

Carte d'acquisition de don

José Luis Rupérez Fombellida

Cette carte d'acquisition de données se connectant au bus USB possède 8 sorties numériques, autant d'entrées numériques, 2 sorties analogiques 10 bits et 8 entrées analogique pour des excursions de tension allant de 0 à 5 volts. Le coeur (et cerveau) du système est un microcontrôleur USB de Microchip, un PIC18F4550 programmé en C. L'électronique prend place sur un circuit compact et ne requiert pas d'alimentation propre.

Tout au long de son histoire Elektor a décrit nombre de cartes et systèmes de mesure se connectant au PC et ayant connu un succès très honorable. Qu'il ait été question d'un montage autonome pour mesure par le biais d'un port RS-232 ou LPT (savez-vous encore de quoi il est question ?), d'une carte enfichable dans le bus ISA (idem) ou maintenant via le port USB, peu importe, dès l'instant où il permet à nos lecteurs (1) de générer et de lire des signaux numériques et (2) de faire la même chose pour les signaux analogiques, il a toujours, par le passé, fait un tabac. La carte décrite dans le présent article pourrait être au berceau de nombreuses applications ayant trait aux mesures et au contrôle.

We want USB

Il est indéniable que RS-232, ISA et même Centronics sont des notions du passé lorsque l'on a affaire à des signaux numériques et analogiques surtout dans le cas de mesure et de pilotage par le biais d'un PC. L'avenir, tant au niveau de la vitesse que de la simplicité de connexion s'appelle indiscutablement USB. L'aspect logiciel

de cette connexion est cependant sensiblement plus complexe qu'il n'y paraît à première vue. Il faut un certain temps (de réflexion aussi !) avant qu'un PC ne reconnaisse un périphérique USB valide !

Nous espérons, par le biais de cet article, offrir de quoi satisfaire tant les lecteurs que seul intéresse la connexion numérique et analogique au monde réel que ceux qui aimeraient en savoir plus sur les tenants et aboutissants de l'USB dans un microcontrôleur et, très important également, comment lui faire faire quelque chose d'utile - au moindre coût bien évidemment.

PIC18F4550 pour USB

Il existe, heureusement, des microcontrôleurs qui rendent l'interfaçage USB entre le PC (hôte) et l'électronique-cible (notre périphérique) plus ou moins transparente. Ceci tient au fait qu'ils sont dotés d'un matériel (*hardware*) et de progiciel permettant d'implémenter l'USB le plus aisément possible. Tout cela reste caché pour ceux qui ne souhaitent qu'utiliser le périphérique USB sans avoir à s'initier au fonctionnement interne (ce qui n'est sans doute pas votre

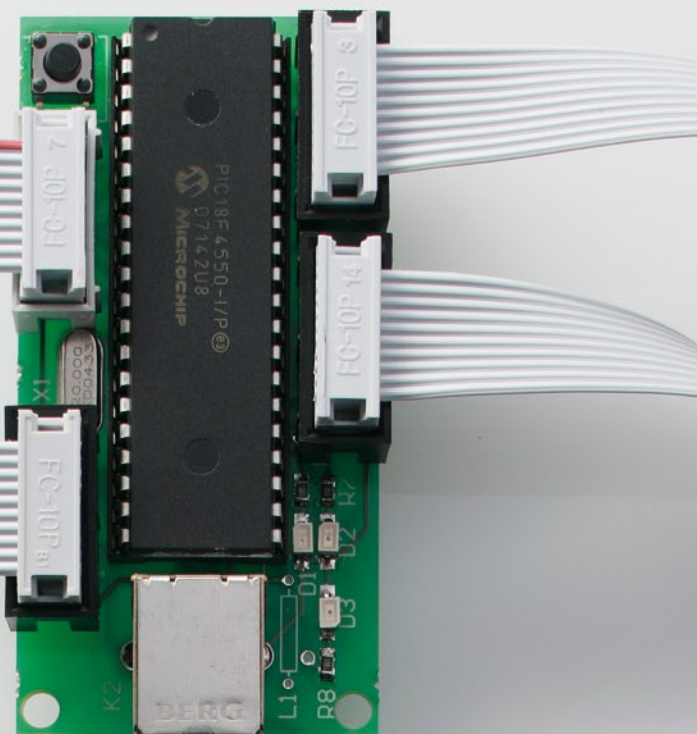
cas !). L'un des processeurs de cette classe, le PIC18F4550 de Microchip, présente en outre l'avantage d'être au coeur d'une nébuleuse de programmes (gratuits). Autre aspect à souligner, ce composant est proposé en boîtier DIP40, ce qui ne manquera pas de recueillir les applaudissements du camp des anti-CMS.

L'électronique

La **figure 1** vous propose le schéma de cette petite merveille de technologie. À y regarder de près il ne s'agit en fait guère plus que d'une CPU puissante, IC1, entourée d'une cour d'embases d'entrée et de sortie, et « enluminée » par quelques LED d'état. Passons en revue la fonction des différentes embases (les lignes de port concernées étant prises entre parenthèses) :

K1 = sortie numérique 8 bits de 0 à 5 V, excursion TTL (RD0 à RD7).

K2 = embase USB pour la connexion



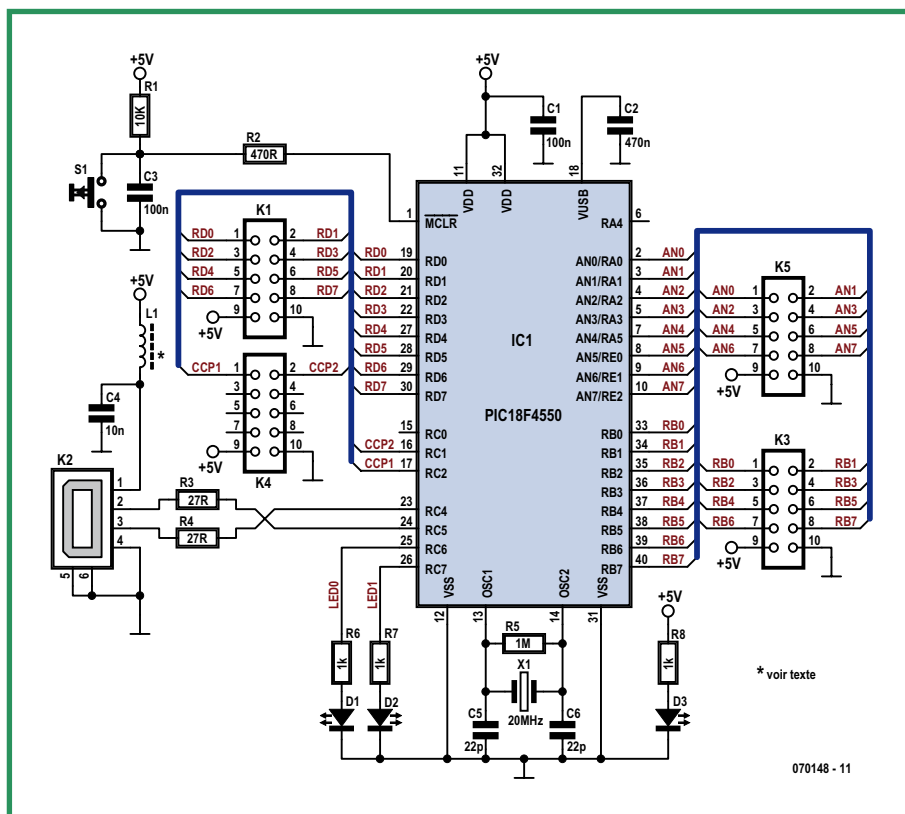


Figure 1. Schéma de la carte d'acquisition de données à connectivité USB.

au PC (RC4 et RC5).

K3 = entrée numérique 8 bits de 0 à 5 V, excursion TTL (RB0 à RB7).

K4 = deux sorties analogiques pour excursion de 0 à 5 V (RC1 et RC2).

K5 = 8 entrées analogiques (AN0/RA0 à AN7/RE2) pour excursion de 0 à 5 V.

Les lignes de port RB (entrées numériques) sont dotées de résistances de forçage au niveau haut (*pull up*). Les sorties analogiques ont une résolution de 10 bits, travaillant en MLI (Modulation en largeur d'impulsion = PWM pour **P**ulse **W**idth **M**odulation) à 2,9 kHz. Le cas échéant, il est possible de filtrer ces sorties à l'aide d'un simple réseau RC.

La tension de sortie continue (CC) V_O obtenue après filtrage répond à la formule suivante :

$$V_O = 5D \text{ [volts]}$$

formule dans laquelle D est le rapport cyclique de la MLI, facteur pouvant prendre une valeur comprise entre 0 et 1.

Les entrées analogiques ont elles aussi une résolution de 10 bits.

L'oscillateur du PIC bat à 20 MHz, fréquence définie par le quartz X1 épaulé par la paire de condensateurs classique assurant une charge en mode parallèle et la résistance de forte valeur, R5, pour la contre-réaction.

En réalité, le microcontrôleur travaille à 48 MHz, fréquence générée en interne par le biais d'une boucle à verrouillage de phase (PLL = **P**hase **L**ocked **L**oop) et d'un diviseur de fréquence interne à partir des 20 MHz fournis par le quartz. Ces 48 MHz sont un multiple rond de la vitesse de bus USB (*Full Speed*, 12 Mbits/s).

Les 2 LED d'état, D1 et D2 visualisent l'état (*status*) de l'USB. D3, à l'évidence, signale la présence de la tension d'alimentation lorsque la carte est connectée au port USB de votre PC.

L'alimentation du montage se fait par le biais de l'embase USB K2, et une petite self de choc L1 épaulée par C4 et destinée à éliminer le bruit.

Il ne nous reste plus à évoquer que S1, R1, R2 et C3 pris à l'entrée MCLR du micro. Il ne s'agit en fait de rien de plus que d'un réseau de réinitialisation (*reset*) classique.

Progiciel du PIC

Qui dit matériel simple sous-entend bien souvent quantité massive de logiciel en contrepartie. Le progiciel (code objet) présent dans le PIC fait exactement ce que souhaite son auteur qui l'a produit en s'aidant d'une paire d'outils gratuits de Microchip, l'environnement (IDE) MPLAB V7.5 et le compilateur C18 Student Edition V3.02. On trouvera sur le site les informations nécessaires et suffisantes d'installation et d'utilisation de ces deux logiciels. Le code-source du progiciel (*firmware*) diffère de l'original de Microchip. L'ensemble du programme de ce projet est à votre disposition sous forme de téléchargement gratuit (**070148-11.zip**) depuis notre site (www.elektor.fr). Vous découvrirez pas moins de 3 dossiers dans le fichier d'archive : « driver », « firmware » et « PC ». Leur contenu est une caverne d'Ali Baba pour les inconditionnels du C, des PIC et de l'USB (un pourcentage non négligeable de nos lecteurs). Nous reproduisons, en **figure 2**, un morceau de code C. Il s'agit du paramétrage des fusibles qu'il vous faut utiliser au cas où vous n'achèteriez pas le contrôleur programmé auprès des sources

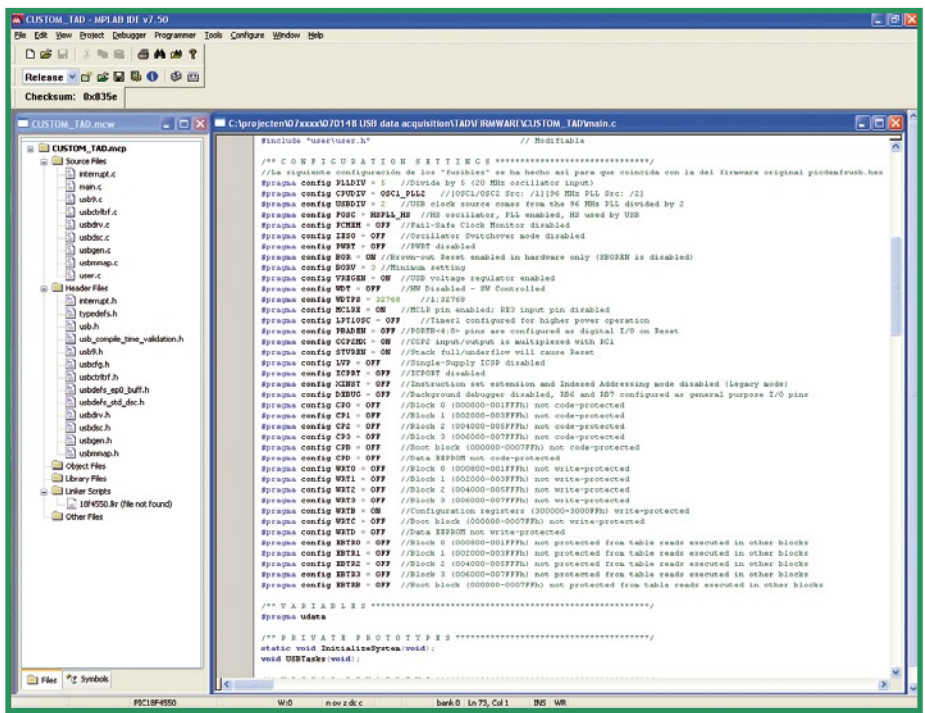


Figure 2. Travaillez vous aussi, comme nombre de lecteurs d'Elektor, sous MPLAB ? Le code C du projet comporte des informations très utiles quant au paramétrage des fusibles - aspect qui reste une véritable source de confusion pour nombre d'inconditionnels des micros (et pas uniquement ceux des PIC).

habituelles. Le dossier « firmware » comprend l'ensemble du projet et le résultat de sa compilation baptisé TAD v1.hex. C'est le fichier à « griller » dans le contrôleur.

Si vous souhaitez en savoir plus quant au fonctionnement profond de l'USB, sachez que la connectivité implémentée sur la carte décrite ici est déterminée par le progiciel propre du PIC18F4550. Il est fait

appel aux blocs suivants : BUS POWER MODE; CUSTOM CLASS; FULL-SPEED (12 MBIT/s) et INTERRUPT TRANSFER.

Réalisation

L'électronique prend place sur une platine double face à trous métallisés compacte dont la figure 3 donne la sérigraphie de l'implantation des composants recto-verso. L'espace laissé aux extrémités de la carte peut servir à sa fixation.

Si la mise en place des composants n'appelle pas de remarque particulière il nous faut faire quelques remarques à l'intention de ceux d'entre nos lecteurs qui se lancent à la découpe de l'USB par le biais de cette carte.

La découpe dans chacune des embases sert de repère d'orientation et devrait être tournée vers le bord de la platine de manière à permettre l'enfichage d'un câble en nappe doté du connecteur adéquat.

Les composants CMS prennent place sur les deux côtés de la platine. Il faudra par conséquent bien identifier la position de chaque composant et s'assurer de l'orientation dans le cas des LED CMS.

Nous recommandons d'enficher le PIC18F4550 dans un support DIL à 40 broches de bonne qualité. Attention à l'orientation de ce gros composant : la broche 1 se trouve à proximité du bouton de RAZ S1.

L1, pour finir, est une perle de ferrite à 2 ou 3 trous traversée par un morceau de fil de cuivre émaillé ou autre conducteur semi-rigide. Une perle de ferrite à un orifice dotée de 3 spires devrait faire l'affaire elle aussi. L'inductance de la self de choc HF ainsi réalisée n'a rien de critique.

Vérifiez, avant connexion au PC et pour éviter tout dommage à ce dernier, l'absence de court-circuit ou d'autre problème au niveau des broches du connecteur USB K2.

Première connexion

Après avoir placé tous les composants sur la carte (et être certain que le contrôleur est bien programmé), connectez-là à un PC par le biais d'un câble USB standard. La LED d'alimentation D3 doit s'allumer et l'une des LED D1 ou D2 clignote alors que l'autre reste éteinte. Parallèlement, le PC vous signalera la connexion d'un nouveau périphérique USB et vous demandera son pilote. Informez Windows de l'emplacement de stockage du pilote (mchpush.inf dans dossier driver\). Une fois le pilote installé les LED d'état clignotent alternativement. La carte est prête à être utilisée.

Set de test pour la carte d'acquisition de données

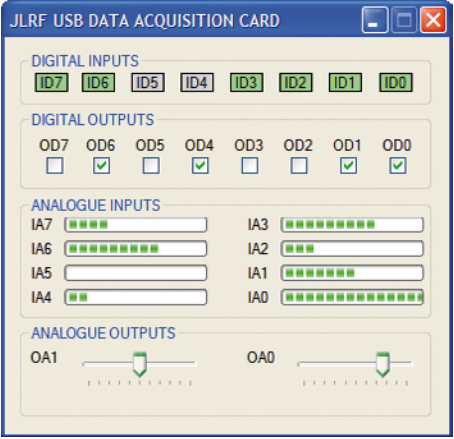
L'auteur a développé 4 cartes-filles simples permettant de tester les fonctionnalités suivantes de la carte d'acquisition de données USB :

- 1. Carte à LED : 8 LED pour visualiser les lignes de sortie en numérique.
2. Carte à boutons et interrupteurs : 4 boutons-poussoirs et 4 interrupteurs pour tester les 8 entrées numériques.
3. Carte voltmètre à LED : 2 LED voient leur intensité varier en fonction des 2 sorties analogiques.
4. Carte à potentiomètres : 8 potentiomètres pour tester le fonctionnement correct des 8 entrées analogiques.

La photo montre la carte d'acquisition de données et les 4 cartes-filles. Bien que l'on voie des circuits imprimés et que l'auteur ait les fichiers de schéma et de platine au format OrCAD, il ne devrait pas être trop difficile de réaliser ces cartes sur des morceaux de platine d'expérimentation à pastilles (Veroboard).

Pour tester l'ensemble, il a été développé un programme en C++ CLR compilé ensuite à l'aide du compilateur Visual C++ 2005 Express (gratuit). Ce programme repose sur des exemples de Microchip.

Nous vous présentons une recopie d'écran du programme. Ce logiciel est inclus dans le fichier d'archive du projet.



VID/PID (Vendor ID/Product ID)

Tous les périphériques USB sont pourvus d'une combinaison unique prenant la forme de 2 nombres de sorte qu'il n'existe pas deux périphériques identiques. Le premier nombre, VID, identifie le fabricant du produit, le second représentant l'identificateur du produit.

La combinaison utilisée dans ce projet utilise le VID de Microchip et comme PID celui d'une carte de démonstration du PIC18F4550 de Microchip. Si vous deviez utiliser la carte d'acquisition USB à des fins commerciales il est essentiel d'obtenir une combinaison VID/PID différente, ce qu'il est possible de faire par, par www.usb.org voire par le biais de Microchip. Cette nouvelle combinaison devra être intégrée dans le code-source du progiciel qu'il faudra recompiler en vue de créer le nouveau fichier .hex qui servira à reprogrammer le microcontrôleur.

Il faudra modifier en conséquence le logiciel tournant sur le PC sachant que progiciel et logiciel doivent avoir la même combinaison VID/PID. Il restera, pour finir, à modifier le fichier du pilote « mchpusb.inf ».

Précautions

Il nous faut faire quelques remarques d'ordre général. Toutes les embases d'extension, K1, K3, K4 et K5 véhiculent tant le +5 V que la masse de manière à alimen-

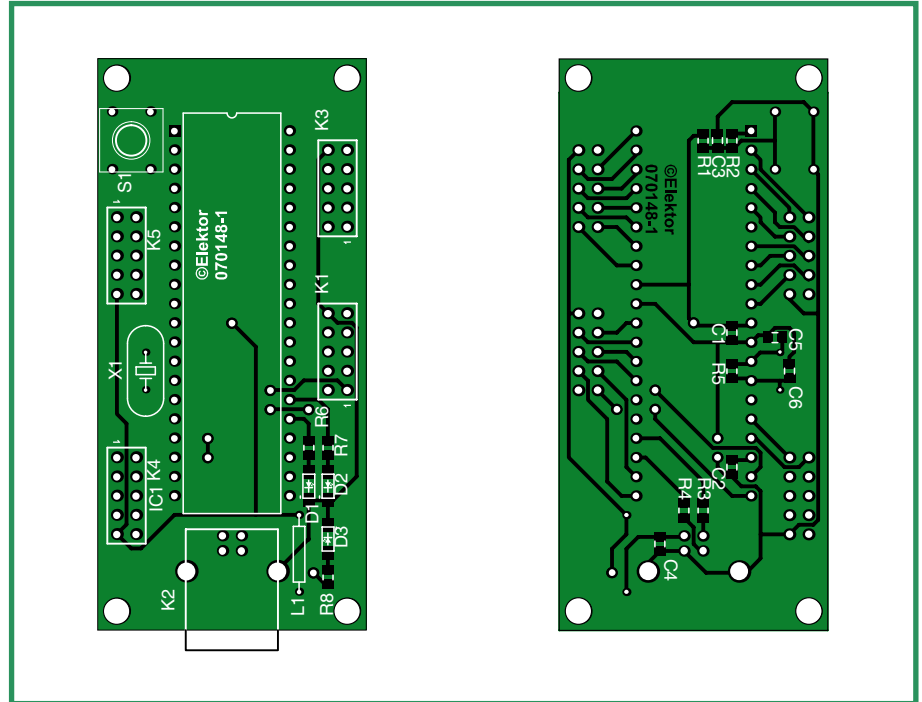


Figure 3. Représentation recto-verso de la sérigraphie de l'implantation des composants de la carte. Ce fichier est téléchargeable gratuitement au format .pdf depuis notre site. Ne pas oublier qu'il s'agit d'un double face à trous métallisés.

ter une carte qui y serait connectée. Il est important de veiller à ne pas provoquer de court-circuit à ces borniers et à ne pas drainer plus de 100 mA de l'un d'entre eux. Il ne faut pas oublier non plus que ces borniers sont reliés directement aux

+5 V et masse du port USB de votre PC (que vous ne souhaitez pas endommager - on évitera donc, en tous cas, d'utiliser l'ordinateur familial ou celui des enfants). Si une application donnée requiert un courant plus important, il faudra envisager

Liste des composants

Résistances :

- (tous CMS de forme 0805)
- R1 = 10 k Ω
- R2 = 470 Ω
- R3,R4 = 33 Ω
- R5 = 1 M Ω
- R6 à R8 = 1 k Ω

Condensateurs :

- (tous CMS de forme 0805)
- C1,C3 = 100 nF
- C2 = 470 nF
- C4 = 10 nF
- C5,C6 = 22 pF

Semi-conducteurs :

- IC1 = PIC18F4550 I/P (programmé **EPS070148-41**)
- D1,D2,D3 = LED CMS forme 1206

Divers :

- K1,K3 à K5 = embase à 2 rangées de 5 contacts
- K2 = embase USB type-B encartable
- X1 = quartz 20 MHz
- L1 = VK200 ou petite perle ferrite avec 2 à 4 spires de fil de cuivre émaillé de faible diamètre



Fichiers et leurs dossiers

L'ensemble du projet de l'application se trouve dans le dossier PC\TAD_V1_win, sous le nom de TAD_V1_win.vcproj. Le programme compilé se trouve dans le dossier PC\TAD_V1_win\Release, son nom : TAD_V1_win.exe (il faut, pour que le programme tourne, que la DLL mpushapi.dll créée par Microsoft se trouve dans ce même dossier).

Le fichier exécutable requiert le .NET Framework. Il est fortement recommandé d'effectuer la mise à jour du système par le biais de Windows Update.

Si vous souhaitez modifier le projet en vue de l'adapter à votre propre cahier des charges, il vous faudra installer le compilateur Visual Microsoft C++ 2005 Express et le mettre à jour à l'aide du Service Pack 1 : Visual C++ 2005 Express SP1. Ultérieurement nous installerons Microsoft Platform SDK for Microsoft Visual C++ 2005 Express. Ce dernier sert au développement d'applications WIN32, nécessaire dans le cas présent pour avoir accès à la DLL (mpushapi.dll). Tout ces programmes sont gratuits et à télécharger depuis le site Web de Microsoft. Vous trouverez également à cet endroit des instructions quant à l'installation et des exemples.

L'utilisation d'une alimentation externe et n'interconnecter alors que les masses de cette dernière et de la carte d'acquisition USB.

Les entrées numériques (K3) demandent certaines précautions : ne pas appliquer de tension ni inférieure à 0 V ni supérieure à 5 V pour éviter tout risque d'endommagement du PIC.

En ce qui concerne les sorties numériques (K1) : chacune de ligne peut fournir un courant maximum de 25 mA pour un niveau logique haut ou bas.

Pour ce qui est des entrées analogiques (K5) : cf. ce qui a été dit pour les entrées numériques.

Dans le cas des sorties analogiques (K4), le courant max est ici aussi de 25 mA pour chaque ligne pour un niveau logique haut ou bas du signal MLI.

Pour finir, la somme de tous les courants drainés par l'ensemble des sorties numériques et analogiques ne doit en aucun cas dépasser 200 mA.

Travaux en cours...

Les applications potentielles de la carte

d'acquisition de données USB sont légions. L'auteur en a développé quelques-unes et travaille sur d'autres encore :

1. Carte à triacs à 8 sorties numériques pour pilotage confortable de charges alimentées par le secteur, carte utilisant des opto-triacs pour l'isolation.
2. Carte de convertisseur résistance-tension donnant une tension proportionnelle à la résistance en entrée, tension appliquée à l'entrée numérique.
3. Carte voltmètre à lecture par barregraphe à LED.
4. Pilotage du régime d'un moteur CC. Cette carte commande la vitesse et le sens de rotation d'un moteur par le biais des sorties analogiques.
5. Carte de commande de moteurs pas à pas capable de micropas via les sorties numériques.
6. Carte de capteurs de distance mettant à contribution les entrées analogiques.
7. Carte à relais pilotés par le biais des sorties numériques.

Elektor et l'auteur seraient heureux de savoir si vous avez développé votre pro-

pre application sur la base de cette carte. N'hésitez pas à nous le faire savoir.

(070148-1)

L'auteur

L'auteur est un ingénieur en techniques de télécommunications travaillant en tant que professeur d'électronique dans un collège technique de Madrid depuis 1984. Il peut prétendre être un incondicional de l'électronique. Il a développé cette carte pour permettre à des étudiants de piloter de petits robots par le biais du port USB d'un PC à l'aide de programmes écrits en C.



Étapes à suivre

Le logiciel pour PC disponible pour ce projet devrait être relativement facile à installer, utiliser et/ou adapter si vous suivez les étapes suivantes :

1. Installer Visual C++ 2005 Express :

<http://msdn2.microsoft.com/en-us/express/aa975050.aspx>

2. Installer Visual C++ 2005 Express SP1 :

<http://msdn2.microsoft.com/en-us/express/aa975050.aspx>

3. Installer PSDK : Microsoft Platform SDK for Microsoft Visual C++ 2005 Express :

<http://msdn2.microsoft.com/en-us/express/aa975050.aspx>

4. Mettre à jour le système d'exploitation par le biais de Windows Update.

5. Dire à Visual C++ d'utiliser PSDK. La séquence pour le faire suggérée par Microsoft est donnée un peu plus loin.

5.1. Mettre à jour les répertoires de Visual C++ dans la section Projects and Solutions dans la boîte de dialogue Options. Ajouter les chemins (paths) vers la sous-section concernée :

Fichiers Exécutables : C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin;

Fichiers Include : C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include;

Fichiers Library : C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Lib.

5.2. Mettre à jour le fichier corewin_express.vsprops.

Il reste une étape additionnelle pour obtenir un modèle (template) Win32 fonctionnel sous Visual C++ Express. Il vous faut éditer le fichier corewin_express.vsprops (trouvé dans C:\Program Files\Microsoft Visual Studio 8\VC\VCProjectDefaults) et modifier la chaîne indiquant : AdditionalDependencies=»kernel32.lib» en : AdditionalDependencies=»kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib».

5.3. Générer et construire (build) une application Win32 pour vérifier les chemins.

Dans Visual C++ Express, le type Win32 Windows Application est désactivé dans l'Assistant (Wizard) Application Win32. Il vous faut, pour activer ce type, éditer le fichier AppSettings.htm se trouvant dans le dossier ?%ProgramFiles%\Microsoft Visual Studio 8\VC\Wizards\AppWiz\Generic\Application\html\1033\». En vous aidant d'un éditeur de texte faites des commentaires des lignes 441 à 444 en les dotant au début d'un // comme illustré ci-après : // WIN_APP.disabled = true; // WIN_APP_LABEL.disabled = true; // DLL_APP.disabled = true; // DLL_APP_LABEL.disabled = true. Enregistrez et fermez le fichier. Ouvrir maintenant Visual C++ Express.