6_1.png")
img.show()
time.sleep(5)
cercle = img.findCircle(canny=250,thresh=200,distance=11)
cercle.draw(color=Color.BLACK, width=4)
cercle.show()
time.sleep(5)
cercle = cercle.sortArea()
cercle[0].draw(color=Color.RED, width=4)
img_with_circles = img.applyLayers()
img_with_circles.save("listing6_2.png")
img_with_circles.show()
time.sleep(15)
```

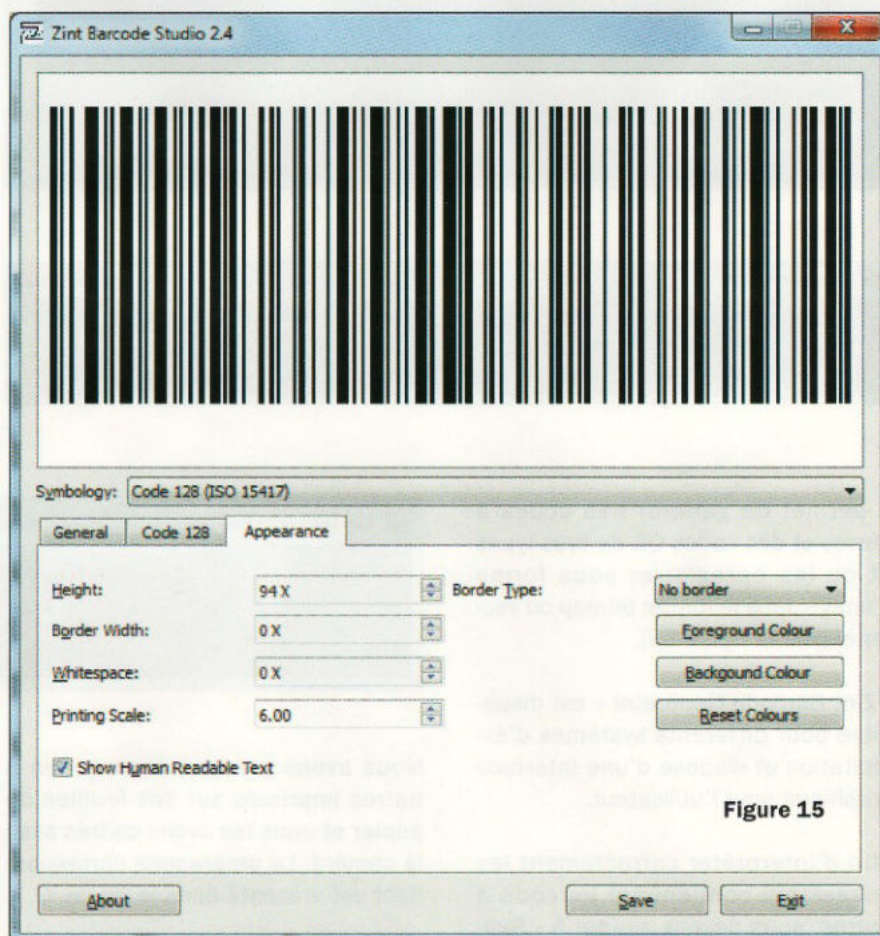


Figure 15

Les **codes QR** (Quick Response code) sont également reconnus. Ce sont des codes à barres spéciaux sous forme de

graphique à 2 dimensions constitué par des modules noirs disposés dans un carré à fond blanc.

L'agencement de ces points définit l'information que contient le code.

Les codes QR sont désormais devenus une norme pour l'information, en particulier grâce à des applications développées pour les appareils mobiles.

Un cryptogramme de type **code QR** peut contenir jusqu'à **7089 caractères numériques**, ou **4296 caractères alphanumériques**, ou encore **2 953 octets**.

Pour la génération des codes QR, nous utilisons le programme « **Zint Barcode Generator** », qui est un programme Open Source.

Listing 7

```
#!/usr/bin/python

import subprocess
from SimpleCV import Image, Barcode
import time

subprocess.call("raspistill -n -w %s -h %s -o listing7_3.png" % (640, 480),
shell=True)
img = Image("listing7_3.png")
img.show()
time.sleep(5)

# charge un code à barres unidimensionnel
barcode = img.findBarcode()
print barcode[0].data
```

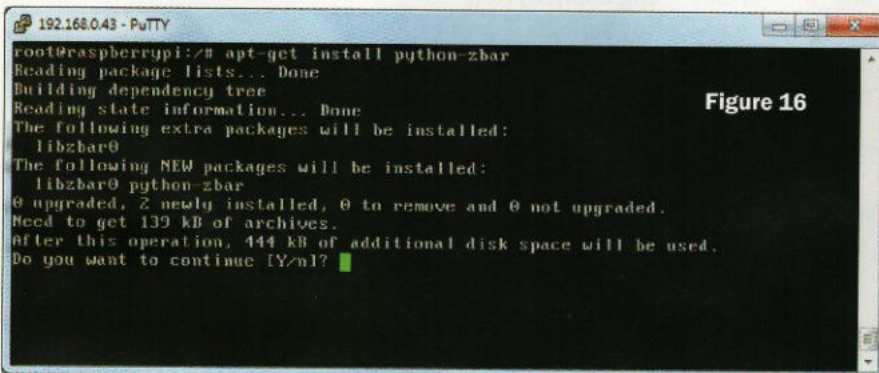


Figure 16

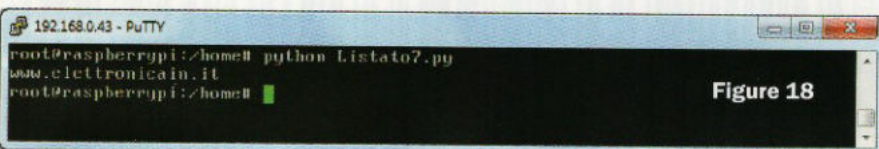


Figure 18



Figure 17



Figure 19

Il permet de générer des codes à barres et des codes QR de tous types et de les enregistrer sous forme d'image dans le format bitmap ou vectoriel (voir la figure 15).

« Zint Barcode Generator » est disponible pour différents systèmes d'exploitation et dispose d'une interface graphique pour l'utilisateur.

Afin d'interpréter correctement les images qui contiennent un code à barres, nous devons ajouter à « SimpleCV » la librairie « **zbar** ». Installons cette dernière avec la commande suivante (voir la figure 16) :

apt-get install python-zbar



Figure 20

Nous avons préparé deux codes à barres imprimés sur des feuilles de papier et nous les avons cadrés avec la caméra. Le programme correspondant est présenté dans le listing 7.

En figure 17 et 18, vous pouvez voir un code à barres classique avec le résultat du traitement, tandis qu'aux figures 19 et 20 vous pouvez voir un code QR et son résultat.

Tous les exemples que nous avons décrits fonctionnent sans la présence d'une caméra.

Vous pouvez aussi traiter des images déjà capturées ou provenant de sources différentes. Il suffit de supprimer les instructions de capture d'image et de « pointer » (diriger) la fonction « image() » vers le chemin/fichier contenant l'image que vous voulez traiter.

CHARGEUR UNIVERSEL INTELLIGENT



Ce chargeur universel intelligent est facilement transportable, il est idéal pour faire fonctionner et recharger des smartphones ou d'autres dispositifs mobiles. Doté de connexions mini-USB et micro-USB, il s'adapte automatiquement au standard de recharge du dispositif auquel il est connecté.

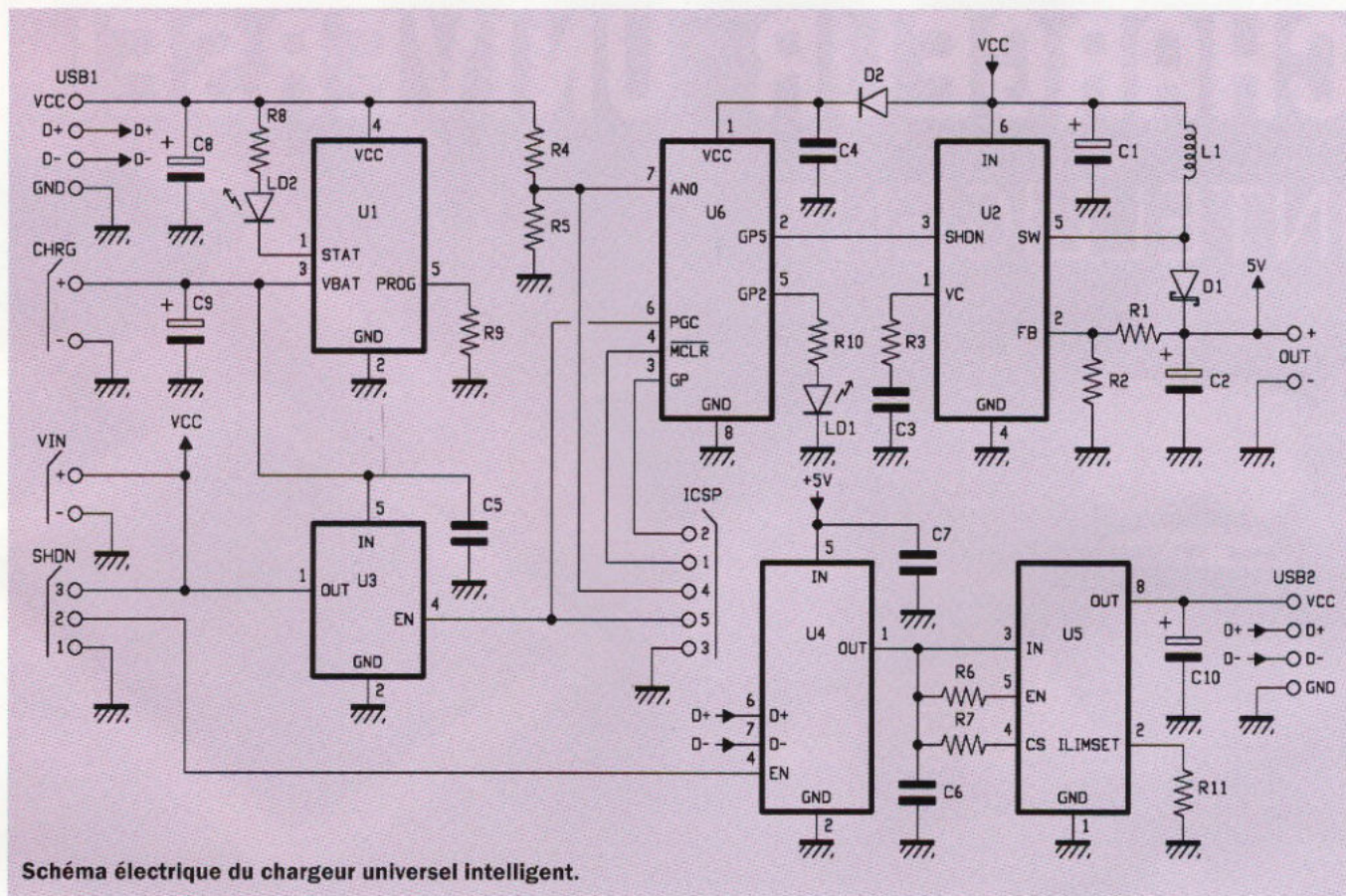
de Vincenzo MENDOLA

La portabilité de nombreux dispositifs a été rendue possible par l'amélioration de l'autonomie d'utilisation, obtenue soit par l'augmentation de la densité d'énergie (énergie par unité de volume) de la batterie, soit par l'utilisation de microprocesseurs et microcontrôleurs plus puissants mais qui consomment moins d'énergie. L'amélioration de la consommation des microprocesseurs et microcontrôleurs modernes est possible en utilisant un mode veille et en désactivant certains registres d'entrées/sorties et d'autres périphériques systèmes lorsqu'ils ne sont pas nécessaires.

En ce qui concerne la batterie, l'augmentation importante de la densité énergétique a été obtenue avec l'utilisation des technologies **Li-ion** et **Li-PO**.

Aujourd'hui, tous les appareils mobiles sont dotés d'une batterie au lithium. Malgré l'augmentation de l'autonomie de fonctionnement des appareils mobiles modernes, il arrive souvent de se retrouver avec une batterie déchargée, parfois cela se produit au milieu d'un appel important ou d'une conversation avec un vieil ami que l'on n'a pas vu depuis longtemps. Pour remédier à ce problème, il existe dans le commerce des chargeurs portables utilisant l'énergie d'une batterie interne, généralement au lithium.

Le projet proposé dans ces pages est un peu différent des chargeurs du commerce. Il utilise l'énergie fournie par une batterie au lithium en élevant sa tension à 5 V, afin de la rendre disponible sur un port USB, s'adaptant ainsi automatiquement aux normes utilisées par les appareils connectés.



Le chargeur intelligent

La recharge de la batterie s'effectue au moyen d'un chargeur secteur ou d'une prise allume-cigare, du moins dans les anciens véhicules. De nos jours, ils sont tous équipés d'au moins une prise USB.

Afin de rendre la vie plus facile pour les utilisateurs et réduire le coût du produit final, nous avons effectué des recherches pour vous proposer un chargeur standardisé. Idéalement, un chargeur doit disposer d'au moins une prise USB et une mini-USB ou une micro-USB.

Cependant, la standardisation représente un danger pour les fabricants d'accessoires compatibles pour appareils mobiles, car ils ne voient pas d'un bon œil l'émergence d'un consortium de la connectique.

Par ailleurs, certains constructeurs comme Apple, proposent des systèmes de recharges propriétaires qui ne sont pas compatibles avec les chargeurs standards équipés de prise

mini-USB, mais nécessitent un chargeur d'origine.

Cela, non pas parce que les chargeurs compatibles ne fournissent pas la tension correcte (une connexion USB doit fournir 5 VDC) ou assez de courant, mais parce que les constructeurs de smartphones utilisent une connectique différente qui dans certains cas sont propriétaires.

Dans ce cas, il est nécessaire de respecter une configuration bien précise de la tension sur les lignes de données USB, configuration indispensable pour que le téléphone soit reconnu et rechargé correctement.

Pour comprendre le fonctionnement de notre circuit, il faut comprendre comment se produit l'identification du standard de recharge que le constructeur utilise et comment l'électronique du smartphone doit s'adapter.

Les constructeurs d'appareils portables de ce genre **exigent que soient présents**, pour un modèle spécifique, des **signaux sur les deux broches de**

données (D+ et D-) de la **prise USB** afin de délivrer la tension d'**alimentation correcte**.

Un port de charge dédié ou **DCP (Dedicated Charging Port)** s'occupe de **monitorer les tensions** présentes sur les lignes de données du port USB et ainsi fournir automatiquement la bonne « **signature** » de la tension sur les broche « **D+** » et « **D-** », en l'adaptant au standard requis et utilisé par l'appareil à recharger.

En général, les périphériques mobiles nécessitent des chargeurs respectant certaines normes.

Lorsque les premiers chargeurs USB secteur (ou voiture) sont apparus, il fallait que le smartphone distingue une connexion USB vers un PC limitée à 500 mA d'une connexion vers un chargeur fort courant.

Une norme, adoptée progressivement par tous les fabricants sauf Apple, a proposé que l'identification d'un chargeur fort courant se fasse de la manière suivante.

Plan de montage du chargeur intelligent

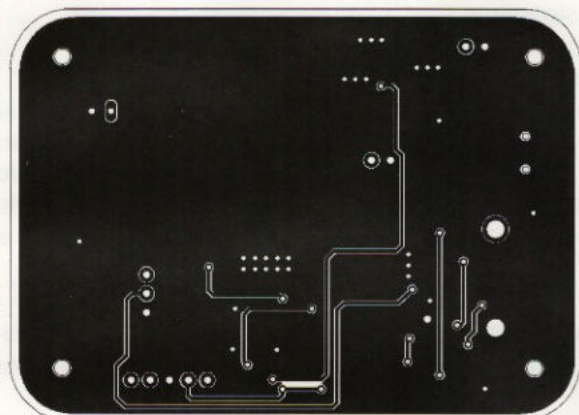


Figure A : circuit imprimé à l'échelle 1 : 1 côté soudures du chargeur intelligent.

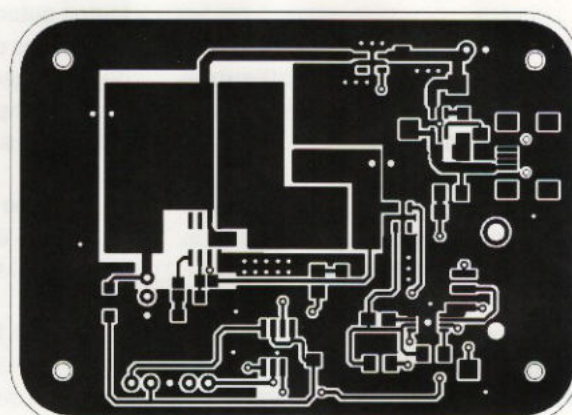


Figure B : circuit imprimé à l'échelle 1 : 1 côté composants du chargeur intelligent.

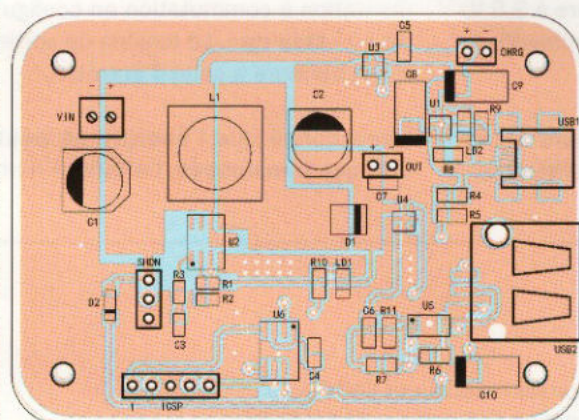


Figure C : plan de câblage des composants du chargeur intelligent.

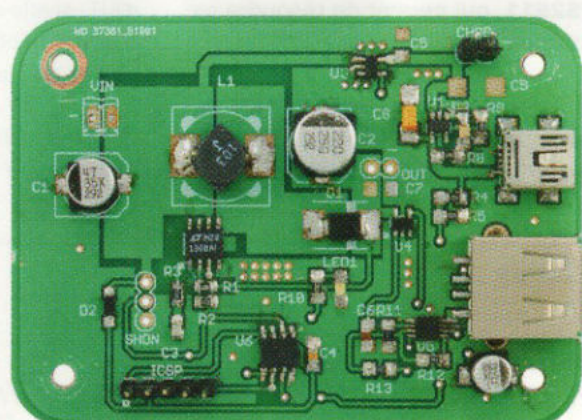


Figure D : photo de l'un de nos prototypes du chargeur intelligent.

Liste des composants du chargeur intelligent

R1..... 1 M Ω boîtier 0805
 R2..... 330 k Ω boîtier 0805
 R3..... 100 k Ω boîtier 0805
 R4..... 4,7 k Ω boîtier 1206
 R5..... 4,7 k Ω boîtier 1206
 R6..... 100 k Ω boîtier 1206
 R7..... 22 k Ω boîtier 1206
 R8..... 470 Ω boîtier 1206
 R9..... 2 k Ω boîtier 1206
 R10.... 470 Ω boîtier 1206
 R11.... 22 k Ω boîtier 1206
 C1..... 47 μ F/6,3 V électrolytique boîtier E

C2..... 220 μ F/6,3 V électrolytique boîtier E
 C3..... 680 pF boîtier 0805
 C4..... 100 nF boîtier 1206
 C5..... 100 nF boîtier 1206
 C6..... 100 nF boîtier 1206
 C7..... 100 nF boîtier 1206
 C8..... 4,7 μ F tantale
 C9..... 4,7 μ F tantale
 C10.... 4,7 μ F tantale

D1..... B540C
 D2..... MBR0520LT

U1..... MCP73831T-2ATI/OT
 U2..... LT1308A

U3..... TPS2061
 U4..... TPS2065C
 U5..... TPS2511
 U6..... PIC12F683-I/SN
 L1 inductance 10 μ H DR125-100-R (www.farnell.fr)
 LD1.... LED verte boîtier 1206
 LD2.... LED verte boîtier 1206

Divers

Bornier 2 pôles au pas de 2,54mm
 Connecteur USB-A Femelle
 Connecteur mini-USB
 Barrette mâle 2 pôles (x2)
 Barrette mâle 3 pôles (x1)
 Barrette mâle 5 pôles (x1)

Les deux lignes de données « D+ » et « D- » sont reliées par une résistance d'une valeur inférieure à 200 Ω et doivent rester flottantes par rapport à l'alimentation.

Apple a décidé d'identifier les sources à sa manière :

- chargeur faible courant (< 500mA)
 D+ = 2,0 V ; D- = 2,7 V ;

- chargeur fort courant (jusqu'à 2,1 A)
 D+ = 2,7 V ; D- = 2,0 V.

Avec l'arrivée de la norme USB 3.0 qui fixe à 900 mA la limite de courant que

doit être capable de fournir un port USB d'un PC, la norme USB BC1.2 est apparue.

Cette norme impose de court-circuiter les 2 lignes « D+ » et « D- ». D'autres normes fixent une tension de 1,2 V sur chacune des deux lignes « D+ » et « D- ».

Dans ces conditions, il n'est pas facile de recharger les batteries au lithium d'un quelconque smartphone, sans la disponibilité des circuits intégrés spécialement conçus pour répondre à toutes ces spécifications de manière automatique et donc transparente pour l'utilisateur.

Cependant, il existe un circuit intégré développé par Texas Instruments, le **TPS2511**, qui permet de résoudre ce problème.

Il s'agit d'un **contrôleur de port de charge dédié USB** avec une limitation en courant de l'alimentation.

Une fonction de détection automatique surveille la tension présente sur les lignes de données du port USB et fournit automatiquement les signatures électriques correctes sur ces lignes afin de recharger correctement les périphériques compatibles avec les systèmes de charge suivants :

- **Divider DCP** : applique 2,7 V sur « D+ » et 2,0 V sur « D- » ou 2,0 V sur « D+ » et 2,7 V sur « D- » ;
- **BC1.2 DCP** : lignes « D+ » et « D- » court-circuitées ou 1,2 V sur chacune des 2 lignes ;

Ce circuit intégré dispose en outre d'une protection de la tension de sortie (OUT) si elle devient inférieure à 3,8 V, d'une protection contre la surchauffe et en cas de surcharge.

Une limitation précise du courant de sortie offre une flexibilité pour différentes

applications. Il est compatible avec les normes USB2.0 et USB3.0.

Notre montage

Tout comme les périphériques que nous voulons recharger ou alimenter, notre circuit utilise l'énergie fournie par une pile au lithium qui, lorsqu'elle est chargée au maximum, délivre à ses bornes une tension de 4,2 V.

Cette tension est insuffisante pour alimenter les périphériques qui nécessitent une tension de 5 VDC (norme USB).

Pour cette raison, nous avons inséré dans notre montage un étage d'alimentation à commutation en configuration « **step-up** ». La tension de sortie est supérieure à celle d'entrée.

Étant donné que **l'énergie ne peut être que transformée**, l'efficacité d'un

Listing 1

```
#include <xc.h>
#pragma config FOSC = INTOSCIO

#pragma config WDTE = OFF
#pragma config PWRTE = ON
#pragma config MCLRE = OFF
#pragma config CP = OFF

#pragma config CPD = OFF

#pragma config BOREN = ON
#pragma config IESO = OFF

#pragma config FCMEN = OFF

#define LED GPIObits.GP2
#define USB GPIObits.GP5
#define MINIUSB GPIObits.GP1
#define STAT GPIObits.GP0

#define _XTAL_FREQ 32000

void main (void)
{
    OSCCON=0b00000011;           // 32 KHz
    ANSEL=0;                     // toutes les broches I/O en digital

    CMCON0bits.CM2=1;
    CMCON0bits.CM1=1;
    CMCON0bits.CM0=1;           // comparateur désactivé
    TRISIO=0b0000001;          // Toutes les sorties GPIO excepté GP0, sont configurées en entrée

    // Bits de sélection de l'oscillateur (INTOSCIO)
    // oscillateur : fonction E/S activée RA4/OSC2/
    // CLKOUT pin, I/O fonction activée RA5/OSC1/CLKIN)
    // Watchdog Timer Enable bit (WDT désactivé)
    // Power-up Timer Enable bit (PWRT activé)
    // MCLR broche intérieurement mise à VCC
    // Code Protection bit (Protection de lecture
    // du code désactivée)
    // Data Code Protection bit (Protection du code
    // de la mémoire de données désactivé)
    // Brown Out Detect (BOR activé)
    // Internal External Switchover bit
    // (Internal External Switchover mode est désactivé)
    // Fail-Safe Clock Monitor Enabled bit
    // (Fail-Safe Clock Monitor est désactivée)

    // PIN 5 LED
    // PIN 2 SHDN + TPS2069 (USB)
    // PIN 6 ICSPCLK ENABLE TPS 2051 (connecte la batterie au MCP)
    // PIN 7 STAT vérifie si la batterie est chargée
```

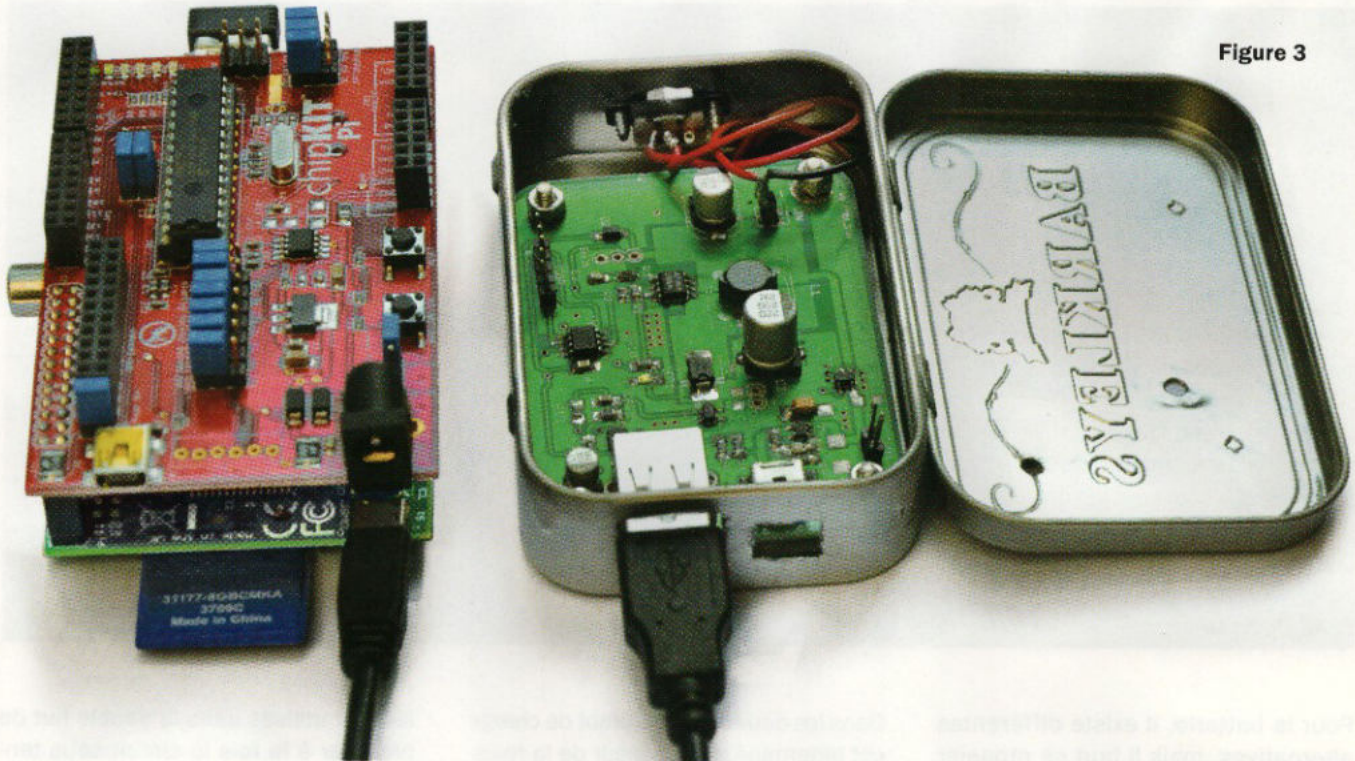



Figure 3

convertisseur DC/DC, même si elle est élevée, reste toujours **inférieure à 100 %**.

Cela signifie que le courant absorbé à l'entrée est toujours supérieur à celui disponible en sortie.

Pour cet étage, nous avons choisi un circuit intégré **LT1308A** de Linear Technology qui nous permet d'obtenir de manière fiable une tension de sortie de 5 VDC, avec un courant de sortie supérieur à 1 A.

Le fait d'avoir une tension de 5 VDC avec une polarité correcte n'est pas

suffisant, même si vous l'appliquez à un périphérique mobile via un connecteur micro-USB, la charge ne s'effectuera pas.

Les chargeurs génériques et adaptateurs, qui se branchent sur l'allume-cigare et équipés de prises USB, semblent fonctionner uniquement lorsqu'un smartphone moderne y est connecté.

Comme évoqué précédemment, cela est dû aux différentes « signatures » des tensions présentes sur les lignes de données « D+ » et « D- » du port USB.

Comme notre montage est alimenté par une batterie, normalement nous devons l'extraire pour la charger, puis la réinsérer pour l'utilisation, mais cela n'est pas pratique.

Nous avons prévu, pour notre montage, un étage basé sur un circuit intégré qui gère l'ensemble du cycle de charge.

La batterie utilisée pour nos tests est un modèle lithium-polymère, qui dispose d'une capacité de 2000 mAh. Cela permet d'alimenter des périphériques mobiles pendant des heures, et pour certains les recharger complètement.

```
IOCbits.IOCO=1;
INTCONbits.GPIF=0;
INTCONbits.GPIE=1;
INTCONbits.GIE=1;
```

```
LED=1;
USB=1;
MINIUSB=1;
__delay_ms(1500);
```

```
if (STAT==0){
__delay_ms(10);
MINIUSB=1;
__delay_ms(10);
LED=1;
USB=1;
__delay_ms(10);
```

```
}
else {
USB=0;
__delay_ms(10);
LED=0;
__delay_ms(10);
MINIUSB=0;
__delay_ms(10);
```

```
while(1){
__delay_ms(100);
asm("sleep");
}
}
```

```
void interrupt pinchange (){
if (INTCONbits.GPIF==1){
```

```
INTCONbits.GPIF=0;
if (STAT==0){
MINIUSB=1;
__delay_ms(10);
LED=1;
__delay_ms(10);
USB=1;
}
else {
USB=0;
__delay_ms(10);
LED=0;
__delay_ms(10);
MINIUSB=0;
}
}
```

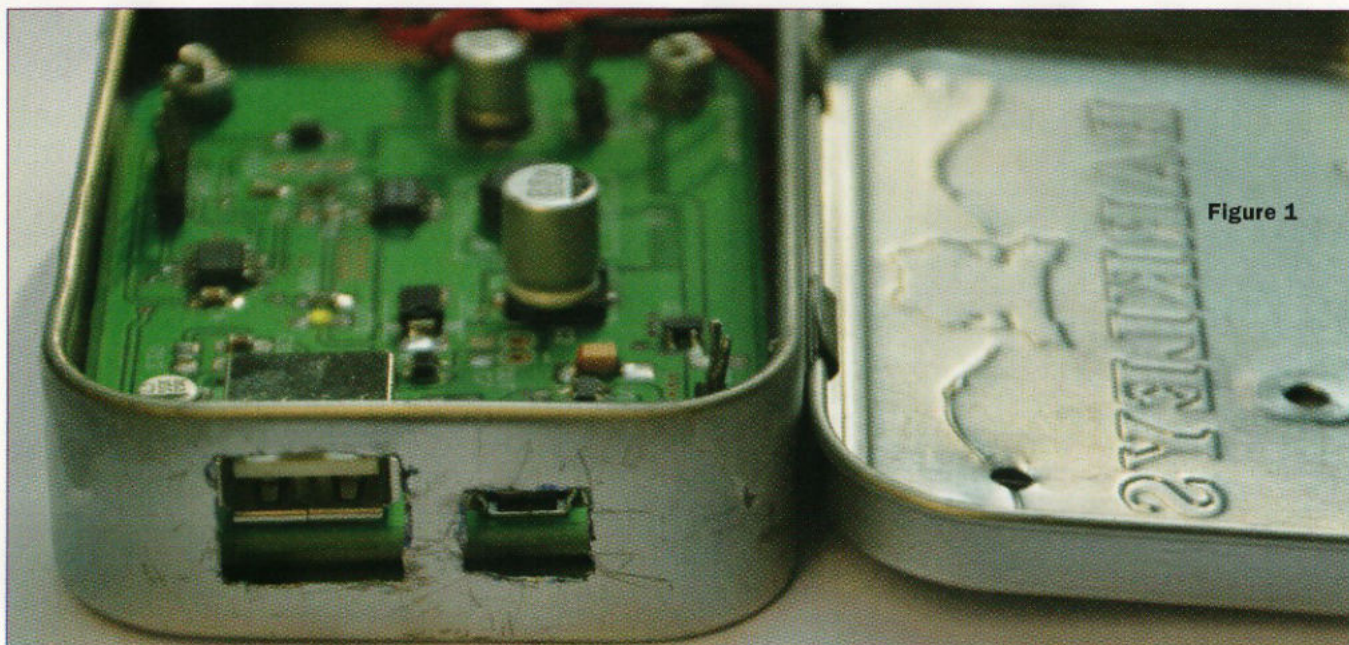



Figure 1

Pour la batterie, il existe différentes alternatives, mais il faut se rappeler que la capacité d'une batterie est caractérisée par le type d'élément chimique qui la constitue. Il faut aussi tenir compte du poids, du temps de recharge et du coût.

Vous pouvez également utiliser d'autres types de batteries basées sur des chimies différentes, par exemple, un accumulateur de type NiMH.

Cependant il nécessite un circuit spécial et vous devez tenir compte de la différence de la tension fournie (1,2 V pour le NiMH contre 4,2 V pour le lithium).

Comme nous nous sommes orientés vers une batterie de type **lithium-ion**, le choix du circuit intégré gérant la charge de la batterie s'est effectué entre deux composants : le **MCP73831** de chez Microchip et le **LTC4054ES5-4.2** de chez Linear Technology.

Ces deux composants ont le même brochage, de sorte que vous pouvez monter l'un ou l'autre en fonction de vos besoins.

Rappelez-vous, cependant, que le MCP73831 est capable de fournir un courant de charge de 500 mA au maximum, alors que le LTC4054ES5-4.2 peut supporter des courants de charge allant jusqu'à 800 mA.

Dans les deux cas, le courant de charge est déterminé par la valeur de la résistance qui est reliée à la broche 5 (R9 sur le schéma électrique).

En général, les câbles utilisés pour relier le smartphone à un ordinateur ou à un chargeur nomade sont de type « USB A » d'un côté et « micro-USB » de l'autre. Par contre, la connexion des chargeurs secteur dotés d'une sortie USB s'effectue au moyen de câbles « USB A/mini-USB ».

Pour éviter d'avoir à chercher des câbles USB de différents formats, nous avons prévu d'installer un connecteur USB femelle de type A sur le circuit imprimé pour y relier le périphérique portable à recharger et un connecteur mini-USB pour recharger la batterie au lithium qui alimente l'ensemble.

Bien évidemment, si vous avez des besoins différents, vous pouvez opter pour des choix différents.

Eh bien, maintenant, nous avons tous les schémas blocs dont nous avons besoin : l'étage « step-up », l'étage de gestion des niveaux des tensions sur les lignes de données USB, et l'étage de charge.

Afin d'améliorer la gestion et l'interconnexion des différents systèmes, nous avons inséré deux commutateurs électroniques capables de limiter le courant maximum disponible à leur sortie.

Ils sont utilisés dans le double but de protéger à la fois le circuit sous tension (l'alimentation externe pendant la phase de recharge) et le périphérique connecté à la batterie au lithium (le TPS2511 en fonctionnement normal).

Dans ce cas, les solutions possibles sont multiples.

Nous avons choisi d'utiliser un petit **microcontrôleur** à 8 broches (un **PIC12F683**) pour une gestion plus souple. Le code source du PIC est présentée dans le listing 1.

L'utilisation d'un microcontrôleur complique un peu la réalisation globale car il est nécessaire d'écrire le code, mais permet de modifier certaines caractéristiques finales sans toucher à l'électronique.

Pour ceux d'entre vous qui êtes habitués à Arduino, rien ne vous empêche d'utiliser un Atmega 328P avec le boot-loader Arduino.

Cependant, nous vous déconseillons cette approche, car elle est beaucoup plus coûteuse et encombrante que la solution basée sur le PIC12F683.

Il s'agit d'un microcontrôleur économique à 8 broches, facile à programmer en C avec le compilateur officiel XC8 libre, mais aussi en BASIC avec le PICBASIC PRO.

Il est évident que vous devez lire attentivement la documentation technique du microcontrôleur afin de configurer correctement les registres et les périphériques pour une utilisation correcte.

Le PIC12F683 contient un oscillateur interne RC, configurable de 32 kHz à 8 MHz et qui peut être utilisé pour générer la fréquence d'horloge sans devoir recourir à un quartz externe.

Dans ce cas, la précision de la fréquence est de l'ordre de 1 %, ce qui est plus que suffisant pour notre application.

La consommation du PIC est liée à la tension d'alimentation, à la fréquence d'horloge et d'autres paramètres, y compris les périphériques utilisés.

Nous avons choisi comme fréquence d'horloge la plus basse possible soit **32 kHz**, ce qui convient parfaitement pour notre usage.

Cette fréquence est sélectionnée à travers le registre « **Special Function Register (SFR)** » **OSCCON**.

Le reste du code est relativement simple. Ce n'est qu'une mise en œuvre possible parmi d'autres, notamment en BASIC.

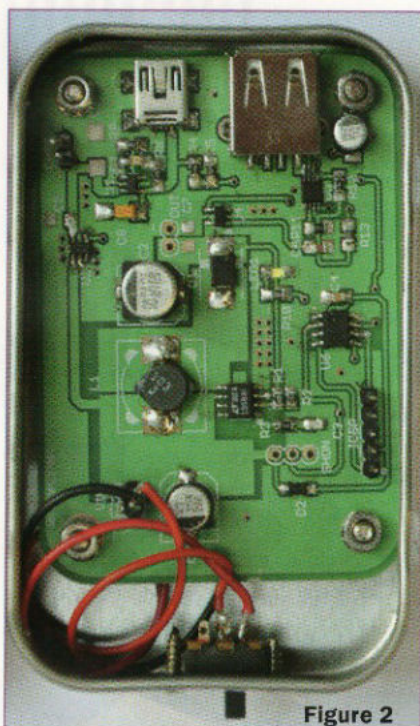


Figure 2

Pour réduire au maximum la consommation d'énergie, il a été décidé de mettre en veille (sleep) le PIC après avoir vérifié l'état de l'entrée GPO (broche 7). Elle est à un niveau haut lorsque la tension de 5 V est appliquée à l'entrée du circuit de recharge.

Le diviseur de tension formé par les résistances R4 et R5 sert à réduire la tension à 2,5 V, de manière à éviter d'endommager le PIC.

En effet la tension présente sur une entrée ne doit pas dépasser la tension d'alimentation du microcontrôleur, qui provient de la batterie au lithium (4,2 V).

Pour « réveiller » le PIC, nous faisons appel à une **interruption sur la broche 7** (GPO), activée à l'aide du bit « **IOCO** » du registre **IOC** (IOCbits.IOCO=1;) et du registre de contrôle d'interruption (**INTCON**).

En particulier, par l'intermédiaire du GPIO « Change Interrupt Enable bit (GPIE) », et des bits d'interruption « Global Interrupt Enable bit (GIE) » ainsi que GPIO « Change Interrupt Flag bit (GPIF) » (INTCONbits.GPIF=0; INTCONbits.GPIE=1; INTCONbits.GIE=1;).

En ce qui concerne le schéma électrique des circuits intégrés LT1308A, TPS2511 et MCP73831 (ou LTC4054) et les composants annexes, nous avons suivi les instructions indiquées par les constructeurs dans les documentations techniques respectives.

Le TPS2065C garantit en sortie (broche 1) un courant continu de 1 A. Il est **actif** lorsque sa broche de validation « **EN** » (broche 4) est à un **niveau haut**.

Le TPS2061 est identique au TPS2065C, sauf qu'il est **actif** lorsque sa broche « **EN** » est à un **niveau bas**.

Cependant, une différence importante existe entre les deux commutateurs (les 2 circuits intégrés).

Il s'agit de la plage de la tension d'entrée. Elle est de **4,5 V à 5,5 V** pour le **TPS2065C** et de **2,7 V à 5,5 V** pour le **TPS2051**, ce qui rend le premier circuit adapté au fonctionnement élévateur en sortie (step-up) tandis que

le second est idéal pour la recharge de la batterie.

La programmation du PIC s'effectue par l'intermédiaire du connecteur ICSP, dont les connexions standards permettent l'utilisation des programmeurs PICKIT ou ICD3 de Microchip.

Compte tenu du caractère expérimental du projet, pour le boîtier nous avons choisi d'en récupérer un plutôt que d'acheter un produit commercial.

Le résultat obtenu est satisfaisant. Il s'agit d'un boîtier métallique qui contenait à l'origine des bonbons, nous l'avons adapté pour qu'il intègre l'ensemble de l'électronique de notre chargeur intelligent.

Nous avons percé le boîtier afin de permettre l'accès de l'extérieur aux connecteurs USB et mini-USB (voir la figure 1).

L'interrupteur d'allumage se situe à l'extérieur ainsi que les deux LED (la verte pour le fonctionnement normal et la rouge pour la recharge de la batterie (voir la figure 2).

Vous pouvez utiliser un boîtier en plastique du commerce, il suffit de choisir correctement les dimensions pour que l'ensemble de l'électronique prenne place.

Le système a été testé en donnant d'excellents résultats, même avec un RaspberryPi.

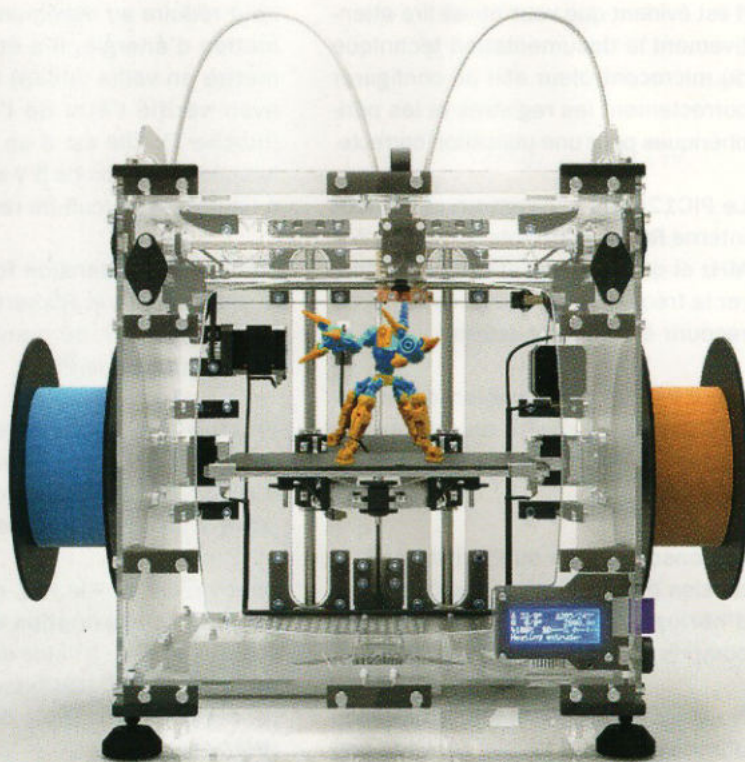
En figure 3, vous pouvez voir notre chargeur connecté à un « chipKIT Pi », afin de rendre le système portable.

Il est évident que, dans ce type d'utilisation, le TPS2511 n'est pas nécessaire, bien que sa présence ne pose pas de problème pour le fonctionnement du RaspberryPi.

Il existe dans le commerce des dispositifs similaires pour répondre aux exigences que nous avons rencontrées.

Nous espérons vous avoir fourni suffisamment d'explications avec cet article afin que vous réalisiez votre propre version du « smart charger ».

Dans le précédent numéro 139 d'Electronique et Loisirs Magazine, nous avons abordé l'utilisation de l'imprimante 3DVERTEX en analysant les différents menus de son firmware. Nous avons notamment, appris à sélectionner le nombre d'extrudeuses, à reconnaître les différentes indications de l'afficheur LCD et enfin à charger et décharger le ou les filament(s). Nous allons maintenant étudier le logiciel de tranchage (Slicer) et d'interpréteur G-code qui se nomme Repetier-Host.



3DVERTEX

L'IMPRIMANTE

3D BICOLORE

6^{ème} partie

..... de Boris LANDONI

Dans ce chapitre, nous expliquerons les principes de base de l'impression 3D, sans entrer dans les détails. Il est recommandé de lire attentivement l'article en entier pour comprendre le processus et obtenir de bons résultats d'impression.

Introduction à l'impression 3D

Pour imprimer un modèle 3D, il est nécessaire d'avoir les éléments suivants :

- un modèle 3D ;
- un logiciel de tranchage (Slicer) et un interpréteur de G-code ;
- une imprimante 3DVERTEX.

Le modèle 3D

Il existe plusieurs façons d'obtenir un modèle 3D, vous pouvez en créer un avec les logiciels suivants qui sont tous gratuits :

- Sketchup : www.sketchup.com ;
- Blender : www.blender.org ;
- OpenSCAD : www.openscad.org.

Vous pouvez aussi télécharger des fichiers d'objets déjà confectionnés sur les sites suivants, dans certains cas les modèles ne sont pas gratuits, il faudra s'acquitter d'un copyright :

- Thingiverse : www.thingiverse.com ;
- MyMiniFactory : www.myminifactory.com ;
- Youmagine : www.youmagine.com.

Le logiciel de tranchage Repetier-Host

Le logiciel que vous pouvez utiliser pour l'imprimante 3DVERTEX est gratuit et s'appelle Repetier-Host. Vous trouverez en téléchargement la version utilisée dans cet article.

Repetier-Host comprend plusieurs logiciels de découpe (Curaengine, Slic3r, Skeinforge) et fonctionne aussi comme un interpréteur de G-code.

Pour imprimer un modèle 3D, il faut « trancher » le modèle, c'est-à-dire interpréter le modèle 3D en parcours ou chemins (tool paths) que la 3DVERTEX peut comprendre. Ces parcours ou chemins sont décrits en G-code.

Le logiciel découpe progressivement le modèle 3D en tranches fines (slices en anglais). Ainsi le logiciel de tranchage traduit le modèle 3D vers un format que l'imprimante 3D peut interpréter.

Lors de ce processus, le logiciel calcule toutes les variables (vitesse, débit, épaisseur de couche, remplissage...), ce qui signifie que les paramètres sont uniques pour chaque type d'imprimante.

Il est donc déconseillé d'utiliser des fichiers G-code prévus pour un autre type d'imprimante que la 3DVERTEX.

Puisque le processus de découpage représente un équilibre délicat entre un grand nombre de variables, nous avons inclus tous ces paramètres dans la version Repetier-Host pour l'imprimante 3DVERTEX. Les réglages de l'ABS, du PLA, et des couleurs sont tous pré-chargés.

La partie interpréteur de G-code du logiciel Repetier-Host permet l'utilisation de fichiers « gcode » pour l'impression d'objets, mais aussi pour le contrôle manuel de l'imprimante.

Le G-code est en fait une séquence de commandes manuelles écrites dans un langage que l'imprimante comprend.

Une fois qu'un objet a été correctement tranché (converti en G-code), ce code peut être chargé et transmis à l'imprimante ligne par ligne. Ce code indiquera à l'imprimante ce qu'elle doit faire pour imprimer l'objet.

La 3DVERTEX est équipée d'un lecteur de carte SD de sorte que vous pouvez enregistrer votre fichier « gcode » sur une carte SD et l'imprimer de manière autonome (sans ordinateur).

Connexion à l'imprimante

Pour utiliser l'imprimante avec un ordinateur et la commander à partir du programme Repetier-Host, vous devez connecter votre imprimante à votre ordinateur via un câble USB de type A-mâle vers B-mâle.

Ensuite vous devez installer le pilote (driver) et le logiciel Repetier-Host. Vous devez vous assurer que le pilote USB pour l'imprimante est installé, avant de la brancher.

Vous trouverez le pilote Virtual COM port (VCP) pour Windows 32/64 bits version 2.12.26, pour Linux 32/64 bits version 1.5.0 et pour MAC OS10.9 et supérieur version 2.4.2 sur notre site dans le sommaire détaillé de la revue.

Une fois le driver installé, mettez la 3DVERTEX sous tension et branchez-la à votre ordinateur via le câble USB. Ouvrez le gestionnaire de périphérique.

Pour cela cliquez sur le bouton « Démarrer » puis tapez la commande « mmc devmgmt.msc ».

La fenêtre du gestionnaire de périphérique doit apparaître. Ouvrez la liste des « Ports (COM et LPT) » et cherchez le port USB série (COMxxx).

Le nombre à côté de l'inscription « COM » correspond

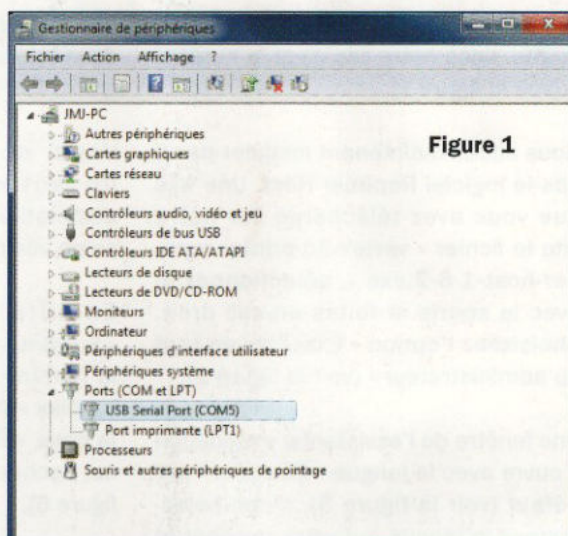


Figure 1

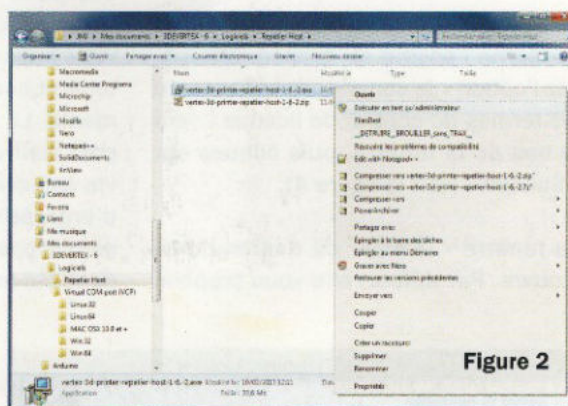


Figure 2

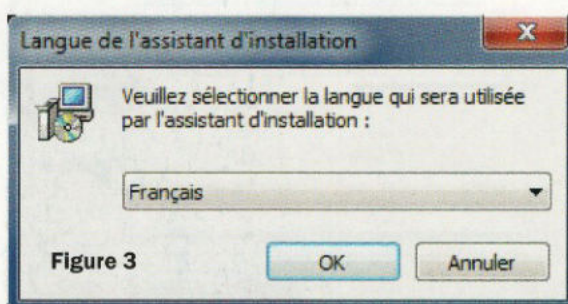


Figure 3

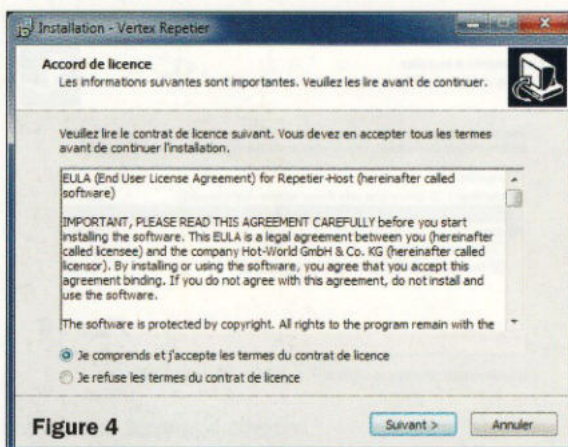


Figure 4

au numéro du port COM de votre imprimante, dans notre cas c'est le numéro 5 (voir la figure 1).

Nous allons maintenant installer pas à pas le logiciel Repetier-Host. Une fois que vous avez téléchargé sur notre site le fichier « vertex-3d-printer-repetier-host-1-6-2.exe », sélectionnez-le avec la souris et faites un clic droit. Choisissez l'option « Exécuter en tant qu'administrateur » (voir la figure 2).

Une fenêtre de l'assistant d'installation s'ouvre avec la langue « Français » par défaut (voir la figure 3), sinon sélectionnez la langue qui vous convient et cliquez sur « OK ».

Une nouvelle fenêtre s'ouvre alors, elle concerne l'accord de licence. Choisissez l'option « Je comprends et j'accepte les termes du contrat de licence » vers le bas de la fenêtre, puis cliquez sur « Suivant » (voir la figure 4).

La fenêtre « Dossier de destination » s'ouvre. Par défaut, elle vous propose

d'installer le programme dans le répertoire « C:\Program Files\VertexRepetier ».

Vous pouvez choisir un autre dossier de destination ou en créer un selon vos besoins.

Ensuite cliquez sur « Suivant » (voir la figure 5), la fenêtre « Composants à installer » s'ouvre, par défaut la case « Repetier-server » est cochée (ligne bleue en figure 6).

Décochez cette ligne et cliquez sur « Suivant », car vous travaillez en local, votre ordinateur étant directement connecté à l'imprimante. Le serveur est utile pour contrôler l'imprimante via Internet ou un réseau d'entreprise. Nous n'utiliserons pas cette possibilité dans cet article.

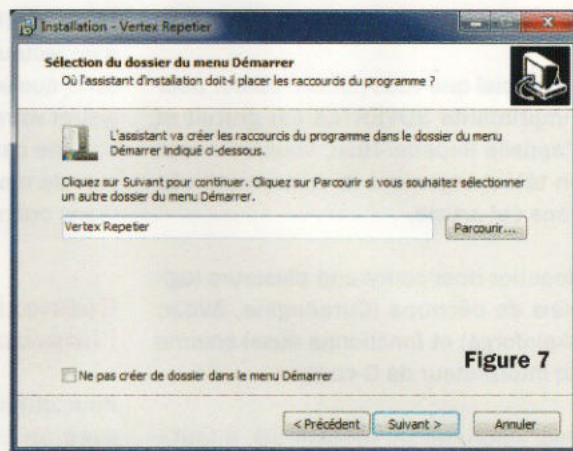


Figure 7

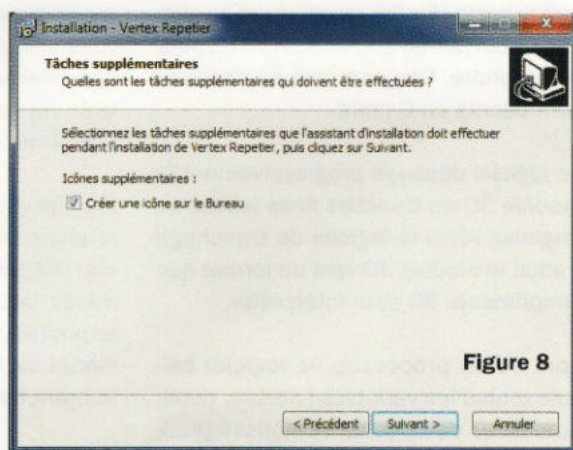


Figure 8

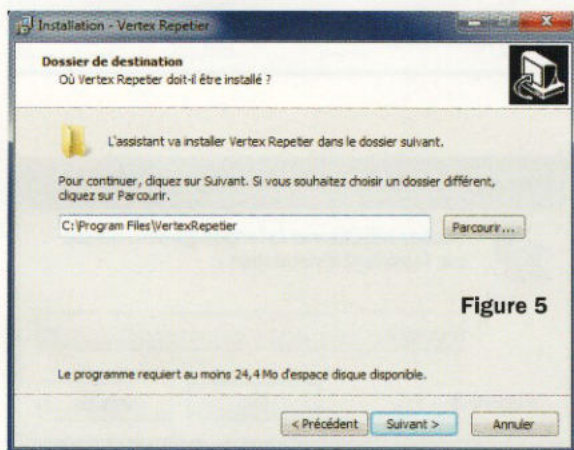


Figure 5

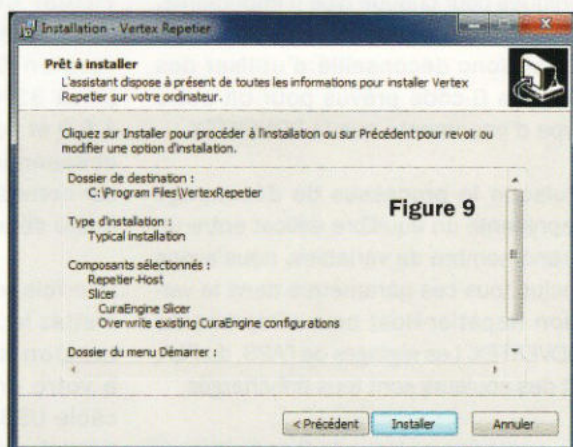


Figure 9

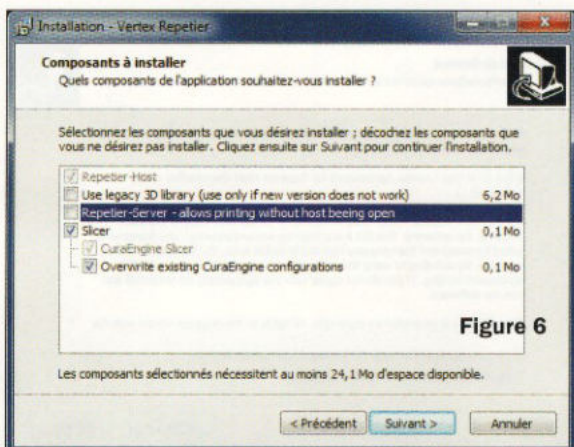


Figure 6

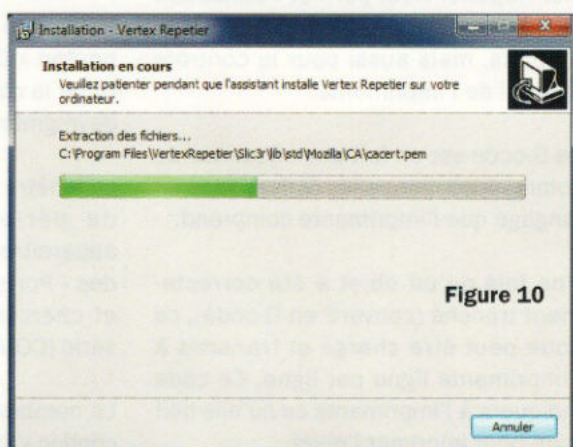


Figure 10

Une nouvelle fenêtre s'ouvre en vous proposant de créer le dossier « Vertex Repetier » dans le menu « Démarrer », vous pouvez modifier le nom ou sélectionner un autre dossier à l'aide du bouton « Parcourir », ensuite cliquez sur « Suivant » (voir la figure 7).

Une nouvelle fenêtre s'ouvre en vous proposant de créer un raccourci sur le « Bureau », selon votre convenance cochez ou décochez la case, puis cliquez sur « Suivant » (voir la figure 8).

Enfin une dernière fenêtre apparaît en affichant un récapitulatif des options d'installation, cliquez sur le bouton « Installer » (voir la figure 9).

Le processus d'installation commence, vous pouvez voir l'état d'avancement (voir la figure 10). À la fin du processus d'installation, une fenêtre apparaît et

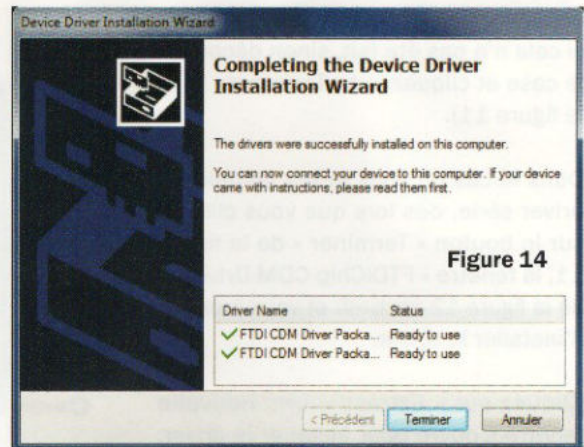


Figure 14



Figure 11

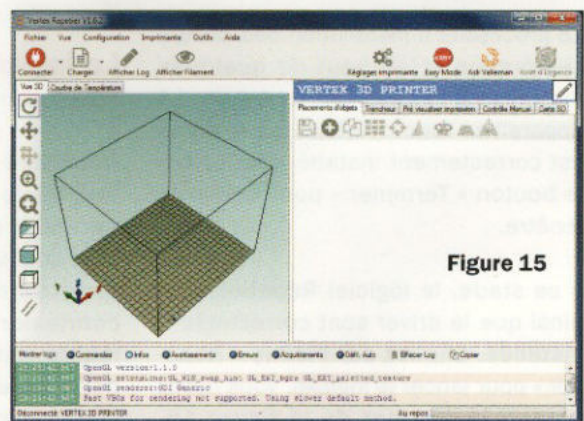


Figure 15

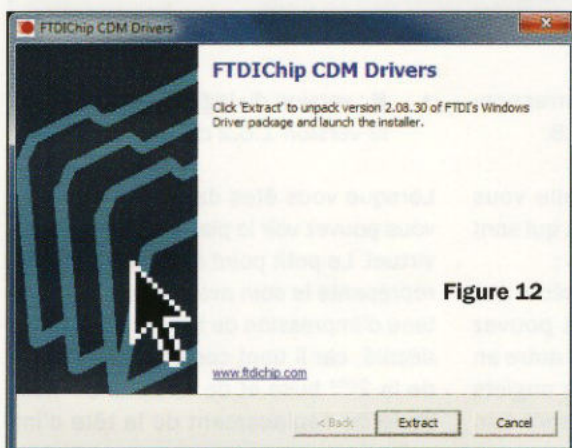


Figure 12

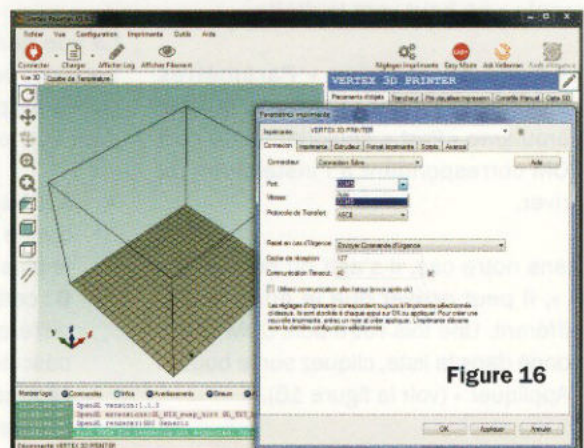


Figure 16



Figure 13

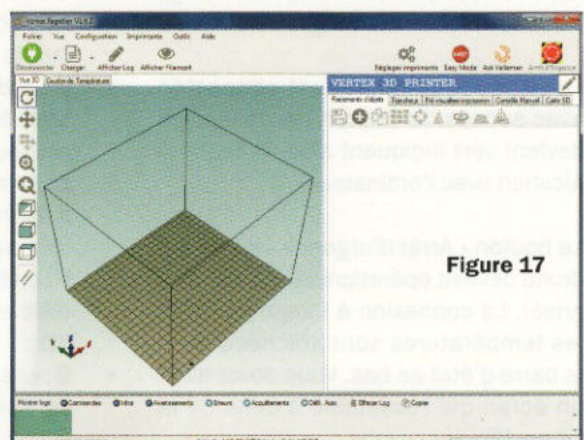


Figure 17

vous propose d'installer le driver série si cela n'a pas été fait, sinon décochez la case et cliquez sur « Terminer » (voir la figure 11).

Dans le cas où vous devez installer le driver série, dès lors que vous cliquez sur le bouton « Terminer » de la figure 11, la fenêtre « FTDIChip CDM Drivers » de la figure 12 apparaît et vous propose d'installer le driver.

Cliquez sur « Extract », une nouvelle fenêtre s'ouvre pour installer le driver, cliquez sur « Suivant » (voir la figure 13).

Le processus d'installation des drivers commence et au bout de quelques instants la fenêtre de la figure 14 apparaît en vous indiquant que le driver est correctement installé, cliquez sur le bouton « Terminer » pour fermer la fenêtre.

À ce stade, le logiciel Repetier-Host ainsi que le driver sont correctement installés. Cliquez sur l'icône « Repetier » pour lancer le logiciel. Vous devez obtenir le résultat de la figure 15. Appuyez sur le bouton « Réglages imprimante » en haut vers la droite.

La boîte de dialogue « Paramètres imprimante » apparaît. Dans la liste déroulante « Port », sélectionnez le port COM correspondant à l'installation du driver.

Dans notre cas, il s'agit du port « COM 5 », il peut arriver que le numéro soit différent. Une fois votre port COM sélectionné dans la liste, cliquez sur le bouton « Appliquer » (voir la figure 16).

Ensuite pour vous connecter à l'imprimante, appuyez sur le bouton rouge « Connecter » en haut à gauche.

Lorsque l'imprimante est connectée avec succès, le bouton de connexion devient vert indiquant ainsi la communication avec l'ordinateur.

Le bouton « Arrêt d'urgence » en haut à droite devient opérationnel (il n'est plus grisé). La connexion à l'imprimante et les températures sont affichées dans la barre d'état en bas. Vous devez avoir un écran qui ressemble à celui de la figure 17.

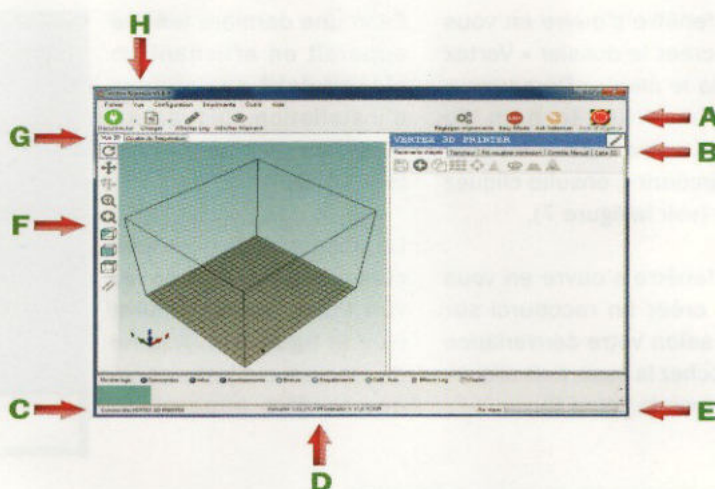


Figure 18

Le logiciel Repetier-Host

Nous allons maintenant étudier le logiciel Repetier-Host dans le détail. Lorsque vous lancez le programme et que la connexion avec l'imprimante est établie, vous devez obtenir une fenêtre semblable à celle de la figure 18.

La vue générale de Repetier-Host peut être décomposée en plusieurs parties correspondant aux lettres sur la figure 18.

- **A** : barre des boutons, elle vous donne accès à des actions qui sont le plus souvent effectuées ;
- **B** : cette zone permet d'accéder aux différents modules, vous pouvez basculer d'un module à un autre en cliquant sur les différents onglets (Placements d'objet, Trancheur, etc.) ;
- **C** : barre d'état, elle vous indique l'état de la connexion avec l'imprimante ;
- **D** : indication en temps réel de la température de(s) la(es) buse(s) active(s) ;
- **E** : barre de progression du temps d'impression, c'est-à-dire le travail d'impression en cours d'exécution ;
- **F** : cette zone permet d'accéder aux différents boutons pour la vue en 3D ;
- **G** : ces 2 onglets permettant d'accéder soit aux différentes vues en 3D, soit aux courbes de température ;

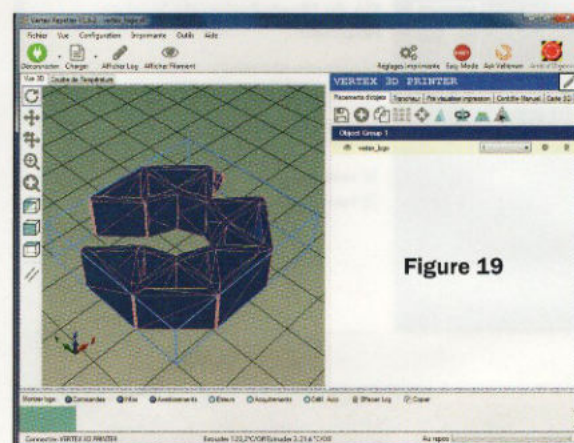


Figure 19

- **H** : version du logiciel, ici il s'agit de la version 1.6.2 de Repetier-Host.

Lorsque vous êtes dans la vue en 3D, vous pouvez voir le plateau d'impression virtuel. Le petit point noir sur le plateau représente le coin avant gauche du plateau d'impression de l'imprimante. Il est décalé, car il tient compte du décalage de la 2^{ème} buse et de la distance maximale de déplacement de la tête d'impression.

Placement de l'objet

Lorsque vous souhaitez imprimer un objet, vous devez charger l'objet dans le logiciel et le positionner sur le plateau virtuel.

Pour cela, vous devez cliquer sur le bouton « Charger » (en haut à gauche) qui se trouve dans la barre des boutons.

Une fenêtre s'ouvre, elle vous permet de sélectionner un fichier sur votre ordinateur, par exemple le fichier « vertex_logo.

stl » (que vous pouvez télécharger sur notre site).

Dans la fenêtre, sélectionnez le fichier et cliquez sur « Ouvrir ». Le logiciel effectue quelques contrôles pour s'assurer que l'objet est multiple, puis celui-ci est affiché sur le plateau virtuel dans la fenêtre « Vue 3D ». Vous pouvez faire un zoom si l'objet est petit. Vous devez obtenir un résultat semblable à celui de la figure 19.

Pour déconnecter l'imprimante, cliquez sur le bouton vert « Déconnecter » en haut à gauche.

Le logiciel Repetier-Host mémorise les réglages, ainsi pour vous connecter de nouveau il suffit simplement de cliquer sur le bouton rouge « Connecter ».

À l'aide de la souris vous pouvez effectuer un zoom avant/arrière et modifier la position de la vue :

- **clic gauche + déplacement de la souris** : modifie la position de la vue ;
- **roulette avant/arrière** : zoom avant/arrière ;
- **clic droit + déplacement de la souris** : déplacement de l'objet.

Pour recentrer l'objet, il suffit d'appuyer sur le bouton « Centrer l'objet » qui se trouve dans l'onglet « Placement objet » (voir la figure 20).

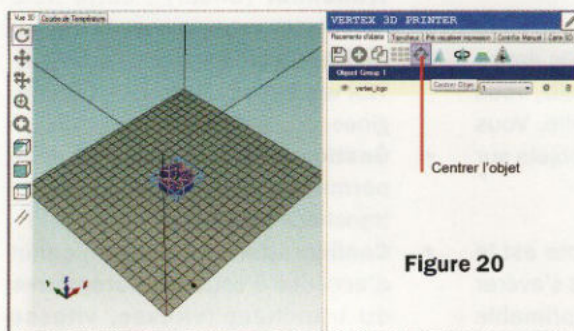


Figure 20

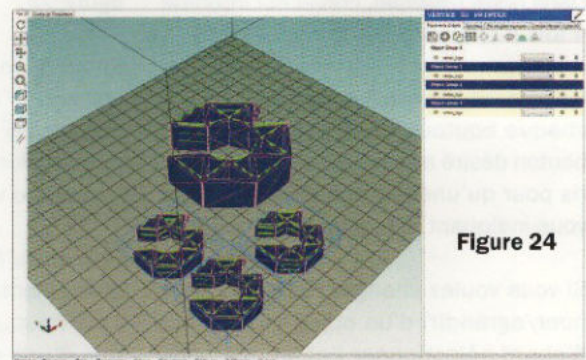


Figure 24

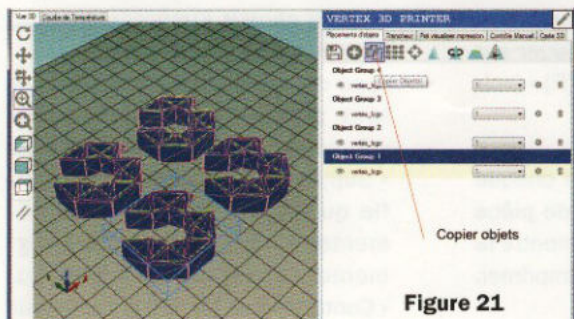


Figure 21

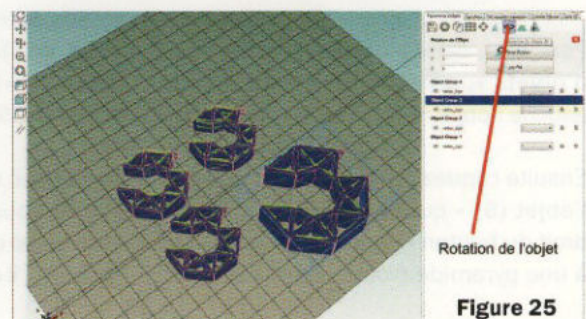


Figure 25

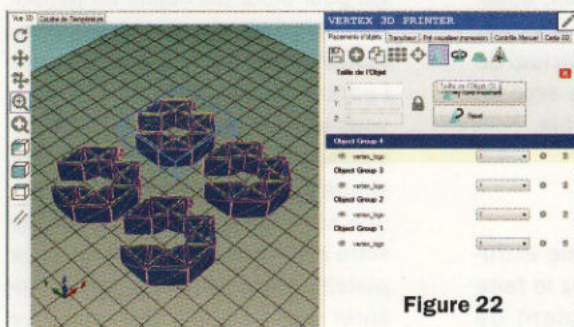


Figure 22

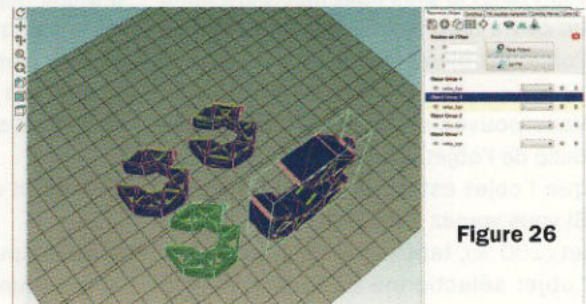


Figure 26

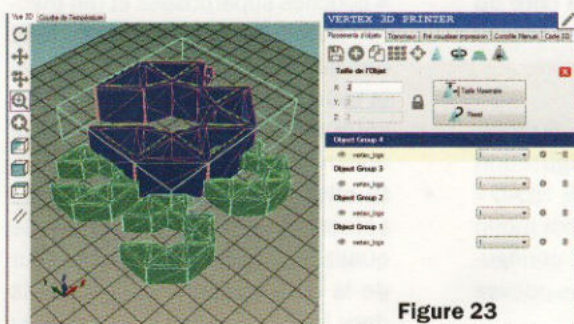


Figure 23

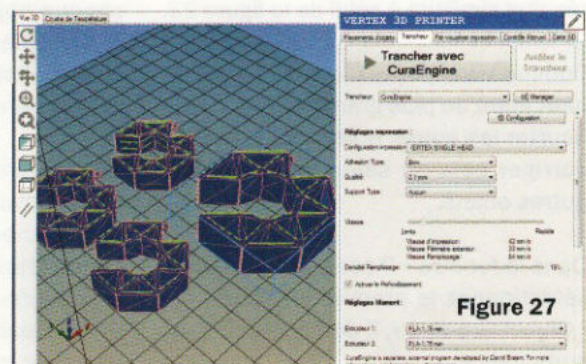


Figure 27

Si vous voulez imprimer plusieurs fois le même objet, vous pouvez appuyer sur le bouton « Copier Objet(s) » dans l'onglet « Placement objet ».

Une boîte de dialogue s'ouvre vous demandant le nombre de copies. Choisissez 3 fois, puis laissez la case « Positionnement automatique » cochée.

Cliquez sur le bouton « Copier », vous devez obtenir le résultat de la figure 21.

Dans notre exemple, l'objet est imprimé 4 fois (l'original + les 3 copies).

NB : pour connaître la fonction de chaque bouton, il suffit de survoler le bouton désiré avec le pointeur de la souris pour qu'une annotation apparaisse vous indiquant la fonction du bouton.

Si vous voulez changer l'échelle (diminuer/agrandir) d'un objet, vous devez d'abord sélectionner l'objet que vous souhaitez modifier.

Pour cela, dans la fenêtre « Placement d'objets », sélectionnez avec un clic gauche l'objet nommé « Objet group 4 ».

Ensuite cliquez sur le bouton « Taille de l'objet (S) » qui se trouve sur le côté droit du bouton « centrer », il ressemble à une pyramide (voir la figure 22).

Notez que dans la vue 3D, vous devez voir l'objet le plus haut incorporé dans un cadre bleu clair. Cela veut dire que l'objet est bien sélectionné.

Vous pouvez maintenant modifier la taille de l'objet. La case « X=1 » indique que l'objet est à l'échelle 1 (100 %). Si vous voulez doubler la taille de l'objet (200 %), tapez 2 dans la case « X », l'objet sélectionné doit apparaître 2 fois plus gros. Vous devez obtenir un résultat similaire à celui de la figure 23.

Comme vous pouvez le voir, l'objet du haut étant de plus grande taille, il vient heurter les autres objets. Vous pouvez corriger cela en sélectionnant les 3 autres objets.

Dans la fenêtre « Placement d'objets », sélectionnez le premier objet nommé « Object Group 1 », il doit apparaître en surbrillance.

Ensuite, appuyez et maintenez la touche « shift » enfoncée (représentée par une flèche à gauche du clavier au-dessus de « Ctrl ») et à l'aide de la souris sélectionnez les 2 autres objets (group 2 et 3).

Positionnez la souris dans la vue 3D sans qu'elle soit sur aucun objet, faites un clic droit + déplacement de la souris afin de déplacer les 3 autres objets vers une zone libre. Vous devez obtenir un résultat semblable à celui de la figure 24.

Ensuite, en faisant « CTRL + clic droit » sur le 4^{ème} objet dans la vue 3D, vous l'ajoutez à la sélection actuelle. Vous pouvez alors centrer tous les objets sur le plateau virtuel.

Une autre fonction intéressante est le pivotement d'objets. Cela peut s'avérer utile lorsqu'un objet n'est imprimable que d'une certaine manière en raison de sa forme. Il faut dans un premier temps sélectionner l'objet que vous souhaitez faire pivoter puis cliquer sur le bouton « Rotation de l'objet (R) » (voir la figure 25).

Entrez des valeurs d'angle pour chaque axe. Si vous tournez une grande pièce d'un angle de 90 °, comme le montre la figure 26, il est très difficile de l'imprimer.

Vous pouvez aussi saisir des valeurs d'angle négatives. Dans ce cas, assurez-vous que l'objet soit en contact avec le plateau d'impression virtuel de manière la plus plate possible avec un espace aussi grand que possible.

Sélection d'une extrudeuse

Si vous souhaitez attribuer une extrudeuse à un objet, vous pouvez le faire en utilisant le menu déroulant de chaque objet. Pour cela, allez dans le menu « Placement d'objet », à côté du nom de l'objet (vertex_logo) se trouve la liste déroulante.

Sélectionnez l'extrudeuse souhaitée. Lorsque vous utilisez la configuration du trancheur « VERTEX SINGLE HEAD » (mono tête), l'imprimante devient mono tête, alors que lorsqu'elle est configurée en double extrudeuse, vous pouvez imprimer en deux couleurs.

Lorsque l'imprimante est munie d'une seule tête, ne configurez pas le « trancheur » avec 2 têtes. Lorsque vous sélectionnez la seconde extrudeuse, la couleur de l'objet change.

Le trancheur

Pour aller dans le menu du trancheur, cliquez sur l'onglet « Trancheur » à côté de « Placement d'objet ». Vous devez régler tous les paramètres comme ci-après (voir la figure 27) :

- **Trancheur** (Slicer) : cette liste déroulante vous permet de choisir les trancheurs installés. La 3DVERTX utilise le trancheur « CuraEngine » ;
- **Gestionnaire** (Manager) : il vous permet de gérer les moteurs du trancheur installés ;
- **Configuration** : ce bouton permet d'accéder à tous les paramètres du trancheur (vitesse, vitesse d'avance, ...) ;
- **Configuration d'impression** : permet de choisir l'impression avec une ou deux extrudeuses ;
- **Type d'adhésion** : ce menu déroulant permet de choisir entre « Aucun », « Contour » (Brim) ou « Support » (Raft). « Aucun » signifie qu'aucune structure supplémentaire n'est ajoutée pour augmenter l'adhésion sur le plateau. « Contour » signifie que le trancheur ajoute une première couche élargie pour aider les objets de petites dimensions à mieux adhérer au plateau d'impression. Utilisez toujours l'option « Contour » (Brim) lorsque vous utilisez de l'ABS. « Support » signifie que le trancheur ajoute plusieurs couches sous l'objet. C'est une sorte de petite plaquette que votre imprimante va imprimer sur le plateau avant votre objet afin d'assurer une meilleure accroche à ce dernier. Il est souvent composé de 4 couches superposées et (en théorie) est censé se retirer facilement une fois l'impression terminée, mais la tâche peut s'avérer compliquée si votre objet est fin et fragile. Cette option n'est pas recommandée ;
- **Qualité** : cette option permet de sélectionner les préréglages de qualité (principalement la hauteur de la couche Z), ils sont stockés dans la configuration du trancheur ;

- **Type de support** : ce menu déroulant permet de choisir entre « Aucun », « Touching Bed » cette option permet de rajouter des structures de support uniquement en contact avec le plateau, et « Everywhere » qui permet de rajouter des structures de support même si la base est l'objet lui-même. Par défaut sélectionnez « Aucun » ;
- **Vitesse** : elle correspond à la vitesse générale du trancher, les réglages des valeurs minimales et maximales proviennent de la configuration du trancher ;
- **Densité de remplissage** : cette option permet de définir la densité intérieure de l'objet. La valeur 100 % signifie que l'objet est solide à l'intérieur et 0 % que l'objet est creux. L'objet sera imprimé comme un solide à partir de 60 %. Des

valeurs plus élevées permettront d'augmenter la quantité de plastique, cependant la précision d'impression diminue ;

- **Activer le refroidissement** : en cochant cette case les ventilateurs entrent en fonctionnement. Ce paramètre doit toujours être activé ;
- **Extrudeur 1** : sélection du matériau chargé dans la première extrudeuse (PLA ou ABS) ;
- **Extrudeur 2** : sélection du matériau chargé dans la deuxième extrudeuse (PLA ou ABS).

Les réglages du trancher

En appuyant sur le bouton « configuration » dans le trancher, vous accédez aux paramètres de configuration du trancher « CuraEngine ». Normalement les paramètres affichés sont chargés et ne doivent pas être modifiés (voir la figure 28).

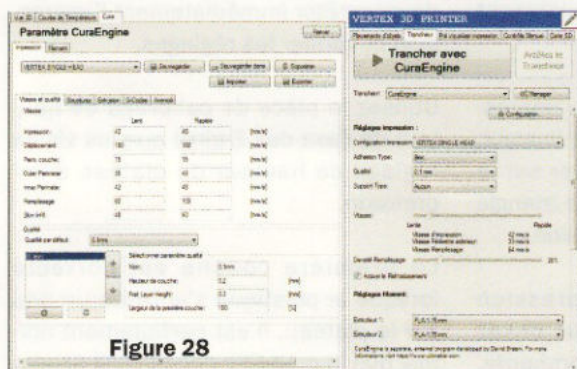


Figure 28

Vous pouvez modifier ou ajouter des profils, cependant l'imprimante 3DVERTEX ne peut supporter que les paramètres chargés par défaut.

Il est possible de faire fonctionner l'imprimante plus rapidement, ou d'utiliser des nouveaux matériaux en modifiant les profils.

Faites attention car vous pouvez causer des dommages à l'imprimante. Procédez avec prudence lorsque vous désirez tester de nouveaux paramètres !

Prévisualisation de l'impression

En cliquant sur le bouton « Trancher avec CuraEngine », le processus de calcul des trajectoires commence (voir la figure 29).

Une barre de progression est visible, selon l'objet à imprimer le processus peut prendre plusieurs minutes. À la fin du processus, le logiciel passe automatiquement

à l'onglet de prévisualisation de l'impression. Une fois le processus de tranchage terminé, vous devez obtenir un résultat similaire à celui de la figure 30.

Vous pouvez voir un résumé qui affiche les données suivantes :

- l'estimation du temps d'impression nécessaire pour imprimer l'objet ;
- le nombre de couches nécessaires pour construire l'objet ;
- le nombre total de lignes de G-CODE ;
- la quantité, en millimètre, de filament nécessaire pour imprimer l'objet.

Vous pouvez également visualiser les trajectoires en sélectionnant « Montrer une couche » ou « Montrer les couches » et en jouant sur les curseurs. Vous devez voir l'objet se modifier dans la vue 3D.

Vous pouvez inspecter le G-Code lui-même, en appuyant sur le bouton « Editer G-Code ». Cela affiche chaque ligne de code. Vous pouvez modifier ce G-Code, mais nous vous recommandons de ne pas le faire, cela est réservé aux utilisateurs expérimentés.

Si vous souhaitez imprimer ce fichier à partir d'une carte SD sans que l'imprimante soit connectée, vous devez enregistrer le G-CODE généré sur une carte SD. Pour cela, appuyer sur le bouton « Sauver » qui est représenté par une disquette (voir la figure 31).

Dans la fenêtre G-Code, il s'agit de la deuxième icône en partant de la gauche. Une fenêtre s'ouvre vous invitant à sélectionner un dossier, choisissez un dossier se trouvant sur la carte SD. Vous pouvez aussi en créer un nouveau.

Parfois, la sauvegarde de grands objets sur une carte SD peut prendre un certain temps et le logiciel peut sembler ne pas répondre.

Ne retirez pas la carte SD, attendez que le programme devienne réactif. Ainsi l'objet sera complètement sauvegardé sur la carte SD.

Pour imprimer l'objet, appuyez sur le bouton en forme de triangle « Démarrer impression » dans la barre des boutons.

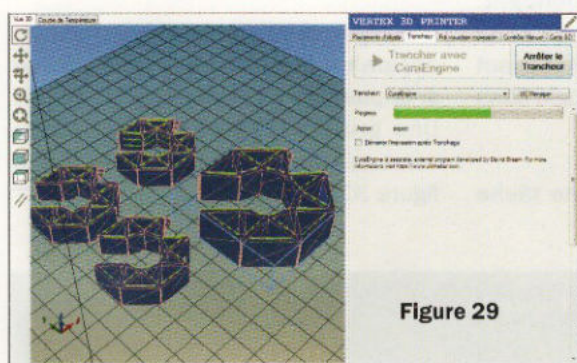


Figure 29

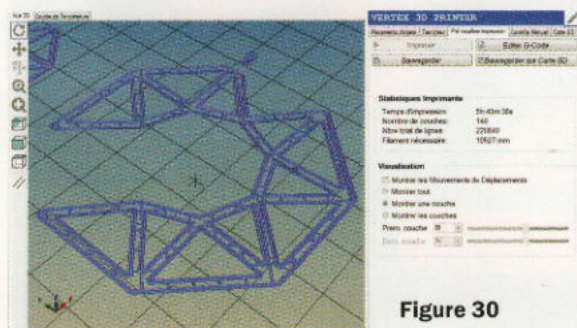


Figure 30

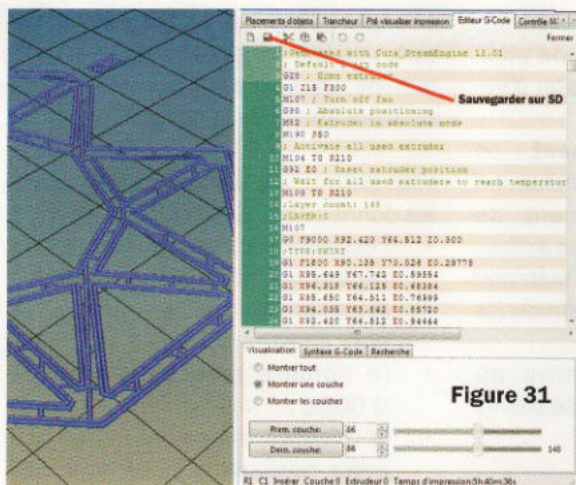


Figure 31

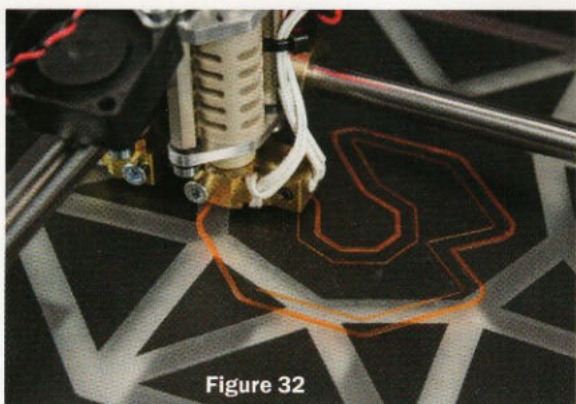


Figure 32

L'impression en 3D

Nous allons maintenant présenter pas à pas le processus d'une tâche d'impression. Nous allons aussi expliquer ce que l'imprimante fait au cours de chaque étape et si cela correspond à un comportement normal (ou anormal). Il est recommandé de lire entièrement ce chapitre avant d'essayer d'imprimer.

Démarrage d'une tâche d'impression

Vous devez vous assurer que le filament est correctement chargé dans l'extrudeuse que vous allez utiliser. Puisque c'est votre première impression, nous vous recommandons fortement d'imprimer d'abord avec une seule extrudeuse et en utilisant du PLA. Eventuellement, reportez-vous aux précédents articles décrits dans les numéros 135 à 139.

Assurez-vous que l'axe des Z est correctement calibré. Assurez-vous également que les buses sont propres et sans résidus de plastique visibles. La meilleure façon de le faire consiste à

préchauffer la buse et l'essuyer avec un morceau de sopalin.

Connectez l'imprimante à votre ordinateur, et établissez une connexion à l'aide du logiciel Repetier-Host. Vous pouvez aussi imprimer votre premier objet à partir d'une carte SD. Dans ce cas, vous n'avez pas besoin de vous connecter à un ordinateur.

Ensuite, vous devez générer le fichier G-CODE. Assurez-vous que dans l'onglet « Trancheur », la configuration d'impression « Vertex single head » est sélectionnée dans la liste déroulante. L'objet sur l'écran est prêt maintenant à être imprimé.

Si vous démarrez l'impression à partir d'un ordinateur, vous devez appuyer sur le bouton en forme de triangle « Démarrer impression ».

Si vous voulez lancer l'impression à partir d'une carte SD, vous devez insérer la carte SD dans l'imprimante, et accéder au menu « Carte SD » en cliquant sur l'onglet correspondant. Sélectionnez le fichier à partir de la carte et lancez l'impression en cliquant sur le bouton en forme de triangle dans le menu « Carte SD ».

Lorsque vous commencez une tâche d'impression, l'imprimante reprend sa position initiale, la tête d'impression reprend sa position de parking et le plateau monte.

Ensuite, l'extrudeuse commence à chauffer jusqu'à la température convenable qui est de 210 °C pour le PLA (245 °C pour l'ABS).

Vous pouvez suivre l'avancement du processus d'impression sur l'afficheur LCD ou dans Repetier-Host. Une fois la température atteinte,

l'imprimante amorce l'extrudeuse en déposant une petite quantité de plastique sur la position de départ.

À partir de là, la tête d'impression se déplace un peu et se rend à la position de départ d'impression de l'objet. Il est préférable de supprimer cette petite quantité de plastique extrudé de la position de départ, avant de commencer une autre impression.

À ce stade, la calibration de l'axe des Z qui a été effectuée est primordiale pour la première couche.

Si la buse dépose une couche, mais se trouve assez loin du plateau, cela signifie que ce dernier doit être un peu plus proche de la buse.

Si par contre la buse touche le plateau d'impression lors de l'extrusion, vous devez arrêter immédiatement l'impression et vérifier les réglages.

Utilisez la pièce de calibrage de hauteur de l'axe des Z ainsi que les vis de réglage de hauteur du plateau d'impression.

La première couche est correcte lorsque le plastique s'entasse un peu sur le plateau. Il est parfaitement normal que la première couche soit un peu liquide.

Il est normal de voir des imperfections apparaître à certains endroits, comme si le plastique s'était un peu liquéfié. Une première couche d'impression correcte doit ressembler à celle de la figure 32.



Figure 33

Arrêt d'une tâche d'impression

Pour arrêter une tâche d'impression dans Repetier-Host, appuyez sur le bouton de forme carrée « Arrêt impression » dans la barre des boutons.

L'imprimante terminera les mouvements enregistrés dans sa mémoire tampon, puis elle positionnera la tête d'impression dans la position de parking qui commencera à refroidir. Le plateau d'impression n'est pas rehaussé.

Pour arrêter une tâche d'impression à partir de l'afficheur LCD, sélectionnez l'option arrêter la tâche d'impression (STOP PRINT JOB) dans le menu. L'imprimante s'arrête immédiatement, mais la tête d'impression reste à l'endroit où elle se trouve.

Déplacez la tête d'impression à l'aide de la commande « déplacer l'axe » (MOVE AXIS) dans le menu de contrôle de l'imprimante (CONTROL PRINTER).

Attention, la buse continuera à chauffer. Si vous souhaitez la refroidir, vous devez utiliser la commande « COOL-DOWN » dans le menu de contrôle (CONTROL PRINTER).

Terminer une tâche d'impression

Lorsque la tâche d'impression est terminée, la tête revient en position « parking » (stationnement) et commence à refroidir.

Retirez le plateau d'impression et utilisez un cutter ou une spatule pour retirer l'objet (voir la figure 33).

Attention à ne pas vous blesser et à ne pas couper la feuille en « buidtak ». Si la première couche a été imprimée trop proche du plateau, l'objet est difficile à enlever. Cela indique un mauvais calibrage de l'axe des Z.

Les problèmes et les solutions

- **le filament est bloqué** : il est possible que le filament reste bloqué dans la buse ou dans le bloc de l'extrudeuse. La première chose à faire, est de chauffer la buse à une température un peu plus élevée

que la température standard. La température normale pour le PLA est de 210° C, vous devez chauffer jusqu'à 220 °C. La température normale pour l'ABS est de 245 °C, dans ce cas vous devez chauffer jusqu'à 255 °C. Vous pouvez aussi extruder le filament manuellement en appuyant sur le levier de desserrage et en faisant passer le filament à travers le bloc de l'extrudeuse. Si cela ne fonctionne pas, démontez la zone où le filament est bloqué et enlevez le morceau de filament en cause ;

- **la buse est obstruée** : démontez la buse et nettoyez l'excédent de plastique à l'intérieur ;
- **l'objet imprimé est déformé** : l'objet commence à se détacher du plateau d'impression, c'est le phénomène de déformation. Cela se produit à cause du facteur de rétrécissement de la matière que vous imprimez. Lorsque chaque couche refroidit, elle rétrécit. Ainsi elle exerce une force sur la couche inférieure. Pour lutter contre les déformations, vous pouvez utiliser la fonction « Contour » (Brim) du logiciel de découpe. Cela agrandit la première couche, et donc la surface de contact avec le plateau. Il est important de positionner l'imprimante loin des zones de la pièce où la température peut varier rapidement (à l'écart des fenêtres, portes, etc.). Les courants d'air refroidissent de manière inégale l'objet et le déforment. Le PLA est beaucoup moins sensible à ce problème que l'ABS. Utilisez toujours l'option « Contour » (Brim) lors de l'impression en ABS ;
- **les couches sont décalées ou déformées** : vérifiez si toutes les poulies des moteurs pas à pas des axes X et Y sont correctement fixées. Si une poulie glisse, la rotation du moteur ne sera pas correctement transmise pour effectuer le mouvement latéral de la tête d'impression. Cela provoque des couches déplacées ou déformées. Le décalage des couches peut également se produire lorsqu'un moteur pas à pas n'a pas suffisamment de force pour déplacer la tête d'impression, comme défini dans le GCODE. En général, cela

se produit à cause des défauts d'alignement des poulies et des barres des axes X et Y. Vérifiez que ces dernières sont bien perpendiculaires et parallèles. Il est possible de résoudre ce problème en ajustant la valeur « VxyJERK ». Cette valeur détermine la vitesse à laquelle la tête d'impression peut changer de direction (sans accélération). Si les barres des axes X et Y ne sont pas parfaitement alignées, l'imprimante peut avoir des difficultés à se déplacer aux vitesses préréglées en usine. Si vous diminuez cette valeur, l'imprimante déplacera la tête d'impression plus lentement lors d'un changement de direction. Pour diminuer cette valeur, allez dans le menu : « Settings » → « Motion » → « VxyJERK ». Donnez la valeur 10 et enregistrez le réglage dans l'EEPROM (« setting » → « Store Memory »). Cela peut aussi être causé par un excès de dissipation de chaleur au niveau des cartes drivers 1,5 A, les moteurs ne sont plus alimentés ou alors avec un courant réduit. Vérifiez que le ventilateur situé au fond de l'imprimante fonctionne et assurez-vous que rien n'entrave la circulation de l'air en bas de l'imprimante.

Conclusion

Nous arrivons au terme de la description de cette imprimante 3DVERTEX qui permet une impression 3D en deux couleurs et ce pour un prix très abordable.

Nous vous rappelons qu'il est primordial d'effectuer de manière précise le montage de la partie mécanique afin de garantir une qualité d'impression optimale. ■



Breakout Board multifonctions !

Petites, polyvalentes, prêtes à l'emploi.



Module Relais

€ 11,¹⁰

Réf. : BREAKOUT005

**Capteur d'humidité
HIH-5030-001**



€ 17,⁸⁰

Réf. : BREAKOUT002

**Capteur de température
DS18B20**



€ 13,⁰⁰

Réf. : BREAKOUT001

**Module
Optocoupleur**



€ 10,⁶⁰

Réf. : BREAKOUT006

Module RS485



€ 9,⁷⁰

Réf. : BREAKOUT007

**Capteur de pression
MPXH6115A6U**



€ 19,⁷⁵

Réf. : BREAKOUT003

Ces cartes de prototypage
Breakout Board sont de véritables
platines de développement
spécifiques à un composant CMS,
elles sont déjà prêtes à l'emploi !



**Module
Alimentation
3,3/5/9 VDC**

€ 11,⁶⁰

Réf. : BREAKOUT008

**Module
capteur de lumière**



€ 10,⁶⁰

Réf. : BREAKOUT009

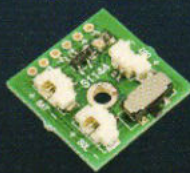
Module DAC



€ 11,⁶⁰

Réf. : BREAKOUT010

Chargeur Lithium-Ion



€ 10,⁶⁰

Réf. : BREAKOUT004

**Module de
température
analogique**



€ 11,⁶⁰

Réf. : BREAKOUT011

Module ADC



€ 11,⁶⁰

Réf. : BREAKOUT012



Cours ARDUINO

Dixième partie

de Mirco SEGATELLO

Dans cette dixième partie du cours Arduino, nous allons étudier et apprendre à utiliser la carte « Ethernet Shield » qui vous permettra de connecter une carte Arduino sur un réseau intranet ou sur Internet.

Nous continuons notre Cours sur le monde Arduino avec cette fois le développement d'un sketch orienté web à travers l'utilisation de la carte « Ethernet Shield ». Connecter une carte Arduino au web signifie l'accès à un grand nombre d'applications intéressantes.

Dans cet article, nous allons étudier l'interaction entre l'électronique et l'informatique afin de créer une solution complète permettant la réalisation de systèmes complexes, avec une variété d'applications allant du domaine de la domotique à celui du traitement de l'information.

La carte « Ethernet Shield »

Ceux qui connaissent le monde d'Arduino savent qu'il existe un grand nombre d'applications mises en œuvre, et parmi celles-ci la connexion à un réseau informatique. Si vous disposez d'une carte Arduino Diecimila, Duemilanove ou Uno, vous avez la possibilité de vous connecter à un réseau simplement en y reliant la carte « Ethernet Shield ».

Officiellement le consortium Arduino a adopté la puce d'interface modèle « W5100 » fabriquée par WIZNET.

Il y a donc une compatibilité totale avec les diverses cartes qui utilisent cette puce. Toutefois, dans le commerce, il existe d'autres cartes équipées de puces différentes.

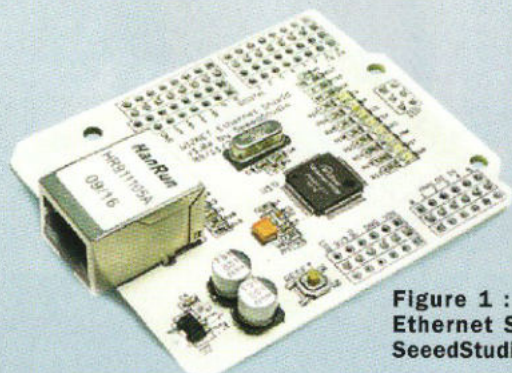


Figure 1 : la carte « Ethernet Shield » de SeeedStudio.

La carte Ethernet que nous utilisons dans cette leçon est la version produite par SeeedStudio (SEEEDWIZNET) et distribuée par la société Comelec (www.comelec.fr). Cette carte est basée sur la puce Ethernet WIZNET W5100 et est compatible avec Arduino (également avec la version MEGA) ainsi qu'avec Seeeduino (également avec la version Mega11).

Il existe une autre carte (ETHW5100), également disponible chez Comelec, basée sur la même puce Ethernet WIZNET W5100 qui est compatible avec Arduino (mais pas avec la version MEGA).

Ces deux cartes utilisent la même puce et supportent jusqu'à 4 connexions sockets simultanées. Le connecteur de liaison est de type standard RJ45.

Un socket est un identifiant unique représentant une adresse sur un réseau. Des processus peuvent s'y connecter pour y envoyer des données ou pour en recevoir. Les processus devront adopter un protocole de communication afin d'assurer un échange cohérent de données.

L'adresse du socket est spécifiée par le nom de l'hôte sur lequel elle a été créée et par le numéro de port. Nous utiliserons le protocole TCP/IP pour assurer le transport des données (paquets) entre les processus serveur et client.

Le serveur est le processus qui détecte toute nouvelle connexion du client et effectue la récupération des données. Le client est le processus qui va tenter de se connecter au serveur et lui envoyer des données.

Les broches utilisées par Arduino pour communiquer avec la puce W5100 sont les broches digitales 10, 11, 12 et 13 (port SPI) qui ne sont donc pas utilisées comme de simples entrées/sorties (I/O). Le bouton de remise à zéro de la carte réinitialise à la fois le circuit W5100 ainsi que la carte Arduino (ou Seeeduino).

La puce Ethernet WIZNET W5100 implémente au niveau matériel une « stack » (pile) IP complète de protocoles de communications TCP et UDP.

Le protocole TCP (Transfer Control Protocol) est basé sur la création d'une communication entre un « Client » et un « Serveur ». Le « Client » peut initier une communication en envoyant une requête au « Serveur » qui écoute sur un port particulier.

Si le « Serveur » accepte, il ouvre une communication et il est ainsi possible de transférer des données. Une des caractéristiques de ce protocole est qu'il connaît l'état de la connexion et qu'il vérifie si les paquets (données) ont été correctement reçus par le destinataire.

Dans le protocole UDP (User Datagram Protocol), à la différence du protocole TCP, il n'y a pas de communication ouverte mais tout simplement l'envoi de données vers un port donné ayant une adresse IP.

Cela signifie qu'il peut avoir le rôle de « Client » ou de « Serveur ». Par conséquent, il n'y a aucun contrôle de l'état de la communication et il n'est pas possible de savoir si les données échangées sont effectivement arrivées au destinataire.

Nous pouvons comparer le protocole UDP à l'envoi d'une simple lettre par la Poste. Il n'y a pas de certitude absolue de la livraison, mais dans la plupart des cas la lettre arrive à destination.

À l'inverse, le protocole TCP peut être comparé à l'envoi d'une lettre en recommandé avec accusé de réception, dans ce cas il y a une vérification de la bonne réception du message. Il est donc plus facile de mettre en œuvre le protocole UDP, bien qu'il impose des limites au système.

Revenons à notre carte, dont voici la signification des LED :

- **TX** : clignote lorsque la carte transmet des données ;
- **RX** : clignote lorsque la carte reçoit des données ;
- **COL** : clignote lorsqu'une collision de paquets se produit sur le réseau ;
- **FDX** : indique que la connexion au réseau est en full-duplex ;
- **LINK** : indique la présence d'un réseau et clignote lorsque la carte transmet ou reçoit des données ;
- **RST** : réinitialisation ;
- **PWR** : indique que la carte est alimentée.

En ce qui concerne l'utilisation de la carte au niveau logiciel, cela ne pose pas de problèmes car l'environnement de développement Arduino contient déjà une librairie et des exemples de sketches pour la gestion de l'Ethernet. Nous pouvons passer directement à l'utilisation pratique.

La carte « Ethernet Shield » (version SeeedStudio) est prête à l'emploi, cependant nous vous recommandons de souder des barrettes femelles afin de pouvoir facilement la connecter à la carte Arduino ou afin de pouvoir insérer une seconde carte. Une fois reliée à la carte Arduino, elle est prête à l'emploi.

Pour les tests, vous pouvez laisser connectés le câble réseau et le câble USB utilisés pour la programmation.

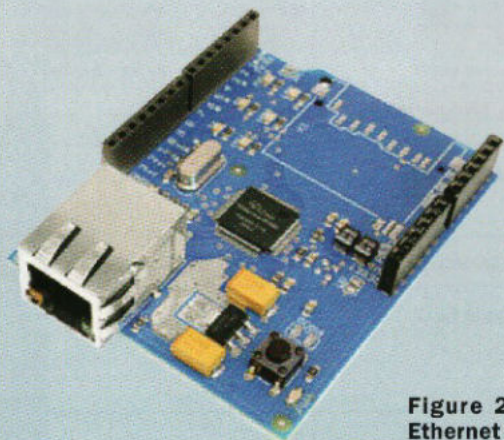


Figure 2 : la carte Ethernet ETHW5100.

La carte Arduino communique avec l'Ethernet via le port SPI, en laissant libre le port UART.

Une attention particulière doit être accordée à la configuration de votre réseau domestique. Pour résumer, le PC principal qui nous permet de programmer Arduino dispose d'un système d'exploitation Windows 7.

Les paramètres réseau sont :

- **adresse IP** = 192.168.0.199
- **masque de sous réseau** = 255.255.255.0
- la passerelle et le serveurs DNS ne sont pas spécifiés.

Ces paramètres sont accessibles via le menu « Démarrer » → « Panneau de configuration » → « Centre Réseau et partage de connexion ». Le PC et Arduino (Ethernet Shield) sont connectés à un routeur D-Link ayant les réglages d'usine.

La version de l'environnement de développement Arduino doit être au minimum la 0021, sinon certaines fonctions, en particulier sur les chaînes de caractères ne seront pas disponibles.

Pour notre premier exemple, nous utilisons notre carte Ethernet pour créer un serveur Web. Nous allons créer, grâce au circuit W5100, une page web accessible via Internet à partir de n'importe quel navigateur.

Cette page contiendra les données relatives à la valeur des entrées analogiques d'Arduino. Il s'agit donc d'une page capable de créer une interaction entre Arduino et le monde extérieur.

Connectez tous les éléments, puis lancez l'IDE d'Arduino. Ouvrez le fichier exemple « ethernet_01.pde ». Pour rendre la page web compatible avec votre réseau, vous devez définir correctement les adresses MAC et IP.

L'adresse MAC (MAC address) est un code de 12 caractères qui identifie de manière unique chaque appareil connecté au réseau. Dans notre cas, la carte réseau et le routeur ont une adresse unique.

Ne modifiez pas l'adresse MAC figurant dans le sketch. Si votre carte doit accéder à Internet, assurez-vous de récupérer une adresse MAC libre.

L'adresse IP (IP Address) est une suite de 4 nombres compris entre 0 et 255 et, de manière similaire à l'adresse MAC, identifie chaque périphérique connecté au réseau. Même si elle est unique pour chaque périphérique du réseau, elle doit être attribuée conjointement avec le masque de sous-réseau. Il s'agit d'une autre séquence de 4 nombres qui vous permettent de spécifier les sous-réseaux.

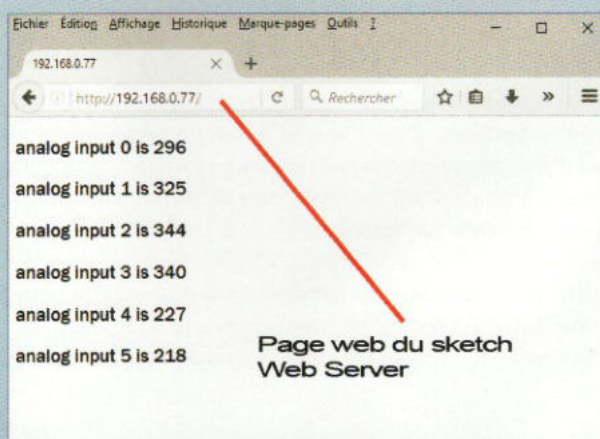


Figure 3 : ici la page web du serveur qui s'affiche.

Dans notre cas, nous avons un masque de sous-réseau égal à 255.255.255.0. Ainsi tous les périphériques seront en mesure de communiquer s'ils font partie du sous réseau dont les 3 premiers nombres sont identiques aux 3 premiers de l'adresse IP et le 4^{ème} pouvant varier de 0 à 255.

L'adresse IP d'Arduino est du type 192.168.0.XXX, où XXX est un nombre compris entre 0 et 255 qui n'est pas utilisé par d'autres périphériques sur le réseau. Pour résumer, la configuration de notre exemple est :

- ROUTER IP = 192.168.0.1
- PC IP = 192.168.0.199
- PC masque de sous réseau = 255.255.255.0
- ARDUINO IP = 192.168.0.77
- ARDUINO masque de sous réseau = 255.255.255.0

Nous laissons tous les autres paramètres non spécifiés.

Dans une connexion TCP, en plus de l'adresse IP du poste distant où doit parvenir l'information, il faut connaître l'application qui demande la transmission des données. Pour cela, les ports de communication sont utilisés. Ils correspondent à une sorte d'allocation de la mémoire de l'ordinateur qui existe lorsque deux ordinateurs sont en communication l'un avec l'autre.

Les ports fournissent un point terminal pour l'application distante. Les ports sont toujours identifiés par un nombre compris entre 0 et 65535.

Les valeurs 0 à 1023 sont affectées à des fonctions spécifiques du système d'exploitation. Voici la configuration utilisée pour notre exemple :

- `byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };`
Adresse MAC physique
- `byte ip[] = { 192, 168, 0, 77 };` Adresse IP du réseau
- `Server server(80);` Port du serveur

Maintenant, chargez le sketch dans Arduino afin que le firmware puisse communiquer avec le routeur. Ouvrez votre navigateur et tapez l'adresse `http://192.168.0.77`. Vous devez voir la page web qui réside dans Arduino se charger. Elle affiche les données des entrées analogiques.

Si les entrées ne sont pas connectées, les valeurs affichées seront aléatoires. Pour lire une nouvelle série de valeurs, vous devez renouveler la requête de lecture de la page. Cela s'effectue en cliquant sur le bouton de rafraîchissement de la page de votre navigateur.

Parfois cette fonction est symbolisée par une flèche enroulée sur elle-même ou 2 flèches vertes (selon le navigateur) et correspond à la commande « Actualiser la page courante ».

Pour tester le sketch, utilisez un trimmer ou un potentiomètre d'une valeur de 10 k Ω . Soudez 3 fils et reliez-les de la manière suivante : une extrémité au + 5 V, l'autre extrémité à la masse (GND) et le point milieu à une entrée analogique d'Arduino. Ainsi le trimmer permet d'imposer (régler) une certaine tension comprise entre 0 et 5 V sur l'entrée analogique.

L'application est intéressante, actualisez la page web pour afficher en temps réel le niveau de la tension présente sur l'entrée analogique.

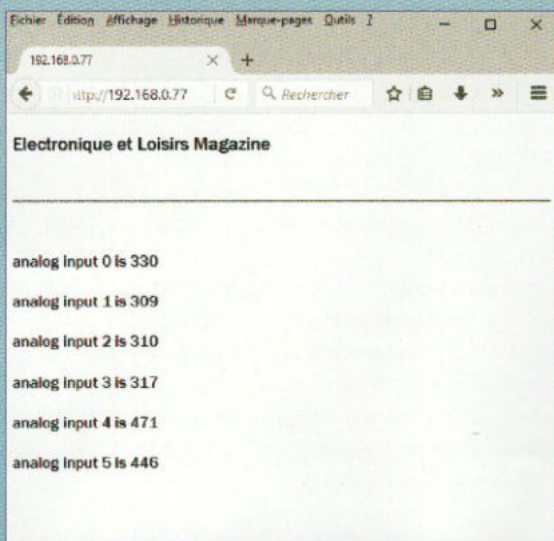


Figure 4 : page web du sketch « ethernet_01.pde ».

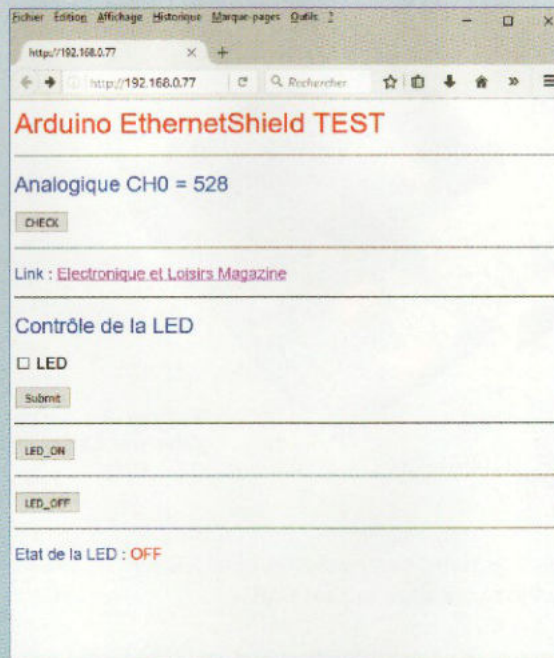


Figure 5 : page web du sketch « ethernet_02.pde ».

Manœuvrez le trimmer et actualisez la page web, vous devez voir une nouvelle valeur de la tension sur l'entrée, vous pouvez répéter l'opération pour d'autres positions du potentiomètre.

Vous trouverez le fichier exemple « ethernet_01.pde » sur notre site dans le sommaire détaillé de la revue.

Maintenant, nous voulons apprendre à gérer les sorties. Pour cela, nous allons insérer des boutons dans la page web, avec lesquels nous pourrions activer une LED.

Etant donné que la broche 13 est occupée, la LED du système n'est pas accessible. Nous allons donc relier une LED en série avec une résistance de 470 Ω entre la sortie numérique 2 et la masse. Le sketch correspondant se nomme « ethernet_02.pde », il est téléchargeable comme le précédent.

Dans ce programme, nous avons ajouté la gestion d'une LED, soit à l'aide du bouton « CHECK » soit par deux boutons. La page web est plus complexe afin de montrer le potentiel du système.

En cherchant à en savoir d'avantage, nous avons trouvé une librairie intéressante créée par Jordan Terrell (<http://blog.jordanterrell.com/post/Arduino-DHCP-Library-Version-04.aspx>) qui permet de mettre en œuvre la fonction DHCP, c'est-à-dire la possibilité que le routeur assigne automatiquement à Arduino une adresse IP libre sans devoir la spécifier à l'avance.

Si nous voulons faire en sorte que notre périphérique soit visible en dehors de notre réseau domestique, il faut nécessairement disposer d'une connexion Internet et activer un canal de communication avec notre application.

En fait, les accès externes seront interceptés par le routeur, qui doit être configuré de telle sorte qu'un certain type de requêtes aient accès à Arduino.

Pour ce faire, vous devez activer le service de réacheminement des ports (port forwarding) du routeur, en spécifiant le numéro de port accessible via Internet ainsi que le port sur lequel Arduino est connecté sur le réseau interne. Pour plus de détails, référez-vous aux spécifications du type de routeur que vous utilisez.

Toujours en accédant à Internet, vous pouvez tester l'exemple « WebClient », qui prévoit que notre application effectue une demande (Client) à un serveur externe, qui est Google dans ce cas. La requête consiste en une recherche du terme « Arduino », la réponse sera visualisée via le Moniteur série et correspondra au résultat de la recherche.

Un autre exemple intéressant est appelé « UdpNtpClient » et prévoit, via Internet, l'interrogation d'un serveur NTP afin de connaître l'heure internationale de manière précise. En fait, le serveur NTP (Network Time Protocol) fournit les informations horaires en les mettant à disposition des utilisateurs de manière facilement accessible. Il suffit d'envoyer des requêtes respectant le protocole UDP. L'heure courante est lue à partir de l'application, puis mise à disposition via le Moniteur série d'Arduino.

Il existe une autre possibilité de diffuser les données acquises par Arduino sur Internet, ou transférer des données vers un serveur de données, tel que le serveur « Pachube » (www.pachube.com).

Pachube est conçu comme un service web qui vous permet de connecter et partager en temps réel les données d'un capteur par exemple.

L'idée de Pachube est de faciliter l'interaction entre les environnements physiques et virtuels, les données d'un capteur physique pouvant alimenter un site web ou un objet et inversement. Ainsi, il permet d'intégrer des données dans une page web à la manière d'un widget, que ce soit pour y donner accès ou pour transformer un blog en capteur de données.

Pachube peut également être utilisé pour faciliter les connexions multiples à un capteur et permet aussi à ses utilisateurs (et plus encore à leurs objets) de se brancher à n'importe quel autre projet. Une solution qui permet donc de faire dialoguer simplement deux objets entre eux.

Dans notre cas, Arduino dialoguera uniquement avec ledit serveur. Les utilisateurs qui voudront voir les données devront se connecter sur le site www.pachube.com, lequel, en plus de rendre disponibles les données, fournira une visualisation graphique ainsi que les informations de géo-localisation.

En ce qui concerne le développement du firmware, Arduino met à disposition des exemples « PachubeClient » et « PachubeClientString ».

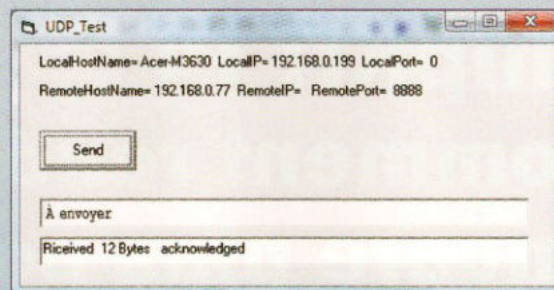


Figure 6 : application du protocole UDP en Visual Basic.

Un autre exemple intéressant consiste à envoyer et recevoir, d'un PC à Arduino, des chaînes de caractères à l'aide du protocole UDP.

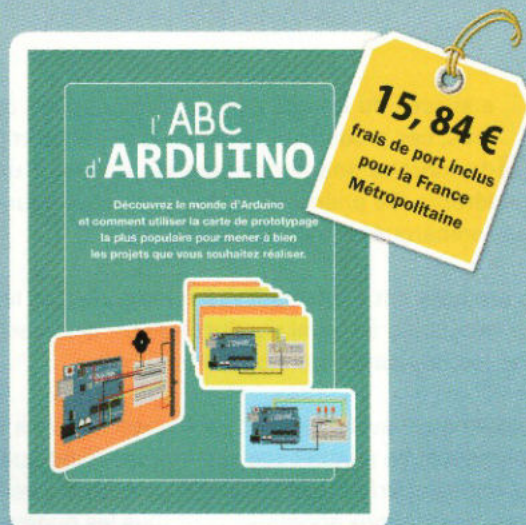
Ouvrez le sketch appelé « udp.pde », modifiez l'adresse IP si nécessaire et téléversez-le dans Arduino. Avec le Moniteur série de l'IDE d'Arduino, vous pouvez voir une chaîne arriver avec la réponse correspondante.

Pour utiliser le protocole UDP, il est nécessaire d'installer la librairie « hypermedia.net », que vous trouverez dans le dossier « UDP_Librairie_Processing » en téléchargement sur notre site.

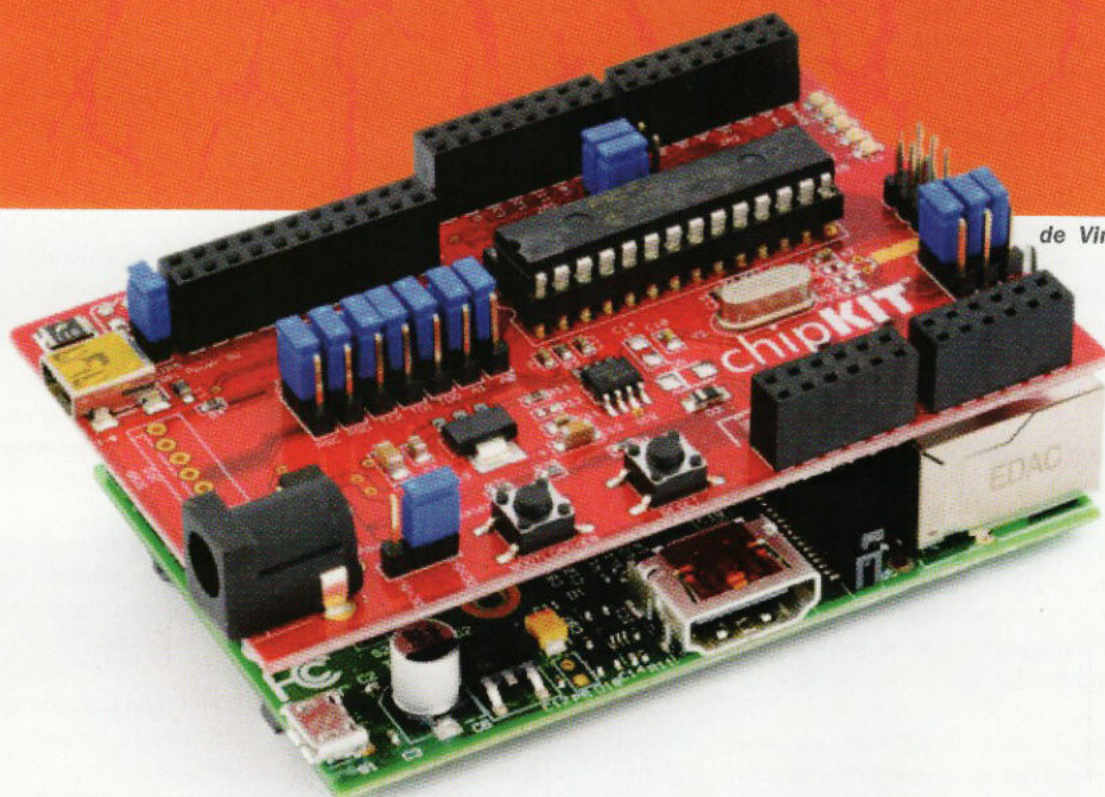
Compte tenu de l'intérêt de communiquer entre un PC et Arduino via Ethernet, nous avons décidé de créer une petite application écrite en Visual Basic et qui implémente le protocole UDP.

L'application envoie une simple chaîne de texte et attend une réponse d'Arduino, vous pourrez facilement la modifier à votre convenance. Le code source de l'application est disponible en téléchargement sur notre site.

Comme vous l'aurez remarqué, les applications mises en œuvre sont très nombreuses et d'une grande simplicité. Une certaine connaissance des réseaux informatiques et de la programmation en HTML permettent d'obtenir dans un court laps de temps des applications très innovantes et professionnelles qui ne manqueront pas de vous donner satisfaction.



chipKIT PI : comment rendre le RaspberryPi encore plus polyvalent



de Vincenzo Mendoia

Voici une nouvelle carte de prototypage qui arrive dans le monde des produits embarqués afin de mettre en valeur les qualités du RaspberryPi et d'Arduino. Découvrons-la !

Le succès du RaspberryPi et d'Arduino est connu de tous, ces produits ont conquis une partie importante du marché des hobbyistes qui a atteint une taille considérable. Ce phénomène d'engouement pour ces systèmes embarqués n'est plus négligeable.

Ces produits ont également été remarqués par les principaux fabricants de semi-conducteurs et, plus particulièrement, les fabricants de microcontrôleurs.

Comme nous l'avons signalé, de nouvelles cartes de développement et de prototypage apparaissent continuellement. Elles ont des performances de plus en plus élevées et complexes, avec un prix abordable pour tous les budgets.

Les modèles les plus usités restent Arduino et RaspberryPi avec une énorme communauté autour de chacun d'eux.

Vous ne serez donc pas surpris de voir que même le célèbre fabricant de microcontrôleur **Microchip** a développé une carte de prototypage compatible avec le monde du RaspberryPi et d'Arduino. Il s'agit de la carte « **chipKIT PI** » que vous pouvez voir en figure 1.

Du fait de l'esprit Open Source avec lequel le projet a été mené, les développeurs ont choisi d'utiliser le logiciel **Eagle** pour la construction de la carte. Ce logiciel est très répandu dans l'industrie. L'utilisation d'Eagle pour la réalisation de cette carte est indiquée sur la face arrière de la carte

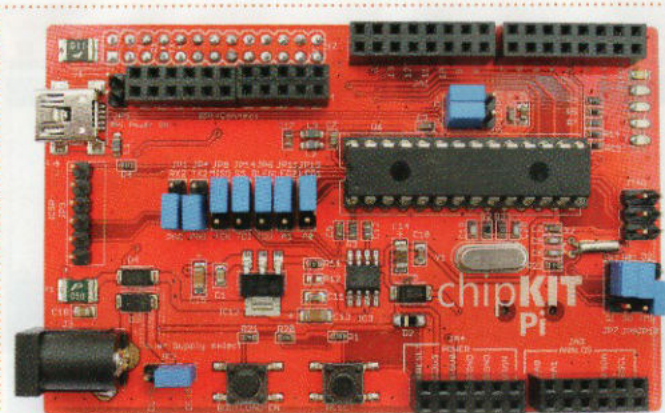


Figure 1

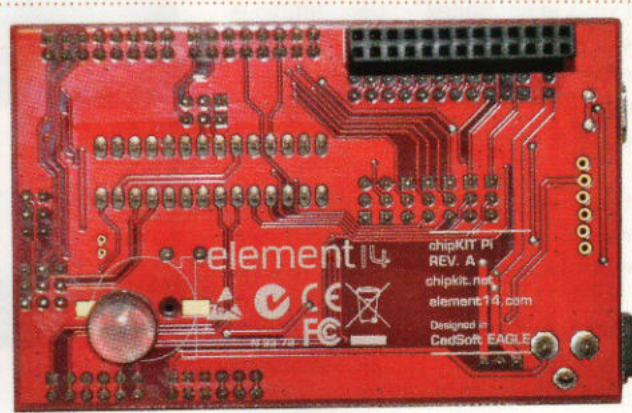


Figure 2

« chipKIT Pi », comme vous pouvez le voir en figure 2.

Le potentiel de la carte RaspberryPi est vraiment remarquable, comme vous le savez grâce à nos nombreux articles décrits dans la revue.

Afin de procurer des fonctionnalités avancées aux hobbyistes, les concepteurs ont prévu de gérer la carte « chipKIT Pi » à l'aide d'un microcontrôleur puissant, de sorte qu'il soit en mesure de mettre en œuvre des applications tactiles et de traitement audio.

Le cœur de la carte « chipKIT Pi » est un **PIC32MX250F128B** qui est un microcontrôleur puissant et flexible. Il s'agit d'un PIC fonctionnant sur 32 bits et encapsulé dans un boîtier plastique

de type « DIP » à 28 broches. Il est vraiment intéressant par rapport aux fonctionnalités offertes.

La carte « chipKIT Pi » est compatible Arduino et est spécialement conçue pour le RaspberryPi.

Elle permet aux utilisateurs de créer, compiler et programmer des applications Arduino sur la plate-forme RaspberryPi.

La « chipKIT Pi » permet le **développement d'applications compatibles Arduino 3,3 V sur le RaspberryPi** (attention aux cartes Arduino fonctionnant en 5 V).

Voici quelques caractéristiques techniques de ce microcontrôleur :

- fréquence maximale de fonctionnement 50 MHz ;
- mémoire de programme 128 ko ;
- RAM : 32 ko ;
- interface CODEC (I2S, AC97) ;
- USB de type « host » ;
- 9 canaux ADC à 10 bits, avec une fréquence d'échantillonnage maximale de 1100 ksp/s (kilo samples per second ou kilo échantillons par seconde) ;
- 5 canaux PWM ;
- 5 timers de 16 bits ;
- 2 oscillateurs internes (8 MHz et 32 kHz) ;
- 3 comparateurs ;
- 2 SPI ;
- 2 I2C ;
- tension d'alimentation de 2,3 à 3,6 VDC.

Grâce à cette carte, il est possible d'ajouter la puissance d'un PIC32 à la flexibilité d'Arduino ou le potentiel de Linux à travers un RaspberryPi, en interfaçant directement la carte « chipKIT Pi » au connecteur d'extension E/S du RaspberryPi.

Le PIC32MX250F128B fonctionne avec une tension de 3,3 VDC comme le RaspberryPi. Il n'est donc pas nécessaire de traduire (adapter) les niveaux logiques.

Par contre, en cas d'utilisation avec une carte Arduino fonctionnant en 5 V, il faudra être prudent en adaptant les niveaux logiques à 3,3 V.

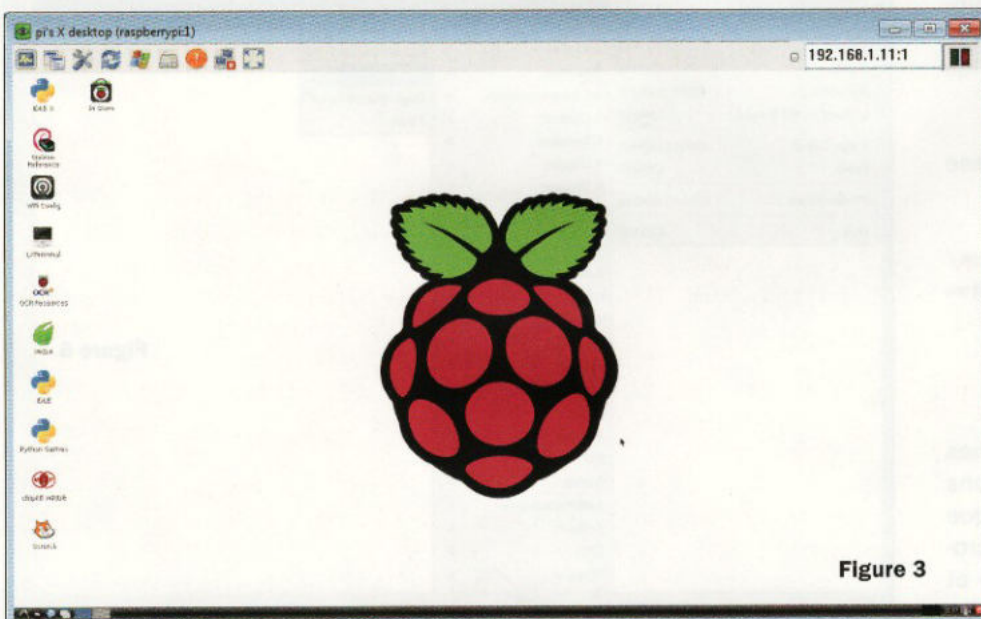


Figure 3

Si vous le souhaitez, cependant, vous pouvez adapter une carte standard Arduino fonctionnant en 5V, en translatant les niveaux logiques comme indiqué dans ce document :

http://www.microchip.com/stellent/groups/techpub_sg/documents/devicedoc/en026368.pdf

Le schéma complet de la carte « chipKIT Pi » est disponible à l'adresse suivante :

<http://www.element14.com/community/servlet/JiveServlet/download/63873-2-118998/chipKIT%20Pi%20Schematic.zip>

Le schéma électrique de cette carte est relativement simple, cela est dû au fait que le PIC32 intègre un grand nombre de périphériques.

Le manuel d'utilisation (en anglais) de la carte « chipKIT Pi » fournit les informations essentielles pour commencer à travailler immédiatement.

Elle peut également être utilisée indépendamment, sans le RaspberryPi, grâce à un connecteur d'alimentation externe ou via le port USB (sélectionnable à l'aide d'un cavalier).

La carte dispose d'un connecteur ICSP de programmation en circuit utilisé par Microchip afin de programmer et déboguer le PIC32.

La carte « chipKIT Pi » peut être facilement programmée par l'intermédiaire d'un environnement de développement Open Source **MPIDE** (voir les figures 3 et 4).

Vous pouvez le télécharger à l'adresse suivante :

<http://downloads.element14.com/downloads/ChipKitPi.zip?COM=-chipKITPi>

La taille du fichier est de 1,7 Go.

L'IDE MPIDE supporte de nombreuses cartes, y compris toutes les versions et certains clones d'Arduino, ainsi que les cartes de développement Microchip, celles de la série « chipKIT » et beaucoup d'autres (voir la figure 5).

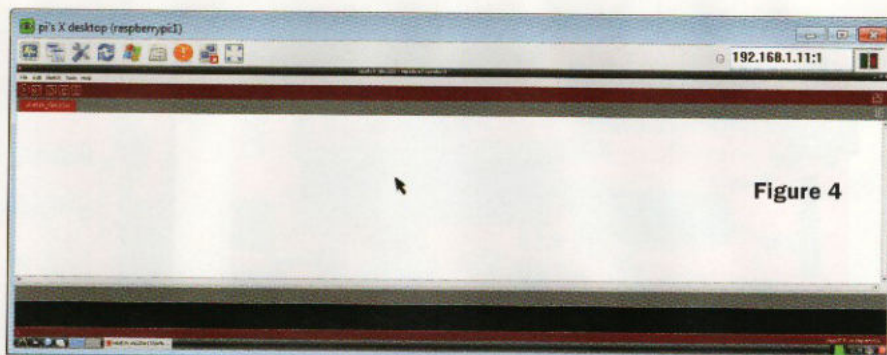


Figure 4

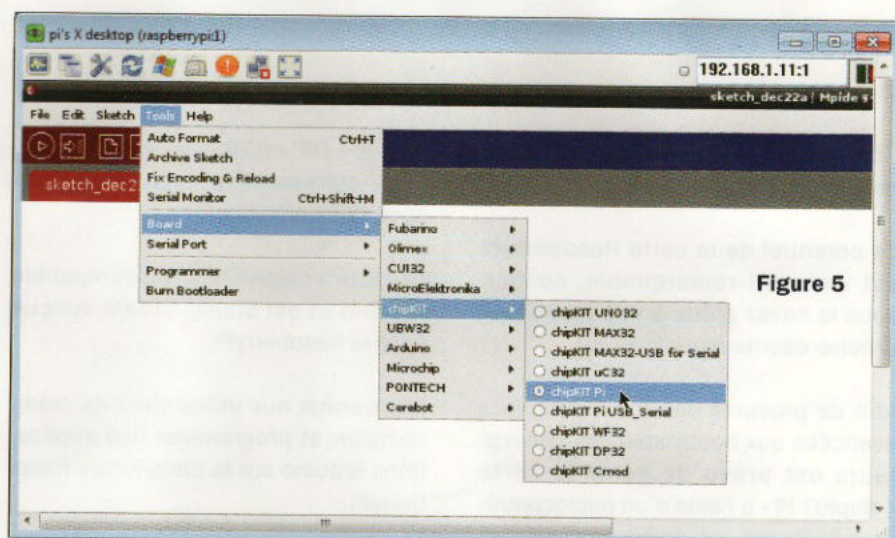


Figure 5

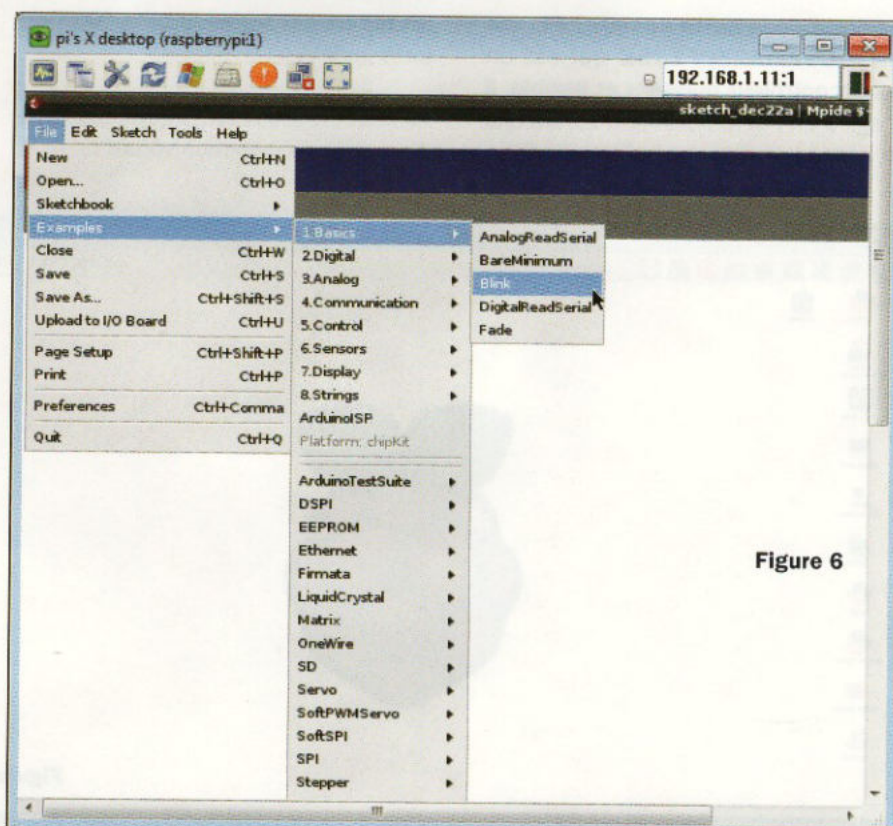


Figure 6

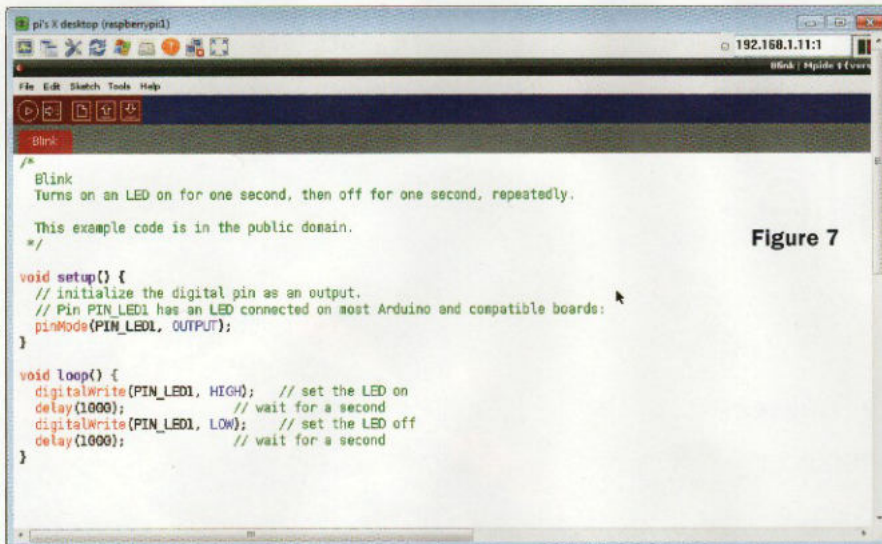


Figure 7

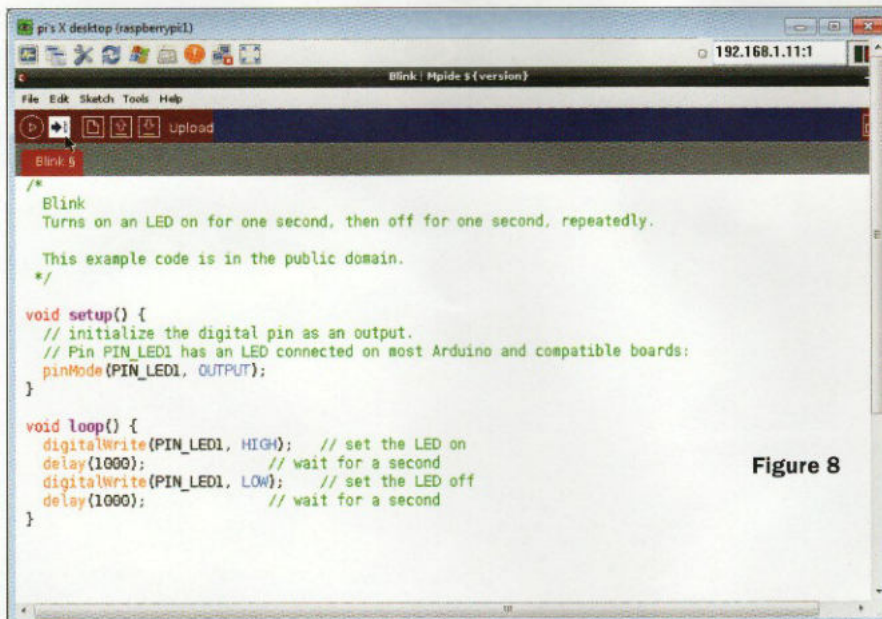


Figure 8

Pour décrire l'utilisation de la carte « chipKIT Pi », partons d'un exemple très simple, comme un petit programme « hello world » ou le clignotement d'une LED, comme nous sommes habitués (voir les figures 6 et 7).

Avant de pouvoir programmer la carte « chipKIT Pi », il faut **configurer le mode « bootloader »**. Pour ce faire, nous devons veiller à ce que les deux cavaliers **JP1** et **JP4** soient positionnés vers le haut (TX et RX), comme indiqué en figure 8.

Une fois ceci fait, nous devons appuyer et maintenir enfoncé le bouton « **BOOTLOAD-EN** » situé en bas à gauche à côté de la prise d'alimentation.

Alors que le bouton « **BOOTLOAD-EN** » est toujours maintenu pressé, appuyez brièvement sur le bouton « **RESET** » situé à droite de ce dernier, puis relâchez le bouton « **BOOTLOAD-EN** ». Le mode « bootloader » est confirmé par l'allumage de la LED TX1 et le clignotement de la LED RX1.

Cette combinaison de signalisations et de boutons permet de charger le code dans le PIC32 à l'aide de l'icône « **upload** » (voir la figure 9).

Une fois le chargement du programme terminé, les deux LED TX1 et RX1 s'éteignent. Dans la barre de commande en bas de la fenêtre de l'IDE Microchip, nous apercevons une expression nous



Figure 9

indiquant que la programmation s'est déroulée avec succès (Done uploading). La LED commence à clignoter à une fréquence de 1 Hz.

Nous avons décrit seulement un exemple d'utilisation de la carte « chipKIT Pi ». Sur le site officiel, vous trouverez d'autres informations ainsi que d'autres applications plus complexes. Visitez l'adresse suivante :

https://www.element14.com/community/community/raspberry-pi/raspberry-pi-accessories/chipkit_pi

Toute l'actualité du Raspberry Pi
www.framboise314.fr



VEILLEUSE A LED universelle

Cette veilleuse à LED peut être utilisée comme un point lumineux pour remplacer une ampoule à incandescence classique fonctionnant sous 230 VAC ou 12/24VDC. Elle est alimentée directement à partir du secteur sans transformateur.

de Boris Landoni



Dans les anciennes armoires électriques ou les panneaux de commande d'équipements anciens se trouvent des ampoules 12 V ou 24 V servant d'éclairage lorsqu'il est nécessaire d'y intervenir.

Les avantages de l'utilisation de LED comme source d'éclairage en remplacement d'ampoules traditionnelles semblent évidents. En effet, les LED permettent d'éclairer parfaitement des petites zones avec une consommation inférieure et une durée de vie bien supérieure à celle des ampoules à filament.

Le circuit que nous vous proposons dans ces pages est, en fait, une LED précédée d'un redresseur double alternance qui permet de la connecter aussi bien sur le secteur alternatif 230 VAC que sur une source de tension continue. Dans ce cas, la présence du pont de diode permet de relier le circuit sans tenir compte de la polarité. Vous pouvez ainsi remplacer n'importe quelle source de lumière classique par cette LED.

Selon vos goûts, vous pouvez utiliser des LED de n'importe quelle couleur, cependant si vous devez vous en servir pour éclairer une zone, vous devrez utiliser des LED blanches (fonctionnant en lumière blanche).

Nous nous chargerons de ces détails plus tard, pour l'instant étudions la conception du circuit. Nous allons utiliser une LED blanche à haut rendement qui permet, en moyenne, de produire un flux lumineux de l'ordre de 50 à 60 lm/W.

Si nous fixons la puissance de fonctionnement à 70 mW, le flux lumineux sera d'environ 3,5 lumens.

Le schéma électrique

Maintenant, examinons le schéma du circuit qui est extrêmement simple. L'ensemble se réduit à une LED alimentée par un **pont de Graetz**, c'est-à-dire un **pont redresseur double alternance**, ainsi que de 2 résistances en série de 3,3 k Ω .

Ce pont de diodes effectue deux tâches, la première est de faire fonctionner les LED à partir d'une tension alternative, et la seconde consiste à maintenir une polarité identique de la tension de la LED indépendamment de la polarité de l'alimentation appliquée entre les points L et N.

Les deux résistances R1 et R2 **limitent le courant circulant dans la LED** et peuvent avoir des valeurs différentes en fonction de l'utilisation du circuit. Vous pouvez aussi en monter une seule. En montant les deux résistances en série de 3,3 k Ω (1/4 W), le montage permet **d'allumer une LED avec une alimentation de 24 VAC ou 24 VDC**. Le courant dans la LED sera d'environ 3,5 mA.

Avec une alimentation de **12 V**, il est nécessaire de court-circuiter l'une des deux résistances, afin d'avoir en série avec le pont redresseur seulement 3,3 k Ω . Il est possible, afin d'augmenter la luminosité de la LED, de réduire les valeurs

de R1 et R2 à 1 k Ω , de manière à obtenir un courant de 10 mA dans la LED avec une alimentation de 24 V et 4,5 mA avec 12 V.

Notre **veilleuse à LED peut fonctionner aussi sur le secteur 230 VAC**, celle-ci se comporte alors comme une lampe témoin ou une veilleuse de nuit. Elle peut aussi éclairer certaines zones afin de trouver son chemin en pleine nuit (couloirs, escalier d'une maison), mais aussi pour trouver les interrupteurs.

La seule précaution à prendre pour faire fonctionner le montage sur le secteur 230 VAC est d'augmenter la valeur des résistances à 47 k Ω chacune. La résistance globale en série avec le pont de Graetz est alors de 94 k Ω , ce qui fixe à 2,4 mA le courant circulant dans la LED, en supposant une tension de 230 VAC.

Il est possible d'augmenter la luminosité de la LED, avec un courant pouvant atteindre 20 mA. Cependant, cela implique, en plus de la réduction de la valeur ohmique des résistances à 6,8 k Ω , une augmentation de la puissance à 2 W pour chaque résistance.

Expliquons maintenant le rôle du pont redresseur dans le circuit. Nous comprenons qu'il permet d'alimenter la LED aussi bien avec une tension alternative que continue.

Le **pont de Graetz** est un **quadrilatère formé par 4 diodes** connectées de la manière visible aux figures 1 et 2.

L'entrée est connectée au secondaire d'un transformateur ou, comme dans notre cas, directement sur le réseau secteur (230 VAC). La caractéristique principale de notre pont est qu'il redresse la tension alternative avec une efficacité maximale.

Un tel circuit fonctionne en exploitant un agencement particulier des 4 diodes, ce qui permet de garder la même polarité en sortie quelle que soit la polarité d'entrée. En d'autres termes, **la charge** (dans notre cas, la diode électroluminescente LD1 ...) **reçoit toujours une tension de même polarité**, lorsque la demi-onde à l'entrée du pont est positive ou négative. Car comme les diodes sont interconnectées, il n'y a qu'un seul couple de diodes à la fois qui conduit.

Pour comprendre le fonctionnement du pont de Graetz, supposons que le courant alternatif du secondaire du transformateur soit positif et appliqué à un instant donné au sommet du quadrilatère. Dans ce cas, la diode **D1 se trouve avec son anode positive par rapport à sa cathode** et devient donc **conductrice**. Un courant circule alors dans la résistance de charge et non dans D2 car cette dernière se retrouve avec une polarité positive sur sa cathode. Le **courant poursuit son chemin à travers D4**, car une **polarité positive est appliquée sur son anode** (sa cathode est négative). Pendant **l'alternance positive, D2 et D3 sont bloquées**.

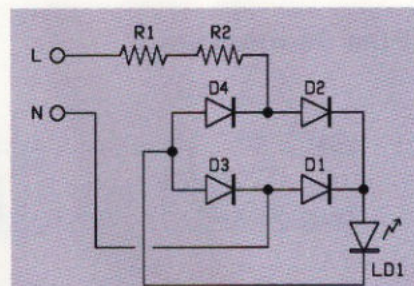


Schéma électrique de la veilleuse à LED universelle.

Maintenant, prenons le cas contraire, c'est-à-dire qu'en sortie du transformateur nous avons une alternance négative. Les diodes **D2 et D3** deviennent **conductrices** alors que **D4 et D1** sont **bloquées**.

En effet, D2 se retrouve avec une polarité positive sur son anode car l'alternance est négative (sa cathode a un potentiel inférieur à celui de son anode). Un courant circule alors dans la charge et pénètre dans l'anode de D3, qui devient positive par rapport à la cathode.

Pour résumer, tout se passe comme si les diodes D2 et D3 étaient connectées en série et étaient polarisées directement par la tension du transformateur (positive sur l'anode de D2 et négative sur la cathode de D3).

À ce stade, la charge reçoit toujours une tension de même polarité et un courant de même intensité quel que soit le type d'alternance, positive ou négative.

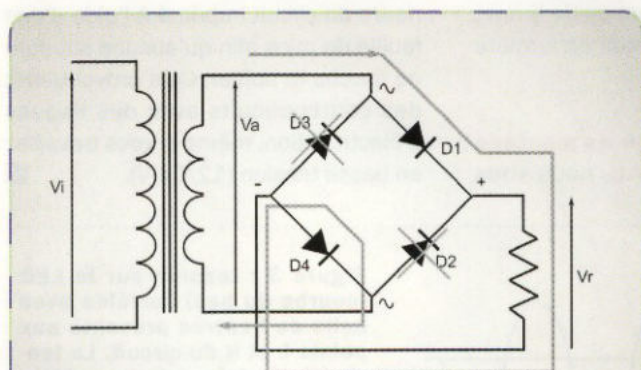


Figure 1 : lorsque la tension alternative est positive sur le sommet du pont, les diodes D1 et D4 se trouvent en série et conduisent car elles sont polarisées directement (l'anode est positive par rapport à la cathode).

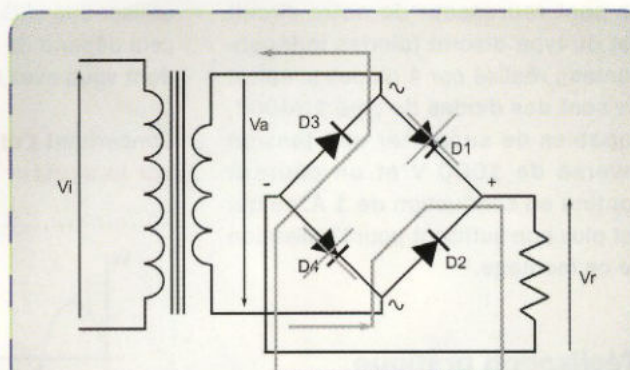
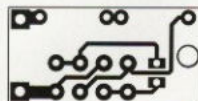


Figure 2 : lorsque la tension est négative sur le sommet du quadrilatère, les diodes D2 et D3 se retrouvent en série avec la charge et deviennent conductrices car elles sont polarisées directement (l'anode est positive par rapport à la cathode).

Plan de montage



Circuit imprimé côté soudures à l'échelle 1 : 1 de la veilleuse à LED.



Plan de câblage des composants de la veilleuse à LED.



Photo de l'un de nos prototypes de la veilleuse à LED.

Liste des composants

R1.....	3,3 k Ω ¼ W (voir texte)
R2.....	3,3 k Ω ¼ W (voir texte)
D1.....	1N4007
D2.....	1N4007
D3.....	1N4007
D4.....	1N4007
LD1....	LED bleue 3 mm

Si le pont de Graetz est alimenté entre les deux points marqués du symbole « ~ » (alternatif) directement à partir du réseau 230 VAC ou à partir d'un secondaire d'un transformateur, **la sortie du pont de diodes** correspondant aux point « + » et « - » **aura une tension dont la fréquence est le double de celle présente à l'entrée du pont.**

Tout se passe comme si le pont faisait « basculer vers le haut » la partie négative de l'onde sinusoïdale pour la faire devenir positive (voir la figure 3), ce qui permet de ne pas perdre de demi alternance, comme cela se produirait avec une simple diode de redressement.

Le pont de Graetz est donc un double redresseur demi alternance, sa caractéristique est de présenter une tension de sortie toujours de même polarité quelle que soit la polarité de la tension d'entrée. Cela nous permet d'utiliser le montage, même avec une tension continue, sans se soucier de la polarité de la connexion de l'alimentation.

Le pont redresseur de notre circuit est du type discret (diodes indépendantes), réalisé par 4 diodes simples. Ce sont des diodes de type 1N4007, capables de supporter une tension inverse de 1000 V et un courant continu en conduction de 1 A, ce qui est plus que suffisant pour l'utilisation de ce montage.

Réalisation pratique

Passons maintenant à la construction de la veilleuse à LED. Le circuit imprimé est du type simple face et peut être facilement reproductible par photogravure.

Vous pouvez aussi utiliser des bandes autocollantes pour dessiner directement le circuit sur la plaque de cuivre.

Comme toujours, le typon au format Acrobat.pdf à l'échelle 1 est téléchargeable sur notre site www.electroniquemagazine.com, dans le sommaire détaillé de la revue.

Pour le montage des composants, commencez par souder la ou les résistance(s) en fonction de vos besoins. Continuez avec les diodes 1N4007 en les insérant verticalement et en respectant la polarité. Aidez-vous du plan de câblage et de la photo du prototype.

Pour simplifier le montage, le circuit imprimé est conçu de telle manière que toutes les diodes doivent avoir leur bague (cathode) positionnée vers le haut.

Terminez le montage en soudant la LED, son côté biseauté qui représente la cathode doit être positionné vers le bord du circuit imprimé. Vous pouvez utiliser une LED de 3 mm ou de 5 mm, cela dépend de la quantité de lumière dont vous avez besoin.

Concernant **l'utilisation du montage sur le secteur 230 VAC**, nous vous

rappelons que **vous n'êtes pas isolé du secteur et qu'il y a risque d'électrocution**. Vous devez disposer le montage dans un boîtier en plastique sur lequel vous aurez au préalable percé un trou pour la LED et un autre pour la câble d'alimentation. Soudez les 2 fils d'alimentation au points L et N, et à l'aide de colle fixez le circuit imprimé au boîtier.

À propos de l'entrée, notez que les points L et N correspondent au marquage des installations électriques monophasées, mais pour ce montage, le sens n'a pas d'importance. En inversant les connexions, le montage fonctionne sans problème.

Si vous voulez utiliser le montage comme point lumineux, prenez une LED blanche ou jaune claire et disposez le circuit dans un boîtier translucide en plastique afin de laisser passer la lumière.

Si vous devez insérer le circuit dans un boîtier métallique, isolez la partie inférieure du circuit imprimé à l'aide d'une feuille de mica afin qu'aucune soudure ne touche le boîtier. Cela provoquerait des courts-circuits avec des risques d'électrocution, même si vous travaillez en basse tension (12/24 V).

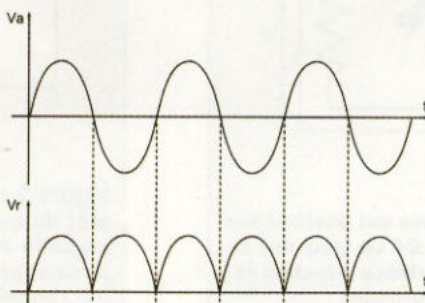


Figure 3 : tension sur la LED (courbe du bas) corrélée avec celle de l'entrée présente aux points L et N du circuit. La tension à l'entrée est sinusoïdale comme vous pouvez le constater. Le pont laisse passer aussi bien les alternances positives que négatives en les redressant avec la même polarité.



Cours de programmation sur iPhone

Sixième partie

de Walter Dal Mut et Francesco Ficili

Dans cette dernière leçon du cours consacré à l'iPhone, nous allons découvrir et apprendre à utiliser l'accéléromètre et le récepteur GPS qui équipent l'iPhone, au moyen de deux exemples d'applications.

Afin de conclure ce cours de programmation pour iPhone, nous allons nous consacrer au développement d'une application utilisant l'accéléromètre.

Nous allons particulièrement étudier la configuration de ce dernier à l'intérieur d'un iPhone et vous proposer un projet pratique d'utilisation.

Un second projet pratique nous permettra d'utiliser un nouveau composant afin d'avoir un retour d'information de type graphique de l'accélération subie par l'iPhone.

Enfin, dans un 3^{ème} projet, nous allons nous servir du récepteur GPS intégré dans le smartphone.

Commençons par l'accéléromètre, qui est fabriqué par « ST Microelectronics » et qui est du type 3 axes.

Il est donc capable de détecter la variation de la vitesse sur un ou plusieurs axes parmi les 3 axes X, Y et Z.

L'accélération maximale qu'il peut supporter sur chacun des axes est de 3 g.

L'accéléromètre est utilisé comme capteur de base pour toutes les applications, car il permet de déterminer la direction d'orientation de l'iPhone et d'ajuster, par exemple, l'affichage.

Rappelez-vous que nous avons parlé des fonctionnalités de rotation automatique de l'affichage (autorotating) et de dimensionnement automatique (autosizing).

L'orientation du dispositif est naturelle, à savoir les coordonnées cartésiennes des 3 axes imaginaires de fonctionnement de l'accéléromètre sont illustrées en figure 1.

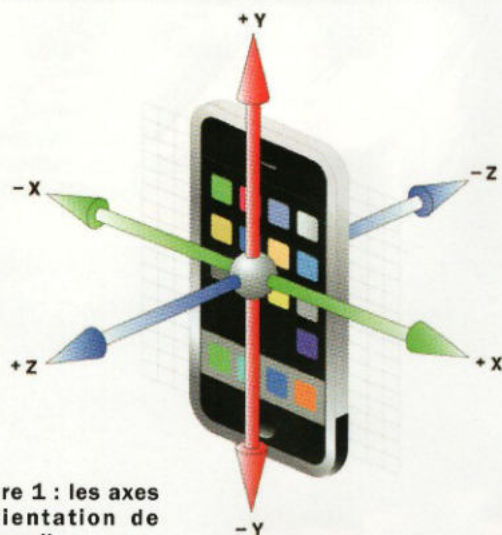


Figure 1 : les axes d'orientation de l'appareil.

Dans la pratique, en regardant l'écran du téléphone avec le bouton « Home » vers le bas, l'axe horizontal est celui de l'axe des **X** (abscisses), l'axe vertical est celui de l'axe des **Y** (ordonnées) et, enfin, l'axe imaginaire pointant vers nous représente la **profondeur**, il s'agit de l'axe des **Z**.

Notez que pour développer des applications utilisant l'accéléromètre, vous devez disposer d'une licence pour développer et exploiter le code sur l'appareil. Le simulateur, malheureusement, ne peut pas exploiter les données d'accélération simulées.

Utilisation de l'accéléromètre sur l'iPhone

L'accéléromètre est modélisé selon un **patron de conception** de type « **Singleton** », c'est-à-dire que nous ne pouvons avoir qu'une seule instance de celui-ci configurée au sein de notre application.

De plus, pour nous informer des données d'accélération, il utilise un protocole dédié semblable à celui que nous avons étudié pour le « **UITableView** » via la méthode de rappel « **cellForRowAtIndexPath** » vue dans les premières leçons. L'utilisation de ce composant, grâce au patron de conception (design pattern), est simple et immédiate.

Nous devons implémenter le protocole du Listing 1 et, activer l'instance singulière comme vous pouvez le voir dans le listing.

Listing 1 : méthode de rappel pour la réception des données brutes.

```
-(void)accelerometer:(UIAccelerometer *)accelerometer
didAccelerate:(UIAcceleration *)acceleration {
//corps du code
}
```

Listing 2 : obtention de l'instance de l'accéléromètre et configuration de la réception.

```
self.accelerometer = [UIAccelerometer sharedAccelerometer];
self.accelerometer.delegate = self;
```

Après ces deux étapes seulement, les données seront collectées à partir du périphérique et notre application sera toujours mise à jour pour la restitution des données d'accélération collectées.

Utilisation de l'accéléromètre

L'objectif de notre premier projet pratique est de créer une application qui peut lire les données brutes de l'accéléromètre et les rendre disponibles en temps réel sur une vue (view) dédiée.

Tout d'abord, nous lançons « Xcode », et nous créons un nouveau projet basé sur une « view » que nous nommons « **EIAccelerometer** ». Nous avons déjà abordé cette étape dans les leçons précédentes.

Une fois cela fait, vous devez obtenir un résultat semblable à celui de la figure 2.

« Xcode » a créé tous les fichiers nécessaires à l'application : les vues, le contrôleur (controller) et l'application déléguée. Commençons par traiter la « View » (vue) en l'insérant dans trois labels et en les attachants au code à travers « **IB** ».

Pour ce faire, nous devons effectuer l'étape déjà vue dans la dernière leçon. Dans le « controller », nous insérons trois objets de type « **UILabel** » et nous les lions à « **IB** » en utilisant le support de « **IBOutlet** » pour cette instance.

N'oublions pas d'effectuer la **synthèse** des « **getters** » et « **setters** » à l'aide de l'instruction « **synthesize** ».

Dans la « View », nous pouvons ajouter les 3 labels qui indiquent à l'utilisateur quel axe se réfère aux données affichées (voir la figure 3). À ce stade, nous sommes prêts à utiliser l'accéléromètre et l'activer pour la réception de données.

Listing 3 : méthode de rappel utilisée par « Cocoa » pour fournir de nouvelles données.

```
-(void)accelerometer:(UIAccelerometer *)accelerometer
didAccelerate:(UIAcceleration *)acceleration {
//Corps de la méthode.
}
```

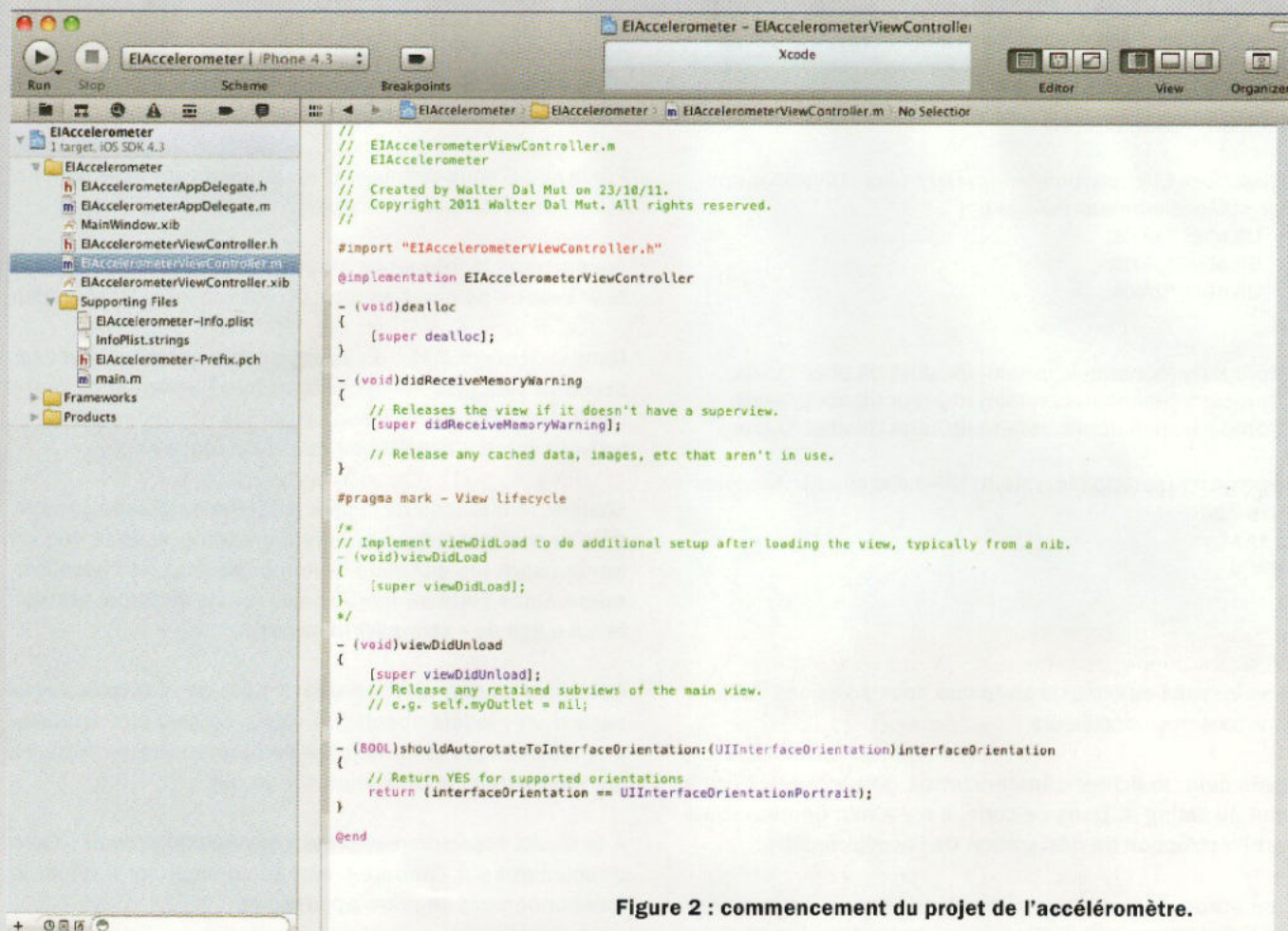



Figure 2 : commencement du projet de l'accéléromètre.

Dans le fichier d'instanciation « EIAccelerometerViewController », nous devons ajouter la déclaration d'utilisation du protocole « UIAccelerometerDelegate », nécessaire pour déclarer la signature de la méthode de réception des données.

Après cette étape, le fichier objet doit déclarer, dans le corps du code, la méthode « accelerometer » comme indiqué dans le listing 3.

À ce stade, nous déclarons l'instance de l'accéléromètre dans le fichier d'instanciation et nous configurons le périphérique lors du chargement de la vue.

Nous déclarons donc un nouvel objet de type « UIAccelerometer » dans le fichier d'instanciation et nous faisons, comme toujours, la synthèse

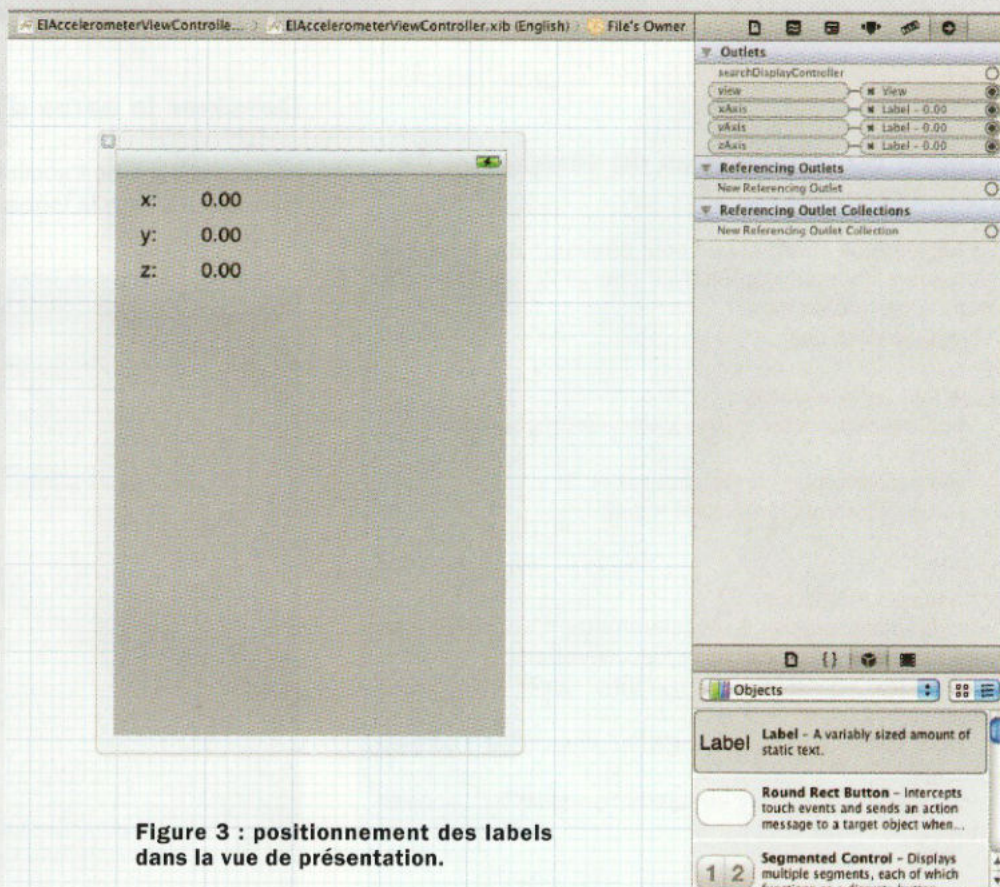


Figure 3 : positionnement des labels dans la vue de présentation.

Listing 4 : fichier d'instanciation pour le contrôleur principal du projet.

```
#import <UIKit/UIKit.h>

@interface ElAccelerometerViewController : UIViewController
<UIAccelerometerDelegate>{
    UILabel *xAxis;
    UILabel *yAxis;
    UILabel *zAxis;
}

@property (nonatomic, retain) IBOutlet UILabel *xAxis;
@property (nonatomic, retain) IBOutlet UILabel *yAxis;
@property (nonatomic, retain) IBOutlet UILabel *zAxis;

@property (nonatomic, retain) UIAccelerometer *accelerometer;

@end
```

dans le corps du code de sorte que nous puissions l'utiliser pour tous nos contrôleurs.

Après cela, le fichier d'instanciation doit être similaire à celui du listing 4. Dans ce code, il n'y a rien de nouveau à part l'instruction de déclaration de l'accéléromètre.

À ce stade, lors du chargement de la vue, nous ne faisons rien d'autre que d'initialiser le composant et, avec la méthode de rappel (callback), nous travaillons avec les labels présents dans la vue (voir le Listing 5).

Listing 5 : initialisation du composant de l'accéléromètre.

```
// Implémente viewDidLoad pour effectuer une configuration supplémentaire après avoir chargé la vue, généralement à partir d'un nib.
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.accelerometer = [UIAccelerometer sharedAccelerometer];
    self.accelerometer.updateInterval = .1;
    self.accelerometer.delegate = self;
}

//... autres méthodes
- (void)accelerometer:(UIAccelerometer *)accelerometer
didAccelerate:(UIAcceleration *)acceleration {
    xAxis.text = [NSString stringWithFormat:@"%f", acceleration.x];
    yAxis.text = [NSString stringWithFormat:@"%f", acceleration.y];
    zAxis.text = [NSString stringWithFormat:@"%f", acceleration.z];
}
```

Dans le listing 5, vous pouvez voir à la 3^{ème} ligne le patron de conception « singleton », lorsqu'il est proposé en utilisant la méthode statique « sharedAccelerometer » de l'instance partagée de l'accéléromètre.

Par la suite, nous définissons la fréquence de mise à jour à 10 échantillons par seconde (0.1 [s]).

Typiquement la fréquence maximale qu'il est possible d'obtenir avec un iPhone 4 est de 100 échantillons par seconde.

Dans la dernière partie du listing, nous avons attribué à l'application déléguée de l'accéléromètre l'instance de notre contrôleur, afin qu'il puisse traiter les nouvelles données arrivant grâce à la méthode de rappel (callback).

Ensuite, la méthode de rappel, appelée automatiquement, gère la configuration des labels présents dans la vue en introduisant les données brutes provenant de l'accéléromètre. Dans cette méthode, nous voyons un appel particulier, il s'agit de « **stringWithFormat** ».

Cette instruction restitue une chaîne de résultats sur la base d'un modèle spécifique. Dans ce cas, elle retourne une chaîne sous la forme d'un nombre en virgule flottante qui représente l'accélération sur un axe.

À ce stade, nous sommes prêts à synthétiser le code source directement sur l'appareil, afin de commencer à tester le fonctionnement de notre application.

En figure 4, vous apercevez notre première application qui utilise l'accéléromètre en tant que composant interne.

Dessiner la barre d'accélération

Pour ce qui est de l'accélération, nous lui associons des informations sous la forme de graphiques qui se déplacent

```
x: -0.093994
y: -0.431000
z: -1.069992
```

Figure 4 : notre premier projet avec la fonction accéléromètre.

Listing 6 : Introduction de la « progress view » pour les accélérations.

```
#import <UIKit/UIKit.h>

@interface EIAccelerometerViewController :
UIViewController <UIAccelerometerDelegate>{
    UILabel *xAxis;
    UILabel *yAxis;
    UILabel *zAxis;

    UIProgressView *xProgressView;
    UIProgressView *yProgressView;
    UIProgressView *zProgressView;
}

@property (nonatomic, retain) IBOutlet UILabel *xAxis;
@property (nonatomic, retain) IBOutlet UILabel *yAxis;
@property (nonatomic, retain) IBOutlet UILabel *zAxis;

@property (nonatomic, retain) IBOutlet UIProgressView
*xProgressView;
@property (nonatomic, retain) IBOutlet UIProgressView
*yProgressView;
@property (nonatomic, retain) IBOutlet UIProgressView
*zProgressView;

@property (nonatomic, retain) UIAccelerometer *acce-
lerometer;

@end
```

horizontalement afin d'indiquer l'accélération instantanée sur un axe.

Dans ce deuxième exercice, nous allons modifier le premier projet en lui insérant 3 barres de progression (progress bar) de type « UIProgressView », qui permettront de représenter graphiquement l'accélération sur les 3 axes.

Tout d'abord, nous devons insérer dans la vue (View) 3 nouveaux composants. Nous devons prendre dans la librairie le composant « Progress View » et l'ajouter 3 fois dans notre « View ».

Ensuite, dans l'instanciation du contrôleur (controller) nous insérons des instructions pour ces 3 nouveaux composants avec la propriété « IBOutlet » pour indiquer que nous les utiliserons dans le code (voir le Listing 6).

Après cela, nous devons connecter les états à l'interface graphique (voir la figure 5). Rappelons-nous aussi que nous devons réaliser la synthèse des méthodes « getters » et « setters » pour ces nouveaux composants, avec l'instruction de synthèse.

À ce stade, il suffit d'entrer le code nécessaire dans la méthode de rappel (callback) pour faire déplacer la barre de progression en fonction des différentes accélérations détectées.

Avant de poursuivre, nous devons considérer que ce composant graphique, généralement utilisé à d'autres fins, ne fonctionne qu'avec des valeurs positives.

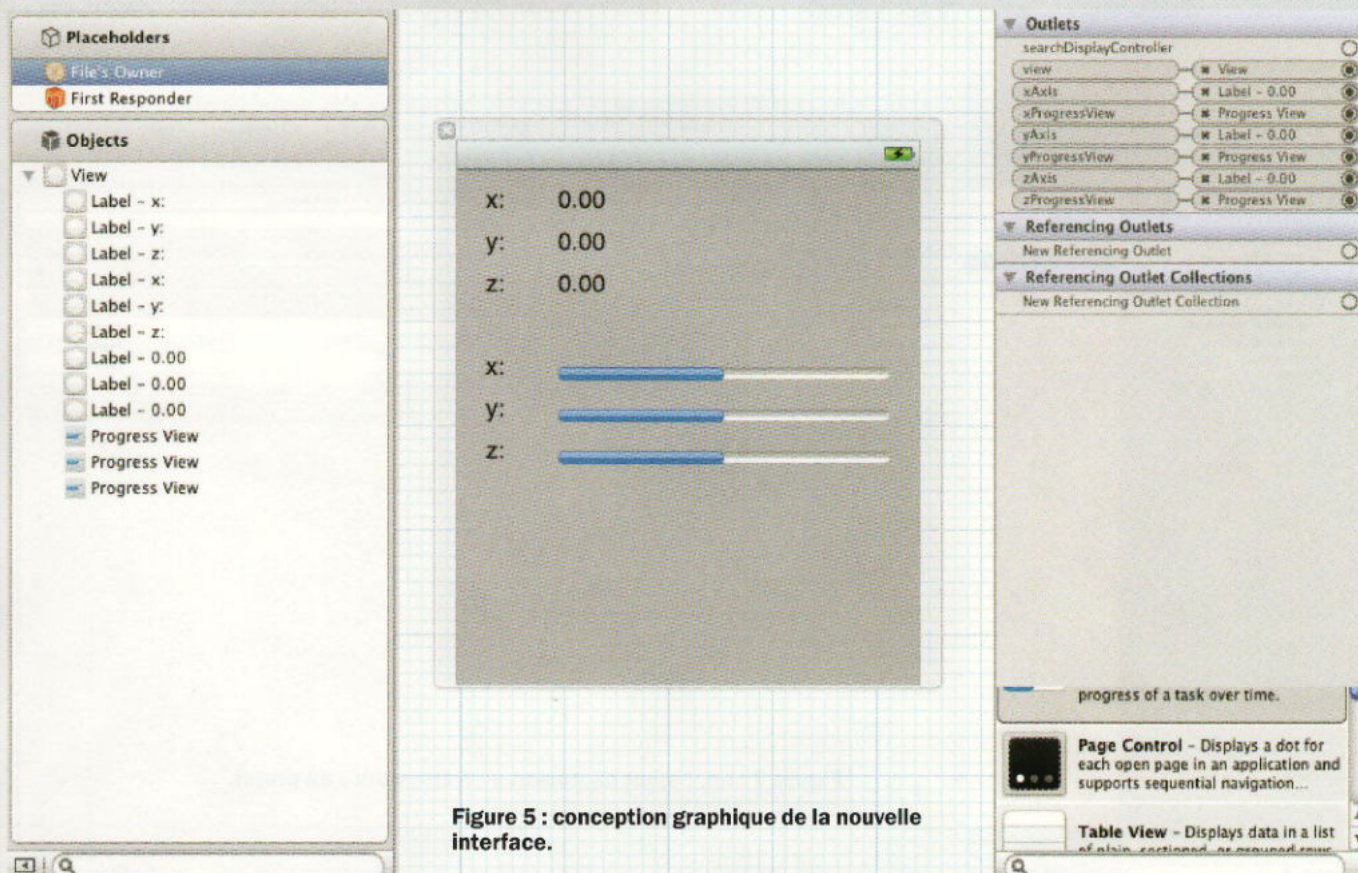


Figure 5 : conception graphique de la nouvelle interface.

Or, nous savons que les accélérations peuvent également avoir des valeurs négatives.

Par conséquent, nous allons effectuer l'opération de « modulo » (valeur absolue) sur les valeurs d'accélération, afin de les rendre compatibles avec l'affichage de la barre de progression.

Le code du listing 7 ajoute à la méthode de rappel (callback) le contrôle à travers le code de la barre de progression.

Comme vous pouvez le voir, il n'y a que 3 lignes qui sont ajoutées. La syntaxe est extrêmement simple.

Listing 7 : méthode de rappel de l'accéléromètre modifiée pour la « ProgressView ».

```
(void)accelerometer:(UIAccelerometer *)accelerometer
didAccelerate:(UIAcceleration *)acceleration {
    xAxis.text = [NSString stringWithFormat:@"%f", acceleration.x];
    yAxis.text = [NSString stringWithFormat:@"%f", acceleration.y];
    zAxis.text = [NSString stringWithFormat:@"%f", acceleration.z];

    self.xProgressView.progress = ABS(acceleration.x);
    self.yProgressView.progress = ABS(acceleration.y);
    self.zProgressView.progress = ABS(acceleration.z);
}
```

x: 0.454987
y: -0.759995
z: -0.498993



Figure 6 : représentation graphique de l'accélération.

Nous accédons à la propriété « progress » de chaque composant et nous leur **assignons la valeur absolue** (modulo), grâce à la fonction « **ABS** », de chaque accélération.

Maintenant, nous pouvons lancer notre nouvelle application. Nous devons apercevoir une information graphique des accélérations présentes sur les 3 axes, comme le montre la figure 6.

Nous constatons que l'accéléromètre est un composant extrêmement simple à configurer et à utiliser, car il n'a pas de structures spécifiques sur lesquelles nous devons agir.

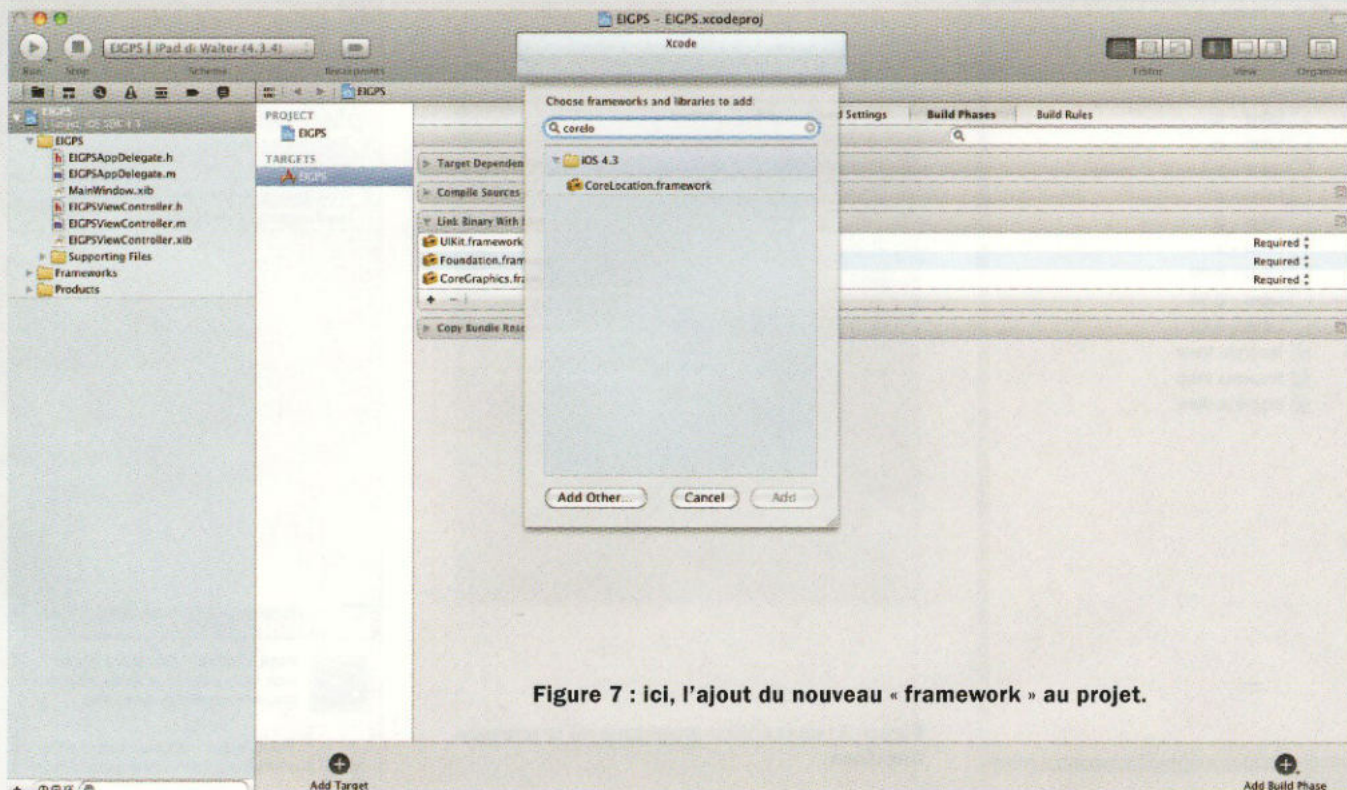


Figure 7 : ici, l'ajout du nouveau « framework » au projet.

Cependant, vous ne devez pas sous-estimer la difficulté d'utiliser les données que l'accéléromètre fournit, car comme cela a été dit précédemment, le terme « brut » signifie qu'avant d'être utilisées, les données doivent être traitées au moyen de filtres numériques.

Introduction au GPS

Un autre composant très intéressant de l'iPhone est le récepteur GPS, qui fournit la latitude, la longitude, l'altitude et la vitesse, en utilisant le « framework » dédié « Cocoa Touch ».

Pour utiliser le GPS, nous devons d'abord ajouter le « framework » « **CoreLocation** » à notre projet. Il est important de souligner que pour exécuter le code sur un périphérique réel, vous devez avoir une licence de développeur.

Donc, ouvrons un nouveau projet basé sur une vue. Ajouter un nouveau « framework » de développement est assez simple. Il suffit de cliquer sur l'ensemble du projet dans la barre de navigation à gauche puis sur « **Targets** ».

Ouvrons l'onglet « **Build Phases** » et cliquons sur l'onglet « **Link Binary With Libraries** ». Une liste de bibliothèques déjà présentes apparaît, cliquons sur le bouton « + » en bas à gauche et ajoutons le « framework » « **CoreLocation** ».

En figure 7, vous apercevez l'ajout du nouveau « framework ».

Réalisons notre application

L'objectif de ce nouveau projet est de créer une application similaire à celle déjà faite avec l'accéléromètre, mais qui affiche les données de latitude et de longitude.

Pour la visualisation des données, nous devons ajouter les labels nécessaires pour les afficher. Nous allons écrire le code à l'aide de « Xcode » en faisant attention aux propriétés « **IBOutlet** », puis nous synthétiserons les méthodes des « getters » et « setters ».

Pour configurer par défaut le nouveau « framework » afin de l'utiliser, nous devons inclure dans le fichier l'instruction « import » afin d'intégrer les informations de la propriété de ce nouveau composant. Dans la ligne de code suivante, nous voyons un exemple d'inclusion du « framework » « **CoreLocation** ».

```
#import <CoreLocation/CoreLocation.h>
```

Bien évidemment, pour ce « framework », il existe plusieurs types d'importations différentes.

Comme pour l'accéléromètre, le récepteur GPS nécessite également une méthode de rappel (callback) pour recevoir les coordonnées de la position. Il peut aussi être activé et désactivé pour réduire la consommation de la batterie.

Listing 8 : fichier d'instanciation pour l'utilisation du GPS intégré.

```
#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

@interface ElGPSViewController : UIViewController <CLLocationManagerDelegate> {
    UILabel *latitudeLabel;
    UILabel *longitudeLabel;

    CLLocationManager *locationManager;
}

@property (nonatomic, retain) IBOutlet UILabel *latitudeLabel;
@property (nonatomic, retain) IBOutlet UILabel *longitudeLabel;

@property (nonatomic, retain) CLLocationManager *locationManager;

@end
```

La méthode de rappel en question est indiquée dans le protocole « **CLLocationManagerDelegate** », il en existe d'autres pour exploiter pleinement le GPS.

D'abord, nous devons déclarer l'instance de « **CLLocationManager** » afin que nous puissions utiliser le contrôleur (controller). Pour cela, nous devons l'insérer dans le fichier d'instanciation, comme indiqué dans le listing 8.

L'activation de ce composant ainsi que la mise en œuvre du contrôleur sont particulièrement simples et rappellent les

Listing 9 : activation et utilisation du GPS.

```
-(void)viewDidLoad
{
    [super viewDidLoad];

    self.locationManager = [[[CLLocationManager alloc] init] autorelease];
    self.locationManager.delegate = self;
    self.locationManager.desiredAccuracy = kCLLocationAccuracyBest;
    [self.locationManager startUpdatingLocation];
}

//... autres

-(void)locationManager:(CLLocationManager*)manager
didUpdateToLocation:(CLLocation*)newLocation fromLocation:(CLLocation*)oldLocation
{
    latitudeLabel.text = [NSString stringWithFormat:@"%f", newLocation.coordinate.latitude];
    longitudeLabel.text = [NSString stringWithFormat:@"%f", newLocation.coordinate.longitude];
}
```


méthodes déjà utilisées pour l'accéléromètre. Cependant, nous devons ajouter une précision.

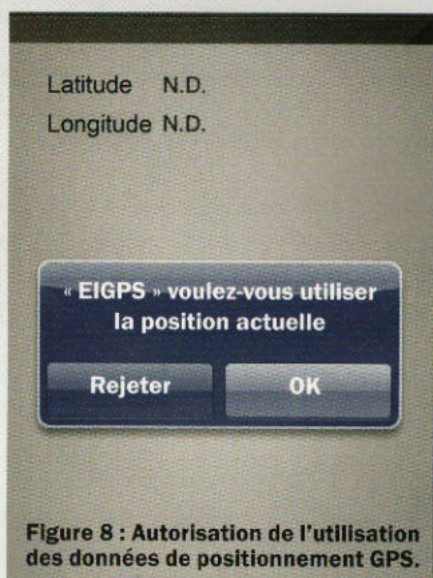
L'allocation du contrôleur est confiée directement au développeur qui utilise le modèle « autorelease » (la libération est effectuée par le SDK lorsque le composant n'est plus nécessaire).

Dans le listing 9, vous apercevez la partie du code qui permet l'**activation** du composant **GPS**.

Dans la méthode « **viewDidLoad** », le composant est alloué et la commande d'activation est envoyée. Successivement, les données entrantes sont mises à disposition par la méthode de rappel (callback) nommée « **didUpdateToLocation** ».

Le « framework » « **CoreLocation** » met à disposition de nombreuses méthodes pour implémenter le composant GPS. Une méthode particulière permettant la gestion des erreurs est intéressante. Il est nécessaire de gérer les erreurs d'une application, car pour pouvoir utiliser le GPS, l'application doit recevoir l'autorisation de l'utilisateur.

Si nous lançons l'application, lors du premier démarrage, nous sommes invités à autoriser l'application (voir la figure 8). Si l'utilisateur refuse l'utilisation du GPS, cela doit être



notifié à l'aide de la méthode « **didFailWithError** ». C'est alors au développeur de gérer ce cas.

Dans ce projet, nous informons l'utilisateur à l'aide d'une fenêtre d'erreur particulière. Pour afficher une fenêtre sur iOS, nous pouvons nous servir du composant « **UIAlertView** », qui met en œuvre toutes les fonctionnalités requises.

Dans le listing 10, vous pouvez voir le code nécessaire pour la **création d'une fenêtre** dédiée contenant un **message d'erreur**. Si nous désactivons l'autorisation d'utilisation du GPS (via les paramètres généraux de l'appareil) et que

Latitude 43.300297
Longitude 5.370113

Figure 9 : la latitude et la longitude de notre position visualisées par notre application.

Listing 10 : gestion des erreurs dans notre application.

```
- (void)locationManager:(CLLocationManager *)manager
didFailWithError:(NSError *)error {
    UIAlertView *errorView = [[UIAlertView alloc] initWithTitle:@"Error" message:[error description] delegate:nil
cancelButtonTitle:@"Fin" otherButtonTitles:nil];
    [errorView show];
    [errorView release];
}
```

nous redémarrons l'application, celle-ci ne pourra pas établir notre position. La méthode de rappel (callback) « **didFailWithError** » du « framework » « **CoreLocation** » sera alors appelée.

Une fois l'autorisation rétablie, l'application affiche la position actuelle via « **didUpdateToLocation** », cela est illustré en figure 9. Lorsque l'un des paramètres du « framework » varie, notre méthode est appelée en nous informant du changement.

Avec cette vue d'ensemble des éléments clés d'iOS, nous concluons la sixième et dernière partie de notre cours de programmation sur iPhone. Au cours de ces leçons, nous avons essayé de vous présenter le système embarqué iOS, en vous montrant les bases de la programmation orientée objet ainsi que des patrons de conception et en vous guidant dans l'écriture de certains programmes. Il vous appartient maintenant d'en faire bon usage. ■

