

ParkAid

Je suis bien *ParkAid*, là ?

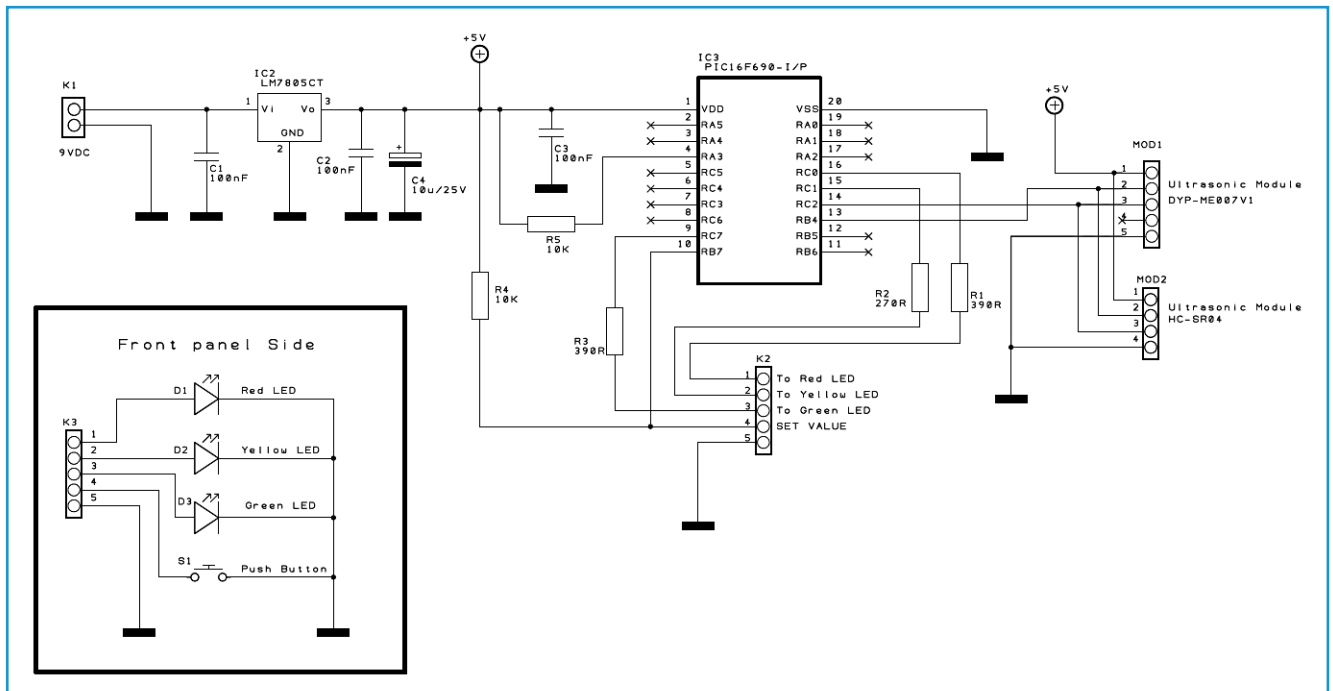
Terry Hinrich
(États-Unis)



Garer sa voiture est de plus en plus délicat à mesure que les places deviennent rares et exiguës. Il y a bien le truc de la balle de tennis suspendue ou de la planche au sol, mais ces repères sont incertains. Voici pour vous assister, plus économique qu'un voiturier puisqu'il ne coûte que 40 € (boîtier et alimentation compris), et moins sujet aux sarcasmes que votre conjoint puisqu'il ne s'exprime que par ultrasons, notre ParkAid. Vous rentrerez votre voiture au garage le coude à la portière !

ParkAid le well nommé

ParkAid est construit autour d'une petite carte qui émet et détecte un signal à ultrasons de 40 kHz. Cette fréquence est de loin supérieure à la limite basse (20 kHz) des fréquences audibles par l'homme (jeune) et l'expérience a prouvé qu'elle était également trop élevée pour être perçue par les chiens. L'émetteur envoie une courte séquence d'impulsions, et le récepteur détecte les impulsions (échos) réfléchies par tout objet – la voiture – situé devant le module. Ce module effectue la majorité du traitement du signal et n'a besoin que d'une impulsion de déclenchement pour que soit lancée une mesure. L'écho est délivré par sa broche de sortie. La distance est calculée à partir de la durée mesurée entre les impulsions émises et leurs réflexions. Si aucun objet n'est présent devant le capteur, une impulsion correspondant à une distance maximale est fournie. C'est un petit microcontrôleur qui



initie l'émission et mesure la durée entre les échos.

La résolution du système est de 2,54 cm. Une impulsion qui se déplace à la vitesse du son met 148 µs pour atteindre un objet situé à 2,54 cm et en revenir. Une fois l'émission initiée, le programme compte donc les intervalles de 148 µs pour déterminer la distance de l'objet réflecteur. Cette méthode donne une bonne précision. Notez que le signal émis a une forme de cône dont la base s'élargit à mesure qu'il s'éloigne de l'émetteur. Il n'est donc pas impossible qu'un objet « indésirable » réfléchisse le signal et limite ainsi la portée maximale à 3 m. En pratique, une voiture est toutefois toujours détectée à une distance suffisante pour que ParkAid puisse tenir son rôle d'assistant. L'auteur, sa femme et leur voiture peuvent en témoigner.

Le circuit (**fig. 1**) peut être téléchargé sous forme de projet *DesignSpark* depuis la page Elektor.LABS [1]. Avant d'expliquer son fonctionnement, passons en revue les « feux de signalisation » (les LED), le microcontrôleur et le logiciel de ParkAid.

Vert, jaune, rouge : un code universel

La distance relative est indiquée au moyen de trois LED. La verte indique que la voiture

est trop loin du module, la jaune que la position d'arrêt est atteinte, et la rouge que la voiture est trop près du module et doit être légèrement reculée.

Ce sont en fait sept distances relatives qui guident la manœuvre de stationnement :

- Le feu vert s'allume lorsque vous entrez dans le garage, et reste fixe pour indiquer que vous pouvez continuer d'avancer.
- Le feu vert clignote lorsque vous êtes à 30,5 cm de la cible.
- Des feux vert et jaune fixes indiquent que vous êtes à 2,54 cm de la position finale. Préparez-vous à stopper.
- Un feu jaune fixe signifie que la position finale est atteinte. Il s'agit d'une zone de 2,54 cm, donc facile à atteindre.
- Les feux jaune et rouge s'allument lorsque vous vous éloignez de la position finale. Reculez légèrement.
- Un feu rouge fixe indique que vous êtes sorti de la zone finale et que vous devez reculer.
- Le feu rouge clignote lorsque vous êtes beaucoup trop près du système. Reculez vite sous peine de l'écraser !

Cette signalisation permet de toujours se garer dans la même position. Son codage

Figure 1. Schéma de ParkAid.

en C n'a pas posé de difficultés, le faire de façon purement matérielle aurait par contre été fastidieux. Modifier les distances relatives stockées dans l'EEPROM ne prendra qu'un instant.

Après la mise sous tension les LED s'allument brièvement pour montrer qu'elles fonctionnent. L'unité peut être débranchée et rebranchée n'importe quand, le système est opérationnel sur-le-champ.

La **figure 2** montre le modèle final de l'auteur avec les trois LED sortant du boîtier. Les deux choses rondes sous les LED sont l'émetteur et le récepteur à ultrasons. Le bouton-poussoir est à gauche, la connexion pour l'adaptateur mural à droite.

PIC au boulot

Le PIC16F690 [2], une puce à 20 pattes, possède plus de fonctions que nécessaire pour cette application. Je ne l'ai pas retenu que parce que je l'avais reçu en cadeau pour l'achat d'un programmeur PICKit 2, mais aussi parce qu'il est rapide, possède de nombreuses broches d'E/S, assez de mémoire, une EEPROM intégrée, est peu onéreux (environ 1,50 €) et facile à se procurer. Ceux qui préfèrent d'autres PIC ou d'autres microcontrôleurs n'auront aucun mal à adapter le programme en conséquence.

Le fonctionnement du PIC est cadencé par une horloge interne de 8 MHz. Une routine d'interruption fait clignoter les LED tous les quarts de seconde. Les autres traitements temporels sont gérés à l'aide de boucles d'attente, et l'anti-rebond du poussoir est lui aussi effectué de façon logicielle.

Un compilateur qui tombe à PIC

J'ai utilisé la version *Lite* du compilateur *HiTech C* pour écrire le code. Cette version gratuite n'a pas les options d'optimisation de la version professionnelle, mais le compromis est équitable pour un amateur au budget modeste. Le code source [1] est simple et peut être transposé dans un autre langage. Si vous le modifiez, soyez assez aimable pour citer l'original. Si vous n'avez pas le compilateur mais disposez d'un programmeur, le fichier HEX vous attend également sur la page [1].

Le code est formaté d'une manière quelque



Figure 2.
Le module de Terry dans son boîtier.

peu inhabituelle. La fonction *main* est placée à la fin du fichier et appelle les fonctions situées au-dessus d'elle, une façon d'éviter le prototypage des fonctions. Ces fonctions ont également été regroupées par fonction (rôle) à l'aide d'un préfixe alphanumérique. Classées par ordre alphabétique, elles sont dès lors faciles à trouver. L'intérêt du procédé est moins flagrant sur un petit programme que sur un gros, mais la méthode reste utile.

Module à ultrasons : les yeux de la carte

Pour le prototype, j'ai choisi le module à ultrasons TS601P01. Son interface a 3 broches : alimentation de 5 V, masse, et broche de données bidirectionnelle. Lorsqu'elle s'est attaquée à la fabrication de son propre prototype, l'équipe d'Elektor.LABS a toutefois remarqué que le TS601P01 n'était pas courant, et que deux autres modules étaient plus faciles à trouver : le DYP-ME007 et le HC-SR04. Tous deux utilisent une interface à 4 broches, même si le premier en a 5. Le connecteur MOD1 de la **figure 1** est pour le DYP-ME007, le MOD2 pour le HC-SR04. Pour plus de détails sur le fonctionnement de ces deux modules, référez-vous à leurs fiches techniques [1].

Le programme a été modifié pour piloter un module à ultrasons à 4 broches.

Alimentation

Un régulateur de tension 7805 sans dissipateur fournit l'unique tension de 5 V dont a besoin l'unité. Avec une LED allumée, le courant mesuré était d'environ 15 mA, et d'envi-

ron 28 mA avec les trois. Ces intensités sont gérables par un 78L05 si vous préférez ce régulateur. Puisque la consommation est faible et qu'il est souhaitable que l'unité fonctionne en permanence, aucun interrupteur d'alimentation n'est présent. Nous avons opté pour un adaptateur secteur de 9 V, mais n'importe quel transformateur délivrant entre 9 et 15 V devrait convenir.

Carte(s)

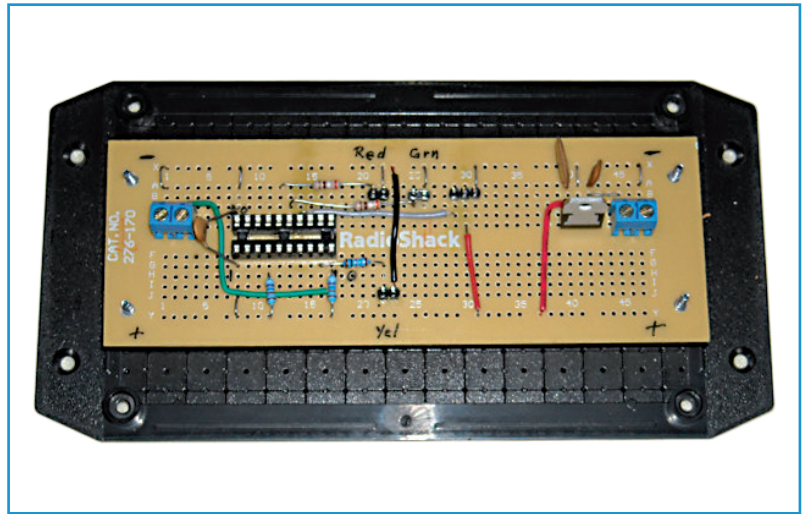
Pour l'assemblage du prototype, j'ai utilisé le modèle 276-170 de Radio Shack, une carte simple face de 2 x 15 cm, pré-percée et qui se prête bien à l'implantation de CI et au câblage manuel. La **figure 3** montre la configuration finale avant la mise en boîtier. La carte est fixée sur le fond amovible avec des espaceurs de 15 mm, tandis que les LED sont sur le dessus.

Elektor a de son côté conçu un système à deux cartes pour faciliter le montage : une carte mère pour loger le module à ultrasons, le microcontrôleur, le régulateur de tension et la plupart des composants, et une carte fille sur laquelle sont placées les LED et le poussoir. Schéma et fichiers *DesignSpark* peuvent être téléchargés [1].

Régler la position finale

Les valeurs par défaut qui déterminent la position finale et les distances relatives ne seront probablement pas adaptées à votre situation. Voyons comment les modifier.

Appuyer sur S1 place l'unité en mode programmation. Le clignotement des LED rouge et jaune indique que le paramétrage de la position finale est en cours. Avant cela, vous devriez placer votre véhicule à l'endroit souhaité. Appuyez deux fois sur S1 en moins d'une seconde pour sauvegarder la position finale « près » (LED jaune). Les distances qui correspondent aux LED verte et rouge sont alors ajustées automatiquement. Aucune autre modification n'est nécessaire. Les LED rouge et jaune clignotent trois fois pour indiquer que la nouvelle position finale a été enregistrée. Vous pouvez ensuite régler la position « loin ». Si vous ne souhaitez pas modifier la position finale, appuyez une fois sur le bouton pour procéder au réglage de la valeur « loin ». Le point « loin » correspond à la distance pour laquelle la LED verte passe de fixe à clignotante. Régler la position finale change



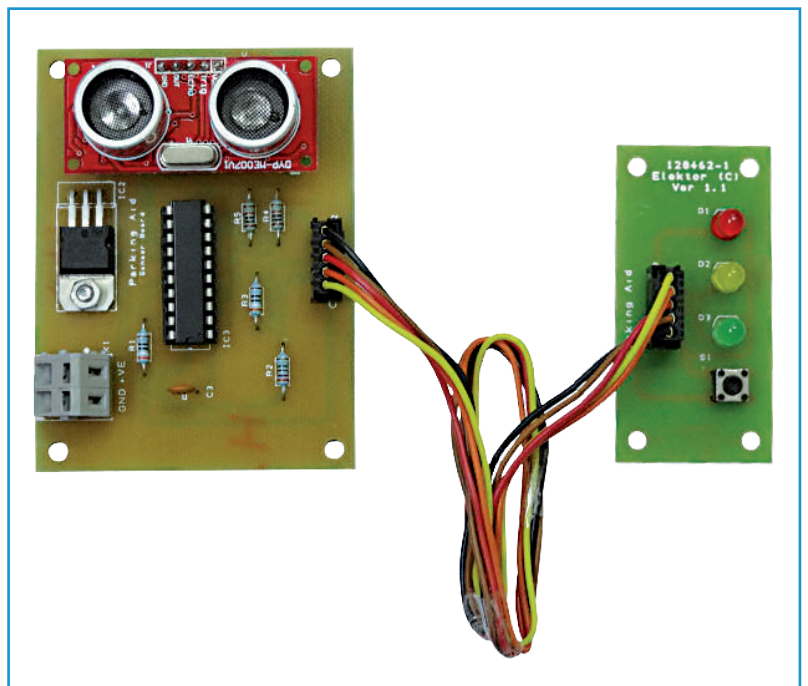
automatiquement la position « loin », située par défaut à 10 cm de la position finale. Pour modifier ce point, éloignez votre véhicule du module et appuyez deux fois sur le bouton en moins d'une seconde ; la nouvelle distance sera alors sauvegardée.

Figure 3. Le prototype de Terry avant fermeture du boîtier. La conception d'Elektor est légèrement différente.

Tout nouveau point réglé est immédiatement sauvegardé. La distance est également recalculée, donc ne vous placez pas devant l'unité à ce moment-là. Une fois ces deux étapes de programmation effectuées, l'unité repasse en mode d'utilisation normale.

Les distances finales « près » et « loin » sont

Figure 4. L'assemblage des cartes façon Elektor. Séparer le circuit permet de placer les LED témoin à un endroit visible, éloigné du module à ultrasons.



LISTE DES COMPOSANTS

Résistances

R1, R3 = 390 Ω, 0,25 W, 5%
 R2 = 270 Ω, 0,25 W, 5%
 R4, R5 = 10 kΩ, 0,25 W, 5%

Condensateurs

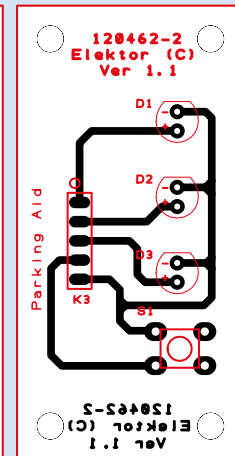
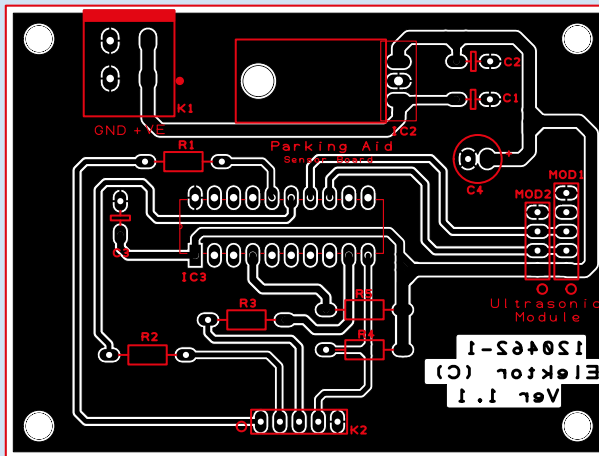
C1, C2, C3 = 100 nF, céramique
 C4 = 10 µF, 25 V, électrolytique, radial

Semi-conducteurs

D1 = LED, 5 mm, rouge
 D2 = LED, 5 mm, jaune
 D3 = LED, 5 mm, verte

Divers

IC2 = 7805 (LM7805CT, Fairchild Semiconductor)
 IC3 = PIC16F690 (Microchip Technology)
 K1 = borne de raccordement à 2 contacts, 5 mm (MC000046, Multicomp)
 K2, K3 = embase à 5 broches, 2,54 mm
 S1 = bouton à effleurement, 1,2 cm x 1,2 cm (Multicomp MCDTS2-1N)
 MOD1 = connecteur SIL à 4 broches, 2,54 mm
 MOD2 = connecteur SIL à 5 broches, 2,54 mm
 Module à ultrasons sur carte, DYP-ME007 ou HC-SR04 (voir texte et [1])



stockées dans l'EEPROM et ne seront donc pas perdues si l'unité est mise hors tension.

Et à part ça, il fait quoi ?

Ce qu'il y a de bien avec les modules à ultrasons, c'est qu'avec un peu d'imagination on leur trouve de nouvelles applications. Faisons-nous part de vos idées !

La consommation de ParkAid est très faible, mais il serait intéressant de le modifier pour l'alimenter par piles. Comme il est censé fonctionner en permanence, ParkAid devrait donc rester en veille la plupart du temps et être activé périodiquement pour voir si la distance a changé.

(120462 - version française : Hervé Moreau)

Liens

- [1] www.elektor-labs.com/120462
- [2] www.microchip.com/wwwproducts/Devices.aspx?dDocName=en023112
- [3] www.htsoft.com

L'auteur

En retraite depuis peu, Terry Hinrich a pour loisir l'électronique. Diplômé en mathématiques, il a passé ses presque 50 ans de carrière dans le monde de l'informatique. Il a écrit des programmes dans de nombreux langages, à la fois pour les grands systèmes et les ordinateurs de moyenne puissance. Il utilise Delphi chez lui. ParkAid aura été sa première incursion dans le monde des très petits processeurs. Ce projet lui a été très utile, et surtout lui aura permis d'apprendre beaucoup de choses !