



Chenillard

Premiers pas sous

Jean-Paul Brodier



Les E-blocks et leur outil de programmation Flowcode ouvrent, vous avez sans doute pu le constater, d'immenses perspectives ; encore faut-il commencer. Par quel bout ? Le cédérom

Flowcode contient un cours très abordable nommé Flowcourse. C'est le moment de vous lancer dans la programmation et de transférer un projet de l'environnement E-blocks vers le monde réel !

Vous aurez installé les logiciels proposés sur le cédérom, en acceptant toutes les options proposées par défaut. En plus de *Flowcode*, on installe MPLAB, l'assembleur et compilateur C de Microchip. Il est quasiment inutile de savoir que ces programmes sont présents, puisque *Flowcode* assure l'interface de façon *transparente*, comme on dit.

Avec un navigateur de votre choix, rendez-vous dans le répertoire D:\flowcourse (si le lecteur de CD s'appelle D: sur votre ordinateur). Ouvrez la page mainpage.html. On suppose ici que vous avez déjà pris connaissance du début du cours et que vous savez en gros comment fonc-

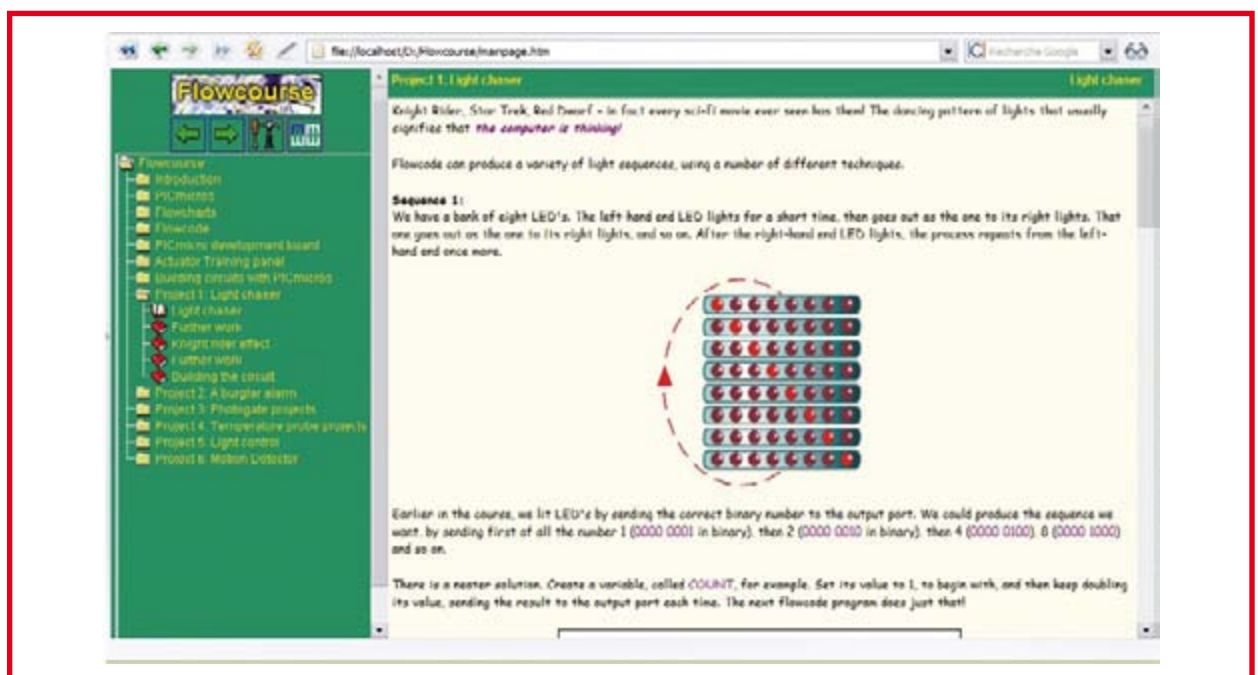


Figure 1. La description du chenillard dans Flowcourse.

d E-blocks

Flowcode

tionne un microcontrôleur et, en théorie au moins, comment il peut allumer et éteindre une LED.

Le projet Chenillard

(Light Chaser)

Ouvrez la page *Project 1: Lightchaser* (figure 1).

Vous trouvez la description d'une application de démonstration parmi les plus simples : un chenillard à LED. Il ne s'agit plus d'une simple LED, mais d'un jeu de lumière à huit LED. Si on veut transposer cette application dans la réalité avec *Flowcode*, les questions embarrassantes commencent à se poser. Voyons plutôt les réponses : non, aucun pont n'est prévu entre cette application de démonstration et les fichiers du répertoire *Tutorials* ; non, le numéro qui sert de nom aux fichiers ne permet pas de les relier à une application et non, aucun fichier du répertoire ne correspond à l'application qui nous intéresse.

Nous allons donc créer avec l'éditeur *Flowcode*, en partant de rien, une application conforme à la description. Commençons par nous faire la main en reproduisant le premier organigramme (*flowchart* en anglais), celui de la figure 2. Cliquez sur File puis New, après avoir saisi le type de microcontrôleur utilisé (ici un PIC16F88), vous avez maintenant une page de l'éditeur avec un programme qui se résume à deux étiquettes DEBUT (BEGIN) et FIN (END) (figure 3).

Pour ajouter des symboles entre le début et la fin, il suffit de cliquer avec le bouton gauche de la souris sur le symbole choisi dans la barre verticale de gauche, puis de traîner le symbole jusqu'à sa place en maintenant le bouton enfoncé (figure 4).

Ensuite, un clic droit ouvre la boîte de saisie des propriétés de l'objet sous le curseur.

Nous rédigeons les étiquettes en français dans la mesure où il ne s'agit pas de mots-clés (figure 5). Le logiciel pose automatiquement la balise de fin de la boucle et y copie le nom que nous avons donné à la balise de début.

En route vers le micro

Pour vérifier le fonctionnement de notre premier programme, nous devons le transposer dans la mémoire de programme du microcontrôleur. Avant d'en arriver à ce stade, il faut transformer l'organigramme en un programme en C, compiler le programme en C en un programme assembleur, assembler ce dernier en code machine à charger, enfin, dans la mémoire flash. Nous pouvons ignorer le détail de ces opérations car elles sont effectuées en chaîne par le programme *Flowcode*, après qu'on aura cliqué sur *Compile to PIC* (figure 6).

La phase « Programmation »

Le programmeur utilisé pour copier les codes machine dans la mémoire du microcontrôleur est la platine MULTI-

Connexion à chaud

Le système Windows XP se débrouille parfaitement avec la liaison USB.

On peut aussi remercier le concepteur d'avoir alimenté par le bus l'interface USB du MULTIPROGRAMMATEUR. Cela évite les problèmes de déconnexion-reconnexion avec le PC quand on coupe l'alimentation pour insérer ou retirer le microcontrôleur ou une carte d'extension.

PROGRAMMATEUR des E-blocks de Matrix Multimedia.

Elle est reconnue automatiquement par les logiciels installés précédemment. Le microcontrôleur utilisé ici est le PIC16F88, qui se révèle largement suffisant.

Le programmeur rend compte de la réussite de l'opération et l'exécution du programme commence immédiatement (figure 7). On voit dans ce compte rendu que le programme n'occupe que 115 mots de programme sur les 4 096 disponibles, soit moins de 3% !

Après cette vérification de la chaîne de programmation du PIC, nous continuons d'ajouter des cases à l'organigramme, jusqu'à avoir reproduit tout le programme. L'organigramme terminé pour le dernier exemple prend l'aspect de la figure 8.

Quelques détails du programme

La boîte « Décalage » demande peut-être une petite explication. L'action exécutée est une opération arithmétique, multiplication ou division, et le décalage physique n'apparaît pas à l'évidence. Il s'agit d'une opération arithmétique parce que c'est tout ce que sait faire un microcontrôleur. Dans tout système de numération, la multiplication ou la division par la base équivaut à un décalage d'un ordre de grandeur à droite ou à gauche (chiasme). Le système binaire ne fait pas exception, ainsi en multipliant le nombre COMPTE par 2, on décale à gauche la LED allumée ; en divisant par 2, on décale à droite.

Vous remarquerez que la durée de l'attente entre les pas est différente de celle de l'exemple ; que la durée pour la « montée » est différente de celle de la « descente ». Il est normal que la descente du mât de cognac soit plus rapide que la montée. Vous avez toute liberté pour accélérer ou ralentir à votre goût un défilement ou les deux.

Strip-tease du projet E-blocks

Maintenant que nous en avons terminé avec l'écriture du programme, de son débogage et de son transfert vers le mi-

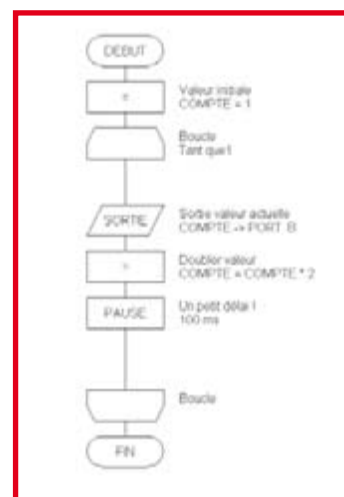


Figure 2. Le programme élémentaire du cours Flowcourse. Nous allons le reproduire pas à pas, avant d'y ajouter des fonctions.

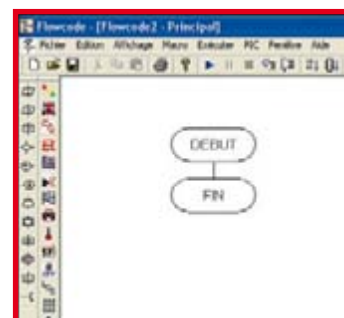


Figure 3. L'organigramme se résume au moment de sa création à ces deux balises, entre lesquelles nous allons ajouter une suite d'instructions.

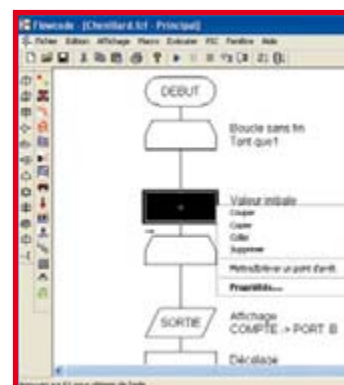


Figure 4. Un clic à droite permet d'accéder aux propriétés de la case que nous venons d'ajouter.

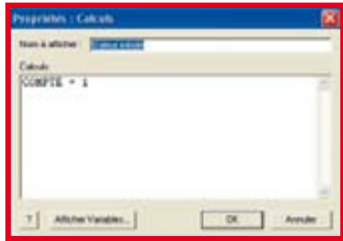


Figure 5. La boîte de saisie des propriétés, qui sont en pratique des calculs arithmétiques.

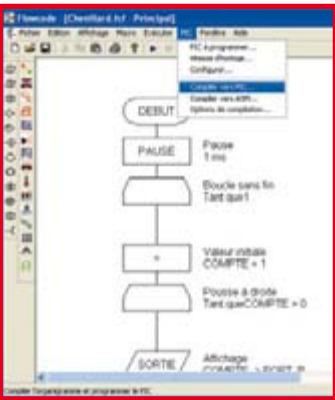


Figure 6. La commande Compile to PIC enclenche le processus de transfert de l'organigramme vers la mémoire de programme du microcontrôleur. On aura précisé auparavant le type de microcontrôleur avec Target PIC.

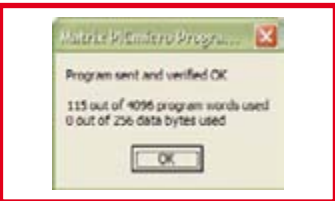


Figure 7. La programmation est l'affaire de quelques secondes. Il reste beaucoup de place disponible en mémoire. L'exécution de notre programme commence aussitôt ce message affiché.

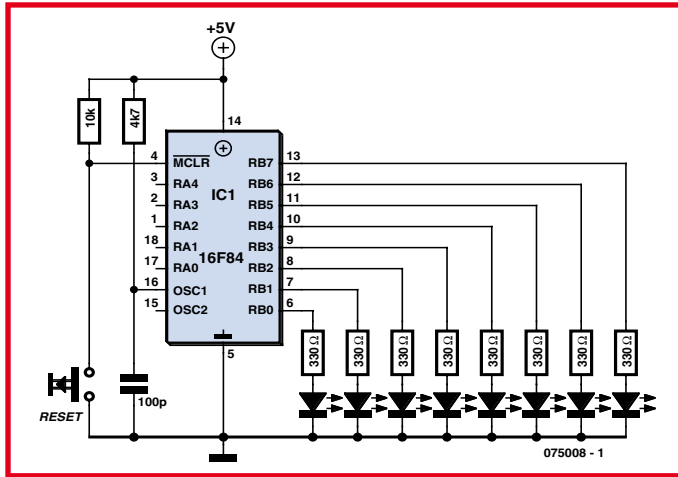


Figure 9. Le schéma de l'application se résume à ces quelques composants.

crocontrôleur, il est temps de faire passer le projet de l'environnement E-blocks vers le monde réel dans lequel il fonctionnera « en autonome », espérons-le !
 Votre méthode de travail globale devrait être d'utiliser les E-blocks pour effectuer tout ce qui a trait à l'écriture du programme, son test et le débogage, pour ensuite « basculer » le total terminé vers une carte totalement distincte. Au cours de cette approche, il vous suffit simplement de revenir à votre système E-blocks en cas de découverte d'erreurs dans le matériel autonome (il y en aura *certainement* !), ou si le « client » requiert des améliorations, modifications, extensions, etc. Dans ce contexte, les capacités de simulation du programme d'E-blocks sera d'un grand secours. Vous ne manquerez pas de constater que certains modules E-blocks tels que la platine à commutateurs et la platine à LED sont uniquement nécessaires au développement et se voient « absorbées » dans l'application finale.
 Dans la ligne de cette approche, nous allons réaliser un montage autonome de façon à rendre la platine MULTIPROGRAMMATEUR et la PLATINE À LED E-blocks à leur destination initiale : le test et la mise au point de programmes.

Le microcontrôleur programmé va prendre place dans un montage conforme au schéma de la **figure 9**. Le montage sera réalisé sur une platine perforée (**figure 10**). L'interruption des pistes conductrices se fait facilement avec un foret de 4 mm monté dans un mandrin de perceuse qui fait office de poignée. Naturellement, le microcontrôleur n'est pas soudé mais enfilé dans un support. Le chenillard est alimenté en 5 V.

(075008-1)

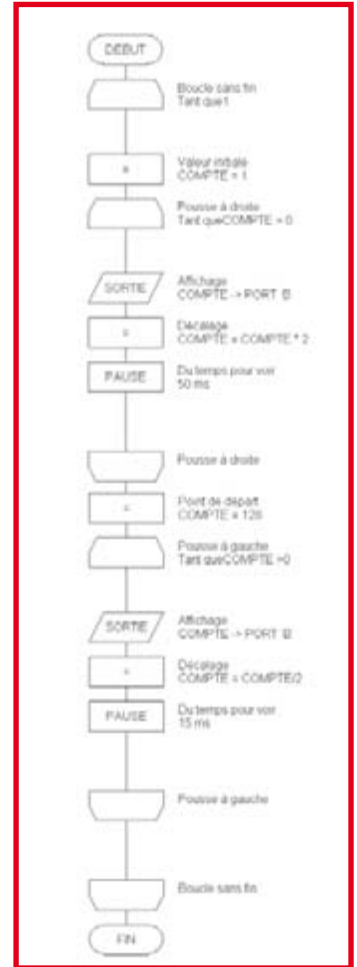


Figure 8. Le point lumineux de notre chenillard effectue un va-et-vient au lieu du balayage simple du programme du début. Pour cela il enchaîne une boucle « Pousse à droite » avec une boucle « Pousse à gauche », puis retourne au commencement.

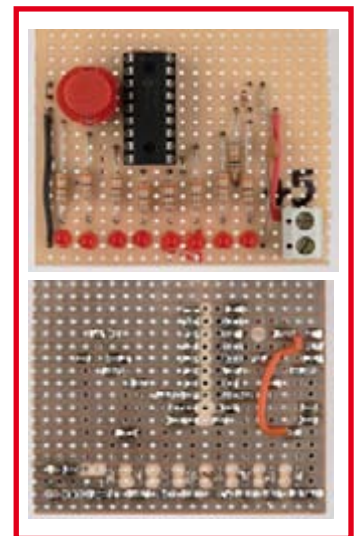


Figure 10. L'implantation des composants sur la platine perforée est conforme à leur disposition sur le schéma de principe. Les traits horizontaux sont reproduits par les bandes cuivrées de la platine, les traits verticaux par les broches de composants ou des morceaux de fil isolé.

Flowcode 3

À l'intention des utilisateurs ayant enregistré leur licence des versions 2.x de Flowcode :

Flowcode 3 est disponible auprès de Matrix Multimedia ou Elektor (e-choppe). Flowcode 3 a fait l'objet d'une description dans le numéro de décembre 2006 d'Elektor.

Pour sa commande, veuillez faire un tour sur www.elektor.fr

