

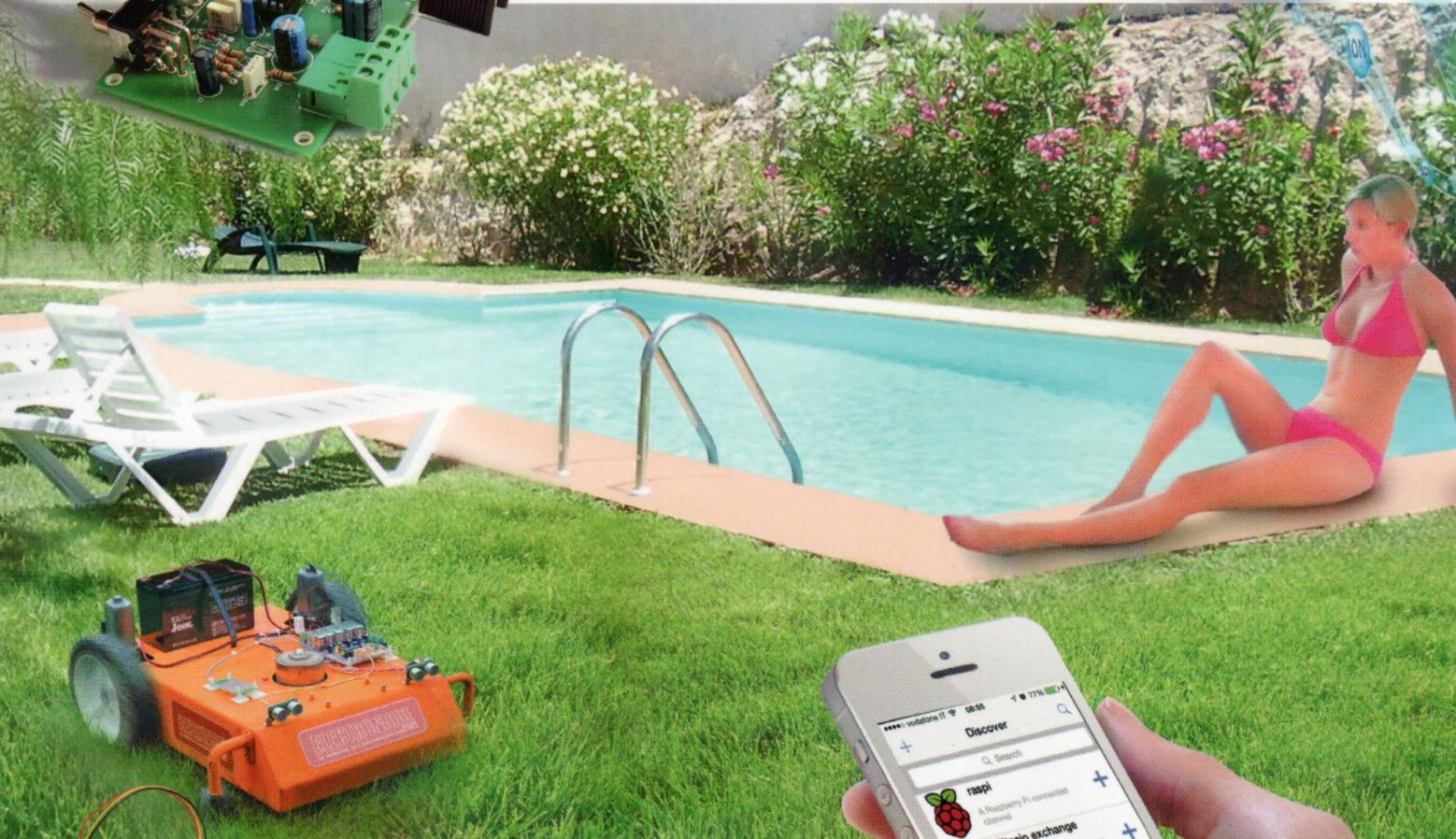
n°139 ÉTÉ 2017

Tondeuse autonome

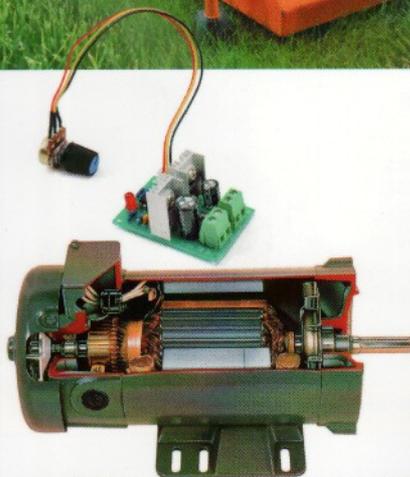
Amplificateur pour guitare



Whiff, purificateur d'air pour la maison



- Notification Push avec Raspberrypi
- Variateur PWM pour moteur DC
- Analyseur de spectre - II
- Imprimante 3D bicolore - V
- Cours Arduino - IX
- Programmez sous iPhone, iPad - V



N° 139 Juin 2017

M 04662 - 139 - F: 8,30 € - RD



Sommaire

ARTICLES

Numéro 139

Été 2017

100 pages



04 MAISON

CONSTRUISONS UNE TONDEUSE AUTONOME

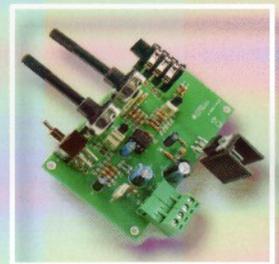
Avec l'arrivée de l'été, la nature s'éveille avec ses lots d'agréments mais aussi de contraintes lorsqu'il s'agit de tondre la pelouse du jardin. Afin de vous éviter cette tâche fastidieuse, nous avons pensé à fabriquer une tondeuse autonome à l'aide de composants de récupérations (c'est le cas des moteurs). Sa particularité réside dans le fait qu'elle est équipée de capteur à ultrasons permettant d'éviter les obstacles, elle est gérée de manière intelligente par un microcontrôleur.



19 AUDIO

AMPLIFICATEUR POUR GUITARE

Cet amplificateur pour guitare dispose d'un réglage de volume, de tonalité et un système de distorsion « fuzz » intégré activable et désactivable si nécessaire. Complété par un haut-parleur ou une mini enceinte et alimenté par une batterie de 12 V, vous pourrez vous exercer partout, à la maison, chez des amis ou bientôt à la fête de la musique. Sa puissance de sortie est compatible avec le voisinage, quelques watts suffisent.



28 INFORMATIQUE

NOTIFICATION PUSH AVEC RASPBERRYPI

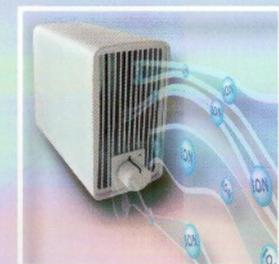
Nous vous proposons dans cet article d'utiliser « Pushetta », qui est un service web permettant d'envoyer des notifications d'événements avec des smartphones sous iOS et Android. Nous allons utiliser la carte d'expérimentation ET1060M pour RaspberryPi afin de simuler des conditions liées à la variation de grandeurs physiques telles que la température, et envoyer des alertes en fonction de seuils prédéfinis.



43 MAISON

WHIFF, LE PURIFICATEUR D'AIR

Cet appareil émet un flux d'air dont l'intensité est réglable, cependant sa particularité est qu'il n'utilise pas des pales, des hélices ou autres ventilateurs. Mais alors comment fonctionne-t-il ? Tout simplement, il produit un flux d'air ionique qui, en plus de souffler l'air, la purifie. « Whiff » se base sur un générateur haute tension qui polarise deux électrodes disposées de façon appropriée et asymétrique



L'EDITO ÉTÉ 2017

Chères lectrices, chers lecteurs,

Avec l'arrivée de l'été, nous avons pensé à ceux d'entre vous qui possèdent un petit coin de jardin qu'il faut malheureusement entretenir. Nous avons donc développé un projet de tondeuse autonome qui peut être réalisé à moindre coût grâce à l'utilisation de composants de récupérations. Grâce à ses capteurs et son micro intelligent, elle vous rendra d'innombrables services.

Vous pourrez ainsi consacrer votre temps à d'autres loisirs, et pourquoi pas, apprendre la guitare grâce à l'amplificateur pour guitare à effet « fuzz », ou encore vous reposer dans votre maison en respirant un air totalement épuré grâce à notre purificateur d'air Whiff. En effet, celui-ci sera d'autant plus efficace en ces périodes chaudes, car il permet de supprimer la pollution à l'ozone de l'air que vous respirez et donc de vous sentir en meilleure forme.

Pour les fans d'informatique, nous proposons de mettre en œuvre un service web d'envoi de notifications vers un smartphone en utilisant le service « Pushetta » et un RaspberryPi.

Enfin, le variateur de vitesse pour moteurs à courant continu ravira les hobbyistes pour la fabrication de maquettes ou de modèles réduits. Il pourra aussi faire varier manuellement la vitesse des ventilateurs d'un PC.

Bien évidemment, nous n'oublions pas les suites des articles du numéro précédent avec la deuxième partie de l'analyseur de spectre, ainsi que les cours Arduino et iPhone.

La Rédaction



53 LABORATOIRE

TRANSFORMEZ VOTRE OSCILLOSCOPE EN ANALYSEUR DE SPECTRE - II

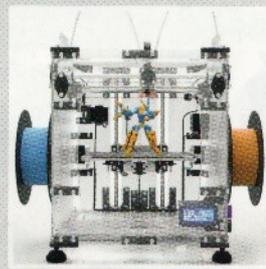
Voici la suite de l'article dont la première partie a été publiée dans la revue N° 138, nous allons aborder la construction et l'utilisation de l'analyseur de spectre. Rappelons qu'il s'agit d'une interface capable de transformer n'importe quel oscilloscope en un analyseur de spectre pour des signaux allant jusqu'à 5 MHz.



61 IMPRIMANTE 3D BICOLORE

L'IMPRIMANTE 3DVERTEX : L'UTILISATION

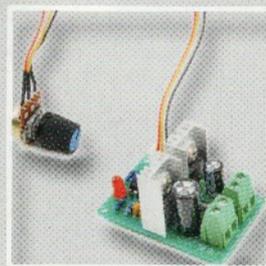
Dans les précédents numéros d'Electronique et Loisirs Magazine nous avons étudié le montage mécanique de l'imprimante 3DVERTEX. Nous allons maintenant aborder dans cet article et les suivants l'utilisation de l'imprimante ainsi que les problèmes techniques que vous pourrez rencontrer lors de la mise en service.



71 ALIMENTATION

VARIATEUR DE VITESSE POUR MOTEUR À COURANT CONTINU

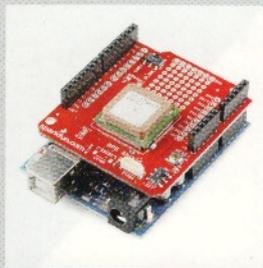
Ce variateur permet de faire varier la vitesse de rotation de n'importe quel moteur électrique à balais de 12 VDC à 35 VDC, à l'aide d'une poignée de composants électroniques dont le célèbre timer NE555. Ce montage convient dans diverses applications, telles que les perceuses et tournevis électriques, mais aussi pour les ventilateurs de PC où il peut être nécessaire de faire varier la vitesse à l'aide d'une commande manuelle afin de réduire le bruit.



77 ARDUINO

COURS ARDUINO - IX

Dans cette neuvième leçon, nous allons vous proposer d'interfacer Arduino avec un récepteur GPS afin de réaliser nos premières expériences de localisation de satellites. Pour cela, nous allons utiliser deux cartes d'extension, une dénommée « GPSSHIELD » et l'autre « ET1017M », dont les composants CMS sont déjà soudés. Ainsi il ne vous restera plus qu'à souder quelques composants classiques ainsi que les connecteurs, le tout pour quelques dizaines d'euros.



89 IPHONE

COURS DE PROGRAMMATION SUR IPHONE : CINQUIÈME PARTIE

Dans cette cinquième leçon, nous allons découvrir comment utiliser la « Navigation Bar » pour gérer la navigation d'un contenu, et apprendre à utiliser les « pickers » qui permettent de sélectionner la date, l'heure ou du contenu comme par exemple une image. Nous aborderons un projet pratique basé sur l'utilisation de « TabBar » et de « Navigation Controller ».



Construisons une tondeuse autonome

de Mirco Segatello



Un beau jardin est sûrement un environnement approprié pour une belle maison ou un immeuble de standing. Cependant, la plupart de l'année, mais surtout pendant l'été, les propriétaires de ces résidences sont contraints de couper l'herbe pour des raisons esthétiques mais aussi de sécurité (en cas d'incendie).

Ainsi chaque week-end, le samedi ou le dimanche, juste au moment où vous souhaitez vous détendre après une semaine de travail intense, vous devez consacrer une partie de votre temps à couper l'herbe à l'aide d'une tondeuse électrique ou à combustion.

Conscients de cette tâche fastidieuse et répétitive, nous avons pensé qu'il serait judicieux de trouver un remède à cette contrainte. La solution évidente est de faire appel à un jardinier, mais cela n'est pas très économique. Nous avons donc développé une **tondeuse autonome**, équipée d'une « intelligence » permettant de couper l'herbe.

Il existe sur le marché des modèles de tondeuses automatiques, mais leur prix est extrêmement élevé. Il nous est donc apparu qu'il était rentable de fabriquer un modèle à partir de pièces de récupération et d'un peu d'électronique. Une tondeuse automatique est relativement facile à mettre en œuvre.



Avec l'arrivée des beaux jours, la nature s'éveille avec ses lots d'agréments mais aussi de contraintes lorsqu'il s'agit de tondre la pelouse du jardin. Afin de vous éviter cette tâche fastidieuse, nous avons pensé à fabriquer une tondeuse autonome à l'aide de composants de récupérations (c'est le cas des moteurs). Sa particularité réside dans le fait qu'elle est équipée de capteur à ultrasons permettant d'éviter les obstacles, le tout étant géré de manière intelligente par un microcontrôleur qui s'adapte à toutes les formes et tailles de pelouses ou jardins. La tondeuse se déplace de façon aléatoire et non répétitive et coupe l'herbe à une hauteur prédéfinie.

Il s'agit d'un petit châssis disposant d'une batterie et de moteurs électriques, le tout géré de manière intelligente par une carte électronique afin d'exécuter les mouvements qu'un jardinier ferait manuellement. Évidemment, il y a des limites à une telle approche du problème, en premier lieu le fait que l'herbe coupée n'est pas ramassée comme le fait un jardinier. En second lieu, afin de maintenir une autonomie de travail importante pour une coupe optimale de l'herbe, la hauteur de celle-ci ne doit pas être trop élevée. Comme les dimensions de la tondeuse sont petites par rapport à celles d'un jardin, la coupe d'une pelouse entière nécessite au moins une journée.

La plupart des produits commerciaux doivent être rechargés sur une base reliée à une prise. Dans notre projet, afin de simplifier les choses, nous avons prévu une prise sur laquelle nous connectons un chargeur de batterie. Compte tenu de ces limites, il faut faire fonctionner la tondeuse en continu (tous les jours) de manière à ce qu'elle coupe que quelques millimètres d'herbe. L'herbe ainsi coupée est de très petite taille et n'a pas besoin d'être ramassée, elle se décompose dans le jardin et aide à la fertilisation de la pelouse restante. D'autres limitations sont dues à la forme du jardin qui doit être bien nivelé, il ne doit pas y avoir de grands trous où la tondeuse pourrait rester bloquée.

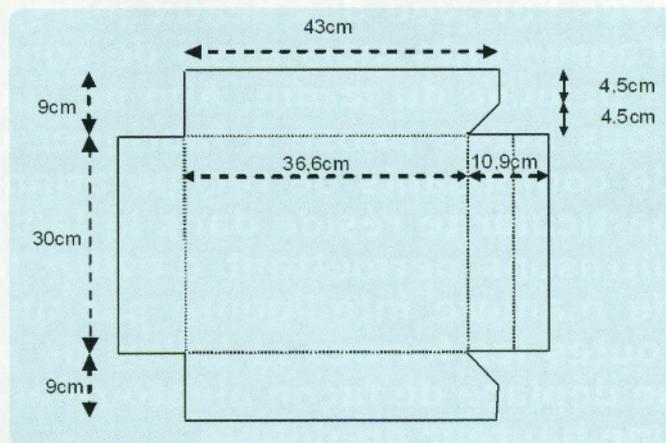
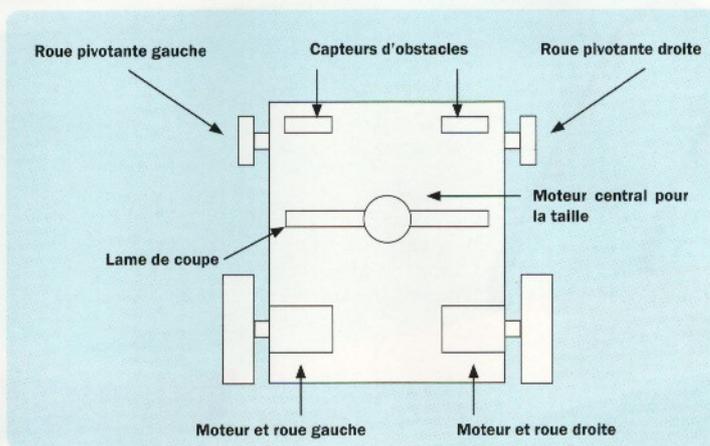
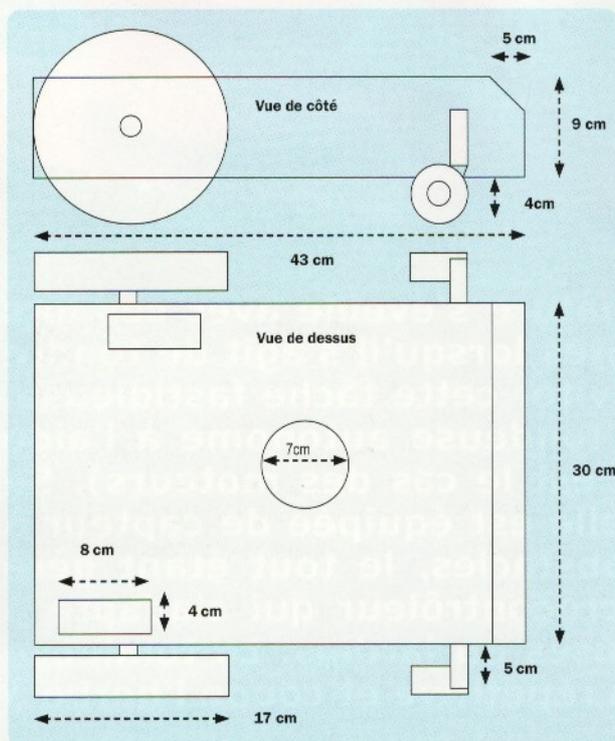
Normalement les plantes, les clôtures et les fleurs ne posent pas de gros problèmes car les capteurs embarqués peuvent les identifier et contourner les obstacles, c'est ce que nous allons expliquer dans cet article.

Le prix d'une tondeuse autonome du commerce n'est pas à la portée de tous, il ne serait pas logique de faire une comparaison avec une tondeuse classique ayant des caractéristiques similaires. Le coût final est justifié par la nouveauté du produit, par le haut niveau de technologie

Caractéristiques techniques :

- Alimentation : batterie au plomb 12 V 12 Ah :
- Diamètre de la taille : 25 cm :
- Radars à ultrasons pour la détection d'obstacles :
- Mesure de la température du moteur avec système de sécurité :
- Mesure de la tension de la batterie :
- Détection de la luminosité ambiante :
- Détecteur de pluie avec verrouillage de sécurité :
- Indicateur de batterie faible.

Le châssis de la tondeuse a été réalisé avec de la tôle d'acier de 1 mm d'épaisseur, convenablement pliée et soudée. Les dessins illustrent la position des différents éléments mécaniques utilisés et les dimensions des trous et des découpes à réaliser. Eventuellement, vous devrez apporter des modifications dans le cas où vous utilisez des pièces mécaniques différentes.



et l'utilisation de pièces chères telles que les moteurs et la batterie.

En fin de compte, effectuer une opération très simple qui consiste à couper de l'herbe, mais avec de la technologie comme la détection d'obstacle, la détermination de la trajectoire pour une coupe optimale, la détection des conditions environnementales, n'est pas aussi simple qu'il n'y paraît.

Le projet présenté dans ces pages ne vise pas à atteindre la performance

d'une tondeuse autonome commerciale, mais décrit plutôt un système simplifié et donc facile à réaliser par le grand public.

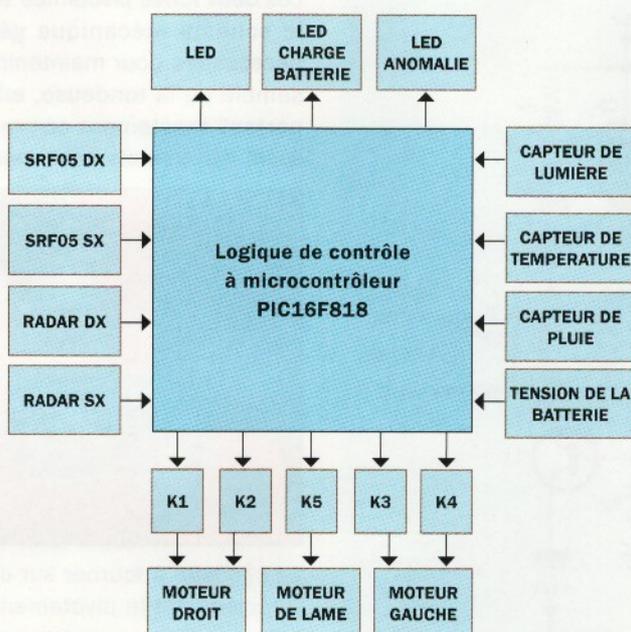
En d'autres termes, notre tondeuse automatique veut juste être un point de départ pour ceux qui se lancent dans la robotique, car elle nécessite des connaissances non seulement en électronique mais aussi en mécanique.

Dans ce projet, la mécanique est certainement l'élément principal de la tondeuse, l'électronique comme nous le

verrons plus loin ne présente pas de difficultés particulières. Il y aura une grande satisfaction dans la réalisation d'un tel projet, qui probablement attirera les regards des voisins et de ceux qui passeront à proximité de votre résidence.

La mécanique

Commençons par la description de la partie mécanique, celle présentée dans ces pages est une solution possible au problème.



La vitesse d'avancement doit être d'environ 50 cm par seconde, il est entendu que le moteur doit tourner très lentement ou devra être équipé d'un réducteur de vitesse. Vous pouvez récupérer des moteurs d'essuie-glace de véhicule ou de vitres électriques dans une casse auto.

Par contre, le moteur qui coupe l'herbe doit tourner très vite en continu. Vous pouvez utiliser un moteur de ventilateur de radiateur de liquide de refroidissement d'un véhicule, toujours dans une casse auto.

Une fois les pièces les plus importantes trouvées, nous pouvons nous consacrer à la réalisation de la carrosserie de manière à ce que les moteurs et les roues soient correctement installés pour que la lame puisse tourner librement par l'intermédiaire du moteur placé en position centrale.

Voyons comment fixer les roues aux moteurs respectifs, ainsi que la lame au moteur central.

Synoptique de l'électronique de commande utilisée dans notre tondeuse autonome. Le dispositif prévoit l'utilisation de capteurs à ultrasons de type commercial ou de fabrication maison. Le contrôle des 3 moteurs (2 pour la traction et 1 pour la coupe) est réalisé au moyen de relais de puissance. Cette solution permet d'utiliser des moteurs de tout type (voire de récupération).

Cependant vous pouvez effectuer des modifications en fonction de vos besoins. Le système de commande électronique, qui sera décrit plus loin, peut être utilisé dans de nombreuses autres applications similaires.

Dans la forme la plus élémentaire, notre tondeuse se compose des éléments suivants :

- un moteur de 12 V à courant continu pour entraîner le côté droit ;
- un moteur de 12 V à courant continu pour entraîner le côté gauche ;
- un moteur central 12 V à courant continu pour couper l'herbe ;
- une batterie au plomb de 12 V pour l'alimentation électrique ;
- deux capteurs à ultrasons pour détecter les obstacles ;
- deux roues directrices pour diriger la tondeuse.

Ces éléments sont assemblés selon les dessins de la page précédente. Le choix du type de moteur et des roues de traction est essentiel, car ce sont les éléments les plus critiques du projet. Les roues peuvent être fabriquées par vos

soins ou vous pouvez les acheter neuves dans le commerce ou encore récupérer des roues d'une tondeuse manuelle en panne.

Les roues doivent avoir typiquement un diamètre de 15 à 20 cm.

Les moteurs doivent avoir une puissance suffisante pour être en mesure de déplacer l'ensemble de la tondeuse sur un gazon qui, habituellement, n'est pas une surface parfaitement lisse. Il est nécessaire que le **moteur développe une puissance d'au moins 3 à 4 W/kg.**



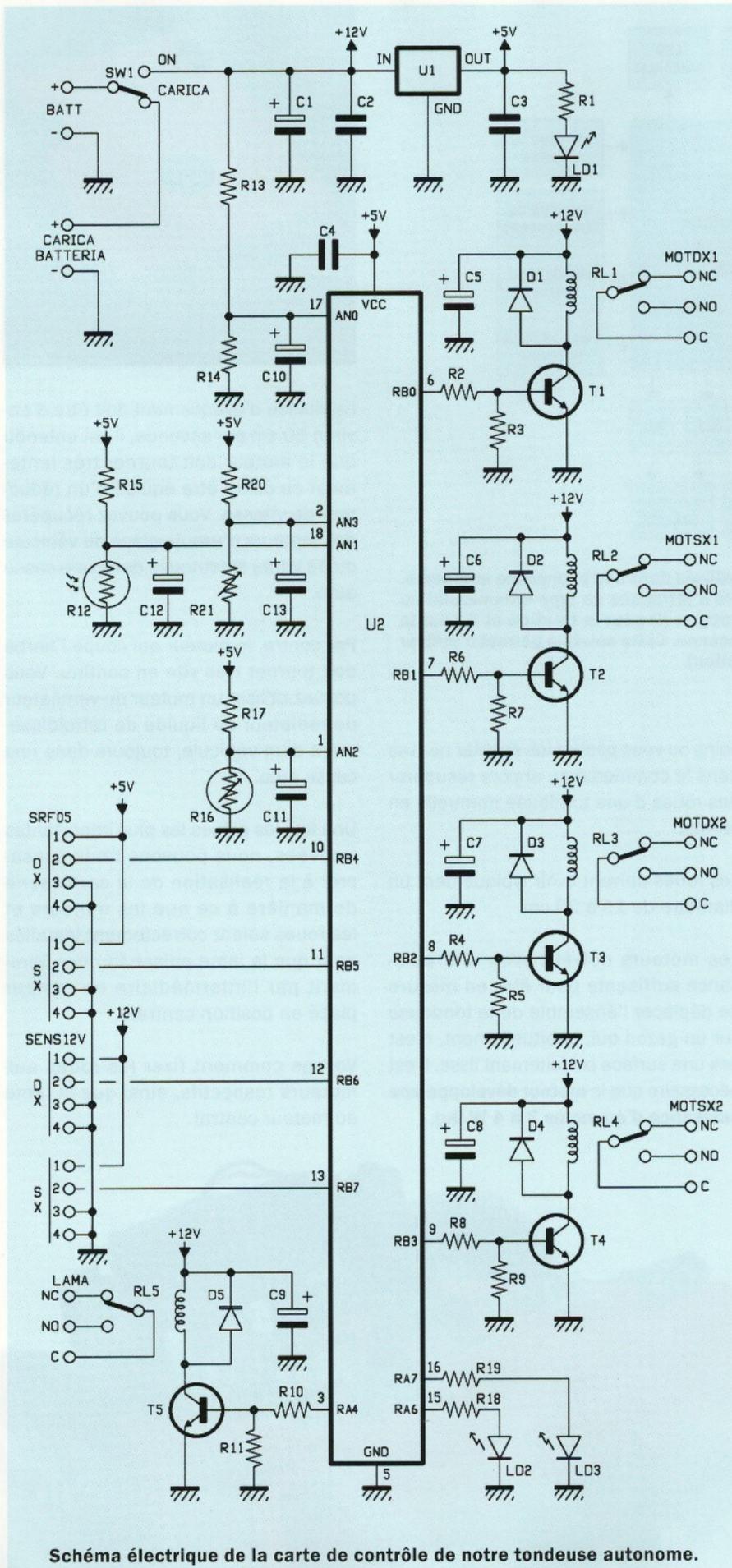
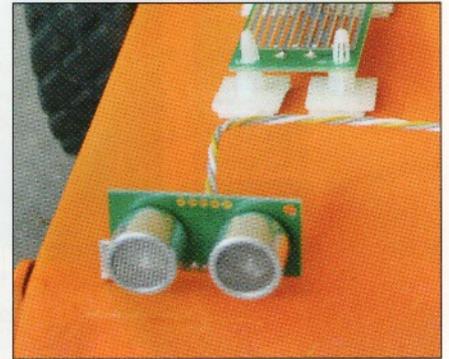


Schéma électrique de la carte de contrôle de notre tondeuse autonome.

Les deux roues pivotantes visibles dans le schéma mécanique général sont nécessaires pour maintenir à plat l'ensemble de la tondeuse, elles se comportent exactement comme les roues avant des chariots des supermarchés.



La capacité à tourner sur elles-mêmes (précisément le pivotement) permet à la tondeuse, avec seulement deux roues directrices, de se mouvoir dans toutes les directions et de faire demi-tour sur place.

Ensuite vous devrez fixer l'ensemble du châssis selon les plans décrits dans les pages précédentes, puis appliquer une couche de peinture vive qui améliorera la visibilité de la tondeuse dans le jardin.

Nous avons utilisé pour le moteur central de notre prototype, un moteur de récupération de ventilateur de refroidissement d'une voiture. Les deux autres moteurs proviennent de mécanismes de récupération de vitres électriques de voiture.

Avec cette configuration, la vitesse d'avancement est trop rapide. Il est nécessaire d'insérer deux résistances en série avec les moteurs, de manière à provoquer une chute de tension et réduire ainsi la vitesse de rotation (donc l'avancement).

Le châssis a été réalisé avec une tôle d'acier, d'épaisseur 1 millimètre, convenablement pliée et soudée. Les roues pivotantes ont été fixées au châssis à l'aide d'un support en forme de « L » afin de permettre une rotation sur 360°. Une belle peinture de couleur orange vif permet de voir facilement la tondeuse sur la pelouse verte.

La batterie a été fixée au moyen de gros colliers isrlans, tandis que pour tous les autres composants nous avons percé des trous sur le châssis afin de les fixer solidement à l'aide de vis et d'écrous.

Le radar à ultrasons SRF05

Comme mentionné précédemment, la tondeuse est conçue pour fonctionner soit avec des capteurs à ultrasons du commerce soit de fabrication maison. Vous pouvez utiliser le capteur « **SRF05** » commercialisé par différentes sociétés, il coûte une vingtaine d'euros mais il a l'avantage d'être déjà monté.

Ce capteur peut être utilisé uniquement en combinaison avec un microcontrôleur, car il nécessite des signaux de contrôle. Il est en mesure de détecter avec précision un objet placé devant lui à une distance comprise entre 1 cm et 4 mètres.

Le principe de fonctionnement sur lequel il repose est très simple et est très proche du système de navigation utilisé par les chauves-souris pour voler. Le principe consiste à générer un signal sonore ultrasonique et détecter les échos provoqués par les obstacles dans son voisinage. Le temps écoulé entre le moment où le signal est envoyé et la réception de son écho permet de déterminer la distance avec l'obstacle, sachant que dans l'air un signal sonore se déplace à la vitesse de 330 m/s.

Le capteur envoie un signal à partir de sa capsule à ultrasons et commute sa sortie lorsqu'il capture le signal d'écho en fonction des diagrammes de synchronisation visibles en figure 2. Lorsqu'une impulsion de 10 μ s est présente sur son entrée de déclenchement, le SRF05 génère 8 cycles d'ultrasons à 40 khz et met sa broche « Echo output » à un niveau logique 1. Lorsqu'il détecte un écho, il abaisse sa ligne d'écho à un niveau logique 0. La ligne d'écho est donc une impulsion dont la largeur est proportionnelle à la distance avec l'objet. Si rien n'est détecté, le SRF05 abaissera sa ligne d'écho après environ 30 ms. Si la largeur de l'impulsion est mesurée en « μ s », alors une division par 58 donnera la distance en centimètres. Le SRF05 peut être déclenché toutes les 50 ms (soit 20 fois par seconde), il est nécessaire d'attendre 50 ms entre 2 déclenchements afin de s'assurer que le signal précédent n'interfère pas avec le suivant.

Pour résumer, le microcontrôleur envoie le signal de déclenchement au capteur, celui-ci démarre la procédure de génération du signal ultrasonique. En fonction des informations reçues, le microcontrôleur mesure la durée du signal de sortie provenant du détecteur qui est proportionnelle à la distance avec l'obstacle.

Il est possible d'utiliser une seule ligne du microcontrôleur pour la génération du signal de déclenchement (trigger) et la mesure de l'impulsion provenant du capteur, par simple mise à la masse de la broche « Mode ».

En utilisant le compilateur PICBASIC pour PIC, il suffit de seulement 4 lignes de code pour gérer le capteur :

Figure 1

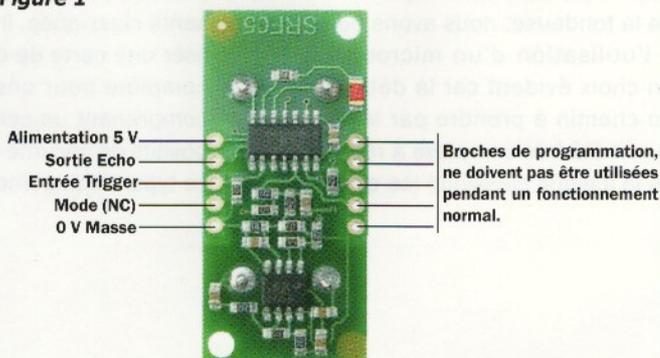


Figure 2

SRF05 Diagrammes de synchronisation, Mode 1

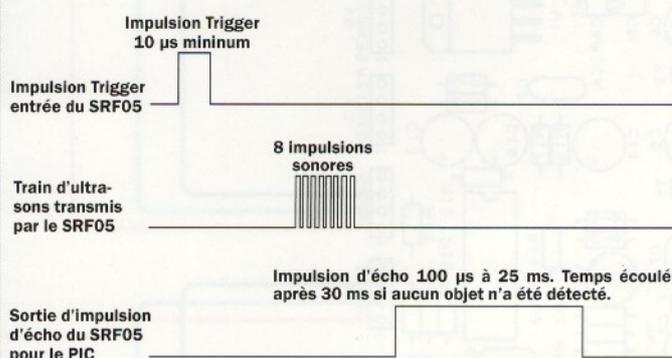
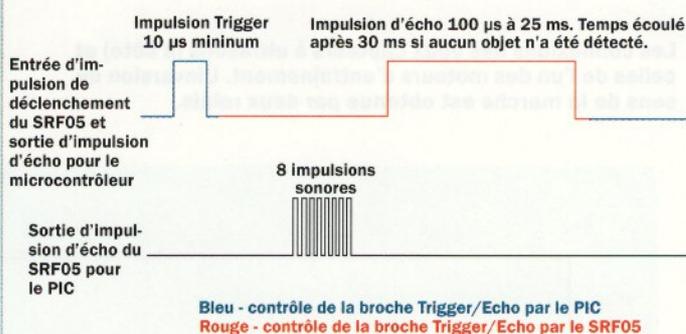


Figure 3

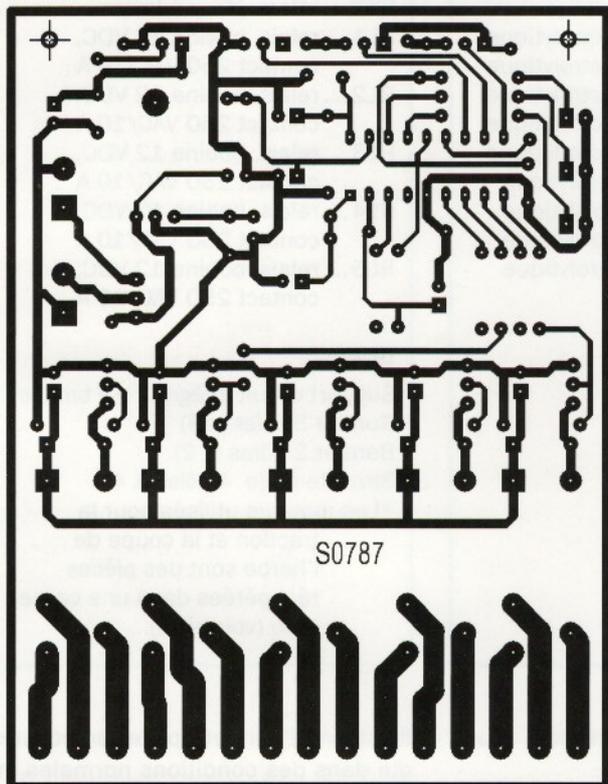
SRF05 Diagrammes de synchronisation, Mode 2



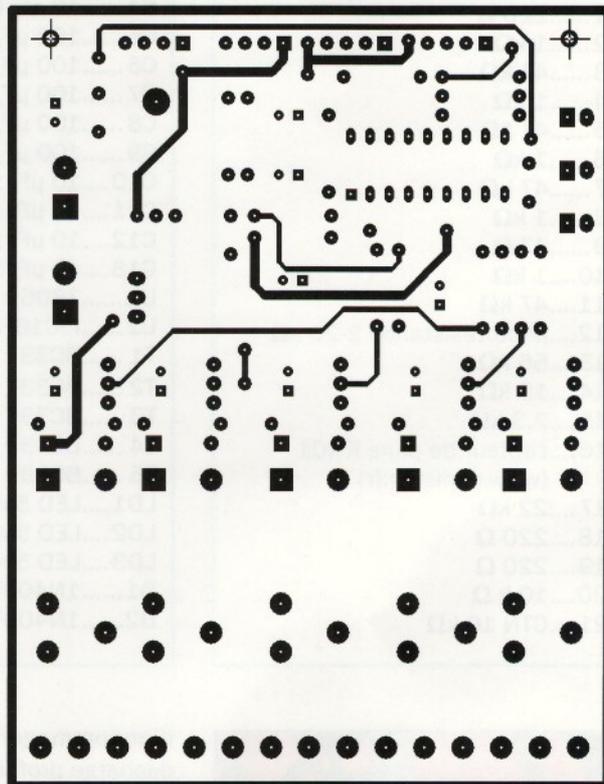
symbol S1 = PORTB.4 ' broche pour le capteur SRF05
time VAR WORD
PULSOUT S1, 2 ' 20 μ s impulsion de départ (start)
PULSIN S1, 1, time ' lit le retard de l'écho

Avec une horloge interne de 4MHz, la valeur de la distance est égale à la variable « time » divisée par 5, le résultat est en centimètres.

Plan de montage de la carte de contrôle de la tondeuse



Circuit imprimé de la carte de contrôle à l'échelle 1 : 1 côté soudures.



Circuit imprimé de la carte de contrôle à l'échelle 1 : 1 côté composants.

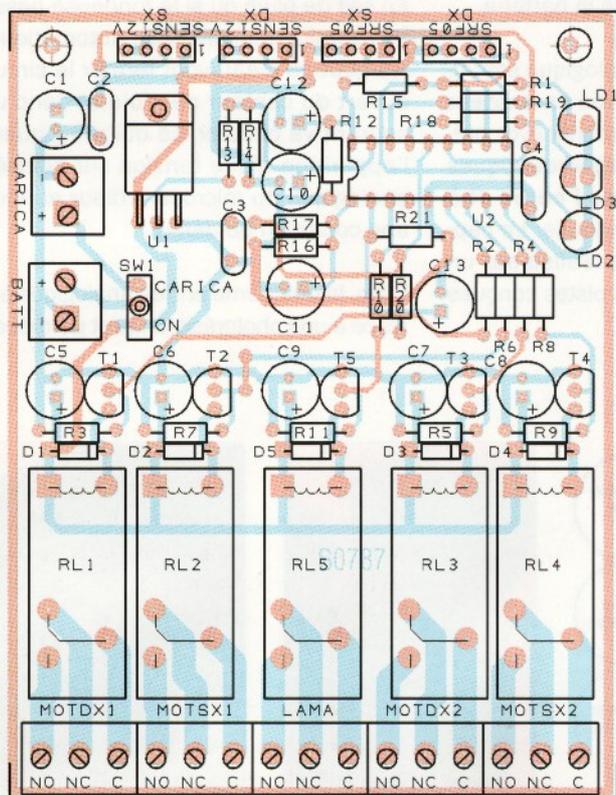


Schéma de câblage des composants de la carte de contrôle de la tondeuse autonome.

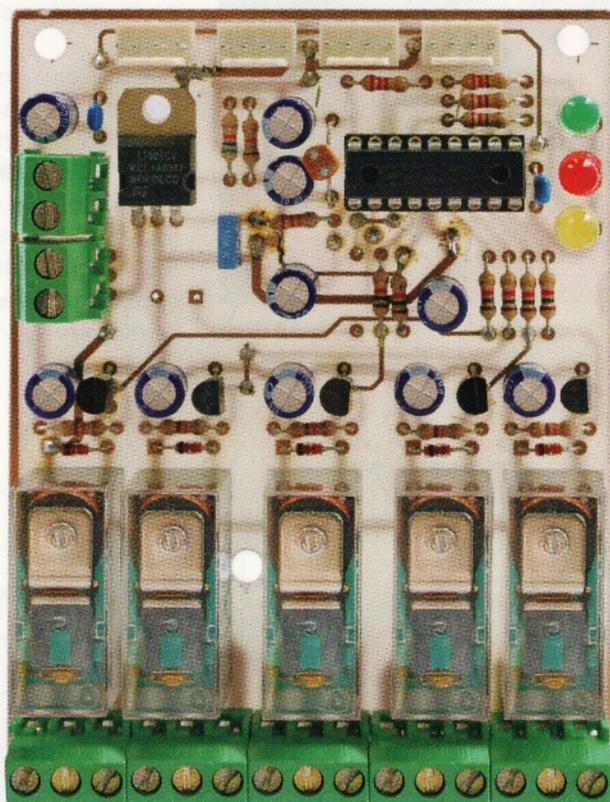


Photo de l'un de nos prototypes de la carte de contrôle.

Liste des composants de la carte de contrôle

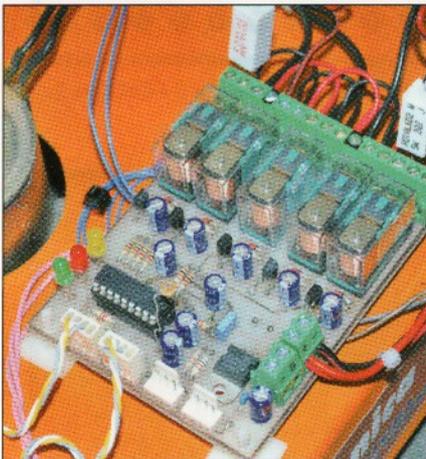
R1.....220 Ω
 R2.....1 kΩ
 R3.....47 kΩ
 R4.....1 kΩ
 R5.....47 kΩ
 R6.....1 kΩ
 R7.....47 kΩ
 R8.....1 kΩ
 R9.....47 Ω
 R10.....1 kΩ
 R11.....47 kΩ
 R12....photorésistance 2-20 kΩ
 R13.....56 kΩ
 R14.....18 kΩ
 R15....2,2 kΩ
 R16....capteur de pluie RN01
 (www.comelec.fr)
 R17....22 kΩ
 R18....220 Ω
 R19....220 Ω
 R20....10 kΩ
 R21....CTN 10 kΩ

C1100 μF/25 V électrolytique
 C2100 nF multicouche
 C3100 nF multicouche
 C410 nF/63 V polyester
 C5100 μF/25 V électrolytique
 C6100 μF/25 V électrolytique
 C7100 μF/25 V électrolytique
 C8100 μF/25 V électrolytique
 C9100 μF/25 V électrolytique
 C1010 μF/16 V électrolytique
 C1110 μF/16 V électrolytique
 C1210 μF/16 V électrolytique
 C1310 μF/16 V électrolytique
 U1.....7805
 U2.....PIC16F819
 T1BC337
 T2BC337
 T3BC337
 T4BC337
 T5BC337
 LD1LED 5 mm verte
 LD2LED 5 mm jaune
 LD3LED 5 mm rouge
 D1.....1N4007
 D2.....1N4007

D31N4007
 D4.....1N4007
 D5.....1N4007
 SW1...interrupteur à levier
 RL1....relais, bobine 12 VDC,
 contact 250 VAC/10 A
 RL2....relais, bobine 12 VDC,
 contact 250 VAC/10 A
 RL3....relais, bobine 12 VDC,
 contact 250 VAC/10 A
 RL4....relais, bobine 12 VDC,
 contact 250 VAC/10 A
 RL5....relais, bobine 12 VDC,
 contact 250 VAC/10 A

Divers

Support circuit intégré 2 x 9 broches
 Bornier 3 pôles (x 5)
 Bornier 2 pôles (x 2)
 Barrette mâle 4 pôles (x 4)
 *Les moteurs utilisés pour la traction et la coupe de l'herbe sont des pièces récupérées dans une casse auto (voir texte).



d'endommager la batterie lors d'une décharge profonde.

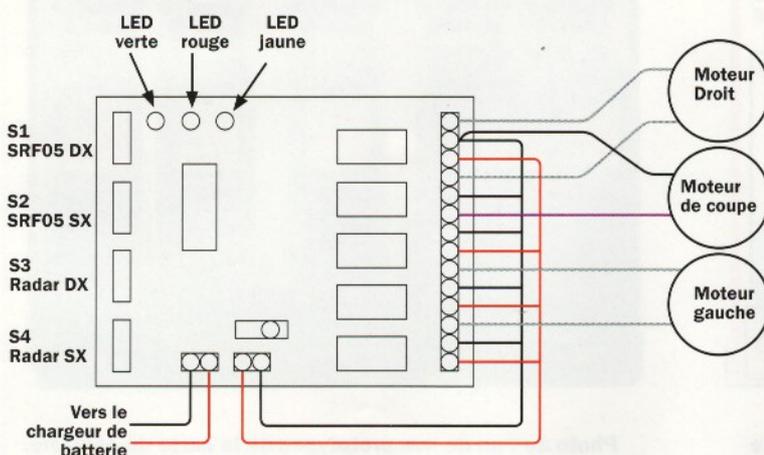
Dans ce cas aussi, **l'alarme est remise à zéro en éteignant et rallumant la carte**, après avoir chargé la batterie.

Une troisième entrée analogique est utilisée en combinaison avec un **capteur de pluie**, elle sert à déterminer avec précision les **conditions météorologiques impropres à la coupe de l'herbe** et à l'utilisation de notre tondeuse. Le capteur de pluie utilisé est réalisé sur un circuit imprimé avec des pistes conçues de manière appropriée.

Il est formé par deux pistes adjacentes, qui dans des conditions normales, ne sont pas en contact et donc se comportent comme un circuit ouvert.

En cas de pluie ou si la tondeuse passe à proximité d'un arrosage automatique, les gouttes d'eau vont fermer le circuit ouvert du capteur de pluie et donc un courant va circuler, ce qui provoquera l'apparition d'une tension sur l'entrée analogique du microcontrôleur et donc une condition de défaut.

Enfin, la quatrième entrée analogique est reliée à une photorésistance et permet de



mesurer le niveau de luminosité externe. Pour le moment cette information n'est pas utilisée, mais pourrait être une évolution intéressante du projet, par exemple avec l'utilisation d'un panneau solaire qui permet de recharger la batterie.

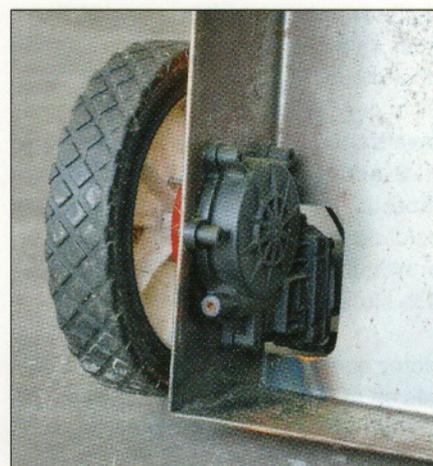
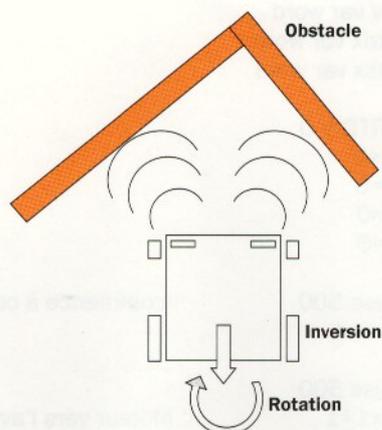
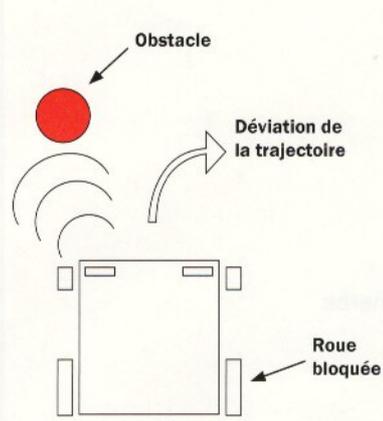
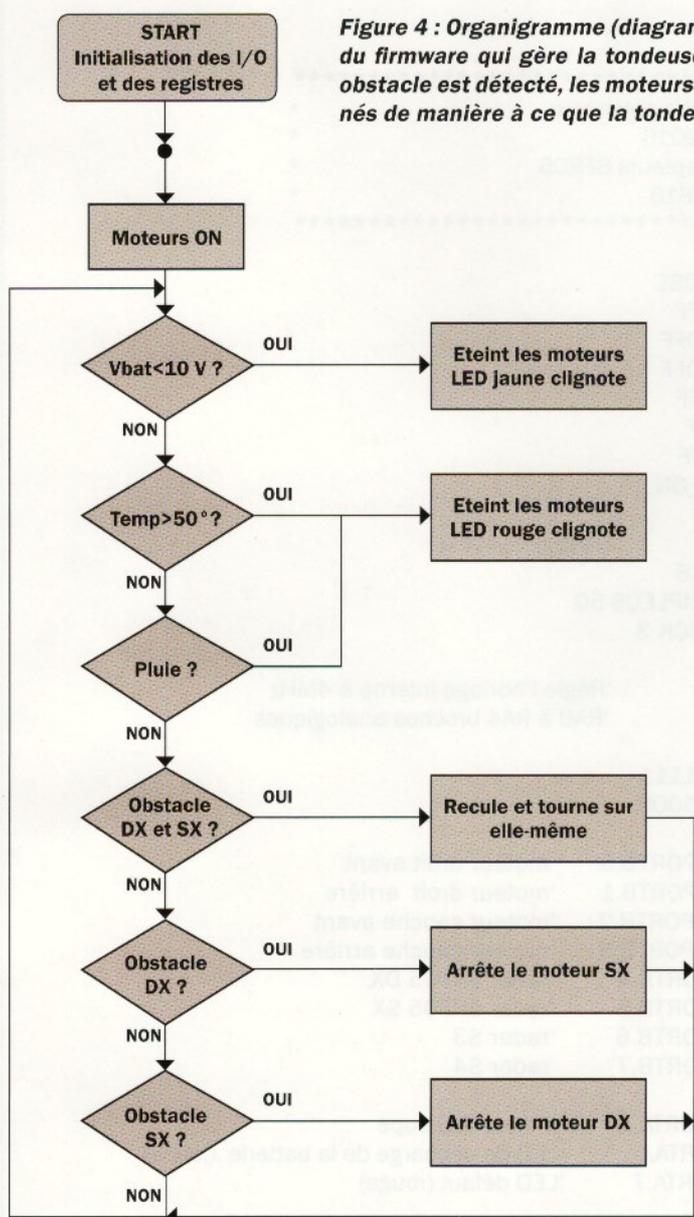
Dans ce cas, la photorésistance pourrait déterminer l'emplacement le plus approprié pour la recharge de la batterie (en d'autres termes l'endroit le plus ensoleillé du jardin). Vous pourrez l'adapter selon vos besoins, vous trouverez plus loin le code source du programme en BASIC, facilement modifiable.

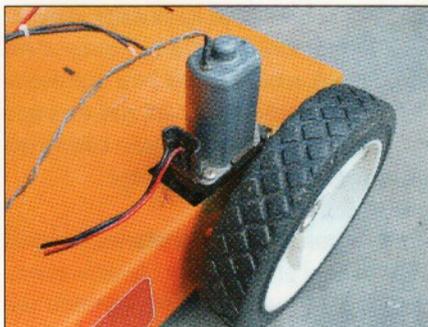
En plus des 4 entrées analogiques, 4 entrées digitales sont utilisées pour la connexion des capteurs d'obstacles à ultrasons. Chaque connecteur comprend 4 contacts, deux pour la masse, un pour l'alimentation et un pour le signal d'entrée provenant du capteur. Pour détecter les obstacles, vous pouvez **utiliser soit des capteurs ultrasoniques commerciaux** ou de **construction maison**. Les deux premiers connecteurs **SX** et **DX** du **SRF05** fournissent une tension stabilisée de **5 V** et sont donc susceptibles d'être reliés au capteur **SRF05**.

Les deux autres connecteurs **SX** et **DX** de « **SENS 12 V** » fournissent une **tension de 12 V** qui est plus appropriée pour des capteurs de type « **fabrication maison** ».

La différence entre les deux solutions réside dans le fait que la version commerciale, en plus d'être déjà assemblée, semble être plus sensible et donc optimale dans toutes les situations très critiques que l'on rencontre dans les jardins (présence d'arbustes ou de clôtures qui sont difficilement détectables).

Figure 4 : Organigramme (diagramme de flux) du firmware qui gère la tondeuse. Quand un obstacle est détecté, les moteurs sont actionnés de manière à ce que la tondeuse l'évite.





Deux sorties du PIC sont reliées à 2 LED, une jaune l'autre rouge, qui signalent les défauts ; 5 sorties sont utilisées pour commander les relais via un étage amplificateur à transistor. Le relais **RL4** est utilisé pour alimenter le **moteur de coupe** sur lequel est fixé la lame par l'intermédiaire de son contact « NO ».

Les 4 autres relais sont utilisés pour commander les deux moteurs des roues, à la fois dans le sens avant et arrière de manière indépendante pour chaque moteur. Les contacts des relais doivent être câblés selon le synoptique visible en page 12.

La configuration particulière des contacts des deux relais affectés à chacun des moteurs permet d'alimenter ces derniers dans un sens ou dans le sens opposé, en fonction du relais activé.

Lorsque les 2 relais sont désactivés, le contact se trouve dans la position « NC » (normalement fermé), aucun courant ne circule et le moteur est arrêté. Si le relais RL1 est activé, le contact C (commun) se positionne sur « NO » portant le contact de gauche de l'enroulement du moteur à 12 V tandis que celui de droite se retrouve à la masse (GND), et donc cela détermine le sens de rotation du moteur.

Inversement, si RL1 est désactivé et RL2 activé, le contact de droite du moteur est porté à 12 V tandis que celui de gauche se retrouve à la masse, provoquant ainsi une rotation opposée à la situation précédente.

Notez que si les deux relais étaient activés en même temps, il n'y aurait aucun dommage et le moteur resterait dans un état stationnaire.

Listing 1

```

*****
'* Name : Tondeuse autonome *
'* Date : 27/03/2016 *
'* Version : 2.0 capteurs SFR05 *
'* Notes : PIC16F818 *
*****

@DEVICEINTRC_OSC
@DEVICEWDT_OFF
@DEVICEPWRTOFF
@DEVICEMCLR_OFF
@DEVICEBOD_OFF
@DEVICELVP_OFF
@DEVICECPD_OFF
@DEVICECCPMX_ON

DEFINE OSC 4 ' horloge à 4MHz
define ADC_BITS 8
DEFINE ADC_SAMPLEUS 50
DEFINE ADC_CLOCK 3

OSCCON = $60 'Règle l'horloge interne à 4MHz
ADCON1 = 2 'RA0 à RA4 broches analogiques

TRISA = %00001111
TRISB = %11110000

symbol msx1 = PORTB.0 'moteur droit avant
symbol msx2 = PORTB.1 'moteur droit arrière
symbol mdx1 = PORTB.2 'moteur gauche avant
symbol mdx2 = PORTB.3 'moteur gauche arrière
symbol S1 = PORTB.4 'radar SRF05 DX
symbol S2 = PORTB.5 'radar SRF05 SX
symbol S3 = PORTB.6 'radar S3
symbol S4 = PORTB.7 'radar S4

symbol lama = PORTA.4 'moteur de coupe
symbol lch = PORTA.6 'LED de décharge de la batterie (jaune)
symbol lan = PORTA.7 'LED défaut (rouge)

lux var byte 'variable luminosité
umi var byte 'variable pluie
Vb var byte 'variable de la tension de la batterie
tem Var byte 'variable température

tp var word
mov var word
distdx var word
distsx var word

PORTB = 0
lan=0
lch=0
S1=0
S2=0

pause 500 ' commence à couper l'herbe
lama=1

pause 500
mdx1=1 ' Moteur vers l'avant
mdx2=0

```

```

msx1=1
msx2=0

MAIN

  adcin 3,tem          'lecture de la température du moteur
  adcin 2,umi          'lecture de l'humidité
  adcin 1,lux          'lecture de la luminosité
  ADCIN 0,Vb          'lecture de la tension de la batterie

  if vb<100 then goto standby 'batterie faible

  if tem<80 then goto securite 'surchauffe

  if umi<80 then goto securite 'pluie

  pause 500

  'Détection d'obstacle DX
  PULSOUT S1, 2          '20 µs impulsion de start
  PULSIN S1, 1, distdx  'lit la distance avec l'obstacle DX

  'Détection d'obstacle SX
  PULSOUT S2, 2          '20 µs impulsion de start
  PULSIN S2, 1, distsx  'lit la distance avec l'obstacle SX

  if distdx<300 and distsx<300 then 'Obstacle devant
    mdx1=0                'arrêt moteur
    mdx2=0
    msx1=0
    msx2=0
    pause 500              'recule
    mdx1=0
    mdx2=1
    msx1=0
    msx2=1
    pause 1000
    mdx1=0                'arrêt moteur
    mdx2=0
    msx1=0
    msx2=0
    pause 500
    'fait pivoter la tondeuse

    random mov
    tp = mov/200          'DX où SX de manière aléatoire
    if (mov//2)=0 then
      mdx1=1
      mdx2=0
      msx1=0
      msx2=1
      pause tp            'Pour un temps aléatoire
    else
      mdx1=0
      mdx2=1
      msx1=1
      msx2=0
      pause tp            'Pour un temps aléatoire
    endif
    goto MAIN
  endif

  if distsx < 300 then    'Obstacle sur SX
    mdx1=0                'arrêt moteur DX avant obstacle SX
    goto MAIN
  endif

```

Un autre avantage de cette configuration est que, si les relais sont désactivés, les bornes du moteur sont court-circuitées ce qui provoque un effet de freinage.

Réalisation pratique

Comme d'habitude vous pouvez télécharger les typons et le programme du microcontrôleur dans le sommaire détaillé de la revue. Une fois le circuit imprimé double face réalisé, commencez par souder les composants ayant un bas profil, à savoir les résistances, les condensateurs non polarisés ainsi que le support du PIC.

Continuez en soudant les diodes (faites attention à l'orientation), les condensateurs polarisés (respectez la polarité), les transistors, le régulateur courbé à 90° sur le circuit imprimé, les borniers, les LED, les connecteurs et enfin les relais.

L'interrupteur M/A peut être soudé directement sur le circuit imprimé ou fixé sur le châssis en prolongeant les connexions à l'aide de fils. Fixez sur le châssis à l'avant de la tondeuse les radars à ultrasons, sur le dessus le capteur de pluie et la batterie.

Procédez ensuite au câblage électrique des différents éléments. Il se peut que vous ne connaissiez pas la polarisation correcte des moteurs (en cas de récupération), donc il est nécessaire de faire des tests avant le câblage final.

À la première mise sous tension, faites-en sorte qu'il n'y ait pas d'obstacle devant la tondeuse pour que la rotation des moteurs se déroule correctement, c'est-à-dire dans la direction de l'avant.

Vérifiez aussi que la lame tourne dans le bon sens **en faisant très attention de ne pas la toucher avec les mains** (risques de blessures). Sinon, il suffit d'inverser les deux fils du moteur pour qu'ils tournent dans le bon sens.

Les radars d'obstacles sont au nombre de deux, l'un positionné sur la gauche et l'autre sur la droite du châssis (un peu comme les phares d'un véhicule), afin qu'il y ait une couverture suffisante

```

if distdx < 300 then      'Obstacle sur DX
  msx1=0                 'arrêt moteur SX avant obstacle DX
  goto MAIN
endif

                          'Aucun obstacle, recommence à couper l'herbe

mdx1=1
mdx2=0
msx1=1
msx2=0

goto MAIN

'batteerie faible
standby
  PORTB=0
  lama=0
  lch=1
  pause 500
  lch=0
  pause 500
goto securite

'Surchauffe ou pluie
securite
  PORTB=0
  lama=0
  lan=1
  pause 500
  lan=0
  pause 500
goto securite

end

```

Nous avons donc développé deux programmes différents, téléchargeables sur notre site dans le sommaire de la revue. Le listing 1, publié dans ces pages, se réfère à la version avec des capteurs du commerce et se nomme « **tondeuse2** ».

Dans les deux cas, le principe de fonctionnement est le même. Le firmware du microcontrôleur est relativement simple et son diagramme de flux résume les opérations effectuées.

Plus précisément, le programme teste de manière cyclique les valeurs des entrées analogiques et vérifie l'état de fonctionnement avec un arrêt général en cas d'anomalie.

Il n'y a pas de reprise automatique afin d'éviter des situations potentiellement dangereuses, nous avons préféré simplement signaler l'anomalie et laisser l'utilisateur intervenir pour résoudre le problème.

Si l'arrêt est dû à la décharge de la batterie, éteignez la tondeuse et reliez au connecteur JP9 un chargeur de batterie du commerce pour batteries au plomb/acide.

Les routines successives du programme concernent les tests relatifs à la présence d'obstacles sur la trajectoire de la tondeuse, si cela est le cas une procédure d'évitement est engagée.

Le programme permet de faire prendre à la tondeuse un chemin aussi aléatoire que possible, de sorte que, dans un délai raisonnable elle parcourt tous les endroits du jardin.

Il faut donc éviter les situations qui amèneraient la tondeuse à se déplacer toujours aux mêmes endroits.

pour la détection d'obstacles devant la tondeuse.

Comme nous l'avons déjà évoqué, la carte électronique accepte deux capteurs commerciaux de type SRF05 fonctionnant en 5 V (c'est le type de capteurs utilisés sur notre prototype), ou d'autres types faits maison fonctionnant sous 12 V.

La différence notable entre ces 2 capteurs radars est qu'une version maison fournit en sortie un signal ON/OFF (tout ou rien) en fonction de la présence/absence d'obstacles, tandis qu'une version commerciale est plus sophistiquée avec des signaux de sortie informant le PIC de la distance réelle avec l'obstacle. Il est évident que dans les deux cas, le contrôle par le programme est différent.

Une fois la tondeuse terminée, elle doit avoir un poids d'environ 8,5 kg, qui est dû en grande partie à la batterie 12 Ah. Pour notre prototype, la consommation globale est de 6 A, ce qui correspond à une durée d'utilisation d'environ 2 heures.

En ce qui concerne la coupe de l'herbe, il est conseillé d'utiliser une lame en acier aiguisée de manière appropriée. Lors de nos tests, nous avons utilisé à la place de la lame du fil de nylon, mais les résultats ont été décevants probablement à cause de la faible puissance du moteur.

Le firmware

Comme nous l'avons mentionné précédemment, la tondeuse peut fonctionner avec des capteurs commerciaux ou faits maison.



Style et design dans le jardin

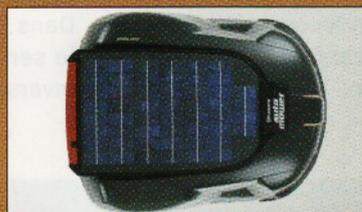
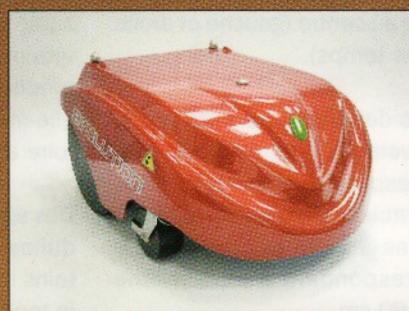
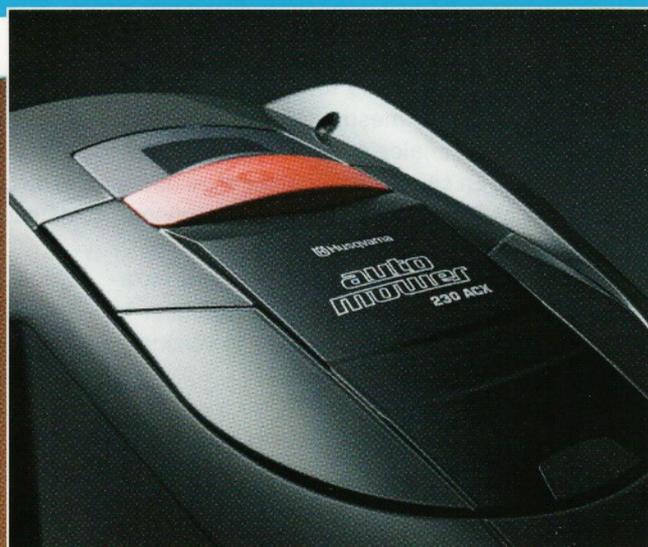
La tondeuse à gazon automatique (ou robot tondeuse, si vous préférez ...) n'est pas un concept nouveau. Depuis quelques années, il existe sur le marché une variété de modèles de tondeuses autonomes électriques, chacune avec sa propre technologie, mais tout à fait similaire du point de vue du fonctionnement. Il s'agit de modèles fonctionnant sur batterie rechargeable qui se distinguent par toute une gamme d'accessoires et donc de fonctions. Certains modèles, en plus d'éviter les obstacles courants (arbres, arbustes, meubles de jardin), savent « reconnaître » une personne ou un animal grâce à des capteurs de proximité.

Parmi les produits intéressants disponibles dans le commerce, nous trouvons la gamme de tondeuses autonomes « **Husqvarna AUTOMOWER** », fabriquées par le leader Suédois du secteur. Ces modèles, en plus de se déplacer en évitant les obstacles, savent quand il est temps de se recharger et où aller chercher l'énergie. En effet, lorsque la décharge de la batterie atteint un certain seuil, la tondeuse se précipite vers sa station de charge qui est aussi son parking. Une fois la charge terminée, elle repart toute seule tondre le reste du jardin. L'AUTOMOWER peut trouver la station de charge, car celle-ci envoie un signal qui peut atteindre une distance de 6 à 8 mètres. Husqvarna produit des tondeuses domestiques à plusieurs lames, capables de couvrir des grandes superficies.

Un autre fabricant de tondeuses autonomes est la société Italienne « **Ambrogio Robot** ». Ces produits sont rechargeables et couvrent tous les besoins, allant de petits jardins de 400 m² jusqu'à des superficies de 3000 m². Le modèle « Linea 300 » est conçu pour couper l'herbe d'un terrain de football.

Portons aussi une grande attention aux produits **Electrolux**, parmi lesquels nous trouvons une tondeuse hybride fonctionnant avec une batterie rechargeable alimentée par panneau solaire. Cette solution réduit les intervalles de charge et, les jours ensoleillés, permet une autonomie inégalée. Le modèle « **Automover Solar Hybrid** » est la première tondeuse à gazon entièrement automatique à énergie solaire. Le produit est adapté pour des superficies allant jusqu'à 2000 m².

Nous concluons avec la gamme de tondeuses automatiques « **Robomow** » de la firme « **Friendly Robotics** ». Ces modèles sont adaptés pour des superficies comprises entre 250 m² et 1600 m² et fonctionnent sur batterie. Selon le modèle, les prix varient de 1 000 € jusqu'à 5 000 €.



Avec deux capteurs, il est possible de déterminer si l'obstacle se situe à droite ou à gauche de la trajectoire. Dans ce cas, nous avons besoin de dévier légèrement la trajectoire de la tondeuse en arrêtant le moteur opposé à la position de l'obstacle. De cette manière, l'obstacle sera évité.

Une situation plus critique peut se produire si l'obstacle est détecté par les deux capteurs en même temps, dans ce cas il est nécessaire d'effectuer une manœuvre plus complexe.

Nous devons d'abord faire reculer la tondeuse puis la faire tourner sur elle-même, et ensuite avancer dans une direction différente.

Il faut donc intervenir à la fois sur le sens de rotation et sur le temps de rotation des moteurs de manière aléatoire par le biais d'une fonction mathématique appropriée incrémentée dans le microcontrôleur. Cette fonction permet d'éviter des situations de blocage qui conduiraient à des mouvements inutiles et répétitifs de la tondeuse.

Les premières lignes du firmware configurent les registres internes du PIC et définissent les variables du programme. Les lignes suivantes gèrent le démarrage des moteurs qui s'effectue selon une séquence déterminée.

D'abord le démarrage du moteur de la lame a lieu, suivi de celui des moteurs des roues afin d'éviter des pics de courant inutiles et dangereux pour l'électronique.

La boucle principale « **MAIN** » démarre la partie du programme qui gère de

manière cyclique toutes les fonctions à partir des mesures des tensions des entrées analogiques et vérifie leur validité. En cas d'erreur détectée, le micro-contrôleur bloque toutes les fonctions.

À ce stade, les deux capteurs à ultrasons sont interrogés pour identifier la présence d'obstacles dans le voisinage de la tondeuse.

Les 3 situations suivantes sont prises en compte :

- obstacle à gauche ;
- obstacle à droite ;
- obstacle au centre (gauche et droite en même temps).

Les mesures des deux capteurs représentant les valeurs de la distance d'un objet sont respectivement rapportées dans les variables « **distsx** » et « **distdx** » et comparées à la valeur numérique « 300 » correspondant à environ une distance de 60 cm.

Vous pouvez modifier ce paramètre, par exemple pour adapter la vitesse de la tondeuse à gazon.

En effet, en considérant l'inertie de la tondeuse et le temps de détection des obstacles, plus la vitesse est élevée plus la distance à laquelle l'obstacle doit être détecté est grande.

Dans les deux premiers cas, il suffit tout simplement de bloquer le moteur opposé à la position de l'obstacle, permettant ainsi de passer juste à côté. La roue est maintenue bloquée tant que l'obstacle est détecté.

Dans la 3^{ème} situation, les deux capteurs détectent la présence d'un obstacle, une simple déviation de trajectoire ne suffit pas. Il est nécessaire d'effectuer une manœuvre au cours de laquelle la tondeuse recule puis tourne sur elle-même, à la recherche d'une trajectoire alternative.

Des temporisations aléatoires sont introduites afin d'éviter la répétition de certains mouvements qui pourraient mettre la tondeuse dans une impasse.

Dans cette situation, pour éviter que le sens de rotation des moteurs soit inversé instantanément, des pauses

dans le programme ont été insérées sur chacun des relais. De cette manière, cela évite un « stress » à l'électronique et à la batterie.

Les limites sont essentiellement dues au fait que la tondeuse ne reconnaît pas un objet parallèle à la trajectoire.

Par exemple les bordures ne sont pas identifiées, ou les petits objets comme un tuyau d'eau se trouvant dans le jardin ! Dans certains cas, les rugosités du terrain peuvent empêcher son bon avancement.

Conclusion

Avec cet article, nous voulions vous proposer une application électronique différente, dans le sens où en profitant de nos connaissances dans ce domaine, il était possible de vous rendre la vie quotidienne plus confortable.

Bien que ce projet soit relativement simpliste, nous espérons qu'il attirera l'attention des curieux et sera le point de départ d'autres solutions techniques encore plus efficaces.

La construction de la tondeuse n'est pas complexe. Dans tous les cas, il est nécessaire d'avoir des connaissances en électronique mais aussi quelques aptitudes en mécanique.



Cet amplificateur pour guitare dispose d'un réglage de volume, de tonalité et un système de distorsion « fuzz » intégré activable et désactivable si nécessaire. Complété par un haut-parleur ou une mini enceinte et alimenté par une batterie de 12 V, vous pourrez vous exercer partout, à la maison, chez des amis ou bientôt à la fête de la musique.

AMPLIFICATEUR 4W POUR GUITARE

de Davide Scullino

Les musiciens savent que la guitare électrique, contrairement à son homologue acoustique, ne dispose pas d'une caisse de résonance suffisante, et donc pour entendre le son produit par les cordes il est nécessaire d'intégrer des capteurs (micros) qui transforment les vibrations des cordes en un signal électrique. Ensuite pour entendre celui-ci, il suffit de l'amplifier. Voilà pourquoi l'exercice peut devenir un problème, parce que sans amplificateur vous obtenez un son pauvre, tandis qu'avec un amplificateur conventionnel utilisé dans les studios d'enregistrement ou sur une scène vous risquez d'ameuter tout le quartier.

Nous vous proposons avec cet article de pratiquer la guitare électrique avec un amplificateur spécifique, qui dispose des réglages de base, mais dont la puissance de sortie est compatible avec un environnement familial, à savoir pas trop élevée pour ne pas déranger le voisinage. Quelques watts suffisent.

Cet amplificateur possède toutes les caractéristiques pour jouer à la maison. En plus des réglages classiques de volume et de tonalité, il intègre une distorsion du signal de type « **fuzz** », à savoir un cisaillement du signal. Cet effet est activé/désactivé par une commande spécifique.

Cet effet de distorsion est de type « **clipping** » (écrêtage) et ne doit pas être confondu avec l'effet « **treble booster** » qui applique une distorsion uniquement dans les fréquences aiguës. La saturation du son est due à un fort écrêtage du signal électrique d'entrée provenant de l'instrument. Plus l'écrêtage est important, plus le son est distordu.

De plus, la tonalité des notes varie en fonction de la vibration de la corde, car la tension (mécanique) appliquée sur celle-ci fait varier la longueur des oscillations (plus la tension est importante plus la fréquence produite est grande et vice-versa). L'effet « **fuzz** » correspond à une **distorsion de l'amplitude** et de la **forme du signal**.

Schéma électrique

Sans plus tarder, nous allons étudier le circuit de l'amplificateur, dont le schéma électrique peut être décomposé en quatre sections :

- un préamplificateur d'entrée ;
- un réglage de tonalité ;
- un étage excréteur de signal ;
- un étage amplificateur de puissance.

Partons du préamplificateur d'entrée qui, comme son nom l'indique, fournit une première amplification du signal provenant des capteurs (ou microphones) de la guitare électrique. Cet étage est constitué par 2 transistors en cascade de type **JFET à canal N** configurés en **source commune** avec une polarisation automatique. Par polarisation automatique, nous entendons que les JFET stabilisent leur point de fonctionnement grâce aux liaisons particulières de leurs grilles et de leurs sources.

Pour bien comprendre le fonctionnement de l'étage, considérons un transistor JFET comme un composant idéalement constitué par une barre de silicium

dont les extrémités sont reliées vers les bornes du drain et de la source (celles de la partie supérieure et inférieure alignées dans le symbole) et entourée par une bague reliée à la grille (symbole graphique correspondant à la flèche).

Si la barre est dopée N et la bague est dopée P, dans ce cas il s'agit d'un JFET à canal N et dans le second cas si la barre est dopée P et la bague est dopée N il s'agit d'un JFET à canal P. La diffusion dans la bague d'impuretés de polarité opposée à celles dopant la barre de silicium forment une jonction PN, qui se comporte comme une diode, mais qui dans un transistor JFET doit travailler en polarisation inverse ou nulle.

C'est pour cette raison que la **résistance entre la grille et la source** (ou le drain) est théoriquement **infinie**, de sorte que la grille peut être considérée comme isolée (ce qui n'est pas le cas dans la réalité, car la jonction fait passer un courant très faible de fuite lorsqu'elle est polarisée en inverse).

Plus la tension de polarisation inverse est grande, plus le canal se resserre, c'est-à-dire la partie de la section de la barre de silicium dans laquelle sont présentes des charges qui sont à l'origine du courant (et qui le maintiennent) traversant le drain et la source.

Pour simplifier, nous pouvons dire qu'un **transistor JFET est une sorte de résistance variable commandée par une tension**. En fonctionnement normal la tension entre le drain et la source est positive ($V_{DS} > 0$) et celle entre la grille et la source (jonction PN) est négative ($V_{GS} < 0$).

L'augmentation de cette tension inverse fait croître les zones de déplétion (non conductrices) autour du canal jusqu'au pincement de celui-ci. La conduction du canal est donc contrôlée par la tension « grille-source » V_{GS} .

Dans notre circuit, les transistors JFET sont à canal N, donc le canal a une résistance minimale lorsque la polarisation est nulle et une résistance maximale lorsqu'une tension maximale négative admissible est appliquée entre la source et la grille.

La polarisation automatique est obtenue en jouant sur le fait que la grille en théorie n'absorbe pas de courant. En alimentant le circuit entre le drain et la source, un courant circule alors provoquant ainsi une chute de tension entre la source et la masse.

En reliant la grille à la masse à travers une résistance de valeur élevée (ce qui a l'avantage d'avoir une haute impédance d'entrée du circuit car la grille n'absorbe pas de courant), elle se trouve à 0 V et est donc négative par rapport à la source.

Ainsi, cela génère une tension de polarisation inverse nécessaire pour contrôler le canal, celui-ci étant stabilisé à une valeur qui est déterminée par la résistance de source. En fait, plus le courant est grand, plus la tension négative appliquée à la grille devient grande (en valeur absolue), de sorte qu'à un moment donné le courant ne peut plus augmenter, il est autolimité. Il ne peut pas diminuer, car sa baisse provoquerait une diminution au niveau de la résistance de source et donc le canal aurait tendance à s'ouvrir.

Revenons à notre schéma électrique, le signal d'entrée est appliqué à la prise « **IN** » et parvient directement sur la grille du transistor FET Q1. Celui-ci est monté en **source commune** avec une **polarisation automatique** contrôlée par la résistance de source **R3** et la résistance de la grille **R1**, cela correspond à peu près à l'impédance d'entrée du circuit. Le signal audio provenant du capteur de la guitare est amplifié en tension une première fois par un facteur égal à un peu plus de **3 fois** et se retrouve sur le

Caractéristiques techniques de l'amplificateur pour guitare :

- Puissance de sortie sous 4 Ω : 4 W
- Puissance de sortie sous 8 Ω : 2,5 W
- Tension d'alimentation : 12 VDC
- Courant maximale absorbé : 1 A
- Bande passante : de 30 à 16 000 Hz
- Gain en tension : 200
- Distorsion harmonique : 0,15 %
- Impédance d'entrée : 1 M Ω

drain de Q1 avec une **phase inversée** (déphasage de 180°).

Le signal audio atteint alors l'étage du contrôle de tonalité. Cet étage est de **type passif** composé par des cellules filtrantes **R/C du premier ordre**. Chaque cellule est caractérisée par une fréquence de coupure de 20 dB/décade (soit 20 dB chaque fois que la fréquence décuple ou est divisée par 10), c'est-à-dire 6 dB par octave (6 dB pour chaque doublement ou division par 2 de la fréquence).

La première cellule est un filtre passe-bas (il atténue les fréquences supérieures à la fréquence de coupure) constitué par la résistance R4 et le condensateur C2. La fréquence de coupure supérieure (avec R4 = 47 kΩ et C2 = 10 nF) est donnée par la formule :

$$f_{ts} = 1 / (6,28 * R4 * C2)$$

$$f_{ts} = 1 / (6,28 * 47\ 000 * 10 * 10^{-9})$$

$$f_{ts} = 1 / (0,0029516) = 339\ \text{Hz}$$

La deuxième cellule est un filtre passe-haut (il atténue les fréquences inférieures à la fréquence de coupure) composé par le condensateur C1 et la résistance R5. La fréquence de coupure inférieure (avec R5 = 22 kΩ et C1 = 6,8 nF) est donnée par la relation :

$$f_{ti} = 1 / (6,28 * R5 * C1)$$

$$f_{ti} = 1 / (6,28 * 22\ 000 * 6,8 * 10^{-9})$$

$$f_{ti} = 1 / (0,000939488) = 1064\ \text{Hz}$$

Etant donné que les filtres sont montés en parallèle à travers les extrémités du potentiomètre R20, le signal de sortie est disponible sur le point milieu de R20 avec une bande passante allant de **339 Hz à 1064 kHz**, bien que l'atténuation légère des filtres utilisés fait passer faiblement les autres signaux. Cela signifie que toute la gamme audio est disponible, même si certaines fréquences peuvent être renforcées.

En fait, le potentiomètre **R20** joue le rôle de **contrôle de tonalité**. Si le point milieu de R20 est amené vers la sortie du filtre passe-bas (R4/C2), il y a prédominance des notes graves (fréquence basses), si par contre le curseur du potentiomètre se situe vers la sortie du filtre passe-haut (R5/C1), il y a prédominance des notes médiums/aigues. Si le curseur du potentiomètre est à mi-course, le signal ne subit aucun renforcement des graves ou des aigus.

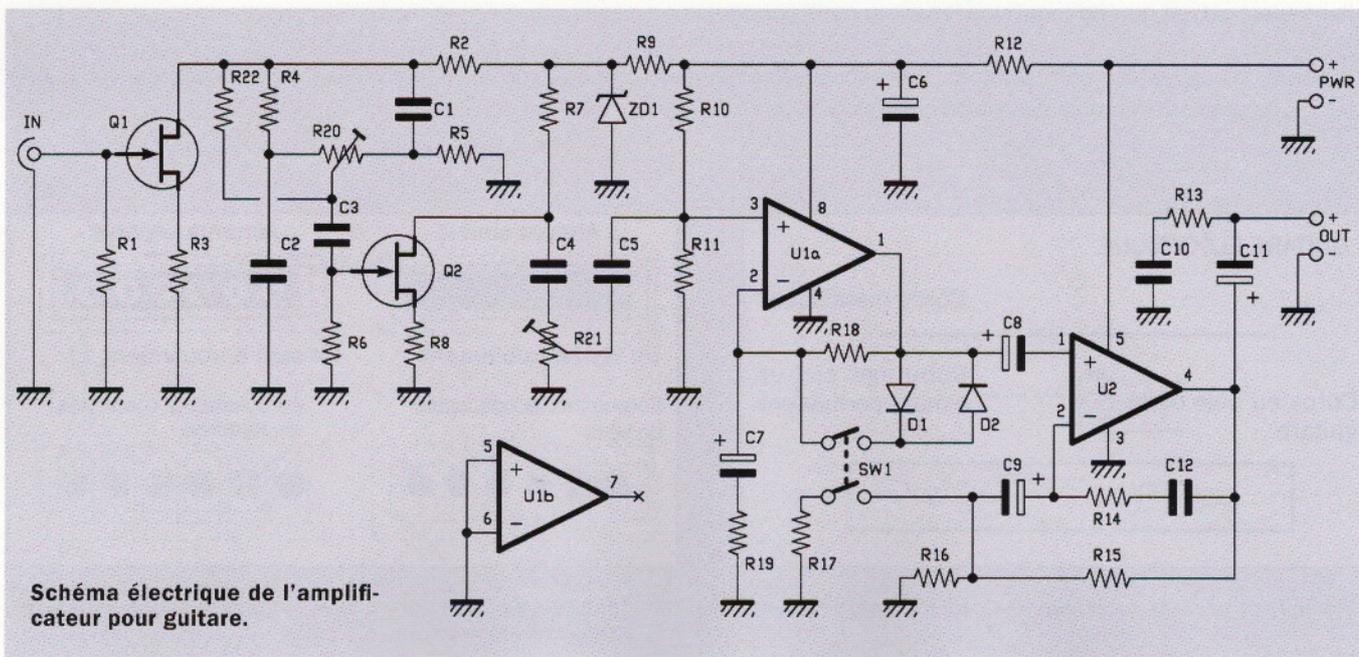
Pour atténuer (à savoir rendre moins accentuée) la courbe de réponse du contrôle de tonalité et éviter ainsi que les fréquences soient privilégiées excessivement à l'intérieur de la bande définie par les fréquences de coupure calculées précédemment, nous avons inséré la résistance R22 de dérivation, dont la valeur est suffisamment élevée pour permettre la mise en avant des tonalités hautes et basses, tout en laissant passer une partie du signal non altéré de la sortie du premier étage JFET vers

l'entrée du second étage JFET. Dans la pratique, R22 réinjecte en sortie du contrôle de tonalité, une partie du signal original, à savoir non filtré.

Le point milieu du potentiomètre R20 est relié, par l'intermédiaire du condensateur de découplage C3 qui sert à isoler le transistor Q1 du réseau de polarisation de celui de Q2, à la grille du second transistor JFET. Celui-ci constitue le second étage amplificateur et fonctionne comme Q1, il est doté des résistances de drain, de source et de grille de valeurs égales à celles de Q1 (R1=R6, R8=R3, R2=R7).

Le transistor **Q2 amplifie encore de 3 fois le signal** afin de compenser l'atténuation se produisant dans les filtres et **déphase de nouveau le signal de 180°** de façon à récupérer **en sortie de Q2 un signal ayant la même phase** que celui arrivant sur l'entrée « IN ». La cohérence de phase est importante, car le signal que le haut-parleur reproduit doit être synchronisé dans la mesure du possible avec celui émis (même très faiblement) par les cordes de la guitare, signal qui est audible même sans amplification.

Le drain de Q2 restitue le signal amplifié à travers le condensateur de découplage C4 pour atteindre le potentiomètre R21. Ce dernier permet de régler le volume d'écoute de l'amplificateur, si le curseur est proche de la masse le niveau d'écoute sera faible et vice versa.



Les capteurs (ou micros) de la guitare

Une guitare électrique comporte au moins deux capteurs dont les sons peuvent être mélangés à l'aide d'un commutateur afin d'amplifier un ou les deux signaux qui se chevauchent et ainsi, obtenir une plus grande quantité d'harmoniques avec un son plus spatial. Le son obtenu étant la somme des vibrations perçues par les capteurs placés à différents endroits de la guitare. Les capteurs d'une guitare électrique sont de types magnétiques, constitués par du fil de cuivre émaillé bobiné sur un noyau ferromagnétique disposés sous les cordes. C'est la forme la plus simple, mais il existe des capteurs constitués d'autant de noyaux qu'il y a de cordes, chacun placé à une distance fixe ou réglable d'une corde.

Dans tous les cas, le son est détecté à travers les variations de la perméabilité magnétique dues aux vibrations de la corde. Par rapport à l'utilisation de capteurs à vibrations acoustiques, une telle solution rend l'amplification insensible à un feedback acoustique (Larsen).

Quel que soit le nombre de noyaux, le capteur peut comporter un seul enroulement (simple bobinage) ou deux enroulements (micro Humbucker).

En ce qui concerne le simple bobinage avec plusieurs enroulements, les bobines peuvent être reliées en série ou en parallèle les unes avec les autres, en fonction des besoins de l'impédance et du niveau du signal de sortie. Cependant les capteurs à simple bobinage sont sensibles à certains parasites dus aux champs magnétiques environnants (transformateurs électriques, etc.).

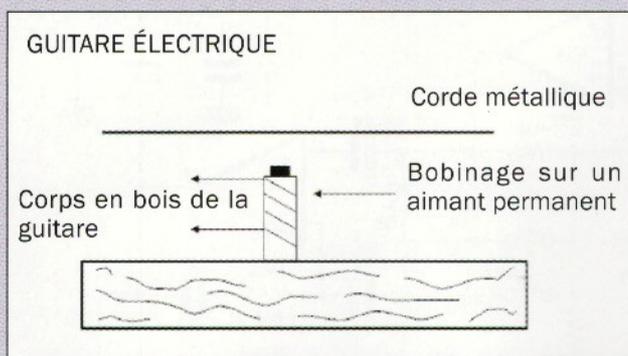
Un ingénieur de chez Gibson, Seth Lover, a trouvé une solution à ce problème en utilisant **deux capteurs « simple bobinage » montés à la fois magnétiquement et électriquement en opposition de phase**. De cette façon, le signal provenant des cordes, qui dépend à la fois de la polarisation magnétique et du sens des bobinages, est en phase sur les deux éléments, et se renforce. Tandis que les perturbations, qui sont présentes sur les deux capteurs, ne dépendent que du sens du bobinage, dans ce cas elles se neutralisent.

Ces capteurs sont nommés « **Humbucker** », ce qui signifie « nettoyeur de bourdonnement ». Ils sont connus pour reproduire un son « chaud » et puissant, contrairement aux micros « simple bobinage » préférés pour leur son clair et cristallin. En raison du niveau de sortie plus élevé, la saturation est atteinte avec un gain plus faible.

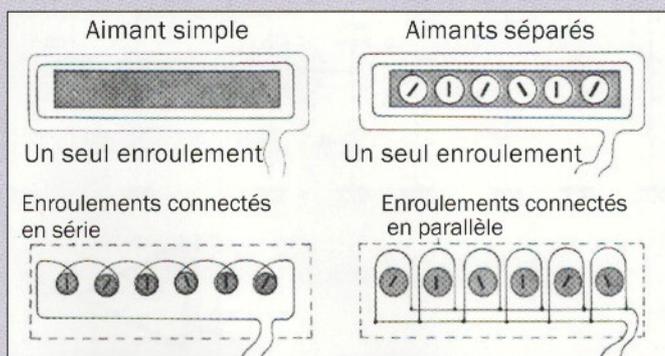
Les caractéristiques d'un capteur (micro) dépendent principalement de la matière du matériau magnétique et des caractéristiques de la bobine. Tout dépend de la section du fil de la bobine et du nombre de tours. Un fil très mince (par exemple 0,05 mm) produit un son très fin (cristallin), tandis qu'un fil de section plus grande (par exemple 0,056-0,06-0,063) produit un son plus riche et ample. Le nombre de spires varie généralement autour de quelques milliers et influence principalement la puissance du signal de sortie et le pic de résonance.

Pour modifier le son d'un capteur, il suffit de jouer sur la hauteur des aimants. Plus la hauteur est importante, plus la largeur de l'enroulement est importante, ce qui a pour effet d'étendre la réponse en fréquence car la valeur de l'inductance de l'enroulement est réduite (pour une longueur de fil donnée).

Souvent, les spires des enroulements sont recouvertes par de la cire ou de la résine afin d'éviter que le capteur recueille les bruits ambiants en plus des vibrations des cordes.



Exemple de capteur magnétique (micro) à simple bobinage.



Configuration du micro à bobine unique (single coil) et à enroulements séparés.

Le matériau utilisé principalement pour les noyaux magnétiques est l'**AINiCo**, il s'agit d'un alliage d'aluminium (Al), de nickel (Ni) et de cobalt (Co), auquel on ajoute certains métaux minoritaires, tels que du cuivre et du titane. Ce matériau est résistant à la corrosion et aux chocs. Les capteurs constitués d'AINiCo ont une longue durée de vie, ils perdent environ 5 % de leurs propriétés magnétiques tous les 100 ans.

Le positionnement des capteurs dans la guitare est crucial pour le son, parce que leurs dispositions par rapport à la longueur des cordes déterminent le contenu harmonique du son restitué. Lorsqu'un seul micro est utilisé, il est généralement monté le plus près possible du chevalet afin de capturer le maximum d'ondes fondamentales.

La guitare solo met davantage l'accent sur les harmoniques afin de donner un son plus percutant. Dans ce cas, le micro est généralement placé vers le manche. Beaucoup de guitares utilisent, pour obtenir une variété de sons, des micros séparés. Ces derniers sont montés, en général, pour les fréquences basses proches du chevalet, pour les fréquences aiguës vers le manche. La plupart des guitares électriques sont équipées d'au moins 2 micros, bien qu'il ne soit pas difficile de trouver des modèles en comportant 3 et plus (moyennant finances).

Parfois, les capteurs arrières sont inclinés d'environ 10° afin d'abaisser la hauteur des cordes par rapport à la caisse. Cela permet de capter plus de fréquences fondamentales et d'avoir une clarté du son des basses jusqu'au aiguës.

Les guitares électriques disposant de plusieurs micros sont généralement équipées de commutateurs qui permettent d'utiliser un seul micro ou différentes combinaisons de micros.

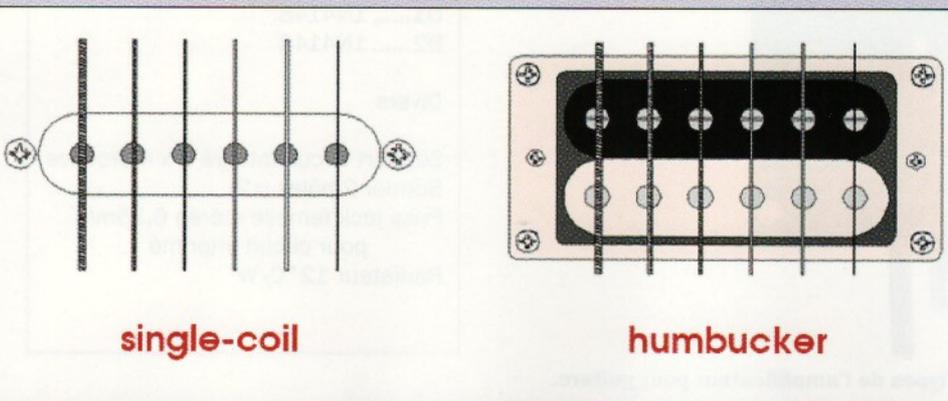
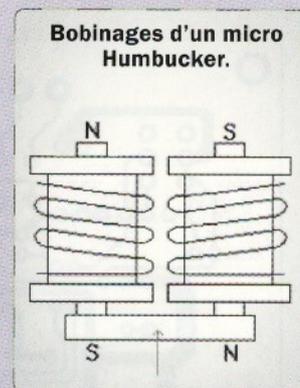
Un facteur très important dans le positionnement du micro est l'attraction exercée sur la corde. Plus la corde est proche, plus le signal de sortie est grand. Cependant, il est nécessaire de rappeler que l'oscillation d'une corde exposée à un champ magnétique modifie le son produit.

Les pièces polaires des aimants des micros sont habituellement réglées à une hauteur de 1,5 mm en dessous des cordes. La bonne disposition des micros le long de la corde dépend d'un certain nombre de facteurs, mais est principalement liée au mode de vibration des cordes.

L'un des problèmes posés par l'utilisation de micros avec des bobines à noyaux magnétiques est qu'ils peuvent détecter les bruits (ronflements) causés par la fréquence du réseau électrique (le 50 Hz ou le 60 Hz). Cela pourrait causer des problèmes pour la reproduction de signaux ayant des niveaux faibles. Pour cela, le micro Humbucker a été inventé, le principe consiste à enrouler deux bobines dans le sens opposé sur 2 noyaux identiques mais magnétiquement opposés l'un à l'autre. Les deux détectent donc le même bruit, mais comme ils sont connectés en série avec des enroulements opposés (opposition de phase), le bruit capté par un micro est annulé par celui capté par l'autre micro, il en résulte un bruit final nul.

Dans un micro Humbucker, le bruit est présent dans les 2 bobines, cependant le signal de la corde en vibration est appliqué à une seule d'entre elle. L'autre bobine est située assez loin de la corde et est également protégée par un écran afin de ne pas capter une quelconque vibration de la corde, ce qui annulerait le signal principal (le son de la guitare) capté par la première bobine !

Les micros Humbucker sont facilement reconnaissables à l'œil, car bien qu'ils aient la même largeur que les micros classiques (7,5 cm), contrairement à ces derniers qui font seulement 2,5 cm de long, leur longueur est de 5 cm.



Aspects des deux configurations d'un micro pour guitare électrique.

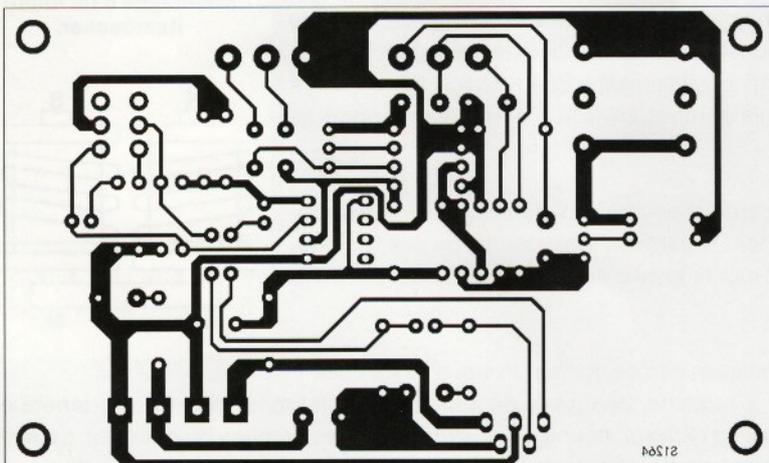


Figure A : circuit imprimé à l'échelle 1 : 1 côté soudures.

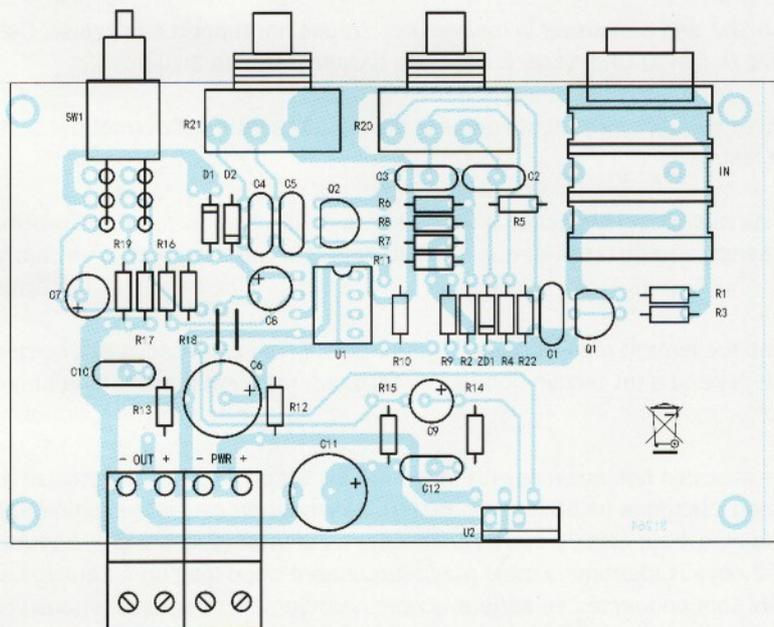


Figure B : schéma de câblage des composants de l'amplificateur pour guitare.

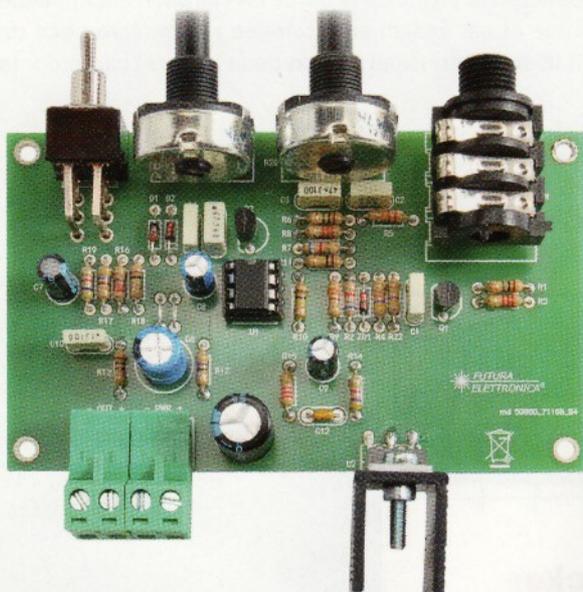


Figure C : photo de l'un de nos prototypes de l'amplificateur pour guitare.

Liste des composants de l'amplificateur pour guitare

- R1..... 1 M Ω
- R2..... 6,8 k Ω
- R3..... 2,2 k Ω
- R4..... 47 k Ω
- R5..... 22 k Ω
- R6..... 1 M Ω
- R7..... 6,8 k Ω
- R8..... 2,2 k Ω
- R9..... 330 Ω
- R10.... 100 k Ω
- R11.... 100 k Ω
- R12.... 47 Ω
- R13.... 10 Ω
- R14.... 4,7 k Ω
- R15.... 2,2 k Ω
- R16.... 220 Ω
- R17.... 470 Ω
- R18.... 100 k Ω
- R19.... 4,7 k Ω
- R20.... potentiomètre 47 k Ω linéaire
- R21.... potentiomètre 47 k Ω logarithmique
- R22.... 470 k Ω

- C1..... 6,8 nF/63 V polyester
- C2..... 10 nF/100 V polyester
- C3..... 47 nF/63 V polyester
- C4..... 47 nF/63 V polyester
- C5..... 470 nF/63 V polyester
- C6..... 220 μ F/16 V électrolytique
- C7..... 1 μ F/16 V électrolytique
- C8..... 10 μ F/16 V électrolytique
- C9..... 4,7 μ F/16 V électrolytique
- C10.... 100 nF/100 V polyester
- C11.... 1000 μ F/16 V électrolytique
- C12.... 3,3 nF/63 V polyester

- Q1..... BF245
- Q2..... BF245

- SW1... interrupteur double coudé à 90°
- U1..... LM358
- U2..... TDA2003

- ZD1.... zener 9,1V/400mW

- D1..... 1N4148
- D2..... 1N4148

Divers

- Support circuit intégré 2 x 4 broches
- Bornier 2 pôles (x2)
- Prise jack femelle stéréo 6,35mm pour circuit imprimé
- Radiateur 12° C/W

Le signal est ensuite dirigé vers l'**étage excréteur** (distorsion « fuzz ») à travers le condensateur C5 qui le découple en continu du réseau de polarisation R10/R11 de l'amplificateur opérationnel U1. Celui-ci fonctionne en configuration « **non-inverseuse** » et amplifie le signal sans l'inverser.

Afin que le montage puisse fonctionner avec une seule alimentation, nous avons amené l'entrée « non-inverseuse » de U1 (broche 3) à la moitié du potentiel de l'alimentation (celle appliquée à la broche 8) grâce aux résistances R10 et R11, tout en découplant le signal du réseau continu.

Ainsi, en l'absence de signal, l'amplificateur opérationnel n'amplifie pas. Son gain en tension G est égal à 1. En présence d'un signal, U1 l'amplifie avec un facteur égal à : $(R18 + R19) / R18$.

Le condensateur C7, calculé pour laisser passer la bande passante de l'amplificateur, présente une impédance négligeable par rapport à la résistance R19, mais en continu il se comporte comme un circuit ouvert, de sorte qu'il possède une impédance (résistance) infinie. Par conséquent en continu, le potentiel appliqué à la broche 3 est amplifié de : $(100\text{ k}\Omega + \infty) / \infty$ ce qui fait 1.

En raison de la rétroaction en continu, le potentiel de sortie de l'amplificateur opérationnel au repos a la même valeur que le potentiel au niveau de la broche 3, ce qui permet au signal d'osciller autour de cette valeur en supposant que l'amplitude maximale soit égale dans les alternances.

Le condensateur de sortie C8 transforme de nouveau le signal de manière alternative en fonction de sa charge et de sa décharge. Le gain du signal n'est pas modifié si le double interrupteur SW1, qui contrôle la distorsion « fuzz », est ouvert.

Si SW1 est fermé, il introduit en parallèle les diodes D1 et D2 sur la résistance de contre réaction R18. Les diodes étant en parallèle, elles coupent (rabortent) une partie de la sinusoïde présente aux bornes de la résistance R18, la tension de sortie se trouve alors écrêtée avec un son caractéristique d'une distorsion par écrêtage (clipping).

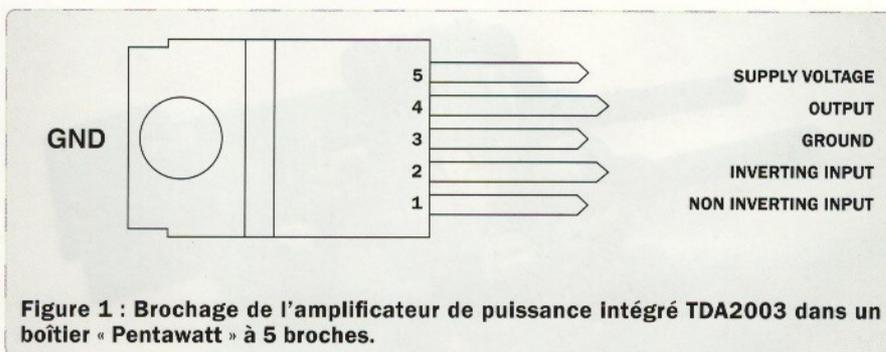


Figure 1 : Brochage de l'amplificateur de puissance intégré TDA2003 dans un boîtier « Pentawatt » à 5 broches.

Etant donné que dans ce cas l'amplitude du signal est limitée lorsque la distorsion est activée, le double interrupteur SW1 relie en même temps à la masse la résistance R17, qui se retrouve en parallèle avec R16. Cela a pour effet d'augmenter le gain en tension de l'étage amplificateur de puissance U2.

Étudions maintenant le dernier étage, à savoir l'amplificateur de puissance. Il est constitué par un circuit intégré **TDA2003** qui se présente dans un boîtier « Pentawatt ». Il est capable de délivrer une puissance de sortie allant jusqu'à 10 W sous 2 Ω avec une tension d'alimentation de 14,4 Vcc et jusqu'à 6 W sous 4 Ω dans les mêmes conditions.

L'alimentation typique du TDA2003 lui permet d'être utilisé comme un booster pour autoradio. En fait la tension de 14,4 VDC correspond à la tension disponible dans un véhicule avec le moteur en marche, c'est-à-dire avec l'alternateur en rotation.

Dans notre cas, nous utilisons seulement une batterie dont la tension est d'environ 12,6 VDC (chargée), ce qui permet d'obtenir en sortie de l'amplificateur une puissance de 4 à 4,5 W sous une charge de 4 Ω. Rien ne vous empêche de connecter en parallèle 2 haut-parleurs de 4 Ω afin de constituer une impédance finale de 2 Ω et d'obtenir ainsi une puissance de sortie de 6 W.

Cet étage amplificateur à base de TDA 2003 est doté d'une contre réaction introduite par les résistances R15 et R16. D'après les données du fabricant, le gain de tension de l'étage en présence d'un signal vaut sensiblement :

$$G = (R15 + R16) / R16$$

Cependant, comme nous avons inséré dans le circuit une seconde résistance (R17) connectée en parallèle avec R16 par l'intermédiaire de l'interrupteur SW1 qui commande la distorsion, lorsque l'effet « fuzz » est activé, le gain augmente.

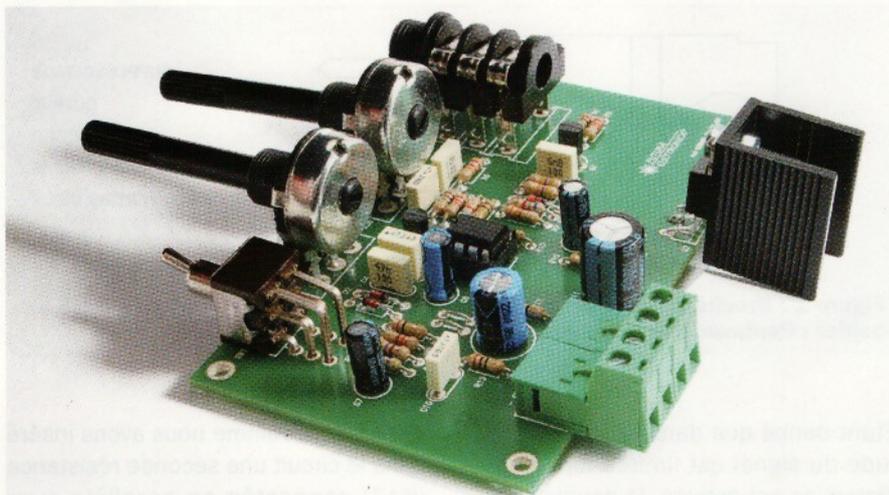
En effet, nous devons remplacer la valeur R16 dans la formule ci-dessus par la valeur $R16 // R17$ (il faut lire R16 en parallèle avec R17), or la résistance $R16 // R17$ est inférieure à la plus petite valeur des deux résistances ($R16 // R17$ vaut environ 150 Ω).

Le condensateur électrolytique C9 permet de faire en sorte qu'en continu, la tension au repos sur la sortie soit définie par le réseau de polarisation interne du circuit intégré TDA 2003 à la moitié de la valeur de la tension d'alimentation.

S'il n'y avait pas le condensateur, l'amplificateur amplifierait le continu même en présence du signal audio et la sortie serait au même potentiel que l'alimentation.

Or nous avons besoin qu'en l'absence de signal, c'est-à-dire au repos, la tension de sortie soit exactement à la moitié de la tension d'alimentation appliquée entre les broches 5 et 3 du TDA2003, étant donné que l'amplificateur est alimenté par une alimentation positive unique.

Avec cet artifice, en présence d'un signal audio, le potentiel sur la broche 4 varie entre des valeurs supérieures et inférieures, mais toujours par rapport à la moitié de la tension d'alimentation. Ainsi le signal de sortie peut évoluer alternativement autour de la moitié de la tension d'alimentation.



Le condensateur de sortie C11, placé entre la broche 4 et le haut-parleur (OUT) reconstitue le signal alternatif. À l'allumage, il se charge à la moitié de la tension d'alimentation.

Lorsqu'une alternance positive arrive, il conduit le courant vers le haut-parleur et se charge en même temps, puis lors d'une alternance négative C11 se décharge vers la sortie du TDA2003, un courant inverse traverse le haut-parleur et ainsi de suite. C11 sert de réservoir pour les alternances négatives, plus sa valeur est élevée plus les fréquences basses seront transmises. Il permet aussi d'éviter de transmettre la tension continue présente sur la sortie du TDA2003, ce qui aurait pour effet la destruction immédiate du haut-parleur.

Notez que le réseau constitué par R13 et C10 sert à compenser la réponse en fréquence, en atténuant l'amplification des fréquences les plus élevées, de façon à éviter des phénomènes d'auto-oscillations qui sont dangereux pour les haut-parleurs.

La nécessité d'un tel réseau s'explique en considérant qu'un haut-parleur est une charge fortement inductive, plus la fréquence du signal est élevée plus son impédance augmente.

L'inductance de la bobine du haut-parleur peut ainsi provoquer une rotation de phase du signal, ce qui peut déclencher une auto oscillation de l'amplificateur en raison du retour par le réseau de contre réaction. Le réseau constitué par R14 et C12 contribue également à la stabilité du TDA2003.

En effet, ce réseau est placé entre les broches 2 et 4. Sa fonction est de réduire l'amplitude lorsque la fréquence du signal audio croît, de manière à contrecarrer tout phénomène d'auto-oscillation des fréquences très hautes, en dehors de la gamme audio qui est celle dans laquelle l'amplificateur doit travailler. Ce réseau exerce également un rôle contre l'effet de rotation de phase provoqué par le haut-parleur aux fréquences élevées.

Concluons l'analyse du schéma électrique avec l'alimentation qui est appliquée au bornier marqué « PWR+ » et « PWR- ». La tension doit être de 12 VDC, de préférence stabilisée. Cette tension est directement appliquée à l'étage de puissance (le TDA2003), tandis que l'étage de distorsion et de préamplification reçoivent cette tension à travers un filtre RC constitué par R12 et C6, avec en plus pour l'étage d'entrée un régulateur constitué par la résistance R9 et la diode zener ZD1.

Le filtre permet de niveler la tension d'alimentation qui risque de chuter lors des pics d'absorptions causés par l'étage de puissance, c'est-à-dire lorsque le TDA2003 fonctionne à des niveaux de puissances élevées.

S'il n'y avait pas le filtre et le régulateur, l'alimentation de l'amplificateur opérationnel et des transistors JFET oscillerait de manière synchronisée avec le signal audio (mais en opposition de phase par rapport au signal d'entrée).

Cela modifierait le signal, sans compter le risque d'auto-oscillations dû au fait que, si la tension d'alimentation varie

en phase avec le signal (ce qui se passe au niveau de Q1), il y a un risque d'oscillation de l'amplificateur, car il se rétro-cède à travers la ligne d'alimentation un signal en phase qui va augmenter le gain au-delà de la limite acceptable.

Réalisation pratique

Passons maintenant à la construction de l'amplificateur, qui nécessite un peu d'électronique et de mécanique. En effet, il doit être placé dans un boîtier proche du haut-parleur. Le circuit imprimé est constitué d'une seule face. Il est facilement reproductible par toute personne ayant un peu de pratique. Vous pouvez télécharger le typon à l'échelle 1:1 ou le photocopier à partir de la revue. Les fichiers Gerber, pour une fabrication professionnelle, sont disponibles sur le site.

Le circuit imprimé a été conçu de manière à ce que toutes les commandes se situent du même côté, ainsi que la prise d'entrée du signal provenant de la guitare. Sur le côté opposé (à l'arrière), se trouvent les borniers des connexions de l'alimentation et du haut-parleur.

Une fois le circuit imprimé fabriqué, commencez par souder les composants ayant un profil bas, c'est-à-dire les résistances et les diodes (attention à l'orientation).

Continuez en soudant le support de l'amplificateur opérationnel, puis les condensateurs non polarisés et les transistors JFET. Pour ces derniers respectez la disposition de leur méplat, ils doivent être orientés vers la gauche vu de dessus avec les commandes vers le haut.

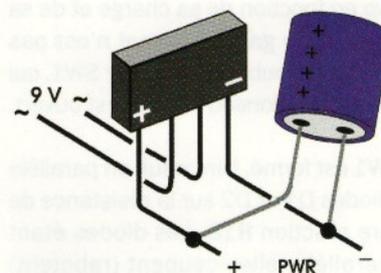


Figure 2 : alimentation non stabilisée avec une sortie de 12 V. Le condensateur doit être un modèle 4700 µF/25 V et le transformateur doit être de 220VAC/9 VAC 20 VA.

Continuez avec les condensateurs polarisés en prêtant attention à l'orientation (le moins est marqué sur leurs boîtiers), les potentiomètres, la prise jack femelle 6,35 mm et les borniers.

Le TDA2003 doit être placé debout sur un dissipateur, ne vous inquiétez pas pour l'orientation en raison de l'agencement des broches du TDA2003, il ne peut aller que dans un seul sens.

Le **dissipateur thermique** doit être d'au moins **12°C/W**, il doit être fixé à l'aide d'un écrou et d'une vis 3MA après avoir appliqué de la graisse silicone sur la partie métallique du TDA2003.

Attention, le dissipateur est en contact électrique avec la masse, étant donné que la semelle métallique du TDA2003 est reliée en interne à la broche 3. Cela ne constitue pas un problème, mais si vous voulez vous pouvez l'isoler afin d'éviter tout contact dans le boîtier. Pour cela, vous devez intercaler une feuille de mica entre la semelle métallique du circuit intégré et le dissipateur.

Les potentiomètres R20 et R21 sont directement soudés sur le circuit imprimé, mais ils peuvent être décalés selon votre boîtier à l'aide de 3 fils dont la longueur ne doit pas dépasser 20 cm.

Le boîtier doit avoir les dimensions suivantes : 20 x 20 x 15 cm (hauteur). Vous pouvez également construire une caisse en bois et intégrer le haut-parleur avec l'électronique de l'amplificateur, comme ceux que vous pouvez trouver dans le commerce.

N'oubliez pas de percer les trous pour le passage des câbles, des axes des potentiomètres et du jack de la prise d'entrée. Pour l'alimentation électrique, vous pouvez utiliser une alimentation stabilisée de 12 VDC/1A.

Si vous prévoyez la construction en bois, vous pouvez intégrer un transformateur, un pont de diode et un condensateur pour faire fonctionner l'amplificateur sur le secteur 230 VAC. Reportez-vous à la figure 2.

Munissez-vous d'un transformateur de 220 VAC/ 9 VAC 20 VA.

En effet la tension en sortie du pont de diode sera de :

$$9 \text{ V} - (2 * 0,6 \text{ V}) * \sqrt{2} \\ = (9 - 1,2) * 1,414 \\ = 12,5 \text{ V environ.}$$

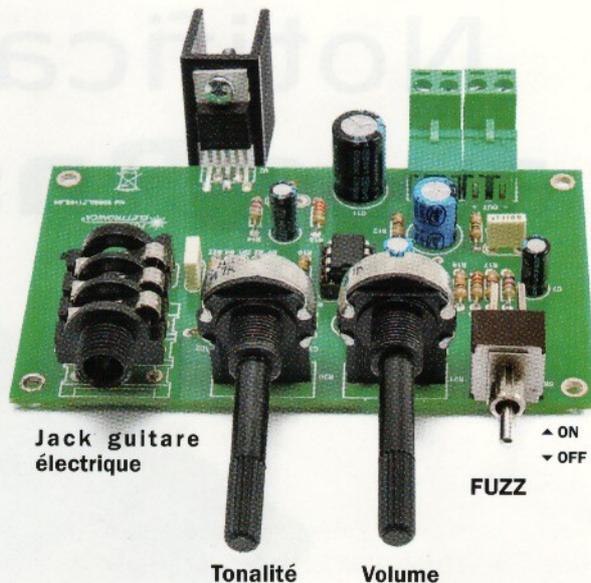
Nous soustrayons 2 * 0,6 V à la tension de sortie du transformateur, car cela provient de la chute de tension introduite par 2 des 4 diodes du pont de diode pour chaque alternance.

Le terme $\sqrt{2}$ provient du lissage de la tension effectuée par le condensateur de filtrage. Le pont de diode doit être un modèle 100 V/2 A et le condensateur doit avoir une valeur de 4700 $\mu\text{F}/25 \text{ V}$.

Vous pouvez essayer avec un transformateur de 220 VAC/12 VAC, ne branchez pas l'amplificateur.

En parallèle sur le condensateur, reliez une résistance de 470 Ω de 1 W ou 2 W afin d'introduire une charge dans le circuit. Testez la tension, elle devrait être autour de 16 V.

Si c'est le cas, vous pouvez brancher l'amplificateur, la puissance de sortie sera plus élevée par rapport à une utilisation sur batterie. **Cependant, ne dépassez pas une valeur de 16 VDC pour l'alimentation.**

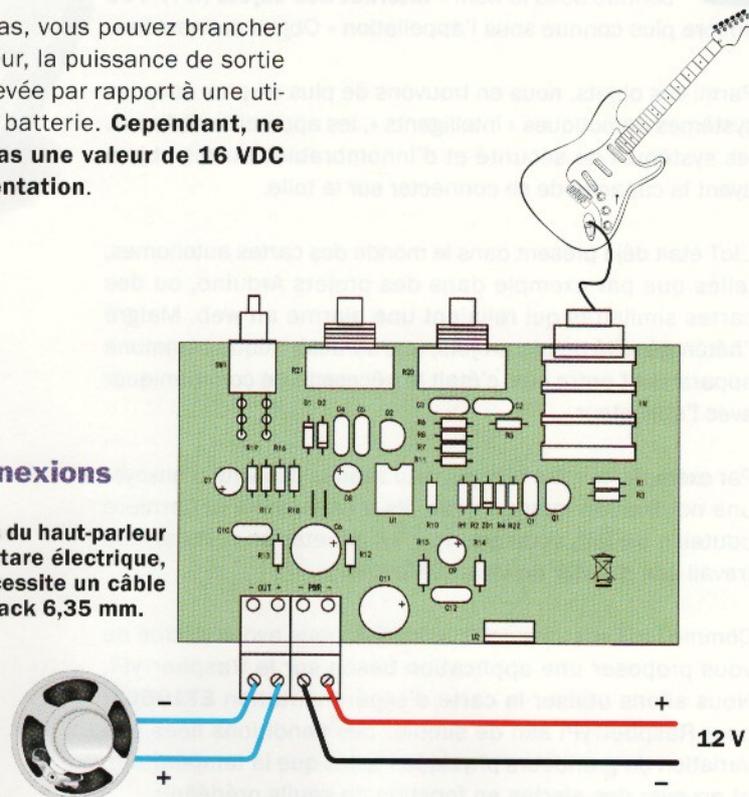


Le haut-parleur doit être un modèle large bande avec un diamètre d'au moins 10 cm, sa réponse en fréquence doit être comprise entre 100 Hz et 5000 Hz. Le modèle Visiton « **KT 100 V - 4 Ω** » convient parfaitement, mais vous pouvez choisir un autre modèle selon vos convenances.

Une fois l'assemblage terminé, insérez le câble de la guitare et mettez sous tension, le volume étant au minimum. Augmentez progressivement le volume et commencez à jouer quelques notes jusqu'à ce que vous entendiez le son dans le haut-parleur. ■

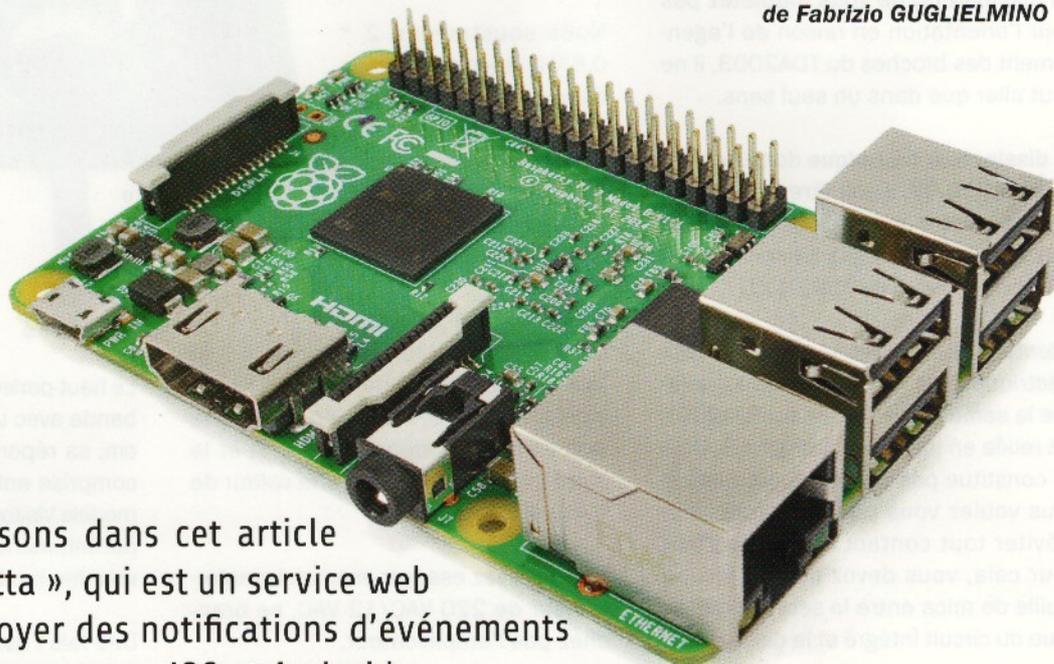
Les connexions

Connexions du haut-parleur et de la guitare électrique, celle-ci nécessite un câble muni d'un jack 6,35 mm.



Notification push avec RaspberryPi

de Fabrizio GUGLIELMINO



Nous vous proposons dans cet article d'utiliser « Pushetta », qui est un service web permettant d'envoyer des notifications d'événements avec des smartphones sous iOS et Android.

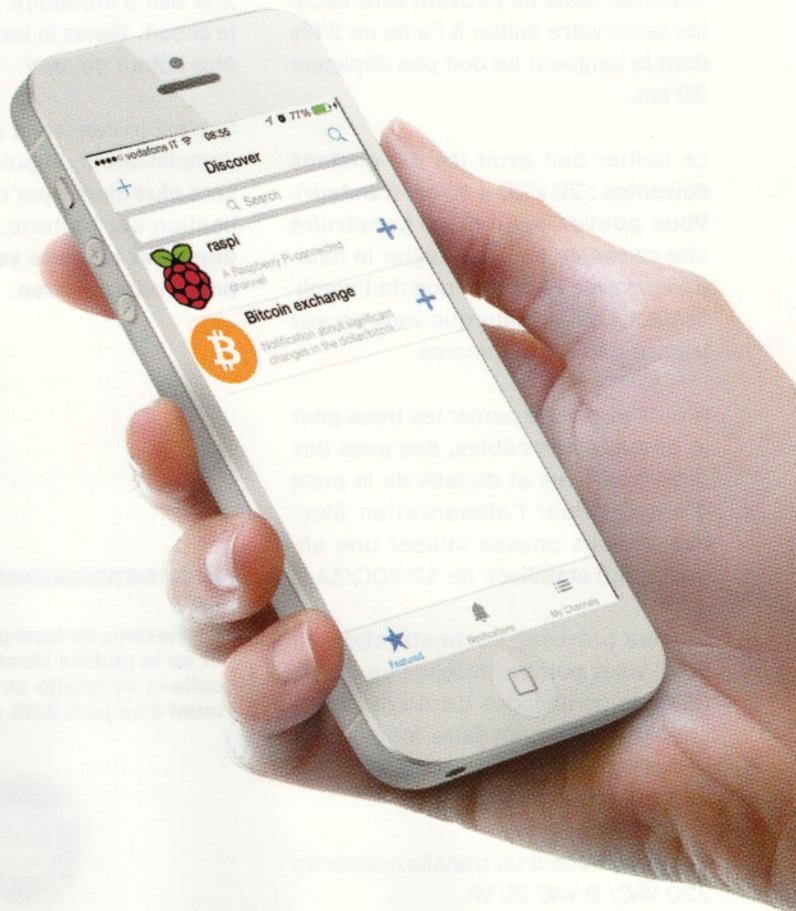
De plus en plus, nous devons réaliser des projets liés à l'Internet, qui rentrent dans la catégorie maintenant connue sous le nom « **Internet des objets (IoT)** », ou encore plus connue sous l'appellation « Objets connectés ».

Parmi ces objets, nous en trouvons de plus en plus dans les systèmes domotiques « intelligents », les appareils médicaux, les systèmes de sécurité et d'innombrables autres objets ayant la capacité de se connecter sur la toile.

L'IoT était déjà présent dans le monde des cartes autonomes, telles que par exemple dans des projets Arduino, ou des cartes similaires qui reliaient une alarme au web. Malgré l'hétérogénéité de ces projets, une caractéristique commune apparaissait entre eux, c'était la nécessité de communiquer avec l'utilisateur.

Par exemple, si votre réfrigérateur avait la capacité d'envoyer une notification lorsque votre fils a consommé la dernière bouteille de lait, vous pourriez en acheter en revenant du travail afin d'éviter de ressortir le soir.

Comme l'IoT est un sujet d'actualité, nous avons décidé de vous proposer une application basée sur le RaspberryPi. Nous allons utiliser la carte d'expérimentation **ET1060M** pour RaspberryPi afin de simuler des conditions liées à la variation de grandeurs physiques telles que la température, et envoyer des alertes en fonction de seuils prédéfinis.



Cette application vous permettra d'acquiescer les compétences nécessaires pour l'adapter à vos besoins et développer ainsi d'autres applications plus complexes. L'idée d'alerter un utilisateur distant n'est pas nouvelle, les systèmes les plus couramment utilisés à ce jour le font à l'aide de SMS. De nos jours, grâce à la diffusion des smartphones, nous avons la possibilité d'envoyer une notification push.

La carte d'extension ET1060M

Cette carte d'extension convient parfaitement pour mettre en œuvre ce projet, elle vient se relier au connecteur GPIO du RaspberryPi. Elle comporte un **convertisseur analogique/digital** (ADC) à **4 canaux** de 8 bits et un **convertisseur digital/analogique** (DAC) sur 8 bits.

Avec cette carte, il est possible de réaliser un système de contrôle à distance et d'acquisition de données à partir par exemple d'une page Web qui gère les protocoles de communication et l'intégration des données dans une base de données.

La carte d'extension utilise deux tensions d'alimentation de 3,3 V et 5 V. La tension de 3,3 V alimente le **PCF8591** qui est un circuit intégré convertisseur ADC/DAC sur 8 bits, les capteurs de température et de lumière.

Étant donné le faible courant fourni par la ligne 3,3 V du RaspberryPi, nous avons préféré intercaler un circuit de puissance entre les sorties digitales et les LED. Les transistors de puissance sont alimentés à partir d'une tension de 5 V. Le bus I2C fonctionne sous 3,3 V.

Analysons le schéma électrique visible en figure 1. La broche 2 du RaspberryPi est capable de fournir 500 mA pour la version « Rev.1 » et 300 mA pour la « Rev.2 ».

Le capteur de température est constitué par une **CTN** de **10 kΩ** qui est reliée à la broche 1 du PCF8591 par l'intermédiaire d'un diviseur de tension formé par R16 et elle-même. La tension de référence « **VREF** », pour la mesure

des entrées analogiques, est assurée par la résistance de limitation R18, par la diode zener DZ1 et le condensateur de filtrage C1 qui maintient la tension stable. Le circuit de thermostat ainsi constitué est complété par 2 LED et les composants de puissance associés. LD2, qui simule un refroidissement, est reliée à la broche 16 (GPIO23) et LD3 connectée à la broche 18 (GPIO24) simule un réchauffement.

Chaque circuit de puissance est réalisé à l'aide d'un transistor BC547. Lorsque le niveau de la broche de sortie du GPIO est à un niveau bas, la **base** du transistor (T2 et/ou T3) **n'est pas polarisée**, le transistor **ne conduit pas**. Lorsque le niveau de la broche prend une valeur élevée (niveau haut), la tension sur la base du transistor dépasse le seuil de conduction du transistor, le **collecteur** de ce dernier **se trouve à la masse** et donc un **courant traverse la résistance** (R6 et/ou R9), ce qui permet à la LED correspondante d'être alimentée par la tension de 5 V, à travers une résistance de 330 Ω.

La **mesure de la luminosité** s'effectue à l'aide de la **photorésistance** qui est reliée à la broche 2 du PCF8591 au moyen du diviseur de tension composé par R17 et la photorésistance. Lorsque la **luminosité varie**, la **valeur de la résistance** de la photorésistance **LDR varie**, ainsi que la **valeur de la tension mesurée sur l'entrée analogique « AIN1 »**.

La LED flip/flop (bascule) est commandée par le bouton poussoir P1, qui est relié à la broche 15 (GPIO22) du RaspberryPi. La LED correspondante LD1 est connectée à la broche 12 (GPIO18) par l'intermédiaire du transistor T1. L'interrupteur SW1 est relié à la broche 11 (GPIO17) du RaspberryPi. La LED correspondante est LD5, qui est reliée à la broche 7 (GPIO4) à travers le transistor T5.

L'existence d'un message est indiquée par l'allumage de la LED LD4, reliée à la broche 22 (GPIO25), par l'intermédiaire du transistor T4. Pour voir le message, il faut appuyer sur le bouton P2 qui est connecté à la broche 13 (GPIO27).

La **sortie du DAC**, qui s'effectue sur la broche « **AOUT** » du PCF8591, est

reliée à la LED LD6 à travers la résistance R19. En fonction du niveau de la tension présente sur la sortie « **AOUT** », la LED LD6 s'allume de manière plus ou moins importante.

Les lignes du **bus I2C**, c'est-à-dire **SDA** et **SCL**, sont respectivement reliées aux broches 3 et 5 du connecteur GPIO du RaspberryPi.

Les **broches d'adressage A0, A1 et A2** peuvent être reliées à la masse à l'aide d'un DIP switch de manière à obtenir **8 (2³) adresses différentes**. Dans notre projet, nous avons relié tous ces broches à la masse à l'aide du DIP1 (tous sur ON).

Les deux entrées analogiques non utilisées « **AIN2** » et « **AIN3** » sont reliées à la masse, afin d'éviter des tensions flottantes.

Concernant la réalisation pratique de cette carte d'extension, nous vous conseillons de l'acheter en kit. Celui-ci contient tous les composants ainsi que le circuit imprimé double face fabriqué pour quelques dizaines d'euros. En figure 3 vous pouvez voir le plan de câblage des composants et en figure 2 une photographie d'un prototype.

Une fois la carte d'extension montée avec soins, vous pouvez la relier au RaspberryPi. Si ce dernier est en fonctionnement, vous devez l'éteindre avec les commandes suivantes :

shutdown -h now

ou encore

halt

Ensuite débranchez l'alimentation. Reliez le connecteur « **RSPY** » de la carte d'extension au connecteur GPIO du RaspberryPi et mettez sous tension.

Maintenant, vous devez activer le module de gestion du bus I2C (dans le monde de Windows cela correspond aux pilotes ou drivers). Le module de gestion du bus I2C est un dispositif qui gère toutes les ressources et les périphériques connectés au RaspberryPi.

Ce module fait partie de ceux compilés dans la distribution Raspian sous la forme d'un module externe et ne fait

Plan de montage de la carte ET1060M

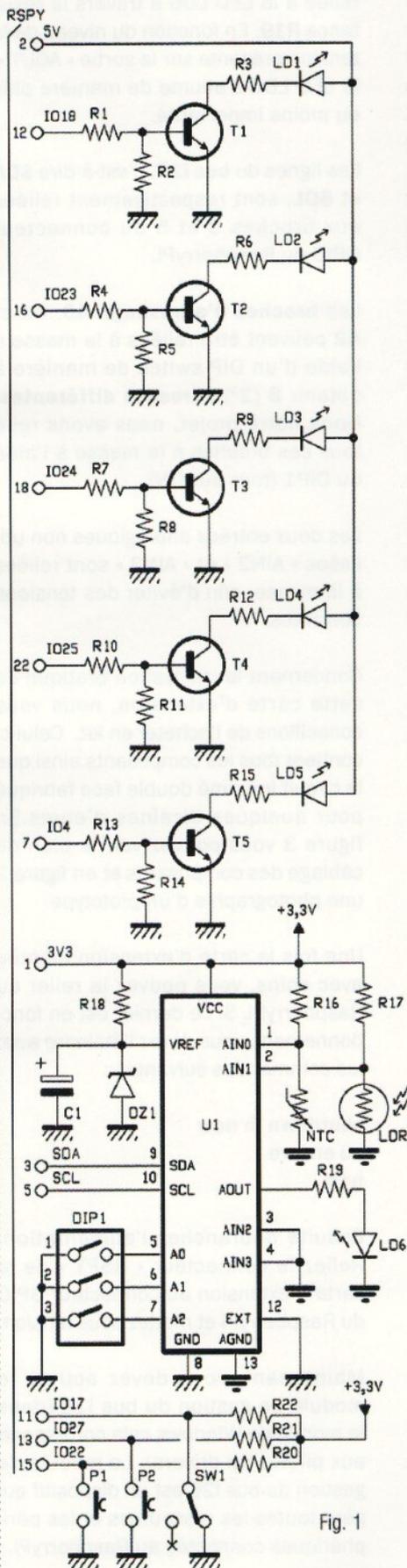


Figure 1 : schéma électrique de la carte d'extension ET1060M pour le RaspberryPi.

Figure 2 : photographie d'un prototype de la carte d'extension ET1060M.

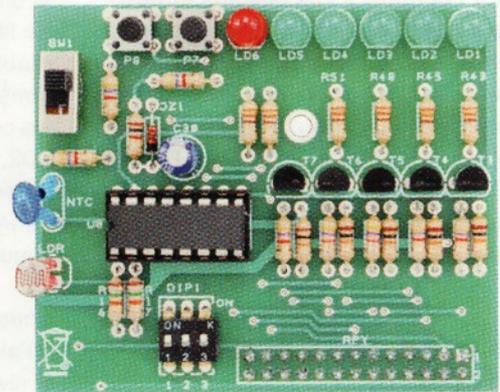
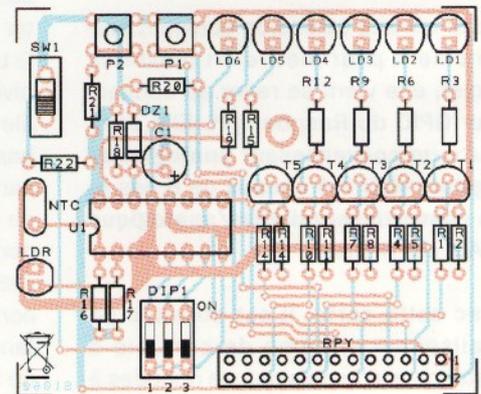


Figure 3 : plan de câblage des composants de la carte d'extension ET1060M.



Liste des composants de la carte ET1060M

- R1..... 4,7 kΩ
- R2..... 10 kΩ
- R3..... 330 Ω
- R4..... 4,7 kΩ
- R5..... 10 kΩ
- R6..... 330 Ω
- R7..... 4,7 kΩ
- R8..... 10 kΩ
- R9..... 330 Ω
- R10..... 4,7 kΩ
- R11..... 10 kΩ
- R12..... 330 Ω
- R13..... 4,7 kΩ
- R14..... 10 kΩ
- R15..... 330 Ω
- R16..... 15 kΩ
- R17..... 15 kΩ
- R18..... 100 Ω
- R19..... 330 Ω
- R20..... 4,7 kΩ
- R21..... 4,7 kΩ
- R22..... 4,7 kΩ

C1..... 4,7 μF/16 V électrolytique

DZ1..... zener 3,3 V 400mW

- T1 BC547
- T2 BC547

- T3 BC547
- T4 BC547
- T5 BC547

- LD1.... LED 5 mm verte
- LD2.... LED 5 mm verte
- LD3.... LED 5 mm verte
- LD4.... LED 5 mm verte
- LD5.... LED 5 mm verte
- LD6.... LED 5 mm rouge

U1.....PCF8591

- P1..... microswitch
- P2..... microswitch

SW1... interrupteur

DIP1... DIP switch 3 voies

NTC..... CTN 10 kΩ

LDR.... photorésistance 2-20 kΩ

Divers

- Support circuit intégré 2 x 8 broches
- Barrette femelle 2 x 13 voies pour RaspberryPi
- Entretoise M/F 16mm
- Vis 8 mm 3 MA (x 2)

pas partie intégrante du noyau (kernel). En outre, dans la distribution Raspbian, le module de gestion du bus I2C est par défaut dans la « blacklist » et n'est donc pas activé en utilisation normale.

En effet, les lignes du bus I2C partagent certaines broches digitales du connecteur GPIO, ces dernières sont prédominantes dans la configuration par défaut.

Donc pour activer le module de gestion du bus I2C, il est nécessaire de le retirer de la « blacklist » (liste noire), puis de l'ajouter à l'ensemble des modules reconnus par le noyau. Pour cela, ouvrez le fichier de configuration qui contient la liste des modules « blacklists » avec la commande suivante (voir la figure 4) :

nano /etc/modprobe.d/raspi-blacklist.conf

« Nano » est un éditeur de texte minimaliste qui fonctionne dans une fenêtre de terminal. Supprimez le module I2C de la liste noire en effaçant la ligne ou, comme nous le préférons, en mettant la ligne en commentaire en insérant un caractère « # » en début de ligne (voir la figure 5).

Appuyez sur « Ctrl + X » puis « Y » lorsque vous êtes invité à enregistrer le fichier après la modification. Vous devez effectuer un redémarrage pour que les modifications prennent effet.

Maintenant, vous devez faire en sorte que le module de gestion du bus I2C soit chargé et qu'il fasse partie intégrante du noyau (kernel). Pour cela, il existe deux possibilités.

La première permet de charger le module afin qu'il soit activé aussi longtemps que le RaspberryPi reste allumé. Au prochain démarrage, le module devra être rechargé à nouveau à l'aide d'une commande. La seconde possibilité permet de charger le module automatiquement au démarrage du système d'exploitation, et de mettre le module de gestion du bus I2C à disposition des applications immédiatement après le démarrage, une condition indispensable dans un système de serveur sans surveillance.

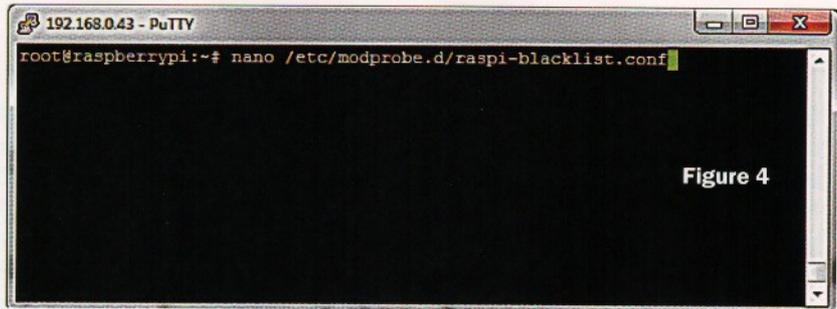


Figure 4

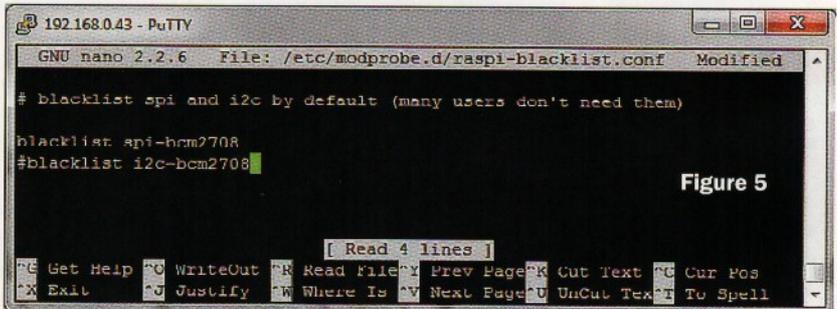


Figure 5

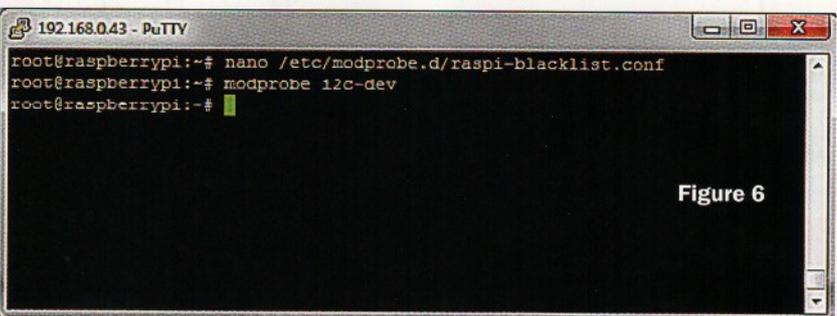


Figure 6

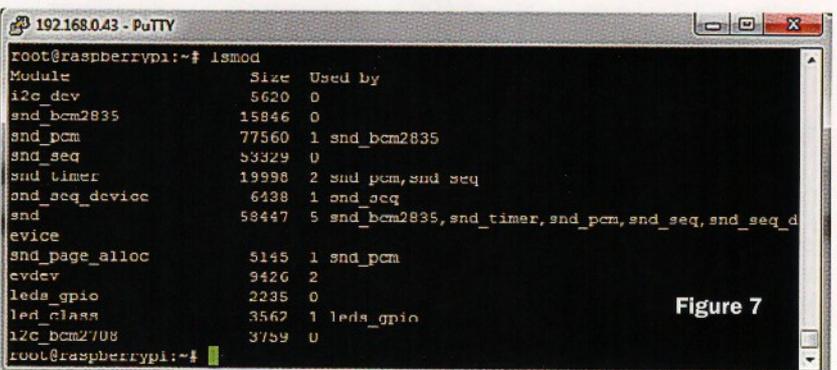


Figure 7

La première option nécessite l'utilisation de la commande « modprobe », tapez la commande suivante (voir la figure 6) :

modprobe i2c-dev

Vous pouvez voir l'activation du driver avec la commande qui affiche la liste de tous les modules installés (voir la figure 7), tapez :

lsmod

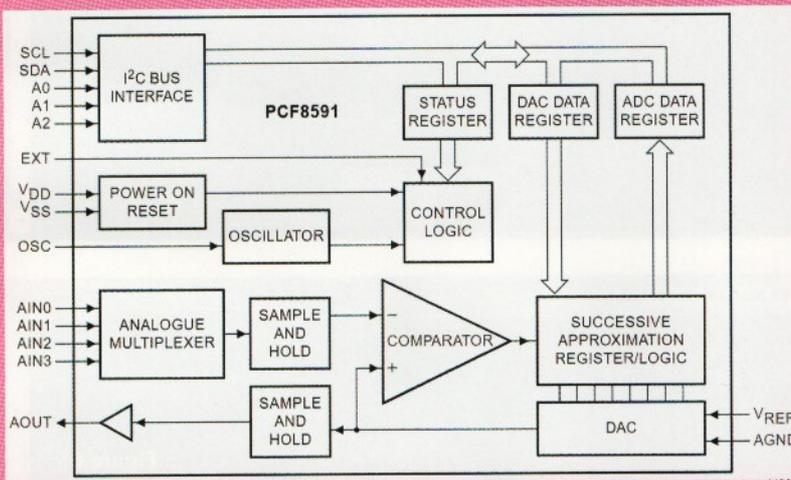
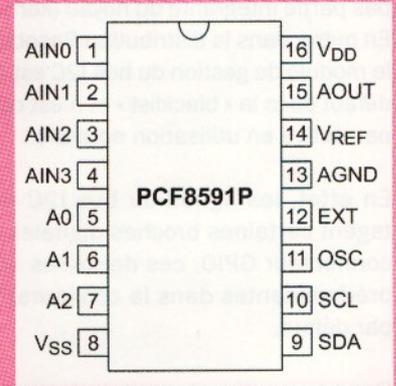
Sous Linux, tout ou presque, est sous forme de fichier. Allez dans le dossier

« /dev » pour voir les fichiers de liaison vers le périphérique « I2c-0 » et « I2c-1 ». La commande « modprobe » permet de charger et décharger les modules en cours d'exécution et maintient ses effets tant que le RaspberryPi reste allumé. Lorsqu'il est arrêté ou en cas de redémarrage, le module devra être rechargé manuellement.

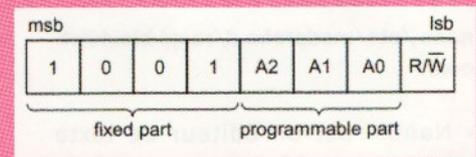
Cette condition n'est pas appropriée pour fonctionner avec une application autonome. La commande « modprobe », avec l'option « -r » (pour remove ou suppression), peut également être utilisée

Le circuit intégré PCF8591

Le PCF8591 est un circuit intégré de type CMOS d'acquisition de données sur 8 bits. Il est alimenté sous une tension unique et dispose de 4 entrées analogiques, d'une sortie analogique et d'un bus de communication I2C. L'adresse physique du circuit est déterminée au moyen de 3 broches, cela permet d'utiliser jusqu'à 8 PCF8591 sur le même bus I2C. La fréquence d'échantillonnage dépend de la vitesse maximale du bus I2C. La tension d'alimentation est comprise entre 2,5 V et 6 V, avec une consommation en veille de l'ordre d'une dizaine de microampères. Les entrées analogiques sont multiplexées, et travaillent avec un échantillonnage continu dont la précision est incrémentale. Elles peuvent être configurées en mode différentiel ou asymétrique. Vous pouvez voir en figure A le brochage du PCF8591 et en figure B son synoptique.



La communication avec le circuit intégré s'effectue via le bus I2C, avec l'envoi des registres d'état, du contrôle du convertisseur ADC et des données du DAC sur deux octets pour chacun d'eux. La configuration de l'adresse physique du PCF8591 s'effectue à l'aide des broches d'adresse A0, A1 et A2 afin qu'il soit reconnu sur le bus I2C.



contrôle, soit un seul canal ou tous les canaux multiplexés. Les entrées, grâce au registre de contrôle, peuvent être configurées soit comme des entrées asymétriques ou différentielles. Les mesures s'effectuent par approximations successives, et sont conservées dans un registre spécial. La valeur du DAC, s'il est activé, est convertie en une tension disponible sur la sortie « AOUT ». Le convertisseur ADC fonctionne sur 8 bits, les valeurs lues correspondent à la tension d'entrée multipliée par (VREF/256), tandis que la tension de sortie du convertisseur DAC correspond à la valeur hexadécimale (comprise entre 1 et 256) multipliée par (VREF/256).

Le registre d'adresses

Le registre d'adresse est utilisé pour identifier le PCF8591 sur le bus I2C. Il se compose de 8 bits, dont les 4 premiers sont fixés par le fabricant et représentent le code unique attribué à ce périphérique, pour le PCF8591 les 4 premiers bits sont : « 1001 » (configuration d'usine). Les 3 bits suivants déterminent la configuration, c'est-à-dire l'adresse que vous allez adopter pour le PCF8591 à l'aide des 3 broches A0, A1 et A2. Le bit « R/W » indique la direction en lecture (1) ou en écriture (0) de l'octet de configuration et de contrôle qui suit dans la communication (voir la figure C).

pour désactiver un module précédemment chargé.

modprobe -r i2c-dev

Si vous voulez que le module soit chargé au démarrage du RaspberryPi, il est nécessaire de charger de manière permanente le module I2C. Cela est effectué en modifiant de manière appropriée le fichier de configuration « etc/modules » qui contient la liste des pilotes à charger au démarrage (voir la figure 9) :

nano /etc/modules

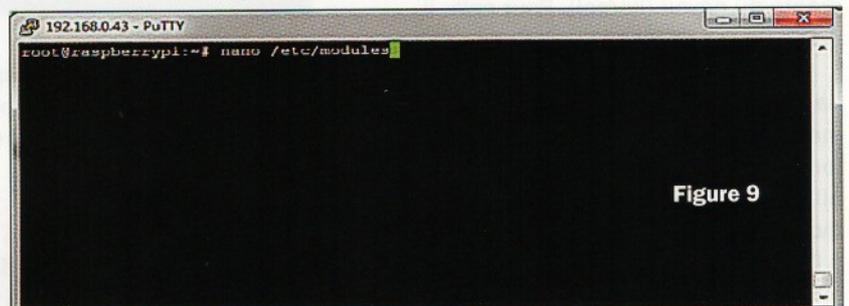


Figure 9

et d'ajouter l'expression « i2c-dev » au fichier de configuration.

Appuyez sur « CTRL + X » puis sur « Y » pour enregistrer la modification apportée au fichier (voir la figure 10).

Maintenant, installez le paquet « i2c-tools » qui fournit une série de fonctions sous la forme de lignes de commandes permettant de vérifier le bon fonctionnement du bus I2C (voir la figure 11) :

Registres de configuration et de contrôle

La gestion du registre de configuration est un peu plus complexe, il est utilisé pour définir les modes de fonctionnement du convertisseur ADC et pour la lecture du buffer d'un canal spécifique. L'octet de configuration est envoyé au périphérique (PCF8591) après l'octet d'adressage et correspond à la concaténation des différents paramètres exprimés dans un format binaire comme visible en haut de la figure D.

Le premier bit à gauche (msb) a toujours la valeur 0. Le deuxième bit indique si la sortie analogique du DAC est activée (valeur égale à 1) ou non (valeur égale à 0), le troisième et le quatrième bit indiquent la configuration des entrées des convertisseurs. La valeur « 00 » configure les 4 entrées en asymétrique (non différentielles).

Le cinquième bit doit toujours être à « 0 » et le sixième correspond au flag d'auto incrémentation. L'activation de l'auto incrément (valeur du bit égal à 1) permet d'incrémenter automatiquement le numéro du canal à chaque conversion analogique/numérique. Dans notre cas, nous fixons la valeur des 2 derniers bits (7 et 8) à 0 et nous rafraîchissons la valeur chaque fois que nous voulons lire un canal. Les deux derniers bits sont donc utilisés pour choisir le canal du convertisseur A/D que nous voulons utiliser, nous avons 4 possibilités :

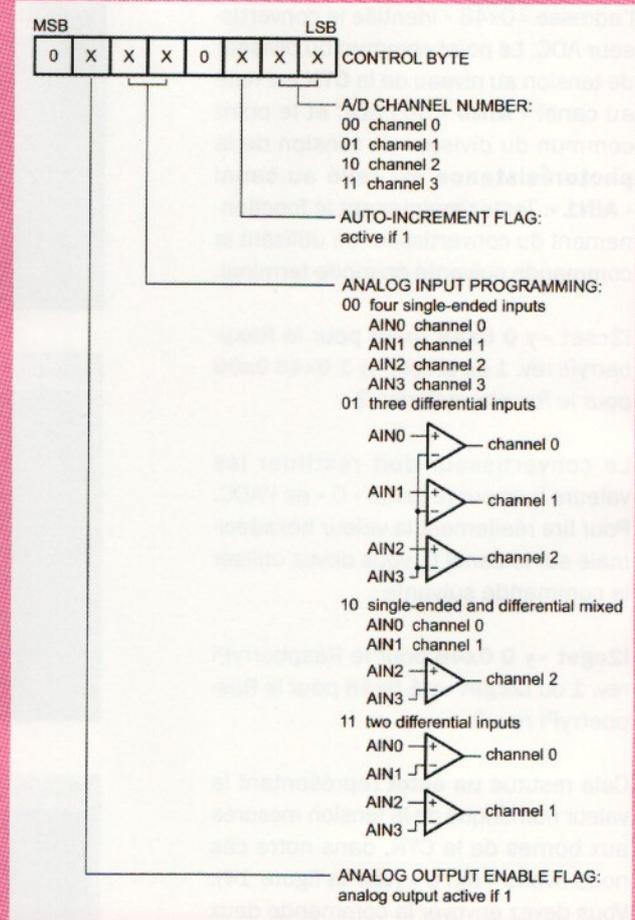
- 00 = canal 1
- 01 = canal 2
- 10 = canal 3
- 11 = canal 4

La conversion digitale/analogique

Le troisième octet envoyé contient la valeur hexadécimale mémorisée dans le décodeur DAC convertie en une valeur analogique correspondante, qui est maintenue jusqu'à la prochaine conversion. La valeur mémorisée est convertie en une tension sur la sortie du DAC si le bit d'activation de la sortie analogique est mis à 1, dans le cas contraire la conversion est désactivée. L'octet peut prendre les valeurs comprises entre 0 et 255, soit 00 et FF en hexadécimal, afin d'obtenir en sortie une tension analogique variant de 0 V à V_{ref} .

Par exemple, si nous voulons lire le canal « 0 » en activant la sortie du DAC, nous devons configurer le registre avec la séquence suivante : 0-1-00-0-0-00.

En valeur hexadécimale cela correspond à la valeur « 0x40 » du registre. Pour lire le canal « 0 » en désactivant la sortie du DAC, nous devons paramétrer le registre avec la séquence : 0-0-00-0-0-00. En valeur hexadécimale cela correspond à la valeur « 0x00 ».



apt-get install i2c-tools

Ajoutez l'utilisateur « pi » au groupe I2C (voir la figure 12) :

adduser pi i2c

Redémarrez le Raspberry Pi pour activer la nouvelle configuration en utilisant la commande suivante :

reboot

Tapez la commande suivante :

modprobe i2c-dev

```
192.168.0.43 - PuTTY
GNU nano 2.2.6 File: /etc/modules Modified
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-dev
```

Figure 10

Après le redémarrage du RaspberryPi et en vous connectant via « Putty » ou « Kitty », vérifiez si sur le bus I2C le convertisseur ADC est visible, tapez la commande : **i2cdetect -y 0** pour le

RaspberryPi rev. 1 ou **i2cdetect -y 1** pour le RaspberryPi rev. 2

Vous devriez obtenir un résultat similaire à celui représenté en figure 13 où

l'adresse « 0x48 » identifie le convertisseur ADC. Le point commun du diviseur de tension au niveau de la CTN est relié au canal « AINO » de l'ADC et le point commun du diviseur de tension de la photorésistance est relié au canal « AIN1 ». Testez maintenant le fonctionnement du convertisseur en utilisant la commande suivante en mode terminal.

i2cset -y 0 0x48 0x00 pour le RaspberryPi rev. 1 ou **i2cset -y 1 0x48 0x00** pour le RaspberryPi rev. 2

Le convertisseur doit restituer les valeurs lues sur le canal « 0 » de l'ADC. Pour lire réellement la valeur hexadécimale sur le canal 0, vous devez utiliser la commande suivante :

i2cget -y 0 0x48 pour le RaspberryPi rev. 1 ou **i2cget -y 1 0x48** pour le RaspberryPi rev. 2

Cela restitue un octet représentant la valeur numérique de la tension mesurée aux bornes de la CTN, dans notre cas nous avons « 0x70 » (voir la figure 14). Vous devez envoyer la commande deux fois car l'ADC renvoie toujours la valeur « 0x80 » à la première lecture.

Pour vérifier le fonctionnement, envoyez plusieurs commandes en tenant la CTN entre vos doigts. Vous devriez voir les diverses valeurs hexadécimales lues par le convertisseur.

Maintenant, installez le package « bc » qui est un programme générique de calcul sous la forme de lignes de commandes. Il permet des calculs et des conversions dans toutes les bases, tapez la commande suivante :

apt-get install bc

Pour convertir une valeur hexadécimale en une valeur décimale, utilisez la commande suivante :

echo 'ibase=16;obase=A;70' | bc

Cette ligne de commande renvoie la valeur « 112 » (voir la figure 15). La commande affiche dans la fenêtre de terminal l'opération de conversion de la base « 16 » en base « 10 » (A en hexadécimal) de la valeur « 70 » (il est obligatoire d'utiliser les majuscules pour les

lettres) et la transmet au programme « bc ». Par exemple, pour la conversion binaire vers hexadécimale, utilisez la commande suivante :

echo 'ibase=2;obase=10000;bbbbbbb' | bc

en remplaçant l'expression « bbbbbbbb » par la valeur binaire que vous

voulez convertir. Pour une conversion hexadécimale vers binaire, utilisez la commande suivante :

echo 'ibase=16;obase=2;hh' | bc

en remplaçant l'expression « hh » par la valeur hexadécimale que vous voulez convertir. Effectuez le même test sur le canal « AIN1 » de l'ADC pour

```

192.168.0.43 - PuTTY
root@raspberrypi:~# apt-get install i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  libi2c-dev python-ambus
The following NEW packages will be installed:
  i2c-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 59.5 kB of archives.
After this operation, 223 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main i2c-tools armhf
3.1.0-2 [59.5 kB]
Fetched 59.5 kB in 1s (55.5 kB/s)

```

Figure 11

```

192.168.0.43 - PuTTY
After this operation, 223 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main i2c-tools armhf
3.1.0-2 [59.5 kB]
Fetched 59.5 kB in 1s (55.5 kB/s)
Selecting previously unselected package i2c-tools.
(Reading database ... 61438 files and directories currently installed.)
Unpacking i2c-tools (from ../i2c-tools_3.1.0-2_armhf.deb) ...
Processing triggers for man-db ...
Setting up i2c-tools (3.1.0-2) ...
root@raspberrypi:~#
root@raspberrypi:~#
root@raspberrypi:~# adduser pi i2c
Adding user 'pi' to group 'i2c' ...
Adding user pi to group i2c
Done.
root@raspberrypi:~#

```

Figure 12

```

192.168.0.43 - PuTTY
root@raspberrypi:~# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@raspberrypi:~#

```

Figure 13

```

192.168.0.43 - PuTTY
root@raspberrypi:~# i2cset -y 1 0x48 0x00
root@raspberrypi:~# i2cget -y 1 0x48 0x48
0x80
root@raspberrypi:~# i2cget -y 1 0x48 0x48
0x70
root@raspberrypi:~#

```

Figure 14

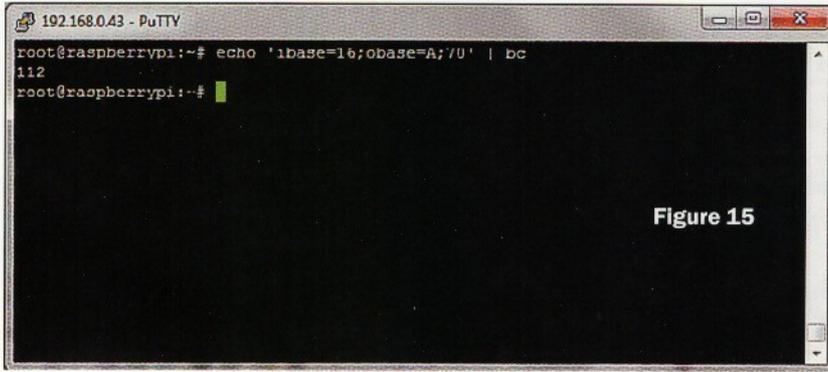


Figure 15

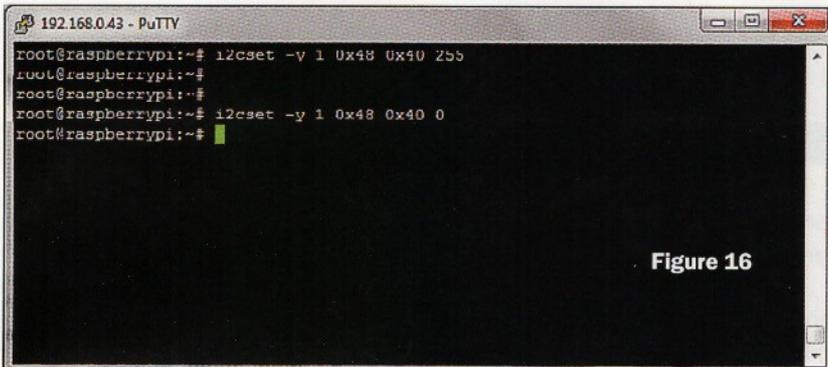


Figure 16

vérifier le bon fonctionnement du circuit de la photorésistance LDR. Pour tester la sortie du convertisseur DAC, utilisez la commande suivante (voir la figure 16) :

i2cset -y 1 0x48 0x40 255

Pour la signification des valeurs des registres, reportez-vous à l'encadré dédié au PCF8591 (voir plus haut dans l'article). Dans cette commande, la valeur « 0x48 » correspond à l'adresse

en hexadécimal du convertisseur sur le bus I2C, la seconde valeur « 0x40 » incrémente le bit qui active le DAC afin d'obtenir sur la sortie « AOUT » une tension correspondant à la valeur hexadécimale prédéfinie, tandis que la valeur 255 correspond à la valeur à transmettre au DAC pour être convertie en tension. Vous devriez voir la LED rouge s'allumer. Pour l'éteindre, tapez la commande suivante :

i2cset -y 1 0x48 0x40 0

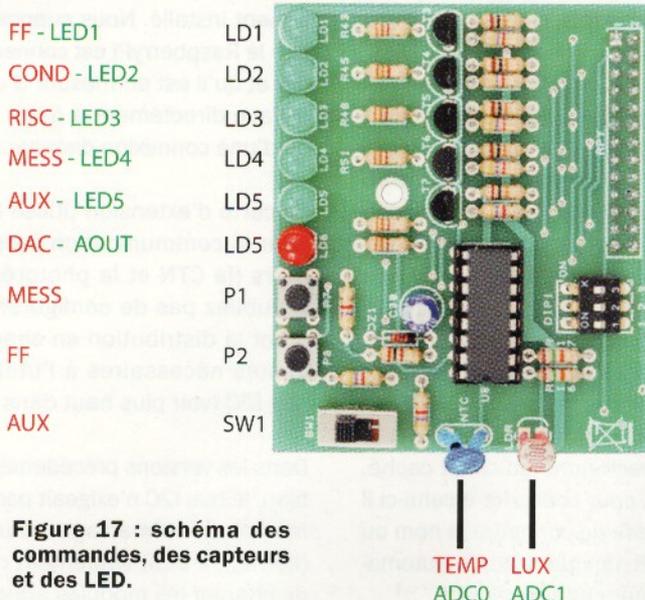


Figure 17 : schéma des commandes, des capteurs et des LED.

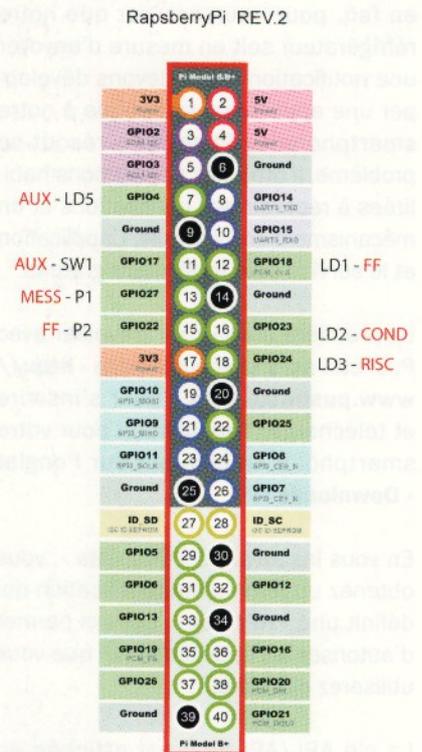
Notification push : mais de quoi s'agit-il ?

Maintenant que nous venons d'étudier et de réaliser notre carte d'interface pour le RaspberryPi, nous allons approfondir le concept de notification push. Le terme « notification push » s'identifie à un mécanisme précis qui permet d'envoyer des messages à des appareils connectés au réseau Internet.

La diffusion des notifications push a été adoptée dans les systèmes d'exploitation des smartphones, en particulier le premier à les introduire a été Apple dans sa version 3.0 d'iOS. La notification push est différente des méthodes mentionnées ci-dessus (SMS, etc.) pour diverses raisons.

La plus importante est sans doute son modèle de fonctionnement. Pour envoyer un SMS, il suffit de connaître le numéro du destinataire, de même que pour envoyer un e-mail, il suffit de connaître l'adresse du destinataire.

Dans le cas de la notifications push, le mécanisme est techniquement appelé « publish/subscribe ».



Attention pour la version REV.1, le brochage est différent.

Figure 18 : connexions avec la carte en fonction des versions du RaspberryPi.

L'utilisateur s'abonne à un **topic** (sujet) pour lequel **il souhaite recevoir des notifications**, il peut à tout moment annuler son abonnement.

Le fonctionnement de la notification push est assez complexe, mais heureusement, nous pouvons décrire l'application sans entrer dans les détails techniques. Cependant, nous allons vous présenter « Pushetta », qui est un service ayant comme tâche spécifique de mettre en œuvre facilement par n'importe quel utilisateur la notification push.

« **Pushetta** » peut être considéré comme une sorte de **passerelle** (gateway) de **communication intermédiaire** entre l'**expéditeur** et le **destinataire** des notifications.

Le problème fondamental des notifications push est qu'il n'y a pas d'application spécifique pour les recevoir, comme par exemple une messagerie e-mail. C'est plutôt un outil de développement d'applications qui configure le format des données à envoyer et qui gère la façon de les interpréter à la réception.

Cela implique une complication majeure pour le scénario initialement décrit, en fait, pour nous assurer que notre réfrigérateur soit en mesure d'envoyer une notification, nous devons développer une application spécifique à notre smartphone ! « Pushetta » résout ce problème, il offre des applications habilitées à recevoir des notifications et un mécanisme d'envoi simple. L'application et le service sont totalement gratuits.

La première étape pour travailler avec Pushetta est d'aller sur la page « <http://www.pushetta.com> » pour s'inscrire et télécharger l'application pour votre smartphone en cliquant sur l'onglet « **Downloads** ».

En vous inscrivant à « Pushetta », vous obtenez un jeton d'authentification qui définit une « **API Key** ». Celle-ci permet d'autoriser les appels de l'API que vous utiliserez plus tard.

La clé API (API Key) est affichée en accédant à votre tableau de bord sur le site (voir la figure 19). Notez soigneusement l'API Key afin de ne pas revenir à chaque fois au tableau de

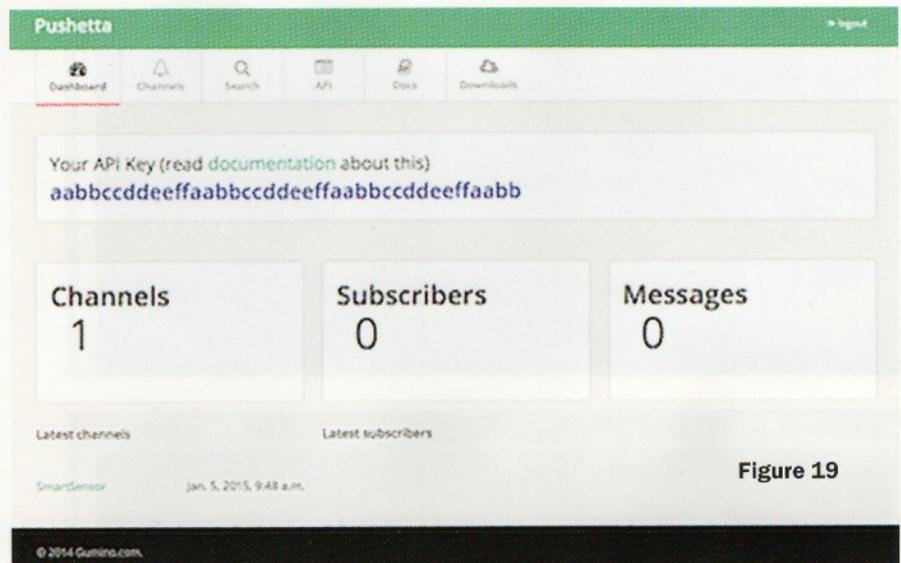


Figure 19

bord lorsque vous écrirez le code pour envoyer des notifications.

L'envoi de notifications avec « Pushetta » est basé sur le concept de « **canal** ». Reprenons un instant l'exemple du réfrigérateur « intelligent ». Nous devons créer un canal qui servira d'identification pour le mécanisme « publish/subscribe ». Par exemple, nous l'appellerons « Frigo ». Lorsque nous enverrons une notification, nous spécifierons l'utilisation du canal « Frigo ». À partir de l'application, nous souscrivons le même canal et chaque notification envoyée sera délivrée automatiquement.

La création d'un canal est une procédure simple. À partir de l'onglet « **Channels** », vous pouvez accéder à votre liste des canaux (lorsque vous vous inscrivez pour la première fois, cette liste est évidemment vide).

À l'aide du bouton « **Add a channel** » (ajouter un canal), vous accédez à une page qui permet de créer un nouveau canal. Pour cela vous devez l'identifier à l'aide d'une image, d'un nom qui n'est déjà pas utilisé par d'autres, et d'une description. Vous pouvez créer des canaux publics ou privés, ces derniers exigent l'autorisation du propriétaire.

Enfin, vous pouvez décider de garder le système de recherche du canal caché. Dans ce cas, pour souscrire à celui-ci il sera nécessaire de connaître le nom ou d'utiliser l'URL unique générée automatiquement pour chaque canal.

Configurons le RaspberryPi

Maintenant passons à la configuration du RaspberryPi, nous n'expliquerons pas celle de base. Pour cela reportez-vous aux explications décrites dans le chapitre consacré à la carte d'extension (un peu plus haut dans l'article). Nous nous limiterons à fournir uniquement les indications spécifiques pour notre projet.

Dans notre exemple, nous utilisons la version de Raspbian distribuée au moment où ces pages sont écrites (début 2017). Gardez à l'esprit que selon votre version, des ajustements mineurs peuvent être nécessaires.

Nous partons de l'hypothèse que le RaspberryPi est parfaitement fonctionnel avec son système d'exploitation correctement installé. Nous supposons aussi que le RaspberryPi est connecté à Internet et qu'il est en mesure d'utiliser l'interface directement à partir du clavier ou d'une connexion distante via SSH.

La carte d'extension utilise le bus I2C pour la communication avec ses capteurs (la CTN et la photorésistance). N'oubliez pas de configurer correctement la distribution en chargeant les pilotes nécessaires à l'utilisation du bus I2C (voir plus haut dans l'article).

Dans les versions précédentes de Raspbian, le bus I2C n'exigeait pas des paramètres de démarrage pour le noyau (kernel), il était seulement nécessaire de charger les modules adéquates.

Dans la dernière version, la situation a un peu changé, car bien que les modules doivent toujours être chargés, vous avez besoin d'un paramètre de démarrage spécifique pour le noyau afin d'activer le bus I2C. Pour cette raison, nous devons modifier le fichier « **/boot/config.txt** » en ajoutant la ligne suivante : **dtparam=i2c_arm=on** comme le montre la figure 21.

Pour modifier les différents fichiers, nous utilisons l'éditeur de texte « Nano ». Sans entrer dans les détails, vous devez savoir comment ouvrir un fichier. Spécifiez « **nano nom du fichier** » et enregistrez vos modifications en appuyant sur « **CTRL + X** ».

L'éditeur possède beaucoup d'autres fonctions que nous ne détaillerons pas ici. Reportez-vous aux anciennes revues d'Électronique et Loisirs Magazine (à partir du numéro 123).

Une dernière remarque avant de poursuivre, certains fichiers ne peuvent pas être modifiés par un utilisateur s'il n'est pas administrateur du système. Dans ce cas, nous utiliserons l'instruction : « **sudo nano nom du fichier** » qui, à la demande du mot de passe, exécutera l'éditeur comme s'il était lancé à partir de « root ».

Comme prévu, nous avons besoin de permettre le chargement des modules. Dans ce cas, nous devons modifier le fichier « **/etc/modules** », dans lequel sont répertoriés les modules à charger au démarrage.

Les modules peuvent aussi être chargés manuellement à l'aide de la commande « **modprobe** », mais cela nécessiterait d'exécuter la commande à chaque (re) démarrage du RaspberryPi. Il est donc préférable d'activer le chargement automatique des modules.

Exécutons la commande « **sudo nano /etc/modules** » et ajoutons les expressions suivantes comme visible en figure 22 :

i2c-bcm2708
i2c-dev

Si nous effectuons maintenant un redémarrage du RaspberryPi, le système

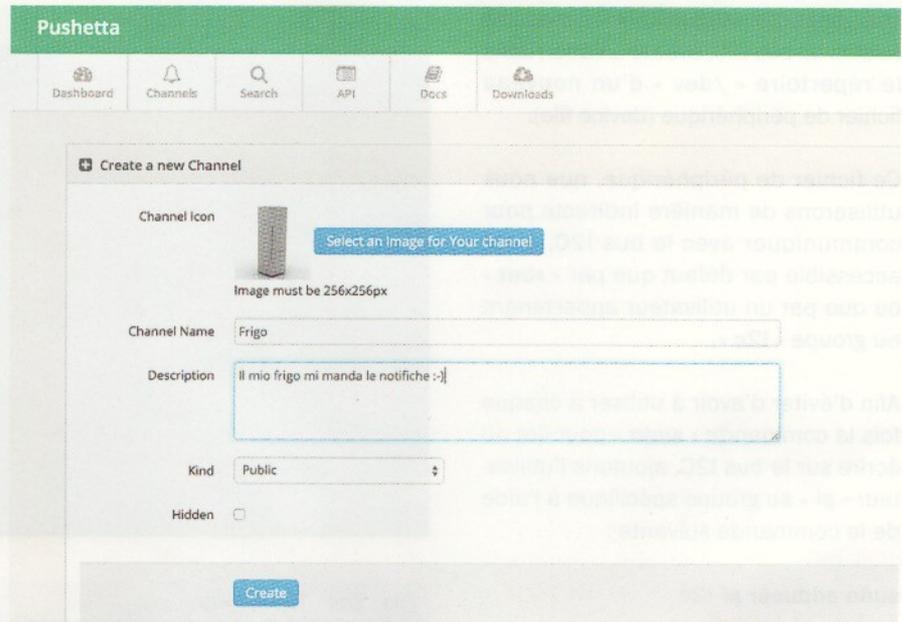


Figure 20 : les différentes étapes pour utiliser le service « Pushetta ». Une fois que vous vous êtes enregistré sur le site, vous pouvez créer un ou plusieurs canaux, le système est alors prêt à envoyer des notifications.

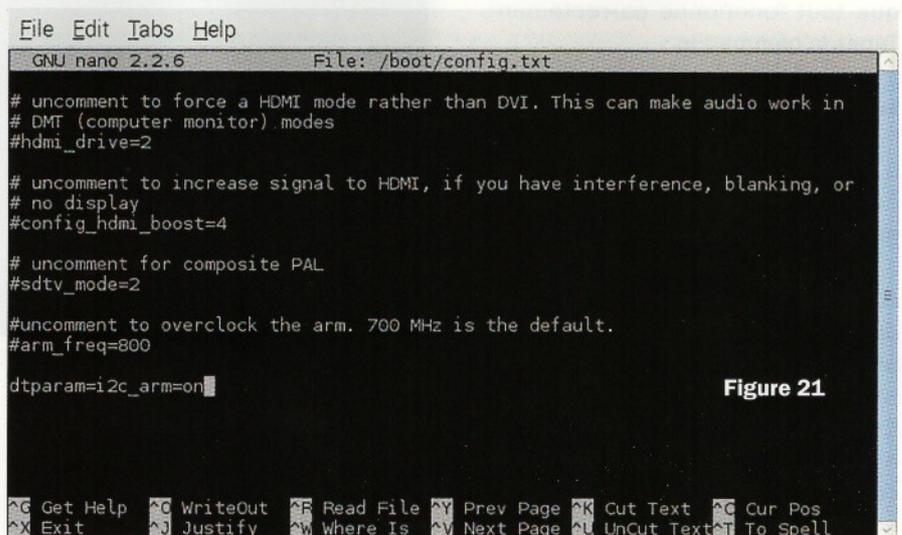


Figure 21

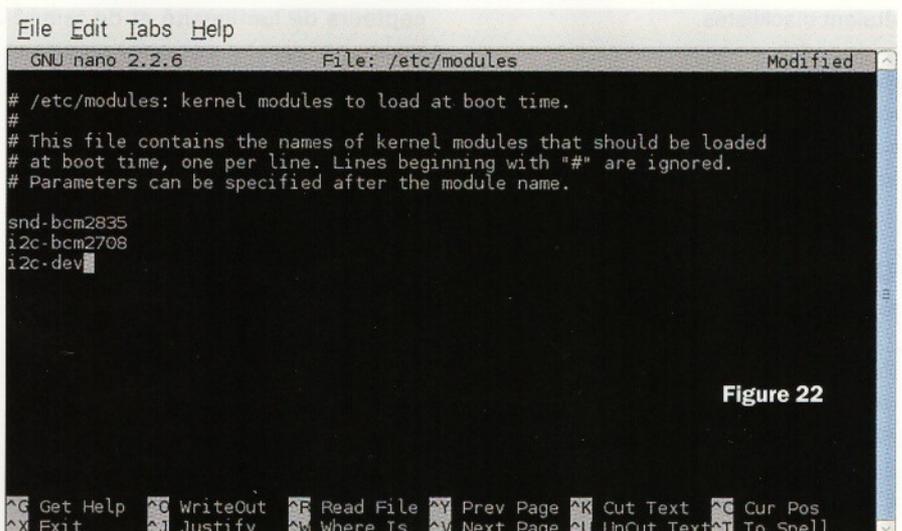


Figure 22

chargera automatiquement les modules relatifs au bus I2C, avec la création dans le répertoire « /dev » d'un nouveau fichier de périphérique (device file).

Ce fichier de périphérique, que nous utiliserons de manière indirecte pour communiquer avec le bus I2C, n'est accessible par défaut que par « root » ou que par un utilisateur appartenant au groupe « i2c ».

Afin d'éviter d'avoir à utiliser à chaque fois la commande « sudo » pour lire ou écrire sur le bus I2C, ajoutons l'utilisateur « pi » au groupe spécifique à l'aide de la commande suivante :

sudo adduser pi i2c

Maintenant, la configuration du bus est terminée. Installons les outils qui nous aideront à effectuer les premiers tests de communication et à vérifier que tout fonctionne correctement. Tapez la commande :

sudo apt-get install i2c-tools

à la fin de l'installation, la distribution est prête pour effectuer nos tests (voir la figure 23).

Comme nous l'avons déjà mentionné, dans les versions précédentes de Raspbian, les étapes de configuration du bus I2C étaient un peu différentes. En particulier, il était nécessaire de modifier le fichier « /etc/modprobe.d/raspi-blacklist.conf » en supprimant les modules « spi-bcm2708 » et « i2c-bcm2708 », afin d'empêcher que leur chargement vienne à se bloquer car ils étaient blacklistés.

Selon la version de Raspbian dont vous disposez, il est préférable de se référer aux indications spécifiques pour la configuration du bus I2C.

Utilisation de la carte d'extension ET1060M

Passons maintenant à la connexion de la carte d'extension ET1060M au RaspberryPi. Cela est très simple car la carte possède un connecteur qui s'enfiche dans le connecteur GPIO du RaspberryPi.

```
File Edit Tabs Help
pi@raspberrypi ~ $ sudo apt-get install i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  libi2c-dev python-smbus
The following NEW packages will be installed:
  i2c-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 59.5 kB of archives.
After this operation, 223 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main i2c-tools armhf 3.1.0-2 [59.5 kB]
Fetched 59.5 kB in 0s (95.2 kB/s)
Selecting previously unselected package i2c-tools.
(Reading database ... 77944 files and directories currently installed.)
Unpacking i2c-tools (from ../i2c-tools_3.1.0-2_armhf.deb) ...
Processing triggers for man-db ...
Setting up i2c-tools (3.1.0-2) ...
/run/udev or .udevdb or .udev presence implies active udev. Aborting MAKEDEV invocation.
```

Figure 23

```
File Edit Tabs Help
pi@raspberrypi ~ $ sudo pip install pushetta
Downloading/unpacking pushetta
  Downloading pushetta-1.0.6.tar.gz
  Running setup.py egg_info for package pushetta

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

Successfully installed pushetta
Cleaning up...
pi@raspberrypi ~ $
```

Figure 24

Selon la version du RaspberryPi que vous possédez (versions B +, A + ou RaspberryPi 2 ou 3), il est nécessaire d'aligner la broche 1 avec celle de la carte. Comme nous l'avons évoqué précédemment, la carte d'extension dispose d'une paire de poussoirs, de capteurs de luminosité et de température, ce dernier est celui que nous allons utiliser.

Réglez les 3 commutateurs du DIP switch sur ON afin de définir l'adresse de bus I2C. Bien que cela ne soit pas obligatoire, cette configuration rendra plus simple l'utilisation du code sans devoir apporter des modifications en fonction de vos besoins.

Nous allons écrire des exemples de code en Python en mettant à profit certaines bibliothèques prêtes à interagir aussi bien avec le service Pushetta qu'avec le bus I2C des capteurs.

Installons la première librairie nécessaire pour interagir avec le service Pushetta. Exécutez la commande suivante visible en figure 24 :

pip install pushetta

Si l'installation s'est correctement déroulée, nous sommes déjà en mesure d'envoyer notre première notification. L'opération ne nécessite que 3 lignes de code dont la première consiste à importer la librairie.

```
from pushetta import Pushetta
p = Pushetta('aabbccddeeff0011223344556677889900aabbcc')
p.pushMessage('Frigo', 'Acheter du lait !')
```

Expliquons ce que le code fait, la première ligne importe la librairie Pushetta à partir du module du même nom, ce qui est assez courant

Listing 1

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

class TemperatureSensor(GenericSensor):

    adc_address = 0x48          # adresse I2C lue avec i2cdetect
    adc_channel = 0x40         # Canal ADC
    V_REF = 3.3                # Tension de référence pour la mesure de l'ADC
    R1 = 15000.0               # Résistance R1
    V_IN = 3.3                 # Tension d'alimentation du diviseur de tension
    R_th0 = 10000.0           # Résistance de référence de la thermistance à 25 °C
    A = 0.00335402            # Steinhart-Hart Constante A
    B = 0.000256985           # Steinhart-Hart Constante B
    C = 2.62013e-6            # Steinhart-Hart Constante C
    D = 6.38309e-8            # Steinhart-Hart Constante D
    pB = 4100.0               # Constante du paramètre B
    K = 6.0                   # Facteur de dissipation K 6 mV C

    def readValue(self):
        bus = SMBus(1) # 0 per le rev.1
        bus.write_byte(self.adc_address, self.adc_channel)
        raw_val = bus.read_byte(self.adc_address)
        raw_val = bus.read_byte(self.adc_address)
        raw_val = bus.read_byte(self.adc_address)

        V = raw_val * self.V_REF / 256.0
        R_th = self.R1 * V / (self.V_IN - V)

        # Calcul des degrés Kelvin avec la formule de Steinhart-Hart

        logR = math.log(R_th / self.R_th0)
        logR2 = logR ** 2
        logR3 = logR ** 3
        Stein = 1.0 / (self.A + self.B * logR + self.C * logR2 + self.D * logR3)

        # Conversion en degrés Celsius
        return round(Stein - 273.15 - V ** 2 / (self.K * R_th), 2)

    def measureUnit(self):
        return ' C'
```

en langage Python. Le code « réel » commence à la ligne suivante, il instancie (initialise un objet à partir d'un ensemble de caractéristiques, appelé « classe ») l'objet pour la communication avec l'API du service Pushetta.

Vous devez **remplacer l'API Key factice** (aabbccdde-ef-f0011223344556677889900aabbcc) de notre exemple **par celle que vous aurez obtenue à partir de votre tableau de bord** du site www.pushetta.com (après inscription).

La dernière ligne effectuée réellement l'appel et déclenche l'envoi de la notification. Ici encore, vous devez remplacer le canal d'exemple « Frigo » par

celui que vous aurez créé après votre inscription sur le site Pushetta.

Si vous avez effectué correctement toutes les étapes, votre smartphone émettra un bip et affichera une notification avec le texte « Acheter du lait ! ». Le premier objectif est atteint.

Étudions le code

Maintenant que nous sommes en mesure d'envoyer des notifications push, nous allons implémenter la lecture des capteurs pour atteindre notre objectif final. Pour simplifier cette tâche, nous devons installer une autre librairie, mais pour Python cette fois.

Dans ce cas, nous utiliserons « apt-get », le gestionnaire de paquets utilisé par Raspbian, car ce que nous voulons mettre en œuvre n'est pas disponible avec « pip ». Exécutons la commande suivante :

```
sudo apt-get python-smbus
```

Une fois l'installation terminée, nous devons implémenter le code. Mais tout d'abord, il est nécessaire d'identifier l'adresse I2C de notre carte d'extension. Lorsque nous avons configuré notre distribution, nous nous sommes préoccupés d'installer également le paquet « **i2c-tools** ». Celui-ci fournit un certain nombre de commandes de diagnostics pour le bus I2C.

Il est temps de l'utiliser, en particulier, nous allons utiliser « i2cdetect », qui effectue une recherche sur le bus en indiquant l'adresse de tous les périphériques détectés.

Nous obtenons un résultat similaire à celui représenté en figure 25. Dans l'exemple représenté, « i2cdetect » a été lancé avec le paramètre « -y 1 ».

Le numéro (1) indique le bus I²C sur lequel s'effectue la recherche. Dans les anciennes versions du RaspberryPi, le bus utilisé était le « 0 », si vous possédez une de ces anciennes versions, la commande à exécuter devra être « i2cdetect -y 0 ».

Une fois que vous avez identifié l'adresse utilisée par votre carte, vous avez tout le nécessaire pour commencer à interagir avec les capteurs. Afin de fournir autant que possible des outils

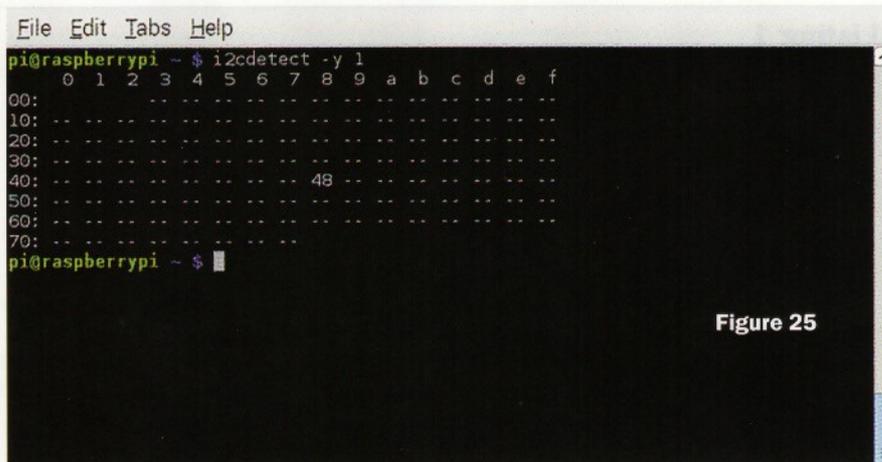


Figure 25

réutilisables, nous allons écrire le code en suivant le mode de programmation orientée objet (OOP).

Cela n'implique pas une plus grande complexité du code, mais pour la mise en œuvre de notre projet, nous n'utilisons pas ces fonctionnalités qui peuvent

avoir un impact sur la compréhension. En simplifiant au maximum, nous pouvons considérer un capteur comme un objet qui permet d'acquérir la mesure d'une grandeur physique.

Nous créons ainsi l'interface qui formalise le comportement du capteur.

Listing 2

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

from smbus import SMBus
from pushetta import Pushetta
import math
import time

# Interface générique pour les capteurs

class GenericSensor:

    def readValue(self):
        raise NotImplementedError('Méthode readValue à implémenter')

    def measureUnit(self):
        class NotImplementedError('Méthode measureUnit à implémenter')

# Implémentation concrète du capteur de température

class TemperatureSensor(GenericSensor):

    adc_address = 0x48
    adc_channel = 0x40
    V_REF = 3.3
    R1 = 15000.0
    V_IN = 3.3
    R_th0 = 10000.0
    A = 0.00335402
    B = 0.000256985
    C = 2.62013e-6
    D = 6.38309e-8
    pB = 4100.0
    K = 6.0

    # Tension de référence pour la mesure de l'ADC
    # Résistance R1
    # Tension d'alimentation du diviseur de tension
    # Résistance de référence de la thermistance à 25 °C
    # Steinhart-Hart Constante A
    # Steinhart-Hart Constante B
    # Steinhart-Hart Constante C
    # Steinhart-Hart Constante D
    # Constante du paramètre B
    # Facteur de dissipation K 6 mV C
    
```

```

def readValue(self):
    bus = SMBus(1) # 0 per le rev.1
    bus.write_byte(self.adc_address, self.adc_channel)
    raw_val = bus.read_byte(self.adc_address)
    raw_val = bus.read_byte(self.adc_address)
    raw_val = bus.read_byte(self.adc_address)

    V = raw_val * self.V_REF / 256.0
    R_th = self.R1 * V / (self.V_IN - V)

    # Calcul des degrés Kelvin avec la formule de Steinhart-Hart

    logR = math.log(R_th / self.R_th0)
    logR2 = logR ** 2
    logR3 = logR ** 3
    Stein = 1.0 / (self.A + self.B * logR + self.C * logR2 + self.D
    * logR3)

    # Conversion en degrés Celsius

    Celsius = round(Stein - 273.15 - V ** 2 / (self.K * R_th), 2)
    return Celsius
def measureUnit(self):
    return 'C'

pushetta = Pushetta('0011223344556677aabbccddeeff001122334455')
sensor = TemperatureSensor()
while True:
    temp = sensor.readValue()
    if temp < 18.0:
        pushetta.pushMessage('Camera', 'Brrrr... il fait froid ici')
        time.sleep(1)

```

```

class GenericSensor:
    def readValue(self):
        raise NotImplementedError('Méthode readValue à implémenter')
    def measureUnit(self):
        raise NotImplementedError('Méthode measureUnit à implémenter')

```

La classe « **GenericSensor** » représente exactement ce que nous avons décrit.

Chaque capteur dispose d'une méthode « **readValue** » qui représente la grandeur physique et « **measureUnit** » qui représente l'unité de mesure.

Nous commençons par l'implémentation de la classe du capteur de température. Le code, visible dans le listing 1, effectue la lecture des capteurs.

« **TemperatureSensor** » correspond à une mise en œuvre concrète de l'interface « **GenericSensor** ». Cette classe est en fait une partie que vous pouvez réutiliser dans vos projets. Son utilisation est triviale, comme le montre l'exemple suivant :

```

sensor = TemperatureSensor()
print "La température est de " + str(sensor.readValue())

```

Maintenant, nous avons tous les éléments pour atteindre notre objectif, écrivons les lignes de code nécessaires :

```

pushetta = Pushetta("0011223344556677aabbccddeeff001122334455")
sensor = TemperatureSensor()
while True:
    temp = sensor.readValue()
    if temp < 18.0:
        pushetta.pushMessage('Camera', 'Brrrr... il fait froid ici')
        time.sleep(5)

```

Le code complet commenté est visible dans le listing 2. Nous créons deux objets pour interagir respectivement avec le service Pushetta et avec le capteur de température.

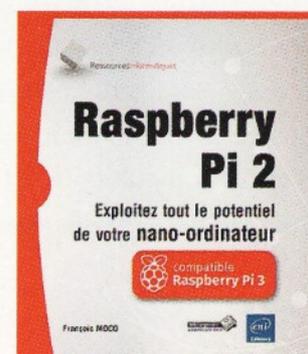
Ensuite nous commençons une boucle infinie qui lit la température, si elle est

en dessous de 18 °C une notification est envoyée au canal « Camera ». Enfin, nous attendons 5 secondes avant de reprendre l'exécution du cycle suivant, et ainsi de suite de manière infinie ou jusqu'à ce que nous interrompions manuellement le programme à l'aide des touches « CTRL + C ».

Vous avez maintenant des structures de code réutilisables et facilement adaptables à vos besoins. Comme pour le premier exemple d'utilisation du service Pushetta, rappelez-vous que l'Api Key ("0011223344556677aabbccddeeff001122334455") et le nom du canal ("Camera") doivent être remplacés par les vôtres.

Conclusion

Nous avons réussi à envoyer des notifications push à partir d'un RaspberryPi vers un smartphone iOS ou Android. De plus, nous avons appris à structurer le code objet selon le paradigme de la programmation orientée objet. Nous avons construit un ensemble de composants facilement réutilisables dans d'autres projets. Nous vous proposons un exercice qui consiste à mettre en œuvre la classe « **LightSensor** », en dessous de la ligne « **TemperatureSensor** », qui lit la valeur du capteur de luminosité. Il est évident que l'utilisation de différents capteurs ne comporte pas de difficultés supplémentaires par rapport à ce qui a été fait dans les exemples précédents. Maintenant que vous avez appris à envoyer des notifications push, vous pourrez réaliser des projets qui ne seront limités que par votre imagination. En particulier dans le domaine de la domotique, où déjà de très nombreux projets associés à la génération de notifications push ont été développés. ■



www.framboise314.fr

TRANSFORMEZ VOTRE OSCILLOSCOPE EN ANALYSEUR DE SPECTRE

DEUXIÈME ET DERNIÈRE PARTIE

de Marco LANDONI

Dans cette seconde partie de l'article paru dans le numéro 138, nous allons aborder la construction et l'utilisation de l'analyseur de spectre. Rappelons qu'il s'agit d'une interface capable de transformer n'importe quel oscilloscope en un analyseur de spectre pour des signaux allant jusqu'à 5 MHz.

Le taux d'échantillonnage maximum est de 10 Ms/s, il est alors possible de traiter le spectre de signaux allant jusqu'à 5 MHz, c'est-à-dire des signaux BF mais aussi vidéos.

L'analyseur de spectre dispose d'une fonction « **Zoom** » allant jusqu'à 4 fois (x4) et d'une fonction de « **scrolling** » (défilement) du spectre visualisé. De plus, 4 échelles permettent de visualiser les amplitudes (2 échelles linéaires en tension et 2 échelles logarithmiques en décibels).

Pour utiliser pleinement l'analyseur de spectre, nous devons disposer d'un oscilloscope double traces configuré en « **mode X/Y** » en reliant les deux sorties aux canaux CH1 et CH2, comme expliqué dans l'article précédent du numéro 138. Comme nous avons largement abordé le fonctionnement dans l'article précédent, nous allons maintenant passer à la réalisation pratique.

Dans le précédent numéro d'Electronique et Loisirs Magazine, nous vous avons proposé un montage capable de transformer un classique oscilloscope analogique à double trace en un analyseur de spectre.

Le circuit effectue une transformée de Fourier du signal à analyser, c'est-à-dire un échantillonnage numérique du signal de manière appropriée, puis restitue le résultat du traitement numérique sur l'écran d'un oscilloscope.

Réalisation pratique

L'analyseur de spectre se compose essentiellement de deux circuits imprimés. Le circuit principal sur lequel se trouve l'ensemble du système, y compris l'alimentation (sauf le transformateur),

et un deuxième circuit imprimé disposé derrière la face avant afin d'accueillir les boutons et les LED.

Dans un premier temps, commencez par télécharger le fichier « **ET855_analyseur de spectre.zip** » sur notre site

www.electroniquemagazine.com dans les sommaires détaillés du n° 138 ou 139 (les 2 fichiers sont identiques).

Le fichier zippé contient les typons au format « pdf » que vous pouvez visualiser à l'aide d'Acrobat Reader.

Plan de montage de l'analyseur de spectre

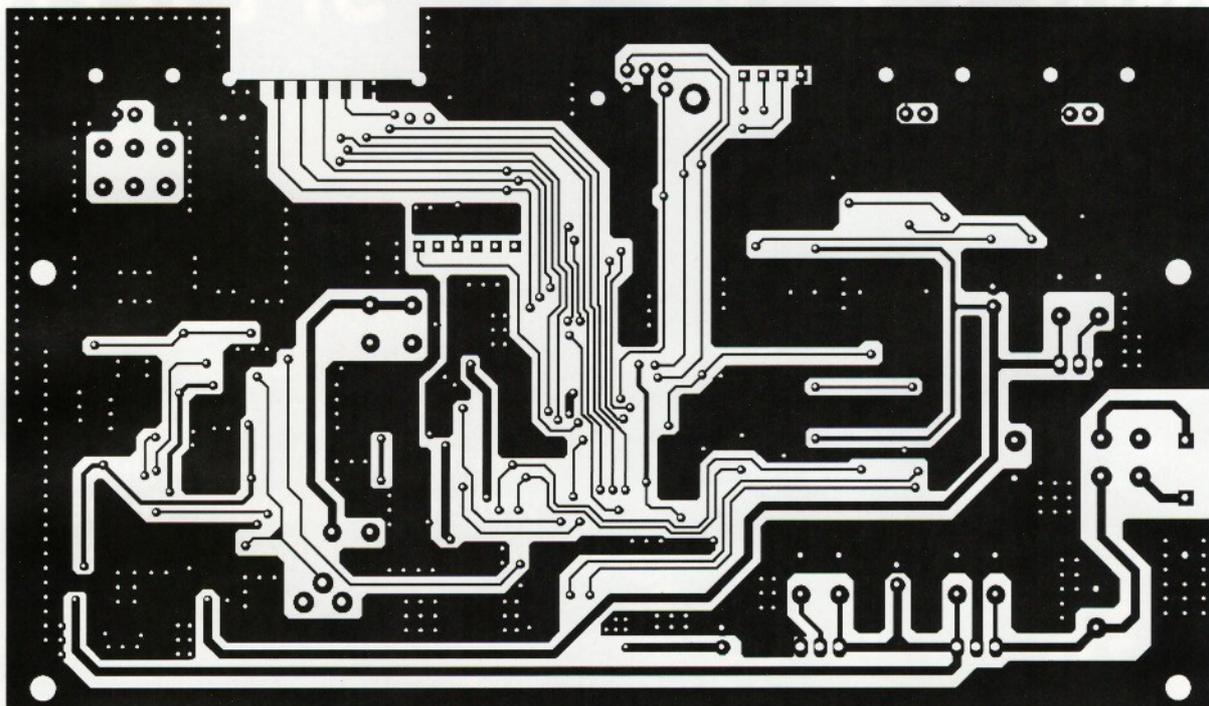


Figure 1 : circuit imprimé principal à l'échelle 1 : 1 côté soudures.

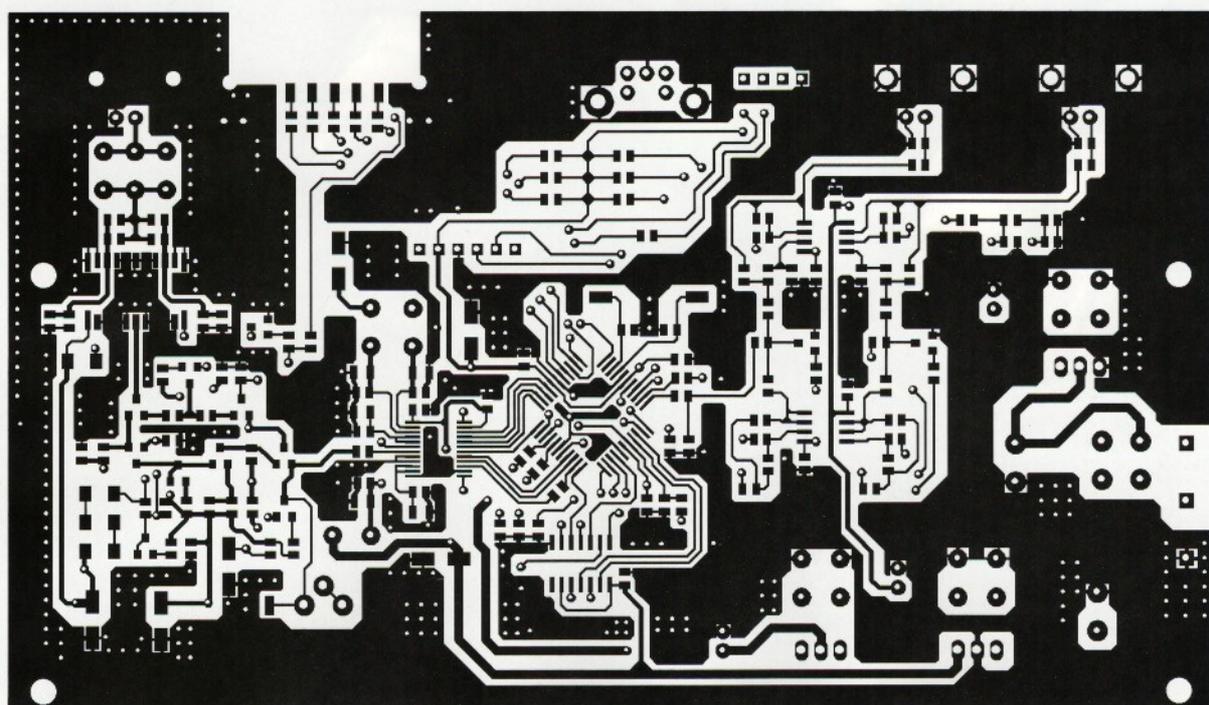


Figure 2 : circuit imprimé principal à l'échelle 1 : 1 côté composants.

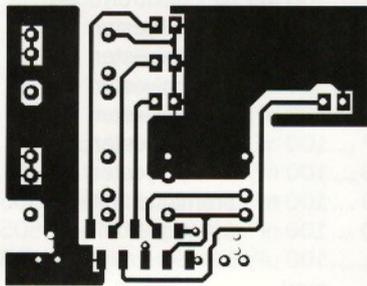


Figure 3 : circuit imprimé de la face avant à l'échelle 1 : 1 côté soudures.

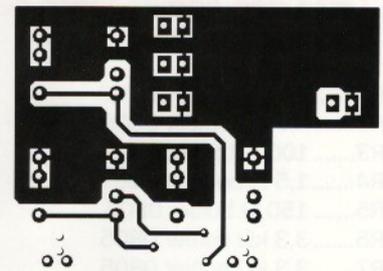


Figure 4 : circuit imprimé de la face avant à l'échelle 1 : 1 côté composants.

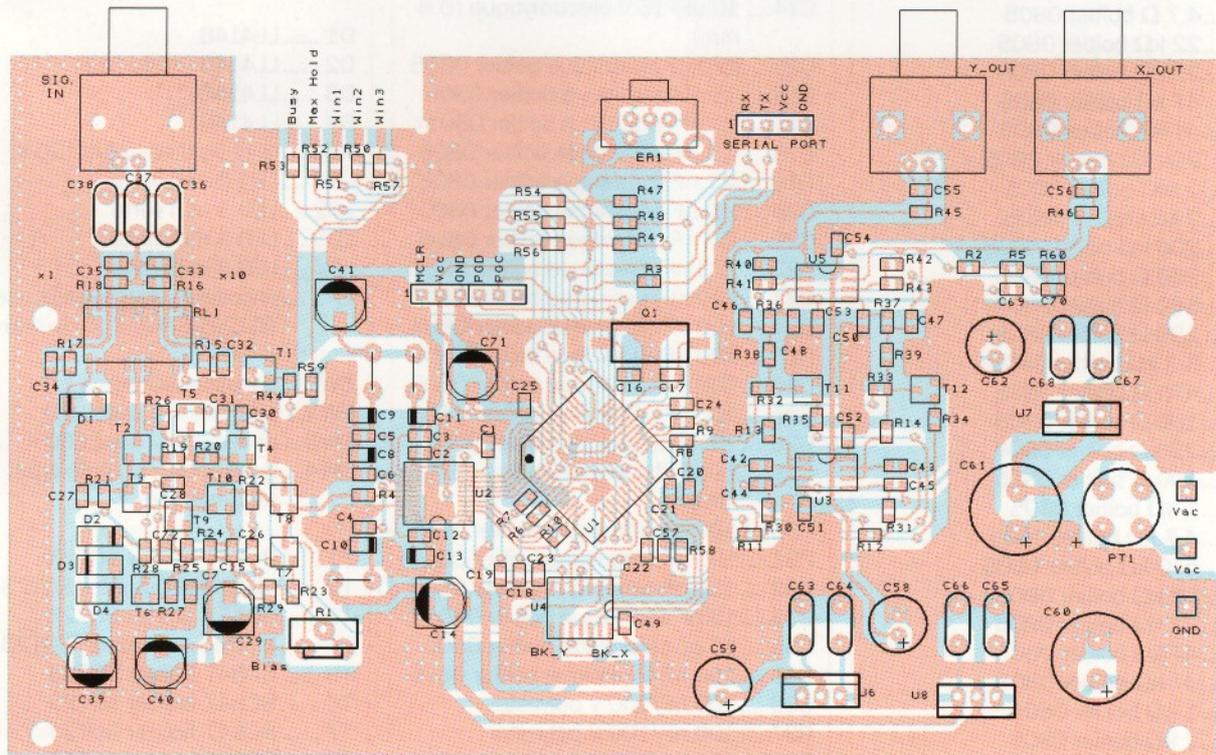


Figure 5 : schéma de câblage du circuit principal.

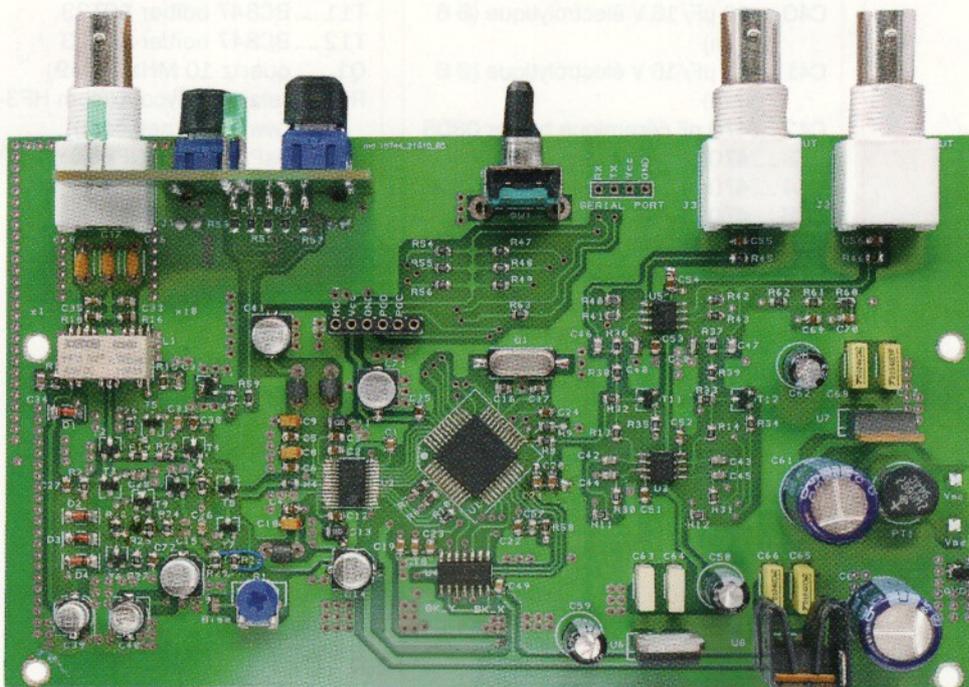


Figure 6 : photo de l'un de nos prototypes du circuit principal de l'analyseur de spectre.

Liste des composants de l'analyseur de spectre

R1..... 500 k Ω trimmer MO
 R2..... 4,7 k Ω boîtier 0805
 R3..... 100 Ω boîtier 0805
 R4..... 1,5 Ω boîtier 0805
 R5..... 150 Ω boîtier 0805
 R6..... 3,3 k Ω boîtier 0805
 R7..... 3,3 k Ω boîtier 0805
 R8..... 3,3 k Ω boîtier 0805
 R9..... 3,3 k Ω boîtier 0805
 R10 4,7 Ω boîtier 0805
 R11 22 k Ω boîtier 0805
 R12 22 k Ω boîtier 0805
 R13 47 k Ω boîtier 0805
 R14 47 k Ω boîtier 0805
 R15 200 k Ω boîtier 0805
 R16 2 M Ω boîtier 0805
 R17 2 M Ω boîtier 0805
 R18 200 k Ω boîtier 0805
 R19 560 Ω boîtier 0805
 R20 560 Ω boîtier 0805
 R21 180 Ω boîtier 0805
 R22 3,3 k Ω boîtier 0805
 R23 470 Ω boîtier 0805
 R24 100 Ω boîtier 0805
 R25 220 Ω boîtier 0805
 R26 100 Ω boîtier 0805
 R27 1 k Ω boîtier 0805
 R28 4,7 k Ω boîtier 0805
 R29 470 k Ω boîtier 0805
 R30 22 k Ω boîtier 0805
 R31 22 k Ω boîtier 0805
 R32 47 k Ω boîtier 0805
 R33 47 k Ω boîtier 0805
 R34 1 k Ω boîtier 0805
 R35 1 k Ω boîtier 0805
 R36 33 k Ω boîtier 0805
 R38 2,2 k Ω boîtier 0805
 R39 2,2 k Ω boîtier 0805
 R40 82 k Ω boîtier 0805
 R41 47 k Ω boîtier 0805
 R42 220 k Ω boîtier 0805
 R43 100 k Ω boîtier 0805
 R44 1 k Ω boîtier 0805
 R45 1,5 k Ω boîtier 0805
 R46 1,5 k Ω boîtier 0805
 R47 100 k Ω boîtier 0805
 R48 100 k Ω boîtier 0805
 R49 100 k Ω boîtier 0805
 R50 330 Ω boîtier 0805
 R51 330 Ω boîtier 0805
 R52 330 Ω boîtier 0805
 R53 330 Ω boîtier 0805
 R54 100 k Ω boîtier 0805
 R55 100 k Ω boîtier 0805
 R56 100 k Ω boîtier 0805
 R57 330 Ω boîtier 0805
 R58 100 k Ω boîtier 0805
 R59 330 Ω boîtier 0805
 R60 1,2 k Ω boîtier 0805
 C1..... 100 nF céramique boîtier 0805

C2..... 100 nF céramique boîtier 0805
 C3..... 100 nF céramique boîtier 0805
 C4..... 100 nF céramique boîtier 0805
 C5..... 100 nF céramique boîtier 0805
 C6..... 100 nF céramique boîtier 0805
 C7..... 100 nF céramique boîtier 0805
 C8..... 4,7 μ F/16 V tantale (B)
 C9..... 4,7 μ F/16 V tantale (B)
 C10..... 4,7 μ F/16 V tantale (B)
 C11 4,7 μ F/16 V tantale (B)
 C12 4,7 μ F/16 V tantale (B)
 C13 4,7 μ F/16 V tantale (B)
 C14.... 10 μ F/16 V électrolytique (\varnothing 6 mm)
 C15 220 pF céramique boîtier 0805
 C16.... 15 pF céramique boîtier 0805
 C17.... 15 pF céramique boîtier 0805
 C18 4,7 nF céramique boîtier 0805
 C19 4,7 nF céramique boîtier 0805
 C20 2,2 nF céramique boîtier 0805
 C21.... 4,7 nF céramique boîtier 0805
 C22 100 nF céramique boîtier 0805
 C23 100 nF céramique boîtier 0805
 C24.... 100 nF céramique boîtier 0805
 C25 100 nF céramique boîtier 0805
 C26 100 nF céramique boîtier 0805
 C27.... 100 nF céramique boîtier 0805
 C28 10 nF céramique boîtier 0805
 C29 10 μ F/16 V électrolytique (\varnothing 6 mm)
 C30 100 nF céramique boîtier 0805
 C31.... 100 nF céramique boîtier 0805
 C32 150 pF céramique boîtier 0805
 C33 15 pF céramique boîtier 0805
 C34 15 pF céramique boîtier 0805
 C35 150 pF céramique boîtier 0805
 C36 100 nF/100 V polyester
 C37.... 100 nF/100 V polyester
 C38 100 nF/100 V polyester
 C39 10 μ F/16 V électrolytique (\varnothing 6 mm)
 C40 10 μ F/16 V électrolytique (\varnothing 6 mm)
 C41.... 10 μ F/16 V électrolytique (\varnothing 6 mm)
 C42 470 pF céramique boîtier 0805
 C43 470 pF céramique boîtier 0805
 C44 470 pF céramique boîtier 0805
 C45 470 pF céramique boîtier 0805
 C46 100 pF céramique boîtier 0805
 C47.... 100 pF céramique boîtier 0805
 C48 220 pF céramique boîtier 0805
 C49 100 nF céramique boîtier 0805
 C50 220 pF céramique boîtier 0805
 C51.... 100 nF céramique boîtier 0805
 C52 100 nF céramique boîtier 0805
 C53 100 nF céramique boîtier 0805
 C54 100 nF céramique boîtier 0805
 C55 2,2 nF céramique boîtier 0805
 C56 2,2 nF céramique boîtier 0805
 C57.... 1 nF céramique boîtier 0805
 C58 100 nF/63 V polyester
 C59 100 μ F/16 V électrolytique
 C60 100 μ F/25V électrolytique
 C61.... 100 μ F/25V électrolytique

C62 100 μ F/16 V électrolytique
 C63 100 nF/63 V polyester
 C64 100 nF/63 V polyester
 C65 100 nF/63 V polyester
 C66 100 nF/63 V polyester
 C67 100 nF/63 V polyester
 C68 100 nF/63 V polyester
 C69 100 nF céramique boîtier 0805
 C70 100 nF céramique boîtier 0805
 C71.... 100 μ F/6,3V électrolytique (\varnothing 6 mm)
 C72 100 nF céramique boîtier 0805
 D1 LL4148
 D2 LL4148
 D3 LL4148
 D4 LL4148
 PT1.... pont de diodes 2A/100 V
 LD1 LED 3 mm verte
 LD2 LED 3 mm verte
 LD3 LED 3 mm verte
 LD4 LED 3 mm verte
 ER1 encodeur rotatif avec poussoir
 EC12E24242A2 (radiospares.fr)

SW1... poussoir NO avec LED
 SW2... poussoir NO avec LED
 SW3... poussoir NO avec LED
 SW4... interrupteur M/A de type rectangulaire

T1 BC847 boîtier SOT23
 T2 MMBFJ310 boîtier SOT23
 T3 BC857 boîtier SOT23
 T4 MMBFJ310 boîtier SOT23
 T5 BC847 boîtier SOT23
 T6 BC847 boîtier SOT23
 T7 BC847 boîtier SOT23
 T8 BC847 boîtier SOT23
 T9 BC847 boîtier SOT23
 T10 BC847 boîtier SOT23
 T11 BC847 boîtier SOT23
 T12 BC847 boîtier SOT23
 Q1..... quartz 10 MHz (HC49)
 RL1.... relais HF Tyco Axicom HF3-52
 (www.fr.farnell.com)
 U1..... dsPIC33FJ64GP804 (TQFP)
 U2..... TLC5510IPW (TSSOP)
 U3..... RC4558 (SOIC8)
 U4..... 74HCT04D (SOIC14)
 U5..... RC4558 (SOIC8)
 U6..... LD1117AV33 (TO220)
 U7..... 7905 (TO220)
 U8..... 7805 (TO220)
 L1 ferrite
 L2 ferrite
 L3 ferrite

Divers

Prise BNC pour ci (x2)
 TF1 transformateur 2 x 9 V/ 500mA
 Barrette mâle 6 pôles
 Boîtier Teko AUS12

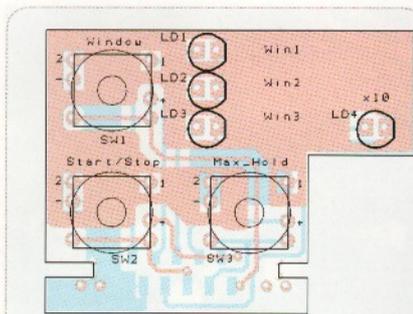


Figure 7 : schéma de câblage du circuit imprimé de la face avant.

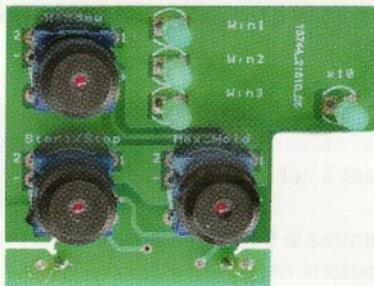


Figure 8 : photo de l'un de nos prototypes du circuit de la face avant de l'analyseur de spectre.

Pour une fabrication professionnelle, vous disposez d'autre part des fichiers GERBER des 2 circuits imprimés doubles faces.

Notez aussi que vous trouverez le programme « .hex » pour le DSPIC mais aussi l'ensemble du projet incluant le code source que vous pouvez modifier et éditer avec l'outil **MPLAB IDE** de Microchip.

Une fois les circuits fabriqués, commencez par l'assemblage des composants du circuit imprimé principal. **Soyez très minutieux pour le soudage du DSPIC U1** et du **convertisseur A/D U2**.

Respectez **l'orientation des composants**. Vous devez disposer d'un fer à souder et de la soudure pour composants CMS. Évitez les courts-circuits sous les broches en minimisant la quantité de soudure.

Continuez en soudant toujours délicatement les autres circuits intégrés, en faisant attention à leur orientation. Ensuite soudez les résistances CMS, les condensateurs CMS non polarisés, le quartz et les condensateurs polarisés

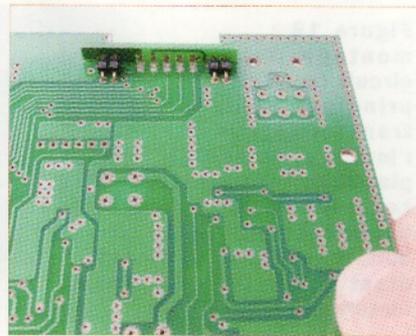


Figure 16 : l'assemblage du circuit imprimé de la face avant sur le circuit imprimé principal s'effectue au moyen de 2 barrettes doubles de chaque côté.

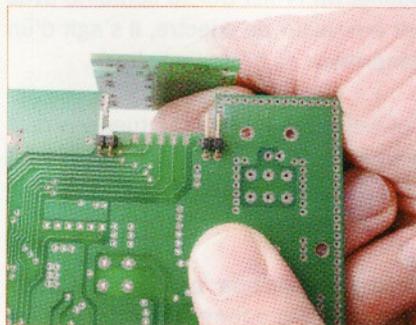


Figure 17 : brochage, vu de dessous, des boutons poussoirs avec LED intégrée utilisés sur le circuit imprimé de la face avant.

des LED sur le circuit imprimé de la face avant. Commencez par souder les 3 boutons poussoirs en prenant soin d'orienter correctement l'anode de la LED incorporée de chaque bouton, sinon elle ne s'allumera pas. Soudez ensuite les 4 LED de 3 mm à une hauteur de 12 mm par rapport au circuit imprimé.

Maintenant, vous devez assembler les deux circuits imprimés. Pour cela, insérez dans la partie inférieure de la carte principale 2 barrettes doubles coudées à 90°, puis insérez dans l'espace prévu à cet effet le petit circuit contenant les boutons poussoirs et les LED (voir la figure 16).

Au cours de cette opération, les barrettes doivent être placées exactement dans les trous situés sur le petit circuit imprimé. Si tout semble correct, soudez uniquement une des 2 barrettes pour maintenir l'ensemble mécaniquement.

Vérifiez que les deux circuits imprimés sont orthogonaux, puis soudez l'autre barrette et toutes les connexions au milieu des 2 barrettes afin d'obtenir un ensemble robuste. Ne mettez pas trop de soudure sur les connexions centrales.

CMS en prêtant une attention particulière à leur polarité.

Finissez en soudant les composants traversants (traditionnels), c'est-à-dire les condensateurs électrolytiques (attention à la polarité) et non polarisés, le pont de diode, les régulateurs de tension U7 et U8, le relais, les prises BNC, l'encodeur rotatif.

Montez enfin, sur le **régulateur U8**, un **dissipateur de 18°C/W**. À ce stade, le montage des composants du circuit imprimé principal est terminé.

Vous devez maintenant procéder au montage des boutons poussoirs et

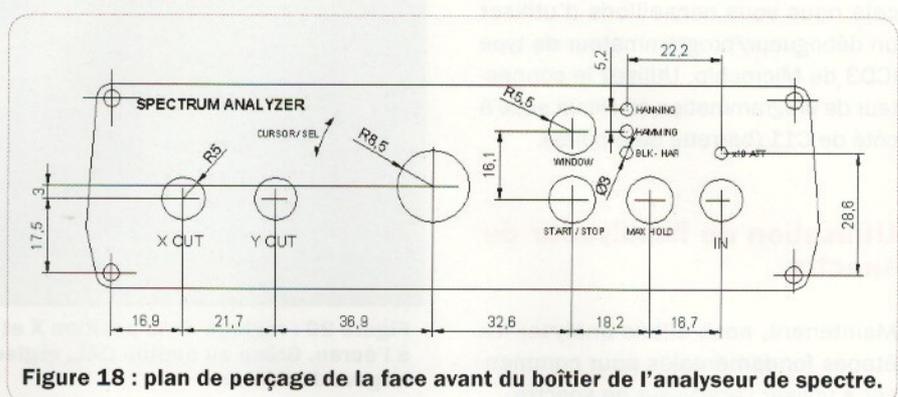


Figure 18 : plan de perçage de la face avant du boîtier de l'analyseur de spectre.

L'ensemble de l'électronique avec le transformateur est conçu pour être monté à l'intérieur d'un boîtier plastique de marque « Teko » référence « AUS12 ». Les trous du circuit imprimé correspondent aux emplacements de ceux du boîtier. En figure 18, vous trouverez un plan de perçage de la face avant du boîtier.

Montez les circuits imprimés à l'intérieur du boîtier, vous devez faire en sorte que le circuit imprimé principal soit distant d'environ 12,5 mm par rapport au fond du boîtier. Pour cela utilisez des entretoises, de préférence en plastique.

N'oubliez pas de percer les trous de fixation du transformateur sur le fond du boîtier. Sur la face arrière du boîtier, vous devez percer un trou pour le cordon d'alimentation sans oublier le passe-câble et faire une découpe rectangulaire pour l'interrupteur M/A.

Une fois cela effectué, reliez le transformateur d'alimentation au circuit imprimé principal en soudant directement aux emplacements prévus sur le circuit les 3 fils provenant du secondaire.

Faites coïncider les LED et les boutons du petit circuit imprimé avec la face avant du boîtier que vous venez de percer, puis serrez les écrous des prises BNC lorsque tout s'emboîte correctement.

Reliez maintenant le primaire du transformateur à l'interrupteur d'alimentation. À ce stade, votre analyseur de spectre est prêt à fonctionner.

Alimentez l'appareil et procédez à la programmation du dsPIC. Attention, pour cela nous vous conseillons d'utiliser un débogueur/programmeur de type ICD3 de Microchip. Utilisez le connecteur de programmation en circuit situé à côté de C11 (barrette de 6 pôles).

Utilisation de l'analyseur de spectre

Maintenant, nous allons analyser les étapes fondamentales pour commencer à utiliser l'analyseur de spectre.

Figure 19 : montage du circuit imprimé principal et du transformateur à l'intérieur du boîtier plastique Teko AUS12. Les 3 fils du secondaire doivent être soudés au circuit, le point milieu (fil noir) étant relié à GND. Au-dessus, l'interrupteur M/A.



Tout d'abord nous vous annonçons que dans le prochain numéro 140 d'Electronique et Loisirs Magazine nous allons publier un projet qui se combine avec cet analyseur de spectre, il s'agit d'un générateur de bruit blanc/rose.

Celui-ci permettra d'effectuer diverses mesures (par exemple sur des filtres) grâce aux performances de cet analyseur de spectre.

Mais procédons par ordre en commençant par l'étalonnage de l'étage d'entrée de l'analyseur.

Mettez sous tension l'analyseur, et à l'aide d'un multimètre réglé sur le calibre VDC (tension continue), **mesurez la tension** entre le point dénommé « **BIAS** » et la masse (à côté R1).

Vous devez mesurer une tension de **1,5 VDC**, si cela n'est pas le cas **réglez le trimmer R1** afin d'obtenir cette valeur.

Maintenant, **connectez l'analyseur de spectre à votre oscilloscope** comme indiqué en figure 1 (voir la page 26 du numéro 138) à l'aide de 2 câbles BNC/BNC.

Réglez la base de temps de l'oscilloscope sur **0,5 V/div**, le couplage **DC sur les deux canaux** et **activez le mode X/Y**.

Vous devriez voir apparaître sur l'écran de l'oscilloscope une image correspondant à celle de la figure 20.

Centrez la trace sur l'écran à l'aide des boutons de positionnement (POS X et POS Y) qui équipent l'oscilloscope et, par le biais de la commande « **CAL** », faites « entrer » l'image entière dans l'écran en ajustant les proportions.

Maintenant, agissez de nouveau sur le trimmer R1 en le tournant très lentement. Vous verrez que la composante continue du spectre visualisé, c'est-à-dire la première et unique ligne verticale positionnée à gauche, changera d'amplitude. Lorsque vous lisez quelques millivolts, votre instrument est parfaitement calibré.

À l'aide d'un générateur de fonction sinusoïdal, injectez un signal d'une fréquence de 100 Hz et d'amplitude 50 mVpp à l'entrée de l'analyseur, vous verrez sur l'écran une ligne.

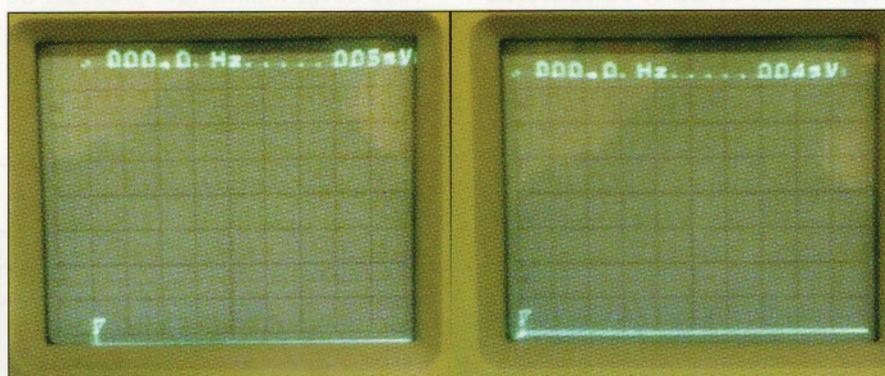


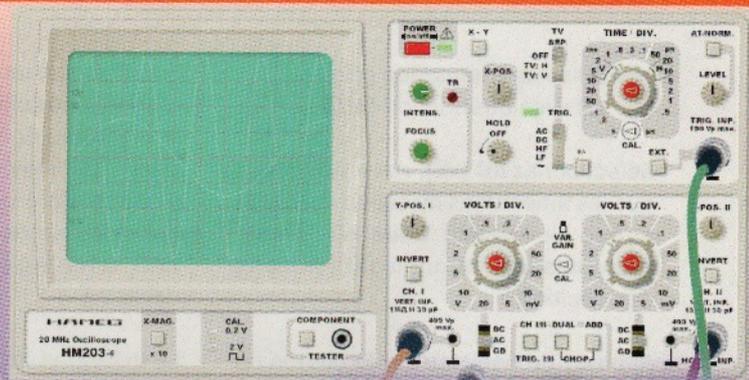
Figure 20 : réglage de la position X et Y de la trace, centrez la trace par rapport à l'écran. Grâce au bouton CAL, régler la trace de façon qu'elle occupe toute la largeur de l'écran.

Un oscilloscope en ligne

Nous avons trouvé sur le web un site qui permet de simuler un oscilloscope virtuel avec un nombre important de commandes et diverses simulations de formes d'ondes.

Si vous voulez vous familiariser avec l'oscilloscope et visualiser les signaux en fonction des différentes commandes, visitez l'adresse : www.virtual-oscilloscope.com. Ce site propose un simulateur parfait du modèle HAMEG HM203-6 avec des tutoriels pour les débutants et pour tous ceux qui veulent rafraîchir les notions relatives à l'utilisation de cet outil très précieux pour le technicien en électronique. Malheureusement, il faudra un peu maîtriser la langue Anglaise, vous pouvez toujours utiliser un traducteur en ligne.

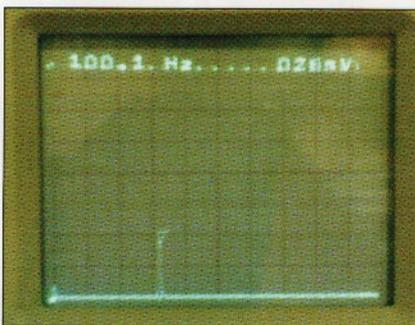
Le simulateur est vraiment réaliste, vous pouvez entendre dans les haut-parleurs de l'ordinateur les sons produits par les sélecteurs rotatifs et les boutons. Pour démarrer la simulation, vous devez activer l'instrument



en cliquant sur le bouton d'alimentation virtuel, vous voyez alors la trace apparaître à l'écran.

Mais ce n'est pas tout, à la base de l'oscilloscope se trouvent 4 câbles avec des BNC (pour les connecter faites glisser le curseur sur la BNC de l'instrument) qui représentent les entrées sur lesquelles vous pouvez appliquer différents signaux. Vous apercevrez alors sur l'écran la forme de l'onde principale et les combinaisons telles que l'addition ou le mode X/Y utilisé par notre analyseur de spectre.

Figure 21 : spectre d'un signal sinusoïdal à 100 Hz. Le curseur indique la fréquence et l'amplitude de la composante visualisée. Pour positionner le curseur, il suffit de tourner le bouton de l'encodeur.



Tournez le bouton de l'encodeur dans le sens horaire afin de déplacer le curseur. Positionnez-vous exactement sur la ligne visualisée, à ce stade les indications à l'écran vous donnent la fréquence et l'amplitude de l'harmonique (voir la figure 21).

Si, au lieu d'injecter une onde sinusoïdale, vous injectez une onde carrée de 100 Hz et de même amplitude, vous visualiserez sur l'écran plusieurs lignes.

Ces lignes correspondent à des harmoniques supérieures qui composent le signal d'onde carrée. Cependant ces parties du signal ne contiennent aucune information pertinente, parce que leur

fréquence est supérieure à la fréquence maximale qui doit être utilisée. Ce phénomène s'appelle « aliasing » ou « repliement ».

Appuyez sur le bouton de l'encodeur pour entrer dans le menu (voir la figure 23).

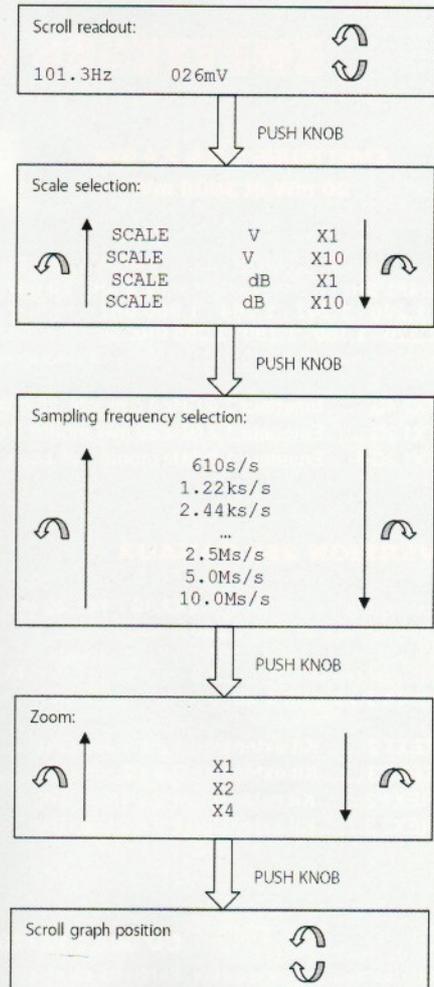
Déplacez le bouton vers la droite ou vers la gauche, vous pouvez alors sélectionner l'échelle de visualisation de l'amplitude entre :

- volts sans atténuation (x1) ;
- volts avec atténuation (x10) ;



Figure 22 : spectre d'un signal carré à 100 Hz. L'échelle est en dB et le curseur pointe vers la 3^{ème} harmonique (700 Hz). La fréquence d'échantillonnage est fixée à 4,88 ks/s pour acquérir correctement toutes les harmoniques du signal.

Figure 23 : navigation à l'intérieur du menu, accessible en appuyant sur le bouton de l'encodeur.



- dB sans atténuation (x1) ;
- dB avec atténuation (x10).

Sélectionnez **dB (x1)**, vous remarquez que les amplitudes du signal changent et que l'indicateur visualise l'amplitude mesurée en utilisant l'échelle des décibels.

Appuyez de nouveau sur le bouton de l'encodeur afin de sélectionner la fréquence d'échantillonnage.

En tournant le bouton dans le sens horaire, sélectionnez 4,88 ks/s.

Vous voyez maintenant que toutes les harmoniques du signal se répartissent de manière ordonnée sur l'écran de l'oscilloscope.

Cela signifie que l'effet de « aliasing », en raison du sous-échantillonnage, est devenu négligeable.

En fait, lorsqu'un signal doit être échantillonné, il ne suffit pas qu'il le soit selon le théorème de Shannon, c'est-à-dire selon l'harmonique fondamentale (100 Hz dans notre cas), mais aussi selon toutes les autres harmoniques (sinon vous aurez toujours le phénomène de « aliasing »).

Le **théorème de Shannon** énonce que pour un signal d'une fréquence maximale « B », il **doit être échantillonné à une fréquence double** soit « 2B » afin de ne pas perdre des informations.

Dans le cas d'un signal carré de 100 Hz, le spectre correspondant s'étend théoriquement jusqu'à une fréquence infinie. Il ne sera jamais échantillonné correctement selon le théorème de Shannon, à moins de limiter la bande passante avec un filtre passe-bas ou accepter un phénomène de « aliasing » qui doit être considéré comme négligeable.

En conclusion, si vous avez besoin d'analyser un signal dont vous ne connaissez pas le spectre, vous devez d'abord le **suréchantillonner** en utilisant une valeur élevée de « fs », puis, le cas échéant, **diminuer** la fréquence d'échantillonnage.

Dans le prochain numéro, nous vous décrirons un montage complémentaire de cet analyseur de spectre, il s'agit d'un **générateur de bruit blanc et rose** avec une fréquence de base jusqu'à 1 MHz. ■



ÉMETTEURS & RÉCEPTEURS 1,2 & 2,4 GHz

ÉMETTEURS 1,2 & 2,4 GHz 20 mW et 1000 mW



Alimentation : 13,6 VDC. 4 fréquences en **2,4 GHz** : 2,4 - 2,427 - 2,454 - 2,481 GHz ou 8 fréquences en **1,2 GHz** 20 mW : 1,112 - 1,139 - 1,193 - 1,220 - 1,247 - 1,264 - 1,300 GHz ou 4 fréquences en **1,2 GHz** 1 W : 1,120 - 1,150 - 1,180 - 1,255 GHz. Sélection des fréquences : dip-switch. Stéréo : audio 1 et 2 (6,5 et 6,0 MHz). Livré sans alimentation ni antenne.

TX2.4G..... Émetteur 2,4 GHz 4 c monté 20 mW 39,20 €
TX1.2G..... Émetteur 1,2 GHz 20 mW monté 8 canaux 48,20 €
TX1.2G-1-... Émetteur 1,2 GHz monté 1 W 4 canaux 79,30 €



RÉCEPTEURS 4 CANAUX 1,2 & 2,4 GHz

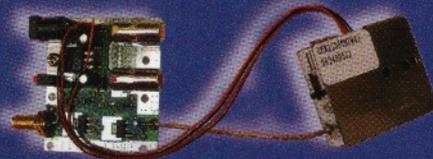
Alimentation : 13,6VDC. 4 fréquences en **2,4 GHz** : 2,4 - 2,427 - 2,454 - 2,481 GHz ou 8 fréquences en **1,2 GHz** : 1,112 - 1,139 - 1,193 - 1,220 - 1,247 - 1,264 - 1,300 GHz. Sélection des fréquences : dip-switch pour le 1,2 GHz et par poussoir pour les versions 2,4 GHz. Stéréo : audio 1 et 2 (6,5 et 6,0 MHz). Fonction scanner pour la version 1,2 GHz. Livré sans alimentation ni antenne.

RX2.4G..... Récepteur monté 2,4 GHz 4 canaux..... 39,00 €
RX1.2G..... Récepteur monté 1,2 GHz 8 canaux 48,20 €

VERSION 256 CANAUX

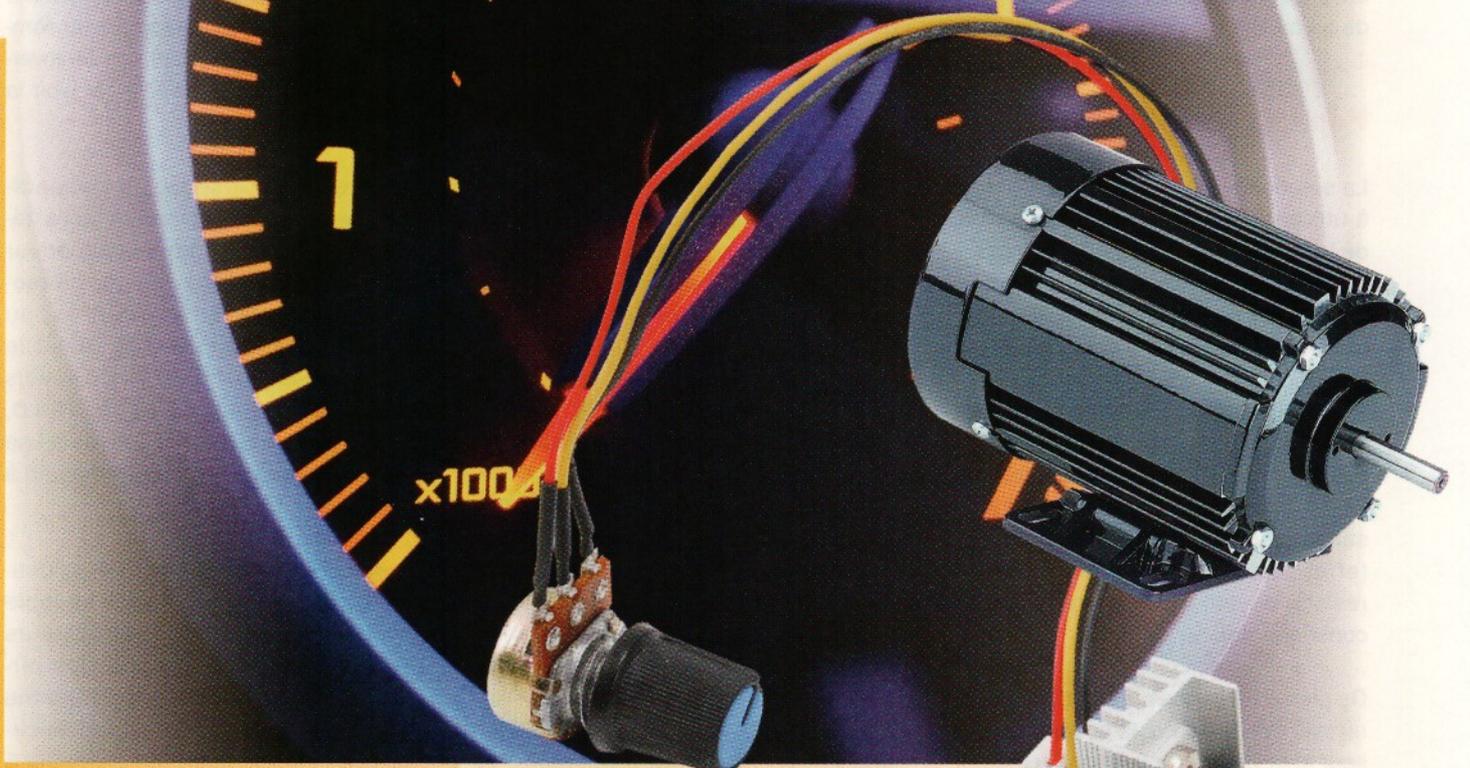
Extensions pour émetteurs **TX2.4G** / **TX1.2G** et récepteurs **RX2.4G** / **RX1.2G** : permettent d'atteindre **256** canaux. Le pas est de 1 MHz. Sélection des canaux par dip-switch. Fréquences de départ : **2,3** pour les versions **TX2,4G** / **REX2,4G** et **1,2** pour les **TX 1,2G** / **RX 1,2G**. Extension vendue sans émetteur ni récepteur.

TEX1.2..... Kit extension 1,2 à 1,456 GHz 19,90 €
TEX2.3..... Kit extension 2,3 à 2,556 GHz 19,90 €
REX1.2..... Kit extension 1,2 à 1,456 GHz 19,90 €
REX2.3..... Kit extension 2,3 à 2,556 GHz 19,90 €



VARIATEUR DE VITESSE POUR MOTEUR À COURANT CONTINU

de Boris Landoni



Ce variateur permet de faire varier la vitesse de rotation de n'importe quel moteur électrique à balais de 12 VDC à 35 VDC, à l'aide d'une poignée de composants électroniques dont le célèbre timer NE555.

Ce montage convient dans diverses applications, telles que les perceuses et tournevis électriques, mais aussi pour les ventilateurs de PC où il peut être nécessaire de faire varier la vitesse à l'aide d'une commande manuelle afin de réduire le bruit.

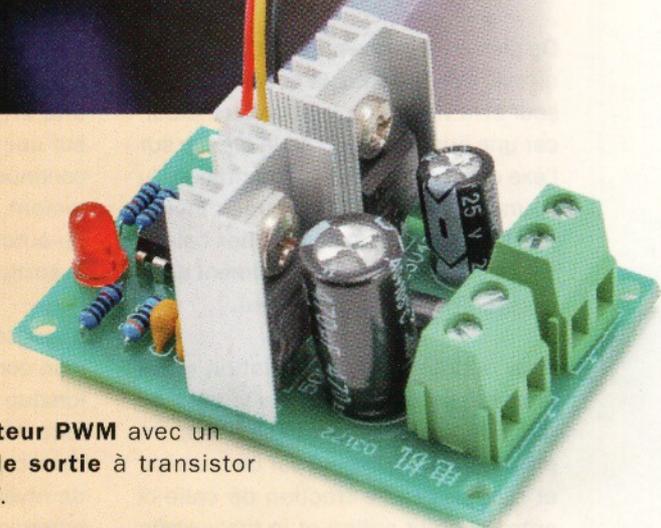
Pour les moteurs à courant alternatif, la technique retenue est le découpage de l'onde obtenue à l'aide d'un variateur dont l'étage de sortie est un TRIAC. Cependant, pour les moteurs à courant continu, en particulier ceux avec des balais, nous devons utiliser une méthode différente pour faire varier la valeur moyenne de la tension appliquée à l'enroulement du rotor.

La solution la plus pratique consiste à réaliser un circuit, comme celui décrit dans ces pages, qui est composé d'un

modulateur PWM avec un **étage de sortie** à transistor **MOSFET**.

Mais avant d'entrer dans le détail du schéma électrique, il est nécessaire de rappeler quelques notions concernant le contrôle d'un moteur DC à l'aide d'un signal PWM (modulation de largeur d'impulsion).

Nous partons du principe que la vitesse de rotation d'un moteur à courant continu dépend de la tension appliquée aux balais, du courant absorbé et de la charge appliquée sur l'axe du moteur (couple de charge). Le couple moteur sur l'axe est proportionnel au courant absorbé. Par conséquent, pour faire varier la vitesse d'un moteur de ce type, il est nécessaire de



Caractéristiques techniques

- Tension d'alimentation : de **12 VDC à 35 VDC**
- Tension du moteur : de **12 VDC à 35 VDC**
- Courant moteur : **5 A**
- Puissance maximale du moteur : **200 W**
- Fréquence PWM : **25 kHz**
- Régulation de la vitesse : de **0 à 100 %**

faire varier la tension qui l'alimente et de lui fournir le courant nécessaire.

Pourquoi la PWM ?

La solution la plus simple consisterait à utiliser un rhéostat ou un régulateur de tension linéaire. Cependant, la puissance dissipée par effet Joule serait importante par rapport à la puissance mécanique fournie par l'axe du moteur.

En fait, un rhéostat est une résistance qui produit à ses bornes une chute de tension proportionnelle au courant qui la traverse. Il faut donc dissiper la puissance produite sous forme de chaleur (effet Joule), ce qui implique que la résistance doit être dimensionnée en conséquence.

De plus, il n'est pas possible de maintenir une relation entre la position du curseur et la vitesse de rotation du moteur, car une augmentation de la charge sur l'axe entraînerait une augmentation du courant absorbé et donc une diminution de la tension appliquée aux balais, ce qui implique un ralentissement de la vitesse de rotation du moteur.

Pour maintenir une corrélation entre la position du curseur et la vitesse de rotation, nous pourrions utiliser un potentiomètre alimenté par une tension et récupérer une fraction de celle-ci entre le point milieu et la masse afin de l'injecter dans un amplificateur de puissance couplé en courant continu dont la sortie alimente le moteur.

Dans ce cas, il reste le problème de l'élément régulateur de tension (l'amplificateur) qui doit dissiper une puissance égale à la différence entre la tension d'alimentation et la tension de sortie, le tout multiplié par le courant fourni au moteur.

Il faut donc trouver une solution qui permette de faire varier la tension fournie au moteur, mais sans pertes ou presque.

Cette solution est la **modulation PWM** qui consiste à fournir des **impulsions rectangulaires** au moteur avec des **amplitudes constantes**, égales à la tension nominale du moteur, mais de **largeur variable** tout en maintenant **constante la fréquence** à une valeur acceptable par le moteur.

Il faut donc faire varier, dans la période, le rapport entre la durée des impulsions de niveau haut et bas. Cette relation en électronique se nomme le rapport cyclique (duty-cycle) qui est exprimé par la formule :

$$dc = th/T$$

où « **th** » est la largeur de l'impulsion de niveau haut et « **T** » la période formée par la durée des impulsions de niveau haut et bas.

La configuration du modulateur PWM, qui est le circuit qui génère des impulsions à une fréquence constante modulées en amplitude, est basée sur un comparateur de tension qui reçoit sur une entrée une onde triangulaire et sur une autre entrée une composante continue qui peut être produite manuellement au moyen d'un potentiomètre ou automatiquement par un circuit ou la sortie d'un microcontrôleur.

Supposons que nous appliquons la tension continue sur l'entrée inverseuse, lorsque le potentiel de l'onde rectangulaire dépasse le seuil du comparateur, celui-ci génère en sortie une impulsion de niveau haut, tandis que lorsque le potentiel est inférieur au seuil, la sortie reste à un niveau bas.

Le schéma électrique

Dans cet article, nous vous proposons un montage un peu différent pour générer le signal PWM rectangulaire. Nous utilisons le circuit intégré polyvalent **NE555** dans une configuration inhabituelle.

En fait, le circuit fonctionne en **configuration astable**, en faisant **varier le rapport cyclique** grâce à un artifice qui exploite les différents modes de charges et de décharges des condensateurs de temporisation C1 et C2 (500 pF).

Pour comprendre le fonctionnement, nous devons d'abord étudier le fonctionnement du NE555. Ce circuit intégré contient deux comparateurs ayant en commun un diviseur de tension multiple qui polarise l'entrée inverseuse du comparateur supérieur et l'entrée non-inverseuse du comparateur inférieur (voir la figure 1).

L'entrée non-inverseuse du comparateur supérieur est accessible de l'extérieur du NE555 (les termes supérieur et inférieur se rapportent au potentiel de référence qu'ils reçoivent du diviseur de tension multiple, la partie supérieure recevant la tension la plus élevée) par la broche 6 (THRESHOLD, c'est-à-dire seuil) et l'entrée inverseuse du comparateur inférieur correspond à la broche 2 (TRIGGER).

Le nœud entre la première et la deuxième résistance du diviseur multiple de référence des deux comparateurs (alimenté comme le reste des circuits internes du 555 par la broche 8) est disponible à l'extérieur du circuit par l'intermédiaire de la broche 5 (CONTROL VOLTAGE).

Cela permet de modifier les tensions de référence des comparateurs, en contrôlant par une tension externe la fréquence de fonctionnement du circuit en configuration astable (c'est-à-dire en effectuant un décalage de fréquence comme cela se fait dans les oscillateurs commandés en tension) ou la durée des impulsions du timer en configuration monostable.

Plus précisément, dans la **configuration astable**, l'abaissement de la tension sur la broche 5 raccourcit la période de commutation de sortie et augmente ainsi la fréquence générée. Inversement, en augmentant la tension cela a pour effet d'allonger la période et donc de réduire la fréquence produite.

Dans une **configuration monostable**, la tension appliquée sur la broche 5

condensateurs **C1** et **C2** sont **déchargés** et que la tension est de 0 V.

En négligeant les effets de C3 qui se charge immédiatement et donc qui ne perturbe pas la tension de référence appliquée par le diviseur de tension multiple aux comparateurs internes du NE555, la condition initiale de ce dernier est telle que les broches 2 et 6 sont à environ 0 V et que la sortie du comparateur inférieur est à un niveau haut.

Cette condition détermine un niveau logique haut sur l'entrée « **SET** » de la bascule et donc un niveau haut sur la sortie Q (Q/ passe à niveau logique bas bloquant ainsi le transistor relié à la broche « **DISCHARGE** », les condensateurs peuvent alors se charger).

À ce stade, les condensateurs C1 et C2 commencent à se charger grâce au courant circulant dans les résistances R1, R2 et R3 (pour le moment nous négligeons la diode D1, la broche 7 n'absorbe aucun courant) jusqu'à ce que la tension de seuil du comparateur inférieur soit atteinte, elle est égale à 1/3 de celle appliquée à la broche 8.

Maintenant, la sortie du comparateur inférieur passe à un niveau logique 0 et réinitialise l'entrée SET de la bascule dont la sortie reste à un niveau haut jusqu'à ce que la tension aux bornes de C1 et C2, continuant à croître, atteigne 2/3 de la valeur de la tension d'alimentation du NE555 sans la dépasser. Cela correspond au seuil de commutation du comparateur supérieur.

Lorsque cela se produit, la sortie du comparateur supérieur passe à un niveau logique haut et remet à zéro la bascule. La sortie Q de celle-ci passe à un niveau logique 0, tandis que la sortie Q/ passe à un niveau haut et sature le transistor interne relié à la broche 7, qui à travers R2 et R3 charge le condensateur C1/C2, car la diode D1 est bloquée.

La situation reste inchangée jusqu'à ce que la tension aux bornes de C1/C2 ne tombe pas en dessous du seuil de commutation du comparateur inférieur. Lorsque la sortie de celui-ci revient à un niveau logique haut, nous obtenons un niveau logique 1 sur la sortie Q et un niveau logique 0 sur Q/.

Dans cette condition, le condensateur C1/C2 recommence à se charger, car la broche 7 est ouverte (transistor bloqué, collecteur ouvert). La charge continue jusqu'à atteindre 2/3 de la tension d'alimentation du NE555, le cycle recommence ainsi.

À la suite de cela, la sortie OUT du circuit intégré est maintenue à un niveau haut pendant la période où la tension aux bornes des condensateurs C1/C2 se situe entre 1/3 et 2/3 de la tension d'alimentation (comme le NE555 est alimenté en 9 V, cette tension est comprise entre 3 V et 6 V, cela correspond respectivement à 1/3 et 2/3 de la tension entre les broches 8 et 1).

La sortie OUT est maintenue à un niveau bas (environ 0 V) pendant la période de décharge des condensateurs C1/C2, c'est-à-dire lorsque la tension à leurs bornes passe de 2/3 à 1/3 de la tension d'alimentation.

Les délais dépendent des valeurs des résistances R1, R2 et R3 et sont d'autant plus semblables que la somme R2 + R3 est faible par rapport à R1. En effet, la charge s'effectue à travers ces 3 résistances tandis que la décharge a lieu seulement à travers R2 et R3.

Étudions maintenant le rôle de la diode D1, qui charge les condensateurs en contournant R3 et une partie de R2 entre le point milieu et l'extrémité reliée à R3. Lorsque le courant de charge des condensateurs est tel qu'il provoque aux bornes des résistances une chute de tension d'au moins 0,6 V, la diode se met à conduire de telle sorte que la charge de C1 et C2 se produit uniquement à travers R1 et une partie de R2.

Pendant la phase de décharge, la diode D1 est toujours bloquée. Nous pouvons dire que D1 intervient pendant la charge des condensateurs en court-circuitant la partie inférieure de R2 et R3. Pour simplifier, disons que C1 et C2 se déchargent à travers R3 et la partie inférieure du potentiomètre R2, et se chargent à travers R1 et la partie supérieure de R2. Cela permet dans certaines limites de faire varier le rapport cyclique de l'onde rectangulaire disponible en sortie du NE555 (broche 3) entre 0 et 100 %, tout en

maintenant constante la fréquence de cette onde lors des variations de la position du curseur de R2. Par conséquent, la durée d'une impulsion de niveau haut en sortie est déterminée de la manière suivante en négligeant la présence de la diode D1 :

$$th = 0,693 * (Ra + Rb) * C$$

C représente la capacité série des condensateurs C1 et C2, qui est égale à $C1C2/(C1 + C2)$ soit 500 pF. Cette période correspond à la charge des condensateurs.

Par contre la durée de la pause vaut :

$$tl = 0,693 * Rb * C$$

et correspond à la décharge des condensateurs.

La durée de la période **T** est égale à « **th + tl** » et le rapport cyclique vaut :

$$dc = th/T$$

Si le curseur de R2 est positionné vers R1, Ra est égal à R1, et Rb est égal à R2 + R3, soit 102 kΩ. En supposant que la diode D1 court-circuite Rb, la durée de l'impulsion est égale à : **th = 0,693 (1 kΩ) 500 pF = 0,3465 μs**

La durée de la pause vaut : **tl = 0,693 * 101 kΩ * 500 pF = 0,035 ms**

Dans ce cas, la période théorique est d'environ 0,353 ms, et le rapport cyclique vaut : **dc = 0,3465 μs / 0,035 ms = 0,0099** soit 0,99 %.

La fréquence est théoriquement de 28 kHz, bien qu'en réalité la diode intervient plus tôt, la fréquence réelle est de 25 kHz. Si le curseur de R2 est positionné vers R3, Ra est égale à 101 kΩ (R1 + R2) et Rb 1 kΩ (R3). La durée d'impulsion est de :

$$th = 0,693 * (101 kΩ + 1 kΩ) * 500 pF = 0,03543 ms$$

Tandis que la pause dure : **tl = 0,693 * 1 kΩ * 500 pF = 0,3465 μs**

Dans ce cas, la période a une durée de 0,03577 ms et la fréquence est égale à 27,95 kHz. Le rapport cyclique vaut : **dc = 0,03543 ms / 0,03577 ms = 0,99** soit 99 %.

Les moteurs à courant continu

Dans sa forme élémentaire, un moteur à courant continu à balais est formé d'une bobine tournante (rotor) située entre les pièces polaires d'un aimant (stator), dont les extrémités sont reliées à un anneau appelé « collecteur » par lequel elle reçoit l'alimentation.

Pour comprendre le fonctionnement, imaginons que nous faisons tourner la bobine de sorte que les deux contacts soient chacun touchés par un balai et que la polarité est telle que le champ magnétique de polarité nord de la bobine soit orienté vers le nord de l'aimant, de même pour la polarité sud. Nous constatons donc, qu'à chaque fois, la bobine est repoussée par les aimants. Elle se met à tourner. En raison de la rotation, les contacts de l'axe tournent et ne touchent plus les balais, de sorte que le champ magnétique cesse et avec lui la force de répulsion. Comme le rotor a une certaine inertie, la rotation continue jusqu'à ce que le balai de polarité opposé touche le contact. Le courant parcourt alors la bobine dans la direction opposée qui présente un champ opposé. Ainsi nous obtenons une situation telle que le pôle sud du champ électromagnétique de la bobine se trouve face au pôle sud de l'aimant inférieur et le pôle nord de la bobine face au pôle nord de l'aimant supérieur. Cela implique une nouvelle répulsion et donc une nouvelle poussée qui fait continuer la rotation du rotor (bobine) jusqu'à ce que les contacts ne touchent plus les balais. L'inertie fait en sorte que la rotation continue jusqu'au retour des conditions initiales. Le cycle se répète ainsi à l'infini, en maintenant le rotor en rotation.

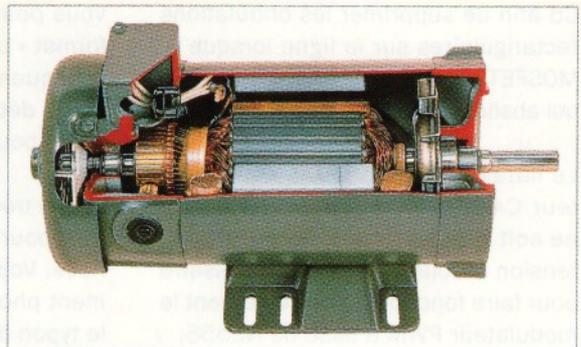
Dans la pratique, le rotor des moteurs est constitué par des pièces polaires en fer ou d'autres matériaux ferromagnétiques et sur celles-ci sont réalisés des enroulements (bobines) connectés de façon appropriée au collecteur. De cette façon, vous obtenez des champs électromagnétiques très intenses et des couples moteurs élevés.

Le collecteur d'un moteur à courant continu est formé, sous sa forme la plus simple, de deux contacts en forme d'arc. Généralement, les contacts sont des multiples de 2 toujours disposés en opposition. Chaque paire représente une bobine et chaque bobine correspond à une paire de pièces polaires. En fait, les pôles du rotor peuvent être plus nombreux, 4 ou plus.

Un moteur à courant continu à aimant permanent a un comportement réversible. Il devient un générateur de courant continu (dynamo) si l'on fait tourner l'axe du moteur. Il est possible de récupérer ainsi l'énergie électrique produite en se reliant sur les balais. Le moteur peut agir ainsi comme un frein, en connectant une résistance entre les balais, l'énergie mécanique transmise par l'axe est dissipée sous forme de chaleur dans cette résistance.

Les balais sont constitués de graphite, ce matériau permet un bon contact électrique et réduit au minimum la friction. Cependant, ils doivent être remplacés périodiquement. Les balais constituent une limitation de la vitesse de rotation maximale. Plus la vitesse est grande, plus la pression exercée sur les balais doit être grande pour maintenir un bon contact.

Les balais, lorsqu'ils ne sont pas en contact avec le collecteur, produisent des phénomènes transitoires dus aux enroulements inductifs. Des étincelles sont produites, ils peuvent perturber des dispositifs électriques (radio, TV, etc.). De tels problèmes peuvent être évités en échangeant la position du rotor avec celle du stator, ou en plaçant dans le stator les enroulements et en disposant sur le rotor deux ou quatre aimants de pôles opposés les uns par rapport aux autres. Il n'y a plus besoin de collecteur ou de balais, les enroulements peuvent dissiper plus facilement la chaleur générée par le moteur. Ces moteurs sont appelés « brushless » (sans balais).



Nous venons donc de décrire le fonctionnement détaillé du NE555, poursuivons l'analyse du schéma. Le signal présent sur la broche OUT pilote, par l'intermédiaire du diviseur de tension formé par R4 et R5, la grille du transistor MOSFET T1.

Ce dernier est un transistor MOS à enrichissement à **canal N** (enhancement-mode) qui est utilisé comme un interrupteur statique.

Il est relié au moteur par les bornes + et - des contacts de la sortie OUT, les impulsions produites par le NE555 sont donc amplifiées en puissance.

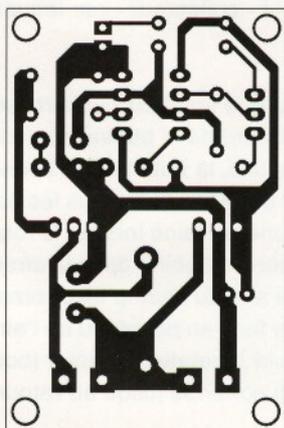
La diode **D2**, connectée en parallèle sur les bornes de sortie, **supprime les impulsions** des tensions inverses générées par le moteur lorsque les balais commutent avec le collecteur et aussi tout pic de tension inverse résultant du pilotage de charges inductives.

Concluons l'analyse du circuit avec la source d'alimentation, elle doit être appliquée aux bornes « PWR+ » et « PWR- ».

La tension d'alimentation arrive sur l'entrée du régulateur U2 et sur le circuit de puissance T1 et D2 de la sortie OUT.

L'alimentation est convenablement filtrée par le condensateur électrolytique

Plan de montage du variateur de vitesse pour moteur DC



Circuit imprimé à l'échelle 1 : 1 côté soudures.

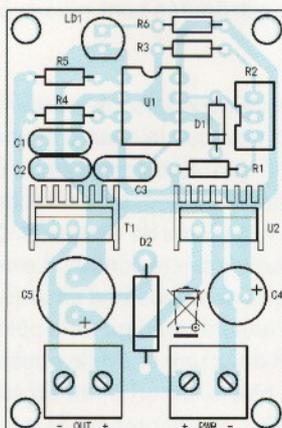


Schéma de câblage des composants du variateur de vitesse pour moteur DC.

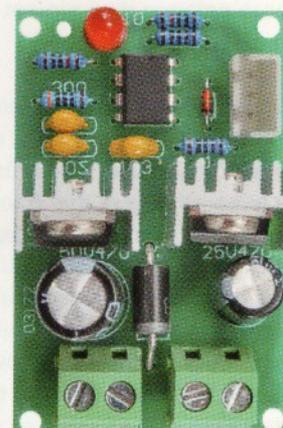


Photo de l'un de nos prototypes du variateur de vitesse pour moteur DC.

Liste des composants

R1 1 k Ω 1%
 R2 potentiomètre logarithmique
 100 k Ω
 R3 1 k Ω 1%
 R4 300 Ω 1%
 R5 10 k Ω 1%
 R6 10 k Ω 1%

C1 1 nF céramique
 C2 1 nF céramique
 C3 10 nF céramique
 C4 470 μ F/25 V électrolytique
 C5 470 μ F/50 V électrolytique
 T1 P75NF75
 D1 1N5408
 D2 BAT41
 LD1 LED 5 mm rouge

U1 NE555
 U2 L7809

Divers :
 Bornier 2 pôles au pas de 5 mm (x 2)
 Dissipateur pour TO220 (x2)
 Vis 10 mm (x2)
 Connecteur enfichable pour circuit
 imprimé 3 pôles

C5 afin de supprimer les ondulations rectangulaires sur la ligne lorsque le MOSFET conduit pour piloter le moteur qui absorbe beaucoup de courant.

Le filtrage effectué par le condensateur C4 assure que la sortie de U2 ne soit pas perturbée et fournit une tension stabilisée de 9 V, nécessaire pour faire fonctionner correctement le modulateur PWM à base de NE555.

La présence de l'alimentation est signalée par la LED LD1 qui est protégée par la résistance de limitation de courant R6.

Réalisation pratique

Passons maintenant à la réalisation pratique du variateur de vitesse pour moteur DC. Le circuit imprimé comporte une seule face, il est facilement réalisable par photogravure.

Vous pouvez télécharger le typon au format « pdf » sur notre site www.electroniquemagazine.com dans le sommaire détaillé de la revue en cliquant sur le bouton « Télécharger ».

Vous y trouverez aussi les fichiers GERBER pour une fabrication professionnelle. Vous pouvez aussi tout simplement photocopier sur un transparent le typon dans l'encadré intitulé

« Plan de montage du variateur de vitesse pour moteur DC » et insoler une plaque de cuivre photosensible.

Une fois le circuit imprimé gravé et percé, vous devez commencer l'assemblage des composants. Ils sont tous de type traditionnel, il n'y a pas de composants CMS.

Comme d'habitude commencez par souder les résistances, la diode D1 et les condensateurs non polarisés (C1 à C3).

Vous pouvez utiliser un support de 2 x 4 broches pour le NE555 ou le souder directement sur le circuit.

Ensuite soudez la diode D2, les condensateurs polarisés, la LED dont le côté biseauté se situe vers U1 et les borniers.

Nous vous conseillons d'abord de fixer le transistor et la diode sur des dissipateurs de résistance thermique 20°C/W. Présentez les deux composants avec leur radiateur aux emplacements prévus en respectant l'orientation (voir la photo du prototype).

Les parties inférieures des dissipateurs doivent toucher le circuit imprimé. Afin de faciliter le transfert de chaleur, en particulier pour le MOSFET, vous pouvez utiliser de la pâte au silicium sur les radiateurs. Le potentiomètre doit être relié à l'aide de 3 fils au bornier.



Cours ARDUINO

Neuvième partie

.....La rédaction

Dans cette neuvième leçon, nous allons vous proposer d'interfacer Arduino avec un récepteur GPS afin de réaliser nos premières expériences de localisation de satellites. Pour cela, nous allons utiliser deux cartes d'extension, une dénommée « GPSSHIELD » et l'autre « ET1017M », dont les composants CMS sont déjà soudés. Ainsi il ne vous restera plus qu'à souder quelques composants classiques ainsi que les connecteurs, le tout pour quelques dizaines d'euros.

Dans cette leçon, nous franchissons un pas important en expérimentant la technologie **GPS** (Global Positioning System) afin de détecter la position des satellites en orbite autour de la terre. Comme vous le savez, au-dessus de nos têtes à environ 20 000 km, se trouvent un nombre important de satellites qui envoient constamment sur terre une série d'informations. Celles-ci sont captées et ensuite traitées afin de déterminer notre position exacte sur le globe avec une précision d'une dizaine de mètres.

Pour recevoir et analyser les données satellitaires, nous devons disposer d'un récepteur GPS doté de son antenne. Celui-ci doit d'abord « accrocher » au signal provenant des satellites, puis

doit fournir en sortie une chaîne de caractères ASCII contenant, en plus de la position, d'autres données importantes.

Le récepteur GPS doit évidemment être en mesure de recevoir des signaux provenant de l'espace et donc il ne fonctionne pas dans des environnements fermés, tels que les appartements ou les magasins, etc.

De plus, comme tous les équipements radio fonctionnant dans la gamme des micro-ondes (au-delà du gigahertz), il reçoit mal les signaux en présence de grands bâtiments ou de constructions, car à ces fréquences la propagation du signal est pratiquement linéaire.

Tableau 1 - Contenu de la trame NMEA

Données NMEA	Description
GGA	Données relatives à la position
GLL	Position géographique
GSA	Liste des satellites actifs
GSV	Satellites capturés
RMC	Données minimales essentielles sur la position
VTG	Direction au sol et vitesse

Si la réception est bonne, après l'allumage du récepteur GPS s'écoule un temps d'acquisition, habituellement entre 30 secondes et 1 minute. Ensuite le récepteur passe à la phase de localisation qui donne la position.

La séquence des données en sortie du récepteur GPS est composée de simples caractères ASCII codés dans une trame dénommée « **NMEA** ». Celle-ci contient la longitude et la latitude du récepteur, la hauteur par rapport au niveau de la mer, la date, l'heure et les indications des satellites accrochés.

Aujourd'hui, dans le commerce vous pouvez trouver une variété de modèles GPS pouvant être facilement utilisés dans vos applications hobbyistes. Si vous devez acheter un module GPS, il y a quelques paramètres à prendre en compte.

Notamment, les dimensions sont des paramètres importants dans des dispositifs où l'emplacement est limité. Cela inclut également l'antenne de réception, qui peut être intégrée ou externe.

Un autre paramètre important est le nombre de lectures effectuées par seconde. Les modèles d'entrée de gamme à faible coût, effectuent une lecture par seconde, tandis que les modèles plus développés (et donc plus chers) peuvent effectuer jusqu'à 10 lectures par seconde.

L'alimentation est un autre paramètre important, les récepteurs sont des dispositifs conçus pour fonctionner dans un équipement portable. Ils sont donc alimentés en 3,3 V ou en 5 V avec une consommation de courant comprise entre 10 et 50 mA (selon le modèle), cette valeur a tendance à diminuer avec l'amélioration de la technologie.

Le nombre de canaux gérables est un autre paramètre généralement mis en avant par les constructeurs, mais dans

Tableau 2 - Description du brochage du EM506

Broche	Nom	Fonction
1,5	GND	0 V de l'alimentation (masse)
2	Vin	Alimentation positive 4,5 V à 6,5 V
3	RX	Signal d'entrée RS232 pour les opérations de programmation
4	TX	Signal de sortie RS232. Sortie des données de navigation au logiciel de navigation. Niveau de sortie 3,3 V.
6	Directive	Cette broche indique les différents états du GPS.

la réalité un nombre inférieur de satellites est accroché et donc les données résultantes sont de moins bonne qualité que celles annoncées.

La sensibilité donne une indication sur la capacité du récepteur à pouvoir détecter des signaux faibles et donc de son bon fonctionnement dans les zones perturbées ou dans lesquelles un faible signal arrive (par exemple en centre-ville).

La précision est un autre paramètre important et est étroitement liée au prix du récepteur. Si nous mettons de côté les versions coûteuses pour des applications professionnelles, les récepteurs commerciaux ont une précision de ± 10 mètres.

Pour ceux d'entre vous qui se demandent à quoi peut servir un récepteur GPS, nous leurs disons que la principale application est de mémoriser un trajet. Par exemple dans les randonnées, en voiture et de plus en plus souvent dans les voitures autonomes (sans conducteur).

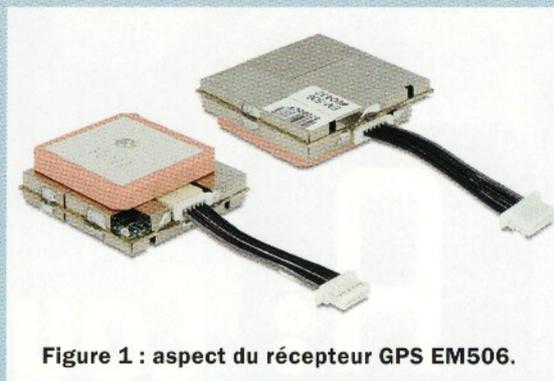


Figure 1 : aspect du récepteur GPS EM506.

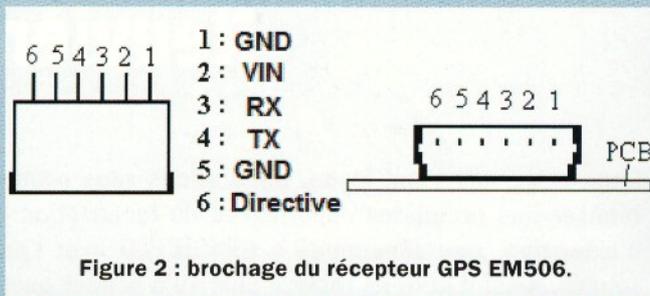


Figure 2 : brochage du récepteur GPS EM506.

Le plus souvent, les récepteurs GPS de dernière génération sont intégrés dans les smartphones haut de gamme. Ainsi, il est possible de disposer de toute une gamme de services, y compris des visites guidées de monuments dans les villes ou encore retrouver le chemin du retour de vacances.

Les smartphones ayant un récepteur GPS intégré peuvent devenir (avec un logiciel approprié) un système de navigation très performant avec même une synthèse vocale. Parmi les modèles disponibles sur le marché, nous avons sélectionné un récepteur entièrement compatible avec une carte Arduino. Compte tenu de la miniaturisation extrême du GPS, l'utilisation d'un circuit imprimé fait maison ou d'une plaque d'essai n'est pas envisageable.

Cependant, il existe plusieurs types de cartes d'extension déjà pré-montées (composants CMS soudés) pour quelques dizaines d'euros. Nous avons choisi comme récepteur GPS le modèle **EM506** doté de 48 canaux et basé sur le chipset « SiRF Star IV » qui lui procure une **haute sensibilité** de l'ordre de -163 dBm.

Ce récepteur GPS dispose d'une antenne intégrée, sa tension d'alimentation est comprise entre 4,5 V et 6,5 V. Il s'interface simplement avec un microcontrôleur. Une LED signale les différents états du module. Ces principales caractéristiques sont les suivantes :

GPS 48 canaux avec le chipset « SiRF Star IV » à haute sensibilité (-163 dBm) ;

- précision de la position : ± 5 mètres ;
- précision du calcul de la vitesse : $\pm 0,1$ m/s ;
- TTFF (Time To First Fix) extrêmement efficace même avec des signaux faibles ;
- supporte les protocoles de données NMEA 0183 V3.0 (GGA, GSA, GSV, RMC, VTG, GLL, ZDA)
- protocole de communication en logique TTL : 4800 bps ;
- technologie « SuperCap » qui permet l'acquisition rapide des données du satellite ;
- antenne intégrée ;
- temps de démarrage à froid < 35 s ;
- temps de démarrage à chaud < 1 s ;
- altitude maximale : 18 000 m ;
- alimentation de 4,5 V à 6,5 V ;
- consommation 35 mA ;
- dimensions : 30 x 30 x 10,7 mm.

Tableau 3 - données en sortie de la séquence GGA.

Nom	Exemple	Unité	Description
Message ID	\$GPGGA		GGA protocole d'en-tête
UTC Time	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicateur	N		N = nord ou S = sud
Longitude	12158.3416		dddmm.mmmm
E/W Indicateur	W		E = est ou W = ouest
Indicateur de correction de position	1		0 = aucune correction 1 = correction OK (GPS SPS Mode) 2 = correction OK (Differential GPS, SPS Mode) 6 = correction OK (mode de détection immédiate)
Satellites utilisés	07		de 0 à 12
HDOP	1.0		précision horizontale
MSL Altitude	9.0	mètres	
Unités	M	mètres	
Séparation des géoïdes	34.2	mètres	
Units	M	mètres	
Age of Diff. Corr.			Champs nuls lorsque DGPS n'est pas utilisé.
Diff. Ref. Station ID	0000		
Checksum	*18		Somme de contrôle
<CR><LF>			Fin du message

Il est possible de paramétrer le récepteur afin de disposer en sortie de différents ensembles de données. Par exemple, la séquence « GGA » est celle indiquée dans le tableau 3.

La chaîne reçue, aussi appelée trame, est la suivante :

```
$GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,,,0000*18
```

La carte d'interface « GPSSHIELD »

Pour interfacier correctement une carte Arduino avec le récepteur GPS, nous avons simplement utilisé la carte d'interface « **GPSSHIELD** » pré-montée et équipée d'un connecteur JST-SH à 6 broches pour le module EM506. La carte dispose d'un bouton de remise à zéro et d'un interrupteur ON/OFF pour le module GPS. Par contre les connecteurs pour relier la carte à Arduino doivent être achetés séparément, puis soudés.

La carte d'interface « GPSSHIELD » possède un emplacement pour loger le module EM506. Elle fournit l'alimentation de 5 V, il faut câbler les 2 lignes TX et RX du GPS avec les lignes D2 et D3 d'Arduino. Avant de mettre sous tension, il est nécessaire de vérifier la position de l'interrupteur « **DLINE-UART** ».

Lorsque l'interrupteur est sur la position UART, le module GPS est relié aux lignes digitales 0 et 1, c'est-à-dire aux lignes TX et RX du module UART d'Arduino. Lorsque l'interrupteur est en position DLINE, le récepteur GPS est relié aux lignes digitales D2 et D3 d'Arduino.

Placez l'interrupteur en position DLINE soit pour programmer Arduino soit pour tester le sketch (programme). De cette façon, il n'y aura pas de conflit avec les données provenant de l'USB, car elles utilisent les lignes TX et RX (broches 0 et 1).

Une fois le module GPS installé dans son emplacement, reliez les 6 fils et soudez les côtés de la partie métallique du module sur les 2 pastilles situées sur le circuit imprimé.

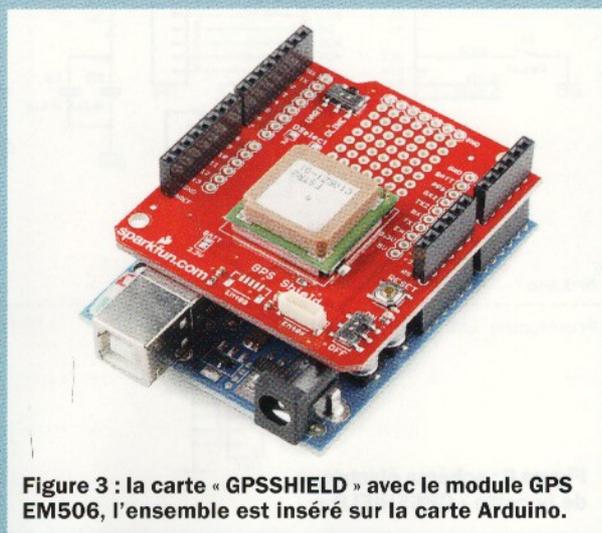


Figure 3 : la carte « GPSSHIELD » avec le module GPS EM506, l'ensemble est inséré sur la carte Arduino.

Ensuite soudez les connecteurs comme vous pouvez le voir en figure 5.

En figure 4 vous pouvez voir les différents éléments se trouvant sur la carte « GPSSHIELD » :

- interrupteur d'alimentation ON/OFF, contrôle l'alimentation du module GPS ;
- interrupteur de sélection « DLINE-UART ». En position UART les lignes séries du module GPS sont reliées aux

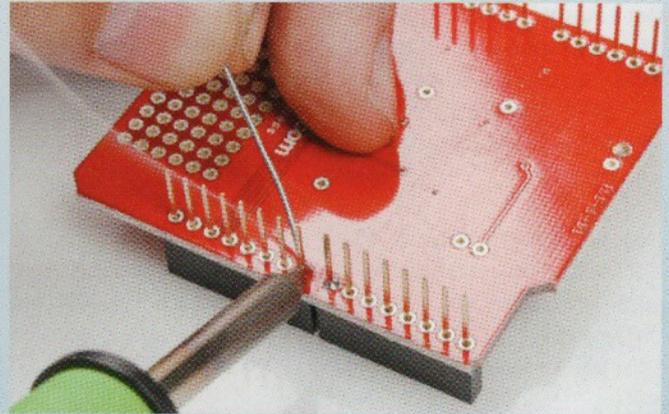


Figure 5 : la carte « GPSSHIELD » avec ses divers connecteurs à souder.

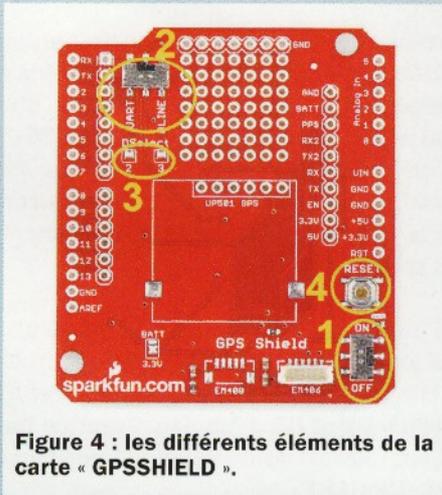
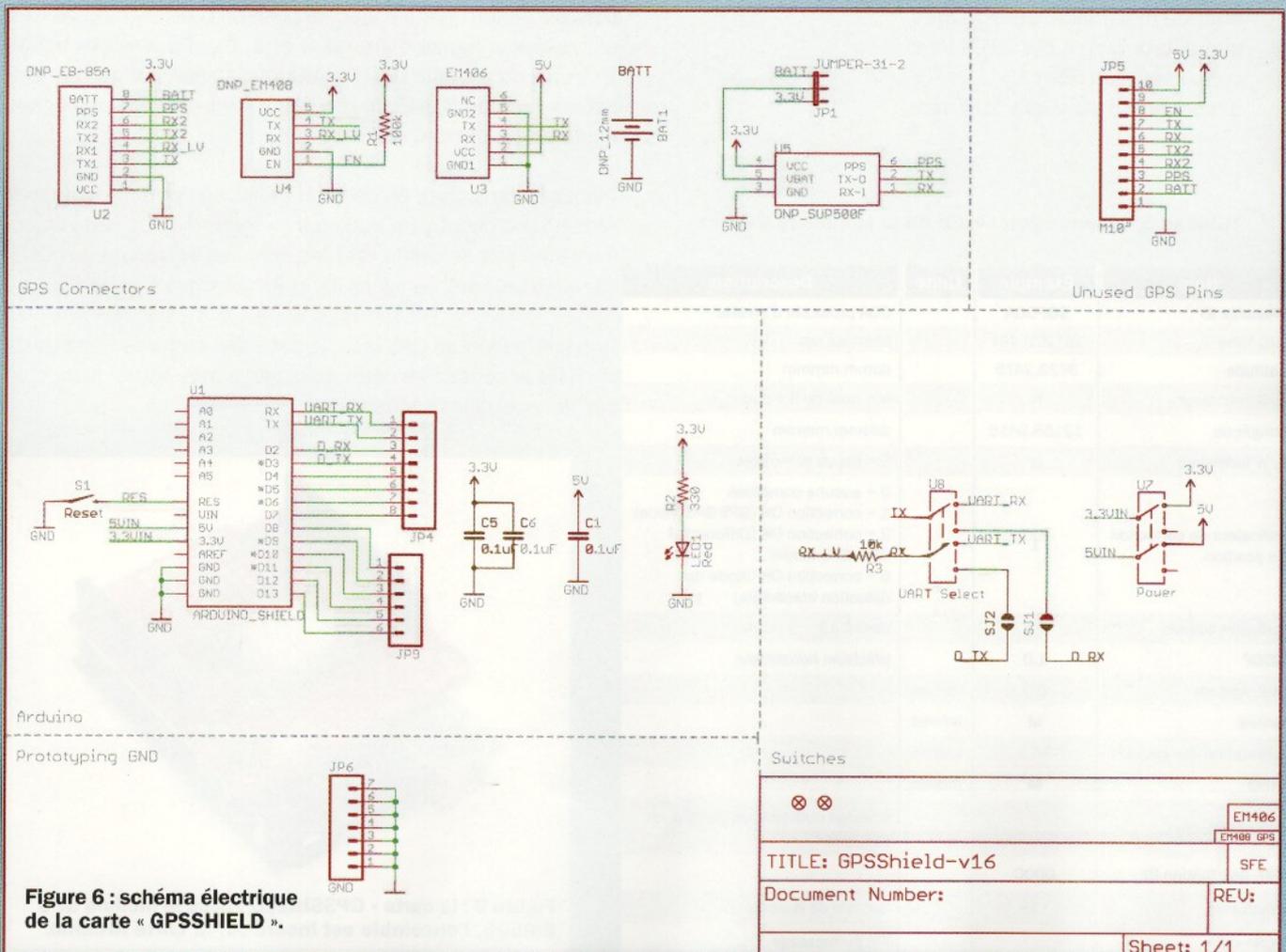


Figure 4 : les différents éléments de la carte « GPSSHIELD ».

broches séries (D0/RX et D1/TX) d'Arduino. En position DLINE (par défaut), le GPS est relié aux broches D2 et D3 d'Arduino ;

- ponts SJ1 et SJ2 (DSelect) à souder pour relier les lignes « D_RX » et « D_TX » aux broches 2 et 3 du circuit imprimé de la carte d'extension ;
- bouton de réinitialisation.

En figure 6, vous pouvez voir le schéma électrique de la carte « GPSSHIELD ». La carte est disponible dans le commerce déjà pré-montée comme nous l'avons indiqué en début d'article.



Etant donné la complexité de la fabrication du circuit imprimé (double face et composants CMS), nous vous conseillons de l'acheter toute faite dans le commerce.

Avant de pouvoir utiliser le module GPS, vous devez installer la librairie « **TinyGPSPlus** » dans le dossier « **libraries** » de l'environnement de développement IDE d'Arduino. Vous devez voir le dossier nommé « **TinyGPSPlus-master** ». Vous pouvez la télécharger sur notre site www.electroniquemagazine.com dans le sommaire détaillé de la revue.

Enfichez la carte « **GPSSHIELD** » sur la carte Arduino et connectez cette dernière à votre ordinateur avec un câble USB. Assurez-vous que l'**interrupteur DLINE/UART se trouve sur la position DLINE** chaque fois que vous téléchargez un sketch, même si vous avez l'intention d'utiliser l'UART.

Si le GPS (ou tout autre périphérique série) est relié pendant la phase de programmation, vous aurez un défaut à la fin du processus (échec du téléversement). Si vous souhaitez utiliser l'UART, mettez l'interrupteur sur la position UART après la fin du téléversement.

Pour l'exemple suivant, assurez-vous que l'interrupteur d'alimentation de la carte GPS soit en position ON et que l'autre interrupteur soit en position DLINE, Arduino étant connecté au PC via le câble USB. Téléchargez l'exemple du sketch décrit dans le listing 1 dans Arduino.

Sur la carte GPS, la LED rouge doit s'allumer indiquant que l'alimentation est correcte. La LED du module EM506 donne les indications suivantes :

- **LED OFF** = récepteur hors tension ;
- **LED ON** = GPS non verrouillé, recherche d'un signal ;
- **LED clignotante** = signal accroché.

Si la LED du module EM506 reste allumée de manière fixe pendant plus d'une minute, cela signifie que vous êtes dans une position pour laquelle le signal GPS n'a pas une intensité suffisante.

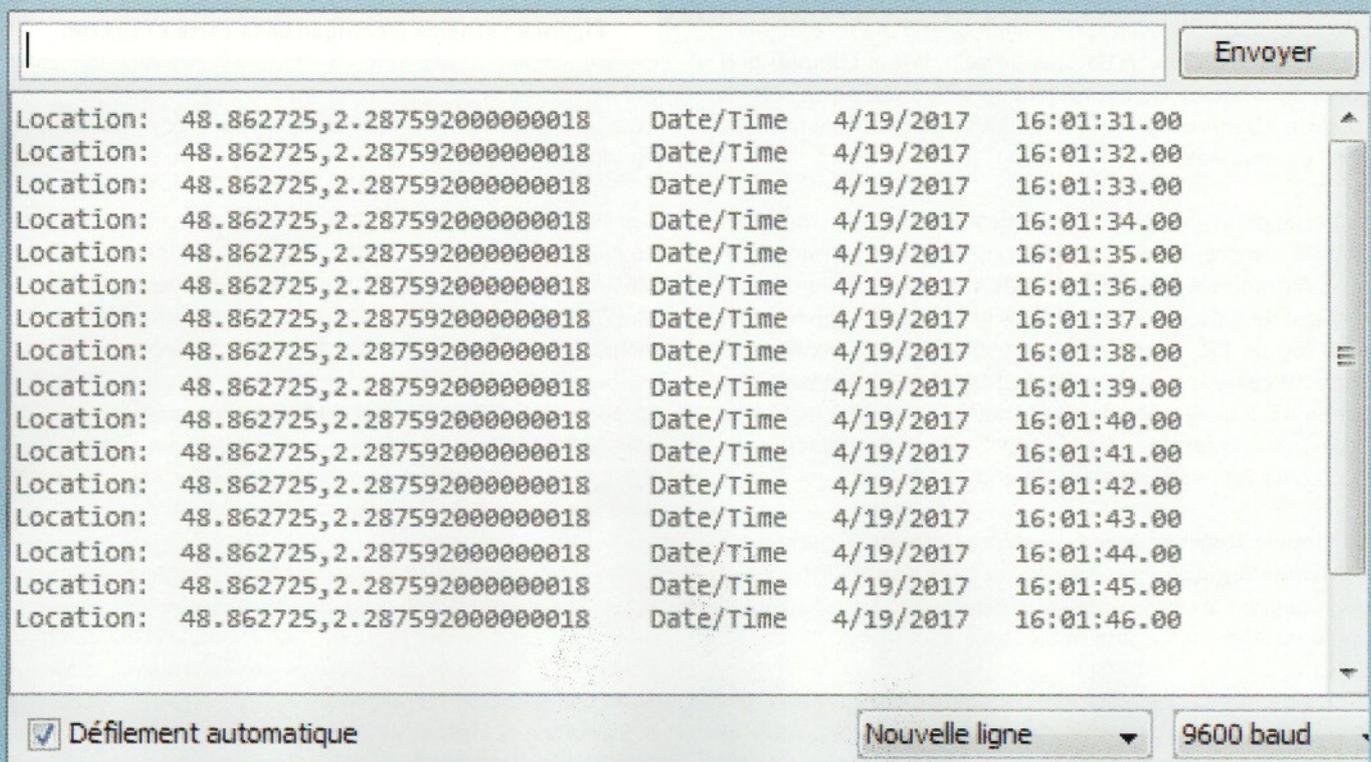
Déplacez-vous vers une zone où les signaux provenant des satellites sont plus puissants, et attendez jusqu'à ce que le GPS accroche un signal correct pour que la trame NMEA puisse être décodée et afficher les données relatives à votre position chaque seconde.

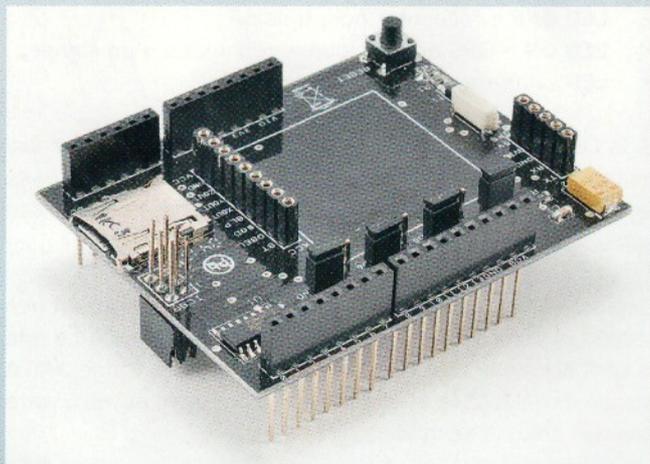
Ouvrez le moniteur série d'Arduino en allant dans le menu « **Outils → Moniteur série** ». Les valeurs de la latitude, de la longitude, de la date et de l'heure doivent s'afficher comme visible en figure 7.

La carte d'extension GPS datalogger ET1017M

En connectant la carte **ET1017M** dotée d'un récepteur GPS de type EM506 à une carte Arduino, nous pouvons créer un **enregistreur de données GPS** simple et efficace. Ce dispositif vous permet de tracer l'itinéraire d'une personne ou d'un véhicule (ou tout autre objet en mouvement) en effectuant simplement une mise en mémoire périodique des points de localisation provenant du GPS. Cette carte **ET1017M sauvegarde les positions enregistrées sur une carte mémoire de type microSD**.

Figure 7 : les valeurs de la latitude, de la longitude, de la date et de l'heure s'affichent chaque seconde dans le Moniteur série.





Vous pouvez ensuite visualiser les données sur votre ordinateur. Cette carte n'est pas un simple datalogger, elle est intelligente dans le sens où elle arrête l'enregistrement des données lorsque la destination est atteinte. Cela permet de ne pas effacer les premières données sauvegardées dans la mémoire.

Cette fonctionnalité est obtenue grâce à un accéléromètre (optionnel) qui, en plus d'arrêter la mémorisation des données lorsque le dispositif ne se déplace plus, permet également d'éteindre le module GPS afin de réduire la consommation. Cela permet de prolonger l'autonomie des batteries.

Le schéma électrique de la carte ET1017M est visible en figure 8. Elle permet d'accueillir un récepteur GPS, un accéléromètre et une carte microSD. L'ensemble se connecte sur une carte Arduino grâce à une série de connecteurs. La carte est alimentée directement à partir d'Arduino.

Nous avons obtenu une consommation d'environ 35 mA sous une tension de 5 V avec le GPS allumé, et 5 mA en éteignant le GPS via la broche 7. En plus de cela, il faut compter une consommation de 40 mA pour la carte Arduino UNO, soit un total de 75 mA en mode d'acquisition et d'enregistrement des coordonnées.

La partie principale de la carte est composée du module GPS EM506 comprenant une antenne intégrée et communiquant avec Arduino via le protocole **NMEA 0183**. La communication est de type série (4800 bps) et utilise les lignes TX et RX à niveau TTL. Ce signal est disponible sur le connecteur série GPS et grâce aux cavaliers GPSRX et GPSTX il peut être acheminé soit vers les broches 0 et 1 (TX et RX c'est-à-dire l'UART physique) soit vers les broches 5 et 6 (utilisation de la librairie SoftwareSerial) d'Arduino.

Le **module GPS est activé** lorsque la **broche 7** est mise à un **niveau logique bas**. À partir de ce moment, il transmet cycliquement la trame NMEA sur le port série jusqu'à ce qu'il soit désactivé en mettant la broche 7 à un niveau haut.

En ce qui concerne le connecteur microSD, la carte est dotée d'un convertisseur de niveaux de type 74HC4050D afin de permettre la communication via le bus SPI en 3,3 V

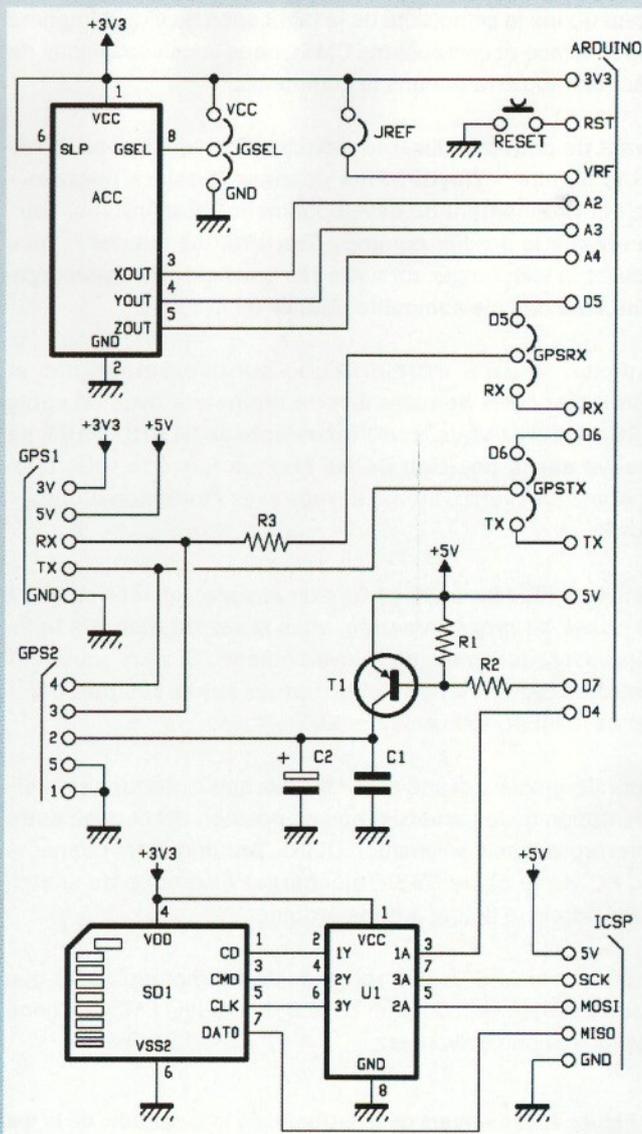


Figure 8 : schéma électrique de la carte ET1017M.

avec la carte mémoire (qui enregistre les données reçues du module GPS).

Les signaux correspondants sont reliés au connecteur ICSP de manière à permettre la compatibilité avec la carte Arduino MEGA. La carte mémoire peut être formatée en FAT16 ou en FAT32, il est nécessaire d'utiliser la librairie SD Arduino incluse dans l'environnement de développement.

Le **connecteur ACC permet de relier un accéléromètre** de type **MMA7361** (non compris). Le capteur est alimenté en 3,3 V et fournit 3 signaux analogiques aux broches 3, 4 et 5 pour les 3 axes X, Y et Z respectivement.

Grâce au cavalier JREF, il est possible de connecter la broche V_{REF} d'Arduino au 3,3 V, de sorte que la sortie de l'accéléromètre, qui est à 1,65 V (0 g), corresponde à une accélération nulle (0 g). Cela coïncide approximativement avec la valeur 512 de la sortie du convertisseur A/D de 10 bits, c'est-à-dire coïncide avec la moitié de l'excursion (la sortie du convertisseur prenant les valeurs 0 à 1023).

Listing 1

```

#include <TinyGPS++.h>
#include <SoftwareSerial.h>

// Cet exemple utilise la librairie TinyGPS++
// Choisissez 2 broches Arduino à utiliser pour le logiciel série (port série virtuel)
// La carte GPS utilise D2 et D3 par défaut en mode DLINE

int RXPin = 2;
int TXPin = 3;

// par défaut 4800 bauds
int GPSBaud = 4800;

// Crée un objet TinyGPS ++ appelé "gps"
TinyGPSPlus gps;

// Crée un port série logiciel (virtuel) appelé "gpsSerial"
SoftwareSerial gpsSerial(RXPin, TXPin);
void setup()
{
  // Démarre le port série matériel d'Arduino à 9600 bauds
  Serial.begin(9600);

  // Démarre le port série logiciel du GPS à défaut bauds
  gpsSerial.begin(GPSBaud);
  Serial.println(F("DeviceExample.ino"));
  Serial.println(F("Une simple démonstration de TinyGPS ++ avec un module GPS lié «)"));
  Serial.print(F("Test de la librairie TinyGPS ++ v. »)); Serial.println(TinyGPSPlus::libraryVersion());
  Serial.println(F("par Mikal Hart"));
  Serial.println();
}
void loop()
{
  // Ce sketch affiche des informations chaque fois qu'une nouvelle trame est correctement décodée.

  while (gpsSerial.available() > 0)
    if (gps.encode(gpsSerial.read()))
      displayInfo();

  // Si 5000 millisecondes passent et qu'il n'y a pas de caractères entrants
  // alors affiche sur le port série logiciel l'erreur «No GPS detected» (pas de GPS détecté)

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected"));
    while(true);
  }
}
void displayInfo()
{
  Serial.print(F("Location: «));
  if (gps.location.isValid())
  {
    Serial.print(gps.location.lat(), 6);
    Serial.print(F("«,»));
    Serial.print(gps.location.lng(), 6);
  }
  else
  {
    Serial.print(F("INVALID»));
  }
}

```

```

Serial.print(F(" Date/Time: "));
if (gps.date.isValid())
{
  Serial.print(gps.date.month());
  Serial.print(F("/"));
  Serial.print(gps.date.day());
  Serial.print(F("/"));
  Serial.print(gps.date.year());
}
else
{
  Serial.print(F("INVALID"));
}
Serial.print(F(" « ));
if (gps.time.isValid())
{
  if (gps.time.hour() < 10) Serial.print(F("0"));
  Serial.print(gps.time.hour());
  Serial.print(F(":"));
  if (gps.time.minute() < 10) Serial.print(F("0"));
  Serial.print(gps.time.minute());
  Serial.print(F(":"));
  if (gps.time.second() < 10) Serial.print(F("0"));
  Serial.print(gps.time.second());
  Serial.print(F("."));
  if (gps.time.centisecond() < 10) Serial.print(F("0"));
  Serial.print(gps.time.centisecond());
}
else
{
  Serial.print(F("INVALID"));
}
Serial.println();
}
    
```

Le cavalier JGSEL permet de sélectionner l'un des deux modes de fonctionnement de l'accéléromètre. Lorsqu'il est relié à la masse (GND), la pleine échelle est de 1,5 g avec une résolution de 800 mV/g, ce qui correspond à une valeur de lecture d'environ 248 pour 1 g.

Lorsqu'il est relié à V_{CC}, la pleine échelle passe à 6 g et la résolution est de 206 mV/g, ce qui correspond à une valeur de lecture d'environ 64 pour 1 g.

La configuration par défaut de la carte ET1017M est la suivante (voir la figure 12) :

- les cavaliers GPSRX et GPSTX sont fermés sur les broches de droite, c'est-à-dire sur D5 et D6 ;
- le cavalier JGSEL (accéléromètre) est fermé sur GND ;
- le cavalier JREF est fermé.

Réalisation pratique

La carte d'extension GPS datalogger ET1017M est livrée avec les composants CMS déjà soudés, il ne vous reste plus qu'à souder les différents connecteurs, les barrettes des cavaliers et le bouton de RESET.

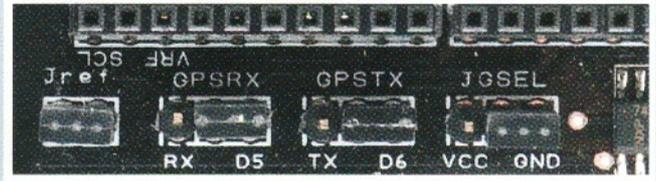


Figure 12 : configuration des différents cavaliers de la carte ET1017M.

Cela ne présente aucune difficulté particulière. Reportez-vous à l'encadré intitulé « Plan de montage de la carte ET1017M ».

Le firmware

Pour la compréhension et la modification du programme, notamment en ce qui concerne les trames mémorisées dans la carte SD, reportez-vous au Cours Arduino 7 traitant des cartes mémoires (n° 137 d'Électronique et Loisirs Magazine).

Le « SoftwareSerial », c'est-à-dire le port série virtuel, nous aidera à gérer la communication avec le module GPS. Si la carte mémoire n'est pas présente, les données GPS seront envoyées sur le port série d'Arduino, de sorte qu'elles puissent être exploitées directement par un ordinateur pour le débogage ou dans des applications avancées.

L'accéléromètre teste chaque seconde si un mouvement se produit. Si un mouvement est perçu dans n'importe quelle direction, l'enregistreur reste actif ou est activé s'il était auparavant en veille.

D'autre part, si les mouvements restent inférieurs à un certain seuil (M_THRESH) pendant plus d'une minute (la durée en millisecondes est configurable grâce à la valeur STOP), le GPS est désactivé.

Les valeurs du programme visible dans le listing 2 ont été vérifiées empiriquement pour fournir des résultats acceptables dans des cas typiques d'utilisation.

Pour des raisons de commodité, lorsque nous arrêtons l'enregistrement des données dans une phase de débogage, nous écrivons une chaîne propriétaire « PXXXX » dans le fichier NMEA, qui sera ignorée par le programme.

La même chaîne est utilisée pour signaler tout débordement du « SoftwareSerial », ce qui serait la cause d'un problème.

Le fichier SD (data.log dans notre exemple) est fréquemment fermé, de sorte qu'à tout moment il est possible de couper l'alimentation et retirer la carte SD afin d'en exploiter les données.

Gestion des trames NMEA

Les trames NMEA enregistrées par notre datalogger peuvent être directement lues par un logiciel nommé **GPSBabel**.

Plan de montage de la carte ET1017M

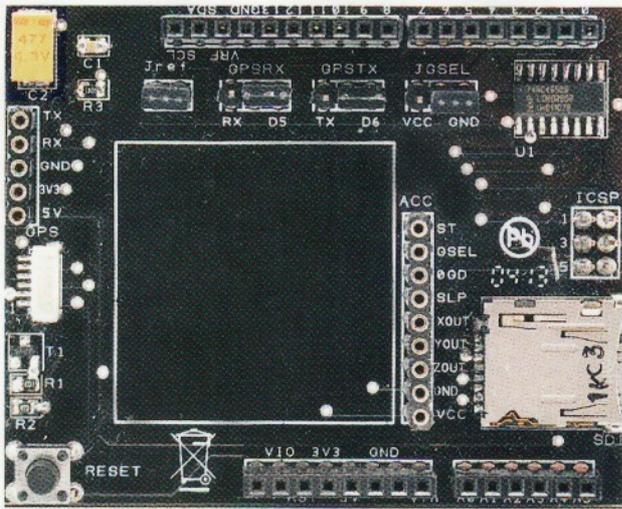


Figure 9 : photo de l'un de nos prototypes avec tous les connecteurs soudés mais sans le module GPS installé.

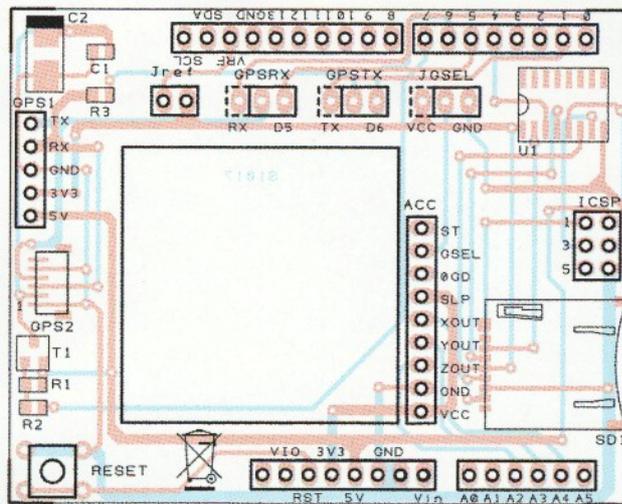


Figure 10 : plan de câblage de la carte ET1017M.

Liste des composants de la carte ET1017M

- R1..... 10 kΩ boîtier 0805
- R2..... 4,7 kΩ boîtier 0805
- R3..... 2,2 kΩ boîtier 0805
- C1..... 100 nF multicouche boîtier 0805
- C2..... 4,7 µF tantale boîtier D
- T1 BC807
- U1..... 74HC4050D
- SD1 ... connecteur microSD
- RESET microswitch
- Connecteur BMO6B-SRSS-TB (www.jst.fr)
- Barrette mâle/femelle 3 pôles (x2)
- Barrette mâle/femelle 6 pôles
- Barrette mâle/femelle 8 pôles (x2)
- Barrette mâle/femelle 10 pôles
- Barrette mâle 2 pôles
- Barrette mâle 3 pôles (x3)
- Barrette femelle 5 pôles
- Barrette femelle 9 pôles
- Cavaliers (x4)

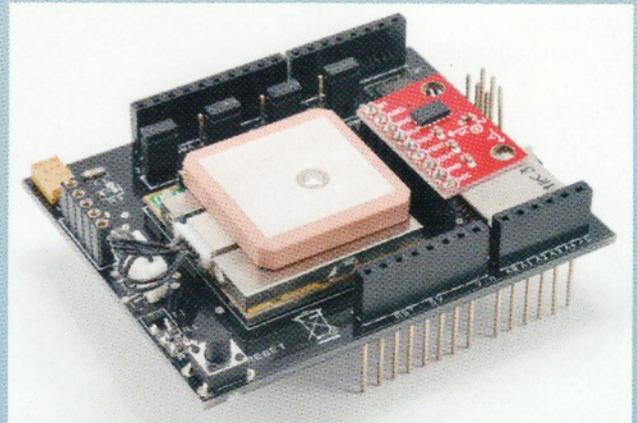


Figure 11 : photo d'un prototype complet avec le module GPS installé.

Ce programme est gratuit et vous permettra de vous initier à la technologie de géolocalisation. Il permet la conversion de la plupart des formats utilisés par les systèmes GPS. Il fonctionne sous Windows et MAC OS X et Linux. Vous pouvez le télécharger à l'adresse suivante : <https://www.gpsbabel.org/download.html>.

Son utilisation est très intuitive, il suffit de sélectionner le format d'entrée (dans notre cas c'est le format NMEA), le nom du fichier à lire (dans notre cas il s'agit du fichier data.log présent sur la SD), le format à générer (le format GPX pour la plupart des programmes ou KML pour Google Earth) et enfin le nom du fichier de sortie. GPSBabel offre plusieurs options pour compléter la conversion, mais les paramètres par défaut sont généralement adaptés à la plupart des utilisations.

Un autre programme intéressant est **GpsPrune**, ce programme est gratuit et multi-plate-forme (Java). Il permet de convertir et manipuler des données GPS, et prend directement en charge le format NMEA.

Vous pouvez le télécharger à l'adresse : <https://activityworkshop.net/software/gpsprune/download.html>.

Pour l'utiliser sous Windows, il vous faudra télécharger le programme Java à l'adresse : <https://www.java.com/fr/>.

Vous trouverez un **mini tutoriel de GpsPrune en français** à l'adresse : https://activityworkshop.net/software/gpsprune/how-tos_fr.html.

Listing 2

```
#include <SD.h>
#include <SoftwareSerial.h>

/* ***** Configuration ***** */
char log_filename[13] = «data.log»; // garder cette FAT

// Combien de temps attendre avant d'arrêter l'enregistrement lorsqu'il n'y a pas de mouvement
#define STOP 60000

/* ***** Fin de la configuration ***** */

#define PIN_GPS_ENABLE 7
SoftwareSerial gpsSerial(5,6);
#define PIN_SD_SS 10
File log_file;
boolean sd_available = false;
boolean moving = true;
unsigned long last_move = 0;
#define ZERO_X 512
#define ZERO_Y 512
#define ZERO_Z 512
#define M_THRESH 60000
#define PIN_X 4
#define PIN_Y 3
#define PIN_Z 2
void setup() {
    Serial.begin(9600);

    // Configuration du GPS
    gpsSerial.begin(4800);
    pinMode(PIN_GPS_ENABLE,OUTPUT);
    digitalWrite(PIN_GPS_ENABLE,HIGH);

    // Configuration de la SD
    pinMode(PIN_SD_SS,OUTPUT);
    pinMode(10,OUTPUT);
    start_sd();
    start_gps();
}
void loop() {
    if(moving) {
        if(sd_available) {
            log_file = SD.open(log_filename,FILE_WRITE);
            if(!log_file) {
                sd_available = false;
            }
        }
        if(gpsSerial.overflow()) {
            if(sd_available) {
                log_file.write(«\r\n$PXXXX,SoftwareSerial overflow!!!»);
            } else {
                Serial.println(«\r\n$PXXXX,SoftwareSerial overflow!!!»);
            }
        }
        while(gpsSerial.available()) {
            int r = gpsSerial.read();
            if(sd_available) {
                log_file.write(r);
            } else {
                Serial.write(r);
            }
        }
    }
}
```

```

    check_mouvement();
    log_file.close();
} else {
    delay(1000);
    check_mouvement();
}
}
void start_gps() {
    gpsSerial.listen();
    digitalWrite(PIN_GPS_ENABLE,LOW);
}
void stop_gps() {
    digitalWrite(PIN_GPS_ENABLE,HIGH);
}
void start_sd() {
    if (!SD.begin(PIN_SD_SS)) {
        Serial.println("SD endommagée ou non présente");
    } else {
        log_file = SD.open(log_filename,FILE_WRITE);
        if (log_file) {
            log_file.close();
            sd_available = true;
        } else {
            Serial.print("Can't open file <»);
            Serial.print(log_filename);
            Serial.println("«> for writing.»);
        }
    }
}
void check_mouvement() {
    int x = analogRead(PIN_X) - ZERO_X;

```

```

int y = analogRead(PIN_Y) - ZERO_Y;
int z = analogRead(PIN_Z) - ZERO_Z;
long acc = (long)x*x + (long)y*y + (long)z*z;
if (acc > M_THRESH) {
    last_move = millis();
    if (!moving) {
        start_gps();
        start_sd();
        // Ignore les données en cas de dépassement
        gpsSerial.overflow();
    }
    moving = true;
} else {
    if (moving) {
        unsigned long now = millis();
        // vérifie le débordement en millisecondes
        if (now-last_move > STOP) {
            moving = false;
            stop_gps();
            if(sd_available) {
                log_file.write("\r\n$PXXXX,Périphérique inactif,
stop logging\r\n");
            } else {
                Serial.println("\r\n$PXXXX, Périphérique inactif,
stop logging\r\n");
            }
        }
    }
}
}
}
}

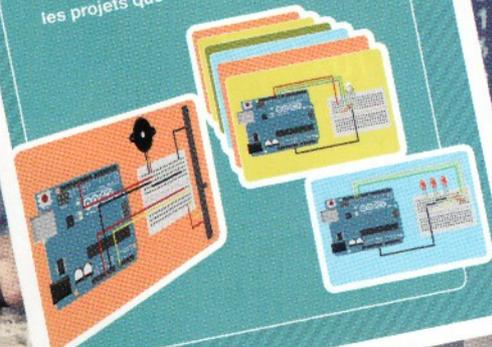
```

Découvrez le monde d'ARDUINO et apprenez à programmer avec le livre l'ABC d'ARDUINO.

Un livre pensé pour les débutants qui abordent pour la première fois l'électronique et pour les hobbyistes qui désirent apprendre la programmation sur une plate-forme Arduino. Ce livre comprend de nombreux tutoriels didactiques conçus pour mettre en pratique les connaissances acquises.

l'ABC d'ARDUINO

Découvrez le monde d'Arduino et comment utiliser la carte de prototypage la plus populaire pour mener à bien les projets que vous souhaitez réaliser.



15,84 €

frais de port inclus pour la France Métropolitaine

Achetez en ligne : www.electroniquemagazine.com