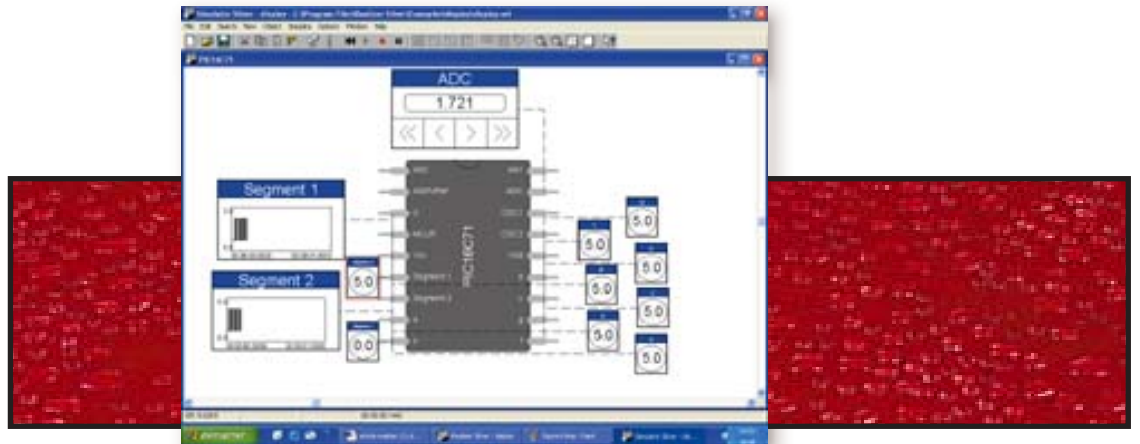


Initiation à la programmation graphique avec REALIZER



Face à l'intérêt croissant pour le REALIZER, le nouvel outil de programmation graphique, nous vous présentons un projet qui expose les différentes potentialités du logiciel. Le REALIZER offre la possibilité aux électroniciens débutants ou professionnels de programmer les microcontrôleurs des différentes familles, sans perdre de temps avec l'apprentissage de l'Assembleur ou d'autres langages comme le C ou le Basic...

Bien entendu, ce n'est pas une déclaration de guerre contre l'Assembleur ou le C, mais la différence est que, s'il vous faut une semaine pour développer votre projet avec le C, il vous faut deux mois avec l'Assembleur, avec le REALIZER vous pourrez réaliser votre projet en moins de 4 jours. Il suffit de quelques heures de formation et d'apprentissage pour entrer dans l'univers de la programmation du microcontrôleur (programmation graphique) alors que l'apprentissage de l'Assembleur ou du langage C nécessite plusieurs mois, voire une année.

Avec REALIZER l'électronicien dispose de tous les outils qui lui sont familiers, nécessaires pour mener à bien toutes les étapes de son projet : de la conception à la réalisation, et d'aller directement à son but, sans pour cela être obligé de se transformer en secrétaire et taper des pages et des pages de code.

Comment est-ce possible ?

C'est très simple, l'utilisateur dispose d'une librairie de composants graphiques que tous les électroniciens connaissent par leurs symboles : logiques, séquentiels, timer, fonctions mathématiques, compteurs, conversion, comparateurs, I2C, UART, entrées, sorties digitales ou analogiques et tant d'autres étant donné que vous avez la possibilité de créer vos propres composants.

Pour cela il suffit de remplir des fenêtres élaborées à cet usage, de deux méthodes différentes, l'une destinée aux non-initiés et l'autre pour les connaisseurs, les deux méthodes sont efficaces.

La **figure 1** présente la librairie, chaque composant sélectionné génère son propre code hexa, binaire, assembleur et même le C et l'Ansi C.

Le logiciel dispose d'outils de connexion pour relier divers composants de la librairie de la même manière qu'un logiciel de CAO électronique. Cette liaison filaire représente le transfert d'un flux d'informations d'un composant de la librairie vers un autre composant. Une fois votre schéma terminé, le REALIZER génère le code qui est votre programme.

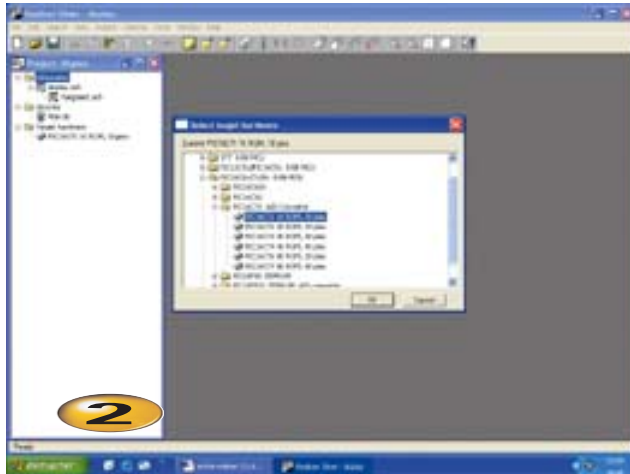
Nous allons matérialiser cela autour d'un projet qui consiste à afficher une grandeur analogique telle que par exemple : la température, la pression ou un simple potentiomètre, libre à chacun de faire son choix et de l'afficher sur 2 digits 7segments.

Pour mener à bien ce projet, nous avons besoin d'un microcontrôleur équipé d'un port analogique digital et d'un port de sortie digitale, pour les afficheurs qui seront externes au microcontrôleur.



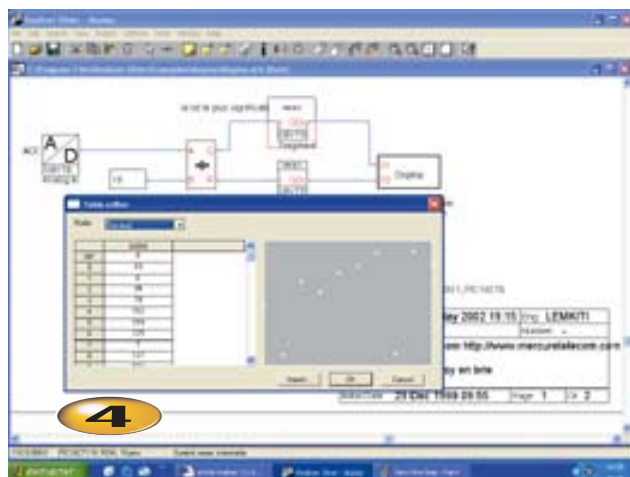
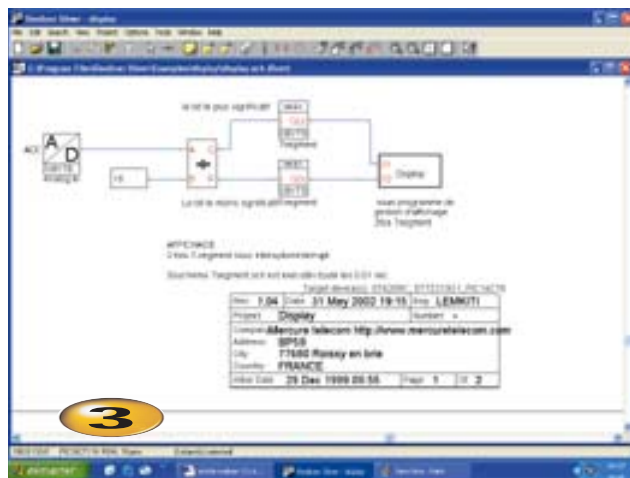
Comme le REALIZER permet de programmer tous les microcontrôleurs de la famille ST6/ST7, soit 27, ainsi que la famille MICROCHIP «les fameux Pic» soit 30 ce qui représente au total 57 microcontrôleurs que nous pouvons programmer au bout de quelques dizaines de minutes sans perte de temps ! Ce qui n'est pas le cas avec les autres langages de programmation !

Figure 2, il suffit de choisir n'importe quel micro parmi cette liste pourvu qu'il soit équipé d'un port de sortie digitale et d'un port d'entrée ana-



logique. Nous avons opté pour un PIC, pour les fans du ST6/ST7, le même projet pourrait être utilisé pour cette famille et, ce, en quelques minutes. Ainsi vous aurez le code pour Pic et pour ST6 alors même que les 2 assembleurs sont totalement différents, vous comprendrez par la suite comment cela est possible, car chaque chose en son temps.

Nous allons réaliser un affichage multiplexé avec interruption Timer, c'est à dire nous allons afficher sur les 2 digits 7 segments les mesures issues de l'entrée analogique. L'affichage de la grandeur analogique se fera par intermittence : affichage des dizaines, des



unités et ainsi de suite, avec une périodicité de 10 ms. De ce fait, l'observateur verra les 2 valeurs affichées sans se rendre compte de cette interruption.

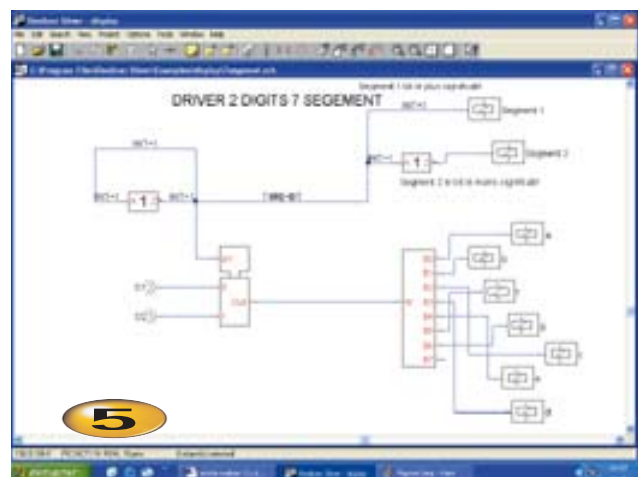
La **figure 3** représente le schéma (le programme) principal de ce projet. Les composants se trouvant sur cette figure A/DC, la valeur 10, la division, les tables indexées se trouvent dans la librairie. La valeur reçue par le port analogique est divisée par 10. Le quotient représente la valeur des dizaines qui sera transmis à l'afficheur des dizaines.

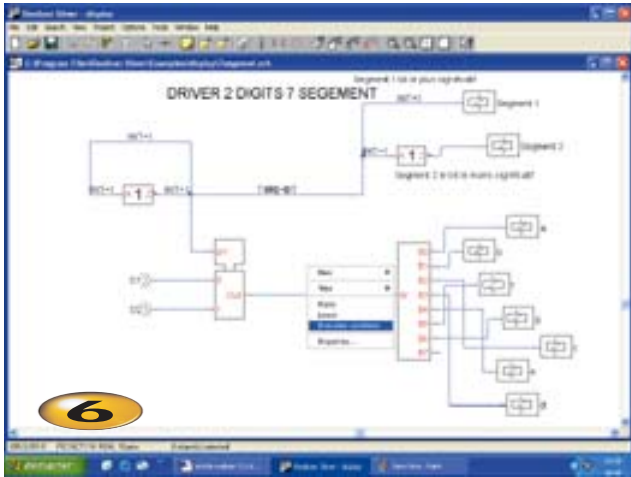
Les résultats de cette division par 10, c'est à dire le quotient et le reste sont stockés dans une table index, le fonctionnement de cette table permet de nous donner le résultat, il suffit de double cliquer sur cette table, de saisir les entrées et les sorties des codes correspondants aux afficheurs 7 segments. Ainsi, par exemple, pour afficher 0 en valeur décimale, nous aurons une valeur 63, 3F en hexadécimale et qui sera en binaire 00111111. La sélection de la base décimale, hexa ou binaire se fait selon votre choix comme le montre la **figure 4**.

Le composant D1/D2 affichage représente le sous-programme qui gère les sorties, il suffit de double cliquer sur ce composant **figure 5**. Comme vous pouvez le constater les données D1/D2 sont multiplexées, l'inverseur qui sélectionne le multiplexage joue le rôle d'un oscillateur, ainsi on a l'affichage par interruption timer une fois sur le segment 1, une fois sur le segment 2, segment qui valide chaque afficheur, à ne pas oublier que segment 1 et segment 2 sont des sorties micro. Le résultat arrive sur un mot qui est éclaté en 8 bits : a,b,c,d,e,f,g,h, qui sont ni plus ni moins que les sorties logiques du microcontrôleur qui sont reliées aux pattes des afficheurs. A ne pas confondre avec les afficheurs.

Pour programmer cette interruption timer ou toute autre interruption, il suffit tout simplement de cliquer droit sur un espace vide dans cette figure, une fenêtre apparaît (**figure 6**) puis cliquer sur la fonction "Exécution Condition" et une autre fenêtre s'affiche qui permet de sélectionner l'interruption timer et vous entrez la valeur de l'interruption qui est de 0,01 s.

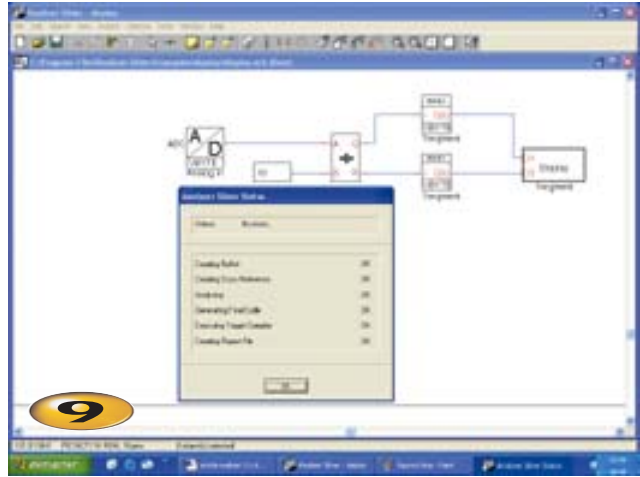
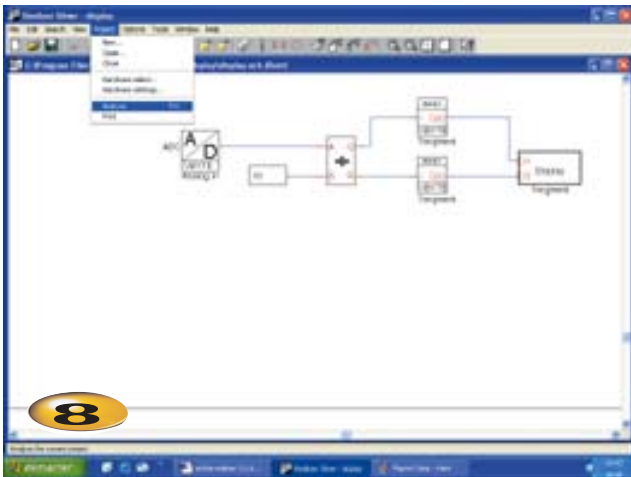
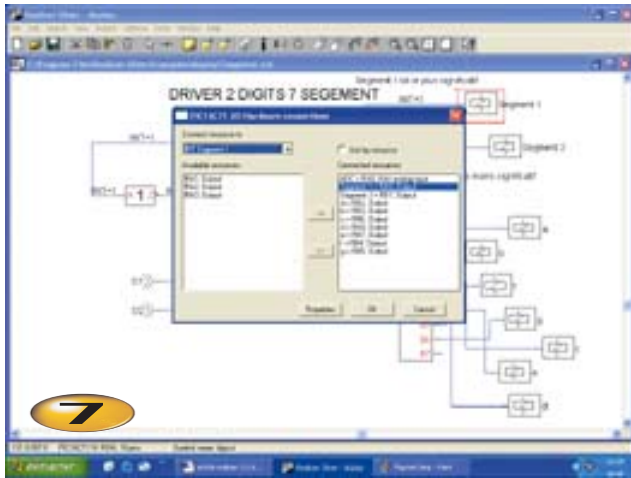
Pour finaliser notre projet, nous devons assigner les entrées et sorties relatives au microcontrôleur en double cliquant sur chacune





des entrées segment 1, segment 2, a,b,c,d,e,f,g,h, par exemple en double cliquant sur segment 1 (**figure 7**), une fenêtre s'ouvre qui permet de sélectionner la sortie du micro voulue, puis double cliquer sur cette dernière pour la valider. Ne pas oublier de faire la même opération pour l'entrée analogique A/D.

A présent le projet est terminé, pour générer le code dans la barre de fonctions, cliquer sur "Projet" et sur "analyse" (**figure 8**). Le résultat de l'analyse s'affiche ; votre projet est terminé !



A ce stade, tous les fichiers sont générés : l'assembleur, hexa. Nous pouvons simuler notre projet en sélectionnant "Affichage simulation" dans la barre de fonctions (**figure 9**). Pour la simulation REALIZER dispose d'outils de simulation, le timer, ainsi vous pouvez simuler aisément votre application pour se rendre compte de son fonctionnement.

Comme nous vous l'avons promis et comme tout arrive à temps, le même projet peut se convertir avec un autre micro, par exemple avec un micro Pic, pour ce faire, il suffit donc de ne pas toucher au composant graphique, par contre, selon le choix de votre Pic, il suffira juste d'affecter les entrées et sorties, et le tour est joué ! (**figure 10**) avec en exemple ST6/ST7.

Nous aborderons, lors de numéros ultérieurs, la gestion d'affichage LCD, claviers, bus I2C et PWM qui nous permettront d'intégrer cette réalisation, étapes par étapes, dans des projets de plus grande envergure. Nous espérons que vous serez au rendez-vous.

Merci de vous adresser à la rédaction pour les programmes hexa. Bonne lecture et bonnes vacances !

A. LEMKITI

