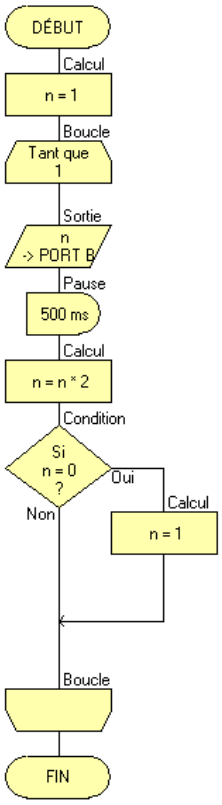
Câblage dans Proteus d'un PIC programmé dans Flowcode

Cette fiche pratique indique les grandes lignes pour utiliser rapidement dans vos montages électroniques sous Proteus un microcontrôleur PIC en tant que composant programmable. Le programme du PIC est mis au point dans Flowcode sous forme d'un algorigramme, puis est ensuite transféré dans le microcontrôleur PIC sous Proteus. Les niveaux logique présents sur les sorties du PIC sont alors exploitable dans le montage.

I – Programme de base



Algorigramme 1

Chacun des 4 paragraphes de cette fiche vous montre un exemple concret d'utilisation d'un PIC dans Proteus. Chaque exemple est composé d'un algorigramme (à réaliser dans Flowcode) et d'un montage électronique (à réaliser dans Proteus). Pour apprendre par la pratique il est conseillé de réaliser une fois chacun de ces exemples sur ordinateur afin de les voir en fonctionnement réel. Une fois ces 4 exemples acquis, libre à vous de les adapter à votre problématique.

Ce premier exemple vous donne les principes de base pour transférer le programme du PIC de Flowcode vers Proteus :

- Réalisez dans Flowcode *l'algorigramme 1*, enregistrez-le sous le nom **exemple1**, puis compilez-le pour obtenir le programme en hexadécimal dans un fichier (cliquez sur **Compiler vers HEX** dans le menu **Puce** de Flowcode)
- Créez dans Proteus un nouveau projet utilisant seulement un PIC16F88 comme unique composant, puis dans les propriétés du PIC réglez l'horloge à **19660800** Hz, et indiquez le fichier **exemple1.hex** dans le champs **Program File** : cela permet à Proteus d'exécuter dans le PIC l'algorigramme créé sous Flowcode
- Lancez sous Proteus la simulation temps réel (touche F12) puis observez l'évolution des sorties du PIC : comme le montre *l'algorigramme 1* les sorties passent à 1 à tour de rôle, tel un chenillard.

Remarques complémentaires (le pic utilisé dans flowcode comme dans proteus sera ici le 16f88 pour les 4 exemples) :

- L'horloge doit être la même dans les 2 logiciels (19660800 Hz par défaut, un copier/coller de Flowcode vers Proteus convient, l'unité n'est pas obligatoire dans Proteus).
- Le type d'oscillateur configuré dans Flowcode (RC, XTAL ou internal) n'a aucune importance pour une simple simulation dans Proteus.
- La borne 4 du PIC (RESET) est active au niveau bas : elle doit être reliée au point d'alimentation VDD (1 logique) pour désactiver le Reset, mais elle peut rester non connectée.
- Aucune borne particulière du PIC ne doit être impérativement câblée : un PIC seul (sans fil ni composants externes) peut très bien être simulé sous Proteus, les niveaux logiques des sorties (les 13 bornes de droite sur le symbole du PIC 16f88) étant indiqués par les petits carrés (rouge ou bleu).
- En indiquant à Proteus le fichier de configuration **.cof** plutôt que le simple code **.hex** on peut faire une simulation pas à pas dans Proteus en suivant l'évolution du code en assembleur.

Montage électronique 1

- Les 13 bornes de droite du PIC (les 5 bits A0 à A4 du port A et les 8 bits B0 à B7 du port B) peuvent être utilisées indépendamment comme une entrée ou comme une sortie : les bits utilisés dans un bloc **sortie** dans Flowcode seront des **sorties** dans Proteus, et les bits utilisés dans un bloc **entrée** dans Flowcode seront des **entrées** dans Proteus. Aucune autre configuration n'est à préciser pour le sens des entrées/sorties.

II – Utilisation des bits A5, A6 et A7

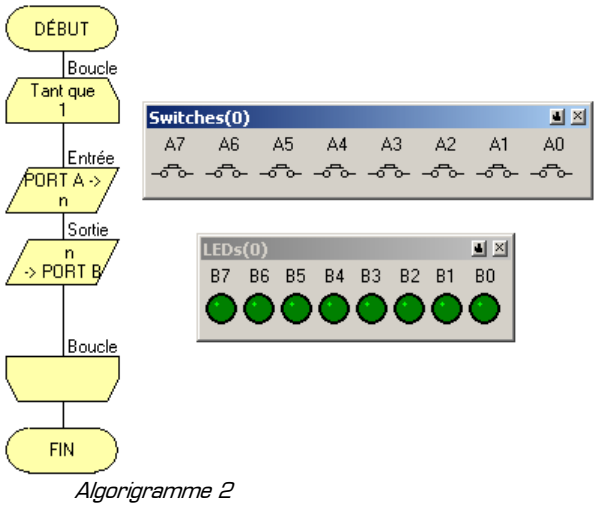
Problème : par défaut la borne 4 du PIC est le RESET (actif au niveau bas), ce qui interdit une utilisation direct du bit A5. De même les bornes 15 et 16 du PIC permettent de câbler un oscillateur externe (quartz ou circuit RC), et interdisent une utilisation directe des bits A6 et A7 (même si les bornes 15 et 16 ne sont pas connectées dans Proteus).

Question : comment utiliser les bits A5, A6 et A7 comme simple entrée/sortie utilisateur ?

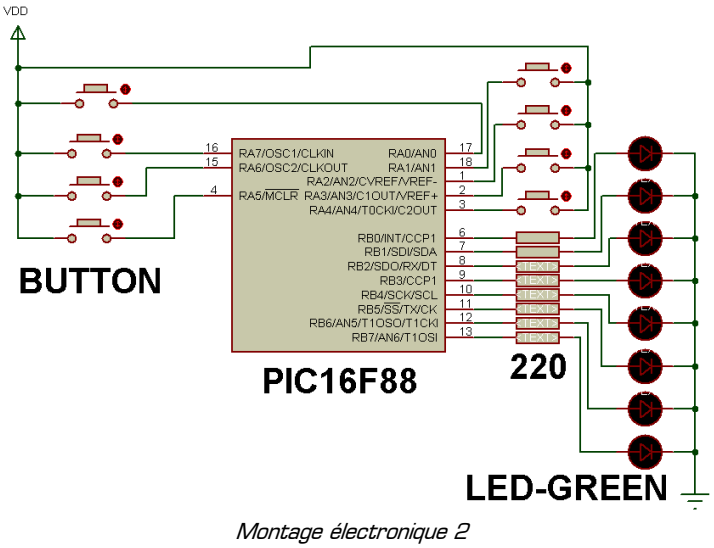
Réponse :

- le rôle de la borne 4 du PIC (choix entre entrée de RESET ou entrée/sortie utilisateur) est configurable dans Flowcode en allant dans le menu ***Puce*** + ***Configurer*** + ***Bouton*** « ***Switch To Expert Config Screen*** », puis choisir **RA5** à la place de **MCLR** sur la 4ème ligne
- pour utiliser les bits A6 et A7 il faut configurer le PIC en oscillateur interne (dans le menu ***Puce*** + ***Configurer*** de Flowcode). Ainsi les bornes 15 et 16 ne sont pas réservées à un oscillateur externe et deviennent des entrées/sorties utilisateur comme les autres

Le PIC 16f88 peut donc être vue en un premier temps comme un simple circuit logique programmable avec 16 entrées/sorties logiques. Il peut par exemple permettre de réaliser très rapidement un séquenceur logique (en lieu et place d'un montage à bascules), sans utiliser un seul composant externe pour le fonctionnement du PIC. Les fonctionnalités avancées du PIC (oscillateur externe, reset, convertisseur analogique/numérique, interruptions, etc.) ne sont pas indispensables à maîtriser et pourront être vue en un second temps et progressivement. Voici l'exemple 2 :



Algorigramme 2



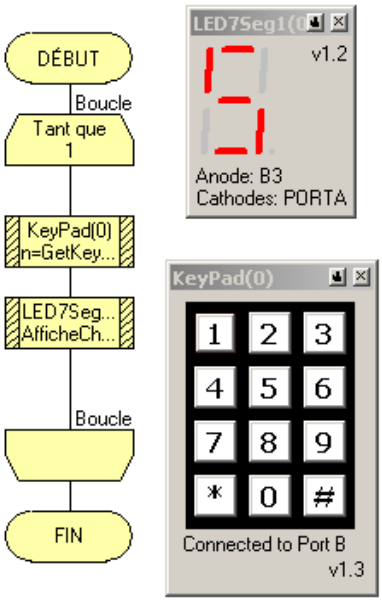
Montage électronique 2

III – Connexion d'un afficheur et d'un clavier à un PIC dans Proteus

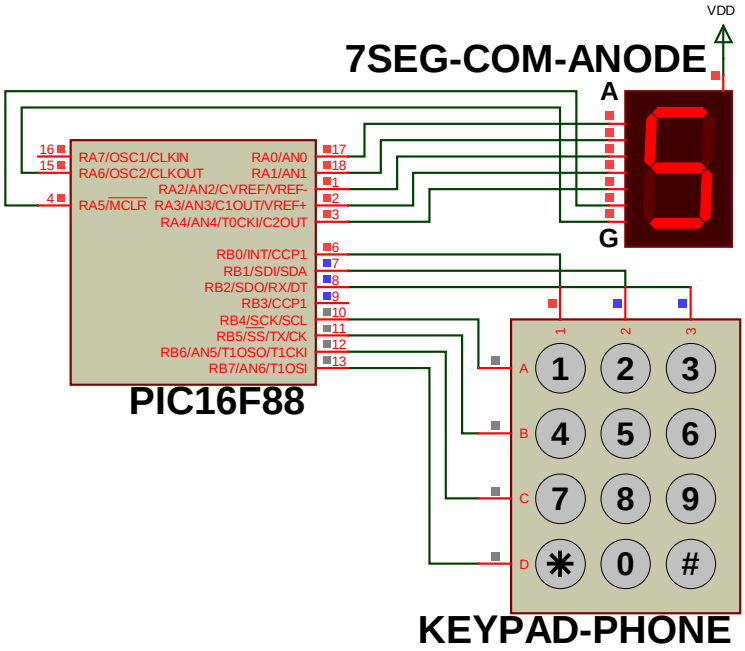
En câblant dans Proteus un afficheur 7 segments à anode commune au PIC comme indiqué dans Flowcode il est possible de le piloter en utilisant les macros de Flowcode. ***Rappel :*** sur un afficheur à anode commune les segments seront allumés s'ils sont reliés à un 0 logique (niveau bas).

Utilisation directe d'un clavier matriciel et d'un afficheur 7 segments dans Proteus avec un PIC 16f88 (sans transistor ni résistance de tirage) :

- Connecter dans Proteus le clavier **KEYPAD-PHONE** au port B du PIC 16f88 comme indiqué dans Flowcode (le bit B3 n'est pas utilisé) : colonne 1, 2 et 3 aux bits B0 à B2 et lignes A, B, C et D aux bits B4 à B7 (voir le *Montage 3*)
- Dans Flowcode, connecter les 7 segments de l'afficheur au bits A0 à A6, l'anode commune à B3 et le point décimal à A7
- Dans Proteus, connecter les 7 segments de l'afficheur **7SEG-COM-ANODE** aux bits A0 à A6, et son anode commune à un point d'alimentation (élément **POWER** du mode Terminal) relié à VDD
- Laisser le bit B3 du PIC non connecté dans Proteus (l'afficheur de Proteus ne possède pas de point décimal)



Algorigramme 3



Montage électronique 3

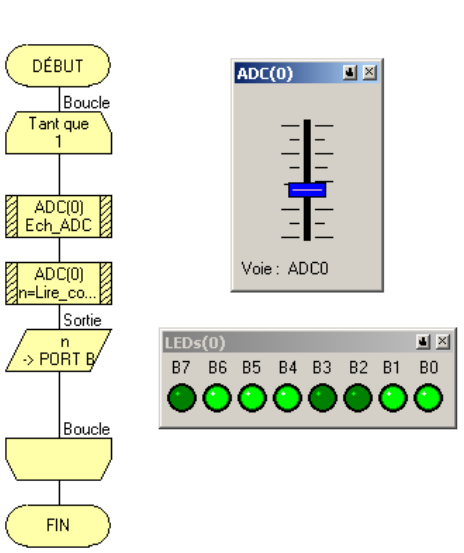
Il est alors possible d'afficher sur l'afficheur 7 segments le chiffre correspondant à la touche enfoncée, en utilisant un simple algorigramme contenant 2 bloc « Routine de composant » (voir *l'algorigramme 3*) :

- Le premier bloc « Routine de composant » appelle la macro **GetKeypadNumber** du clavier et enregistre la valeur de la touche enfoncée dans la variable n (variable de type octet à créer)
- Le second bloc appelle la macro **AfficheChiffre** de l'afficheur en lui communiquant la valeur de la variable n pour afficher le chiffre

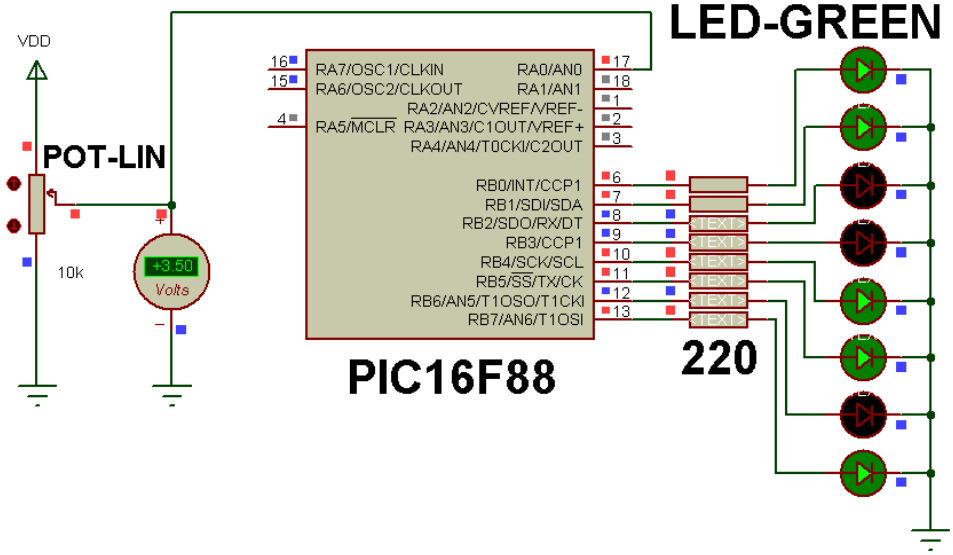
Remarque : pour éteindre l'afficheur 7 segments lorsque aucune touche n'est enfoncée il faut mettre 255 dans le port A (toutes les soties à 1).

IV – Utilisation des entrées analogiques

Ce dernier exemple montre comment utiliser le convertisseur analogique/numérique du PIC (le C.A.N.) afin de mesurer une tension et de la convertir en valeur numérique exploitable par le PIC. Dans le *montage électronique 4* le potentiomètre **POT-LIN** crée une tension variant de 0 V à 5 V (en fonction de la position du curseur du potentiomètre). Cette tension est appliquée sur l'entrée analogique **AN0** du PIC. Dans *l'algorigramme 4* le résultat de la conversion analogique/numérique est enregistré dans la variable n grâce à la macro **lire_comme_Octet** du composant **ADC(0)**.



Algorigramme 4



Montage électronique 4

Remarque : le PIC 16f88 possède 7 entrées analogiques utilisables. Elles sont notées **AN0** (borne 17) à **AN6** (borne 13) sur le symbole du PIC