

# MODE D'EMPLOI (V1.50)

## PIC-01

### Programmateurs de microcontrôleurs PIC de Microchip

#### I) Installation du logiciel et du programmeur :

##### Présentation :

Le PIC-01 permet la programmation des microcontrôleurs PIC séries de chez Microchip, (familles PIC12Cxxx, PIC12CExxx, PIC12Fxxx, PIC16Cxxx, PIC16CExxx, PIC16Fxxx et PIC18Fxxx), ainsi que les EEproms Séries, (famille 24Cxx). Equipé de 5 supports tulipes, il supporte les composants en boîtiers DIP 8, 18, 28 et 40 broches permettant la programmation de plus de 110 références différentes. Il est équipé d'une véritable interface RS232 permettant la connexion sur le port série (cordon fourni) de tout compatible PC, il fonctionne avec le logiciel ICprog et WinPic800 très complet sous Windows 95/98/ME/NT/2000/XP/VISTA/7/8 (32/64bits).

##### Copyright:

Programmeur et mode d'emploi Copyright (C) 1996 - 2016 par SEEIT.

SEEIT est une marque déposée.

SEEIT ne pourra en aucun cas être tenue pour responsable des préjudices de quelque nature que ce soit pouvant résulter de l'utilisation des logiciels, du programmeur, du cordon informatique ou de la documentation.

Les mini-programmateurs SEEIT ont été conçus à titre éducatif dans un but pédagogique ou pour la réalisation de prototypes en vue d'une utilisation légale conforme aux lois en vigueur dans le pays d'utilisation et conforme aux droits de l'homme.

Tous droits réservés. Toute reproduction, intégrale ou partielle, par quelque procédé que ce soit, du programmeur ou de la documentation est illicite, (loi du 11 Mars 1957, article 40, 1er alinéa). Cette reproduction illicite, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

Les mises à jour des logiciels sont cependant téléchargeables gratuitement sur notre site Internet à l'URL suivante:

<http://www.seeit.fr>

##### Contenu du produit:

Un programmeur.

Un câble rallonge port série.

Une disquette 3 1/2.

Un mode d'emploi.

##### Composants supportés par le PIC-01 sous IC-Prog Version 1.05E:

<b>PIC12Cxxx</b>	PIC12C508	PIC12C508A	PIC12C509	PIC12C509A	PIC12CE518	PIC12CE519	PIC12C671	PIC12C672
	PIC12CE673	PIC12CE674						
<b>PIC12Fxxx</b>	PIC12F629	PIC12F675						
<b>PIC16Cxxx</b>	PIC16C433	PIC16C61	PIC16C62A	PIC16C62B	PIC16C63	PIC16C63A	PIC16C64A	PIC16C65A
	PIC16C65B	PIC16C66	PIC16C67	PIC16C71	PIC16C72	PIC16C72A	PIC16C73A	PIC16C73B
	PIC16C74A	PIC16C74B	PIC16C76	PIC16C77	PIC16C84	PIC16C505*	PIC16C620	PIC16C620A
	PIC16C621	PIC16C621A	PIC16C622	PIC16C622A	PIC16CE623	PIC16CE624	PIC16CE625	PIC16C710
	PIC16C711	PIC16C712	PIC16C715	PIC16C716	PIC16C717	PIC16C745	PIC16C765	PIC16C770
	PIC16C771	PIC16C773	PIC16C774	PIC16C781	PIC16C782	<b>PIC16C923**</b>	<b>PIC16C924**</b>	
<b>PIC16Fxxx</b>	PIC16F72	PIC16F73	PIC16F74	PIC16F76	PIC16F77	PIC16F83	PIC16F84	PIC16F84A
	PIC16F88	PIC16F627	PIC16F628	PIC16F628A	PIC16F630*	PIC16F648A	PIC16F676*	PIC16F818
	PIC16F819	PIC16F870	PIC16F871	PIC16F872	PIC16F873	PIC16F873A	PIC16F874	PIC16F874A
	PIC16F876	PIC16F876A	PIC16F877	PIC16F877A				
<b>PIC18Fxxx</b>	PIC18F242	PIC18F248	PIC18F252	PIC18F258	PIC18F442	PIC18F448	PIC18F452	PIC18F458
	PIC18F1220	PIC18F1320	PIC18F2320	PIC18F2585	PIC18F2620	PIC18F4320	PIC18F4455	PIC18F4520
	PIC18F4539	PIC18F4550	<b>PIC18F6620**</b>	<b>PIC18F6720**</b>	<b>PIC18F8620**</b>	<b>PIC18F8720**</b>		
<b>24Cxx</b>	X24C01	24C01A	24C02	24C04	24C08	24C16	24C32	24C64
	AT24C128	M24C128	AT24C256	M24C256	AT24C512	24C515		

\*Composants en boîtiers DIP14 nécessitant une adaptation sur le support 18DIP, (voir plus bas).

\*\*Composants en boîtier TQFP64 nécessitant un adaptateur optionnel.

### **Installation de la carte:**

Connecter le câble informatique entre le connecteur DB-9 femelle de la carte de programmation et un port série de votre PC, COM1 ou COM2. Connecter un bloc d'alimentation secteur continue **non stabilisé** 500 mA sur l'embase alimentation du programmeur et réglez le sur 12V, (celle-ci doit délivrer au moins 16V à l'entrée du programmeur PIC-01 pour une programmation correcte des microcontrôleurs PIC). Si vous utilisez une alimentation continue et régulée, réglez votre alimentation sur 16V. Le jack doit avoir la masse autour et le + au milieu.

### **Installation et configuration du logiciel ICprog V1.05E pour le PIC-01 :**

Lancer l'explorateur Windows et sélectionner le menu « Fichier\Nouveau\Dossier » pour créer un répertoire « C:\PIC-01 » sur votre disque dur. Ensuite sélectionner le lecteur de disquette pour effectuer un « Copier » et un « Coller » pour transférer le fichier PIC-01.ZIP se trouvant sur la disquette vers le répertoire PIC-01 que vous venez de créer sur votre disque dur. Utilisez ensuite Winzip ou Pkzip pour décompresser les fichiers.

Pour lancer le logiciel Icprog cliquer sur « Démarrer », puis sur « Exécuter » et sélectionner le fichier C:\PIC-01\ICPROG.EXE. Cliquer sur « OK » pour lancer le logiciel.

Allez dans le menu « Settings\Options » et cliquer sur l'onglet « Language ». Sélectionner l'option « French » dans le menu déroulant et cliquer sur « OK ».

Cliquer maintenant sur le menu « Configurations\Hardware » et régler les paramètres suivants :

« Programmeur » : JDM programmer.

« Port » : COM1 ou COM2.

« Delay I/O » : 2 ou 6.

Interface : « Windows API » (pour les PIC12Cxxx et PIC16Cxxx) et « Direct I/O » (pour les PIC16Fxxx et PIC18Fxxx).

Communication : ne rien cocher.

Cliquer ensuite sur « OK » pour sortir.

Vérifier que le menu « Smartcard (Phoenix) » n'est pas coché dans le menu « Configuration ».

Si vous utilisez **Windows NT/2000/XP** vous devez également cliquer sur le menu « Configuration\Option », allez dans l'onglet « Misc », cocher l'option « Active driver NT/2000/XP ». Egalement vérifier que le fichier « icprog.sys » est bien installé dans le même répertoire que le fichier « icprog.exe ».

Pour les utilisateurs, n'ayant plus de port série sur leurs ordinateurs, vous pouvez utiliser un câble convertisseur USB-série. Dans ce cas sélectionner dans le menu « Configuration\Hardware » l'interface « Windows API » uniquement. Cependant cette méthode ralentit considérablement le temps de programmation du composant, (temps multiplié par 10 environ).

## **II) Utilisation des composants :**

### **Positionnement des composants sur les supports :**

Une encoche sur le support tulipe ou le symbole 1 sur le circuit imprimé permet de repérer le sens d'insertion du composant. Les composants doivent être orientés dans le même sens que les deux circuits intégrés déjà soudés sur le programmeur. (composants noir à 14 et 16 pattes)

Pour retirer un composant de son support tulipe, utiliser un tournevis plat et faire levier progressivement de chaque côté du composant.

Le composant peut être inséré ou retiré de son support lorsque la carte de programmation est déjà sous tension et le logiciel en fonctionnement.

Ne pas insérer plusieurs composants en même temps sur différents supports tulipes.

Ne pas insérer ou retirer un composant de son support durant une phase de lecture ou de programmation, (lorsque la LED verte est allumée).

### **Positionnement des composants en boîtiers DIP28 large (600mil) :**

Pour programmer un composant en boîtier 28 broches larges (anciens modèles) il faut le placer sur le support 40 broches, décalé vers le côté où se trouve l'encoche du support tulipe 40 broches, ainsi la broche 1 du composant en DIP28 doit se retrouver sur la broche 1 du support tulipe DIP40. Son orientation étant dans le même sens que les autres circuits.

### **Positionnement des composants en boîtier DIP14 (PIC16C505, PIC16F630 et PIC16F676) :**

Les PIC16C505, PIC16F630 et PIC16F676 sont des composants récents et utilisent des boîtiers DIP14, non compatibles avec le PIC-01. Il est cependant possible de les utiliser sur le support DIP18 en effectuant les modifications suivantes. Sous le support tulipe DIP18 ajouter les 4 ponts suivants entre les pattes : relier la 1 sur la 14, la 5 sur la 18, la 12 sur la 16 et la 13 sur la 17. Ces modifications n'empêchent pas de continuer à programmer des composants en boîtiers DIP18. Les composants en boîtiers DIP14 sont à placer décalés vers le côté où se trouve l'encoche du support tulipe, ainsi la broche 1 du composant en DIP14 doit se retrouver sur la broche 1 du support tulipe DIP18. Son orientation étant dans le même sens que les autres circuits.

Une autre solution plus simple consiste à placer le composant en boîtier DIP14 sur le support tulipe 8 broches des 12C508. Placer dans ce cas les broches 1,2,3,4 et 11,12,13,14 du composant sur le support DIP8, les autres pattes du composant restant en l'air, ainsi la broche 1 du composant en DIP14 doit se retrouver sur la broche 1 du support tulipe DIP18. Son orientation étant dans le même sens que les autres circuits. De cette façon aucune modification hardware n'est à effectuer.

### **Affichage des données du composant dans le buffer :**

Lorsque vous sélectionnez un composant l'interface du logiciel change afin de s'adapter. Cette fonctionnalité très souple permet à l'utilisateur d'accéder à toutes les fonctions des microcontrôleurs Microchip.

L'affichage du buffer change en fonction de l'architecture mémoire de chaque composant par exemple, la 16C84 et la 16F84 sont des microcontrôleurs 14 bits, donc lorsque le buffer est vide, à chaque adresse vous devez voir la valeur hexadécimale 3FFFh à l'écran, (en binaire : 11 1111 1111 1111b). De même la 12C508 et la 12C509 sont des microcontrôleurs 12 bits, donc quand le buffer est vide, à chaque adresse vous devez voir la valeur hexadécimale 0FFFh à l'écran, (en binaire : 1111 1111 1111b).

A droite de l'écran, dans le cadre Configuration, apparaît également tous les registres modifiables par l'utilisateur.

Lorsque le composant est équipé de plusieurs mémoires (Flash et EEprom interne) deux buffers apparaissent dans l'interface.

### **Effacement d'un composant :**

Un microcontrôleur PIC12C508/509 ne peut être programmé qu'une seule fois et ne s'efface pas. Pour les microcontrôleurs PIC qui ne s'efface pas électriquement, il existe des versions à fenêtres équipées d'une EPROM. Elles ont le même nom que le microcontrôleur courant mais avec l'extension « JW », ex : PIC12C508A-JW.

Un microcontrôleur PIC12Fxx/PIC16Fxx/PIC18Fxx doit être effacé avec le menu "Erase" avant reprogrammation.

Une EEPROM du type 24Cxx se reprogramme directement par dessus son ancien contenu sans qu'il soit nécessaire de l'effacer préalablement.

### **Résolution de problèmes courants:**

Si la case « CP » code protect est cochée (dans le cadre à droite de l'écran), votre composant sera protégé en lecture lors de sa programmation. Dans ce cas il ne sera pas possible de relire son contenu après programmation et si vous effectuez une vérification après la programmation un message d'erreur surviendra.

Afin de pouvoir faire une vérification du composant après programmation ne cochez pas cette case avant de lancer la programmation. Ensuite une fois la vérification effectuée, sélectionnez le menu « Commande\Programmer configuration » pour programmer les registres et le « code protect » afin de protéger le composant en lecture.

Si le composant ne se programme pas, assurez vous que le bloc d'alimentation que vous utilisez délivre bien au moins 16V à vide lorsqu'il est branché sur le programmeur, à l'aide d'un voltmètre.

### **Aide en cas de panne :**

Si vous insérez un microcontrôleur à l'envers sur le support tulipe et que vous faites une lecture ou une programmation du composant vous risquez d'endommager le transistor BC557. Ce transistor commute la tension de programmation 13V. Pour vérifier qu'il fonctionne bien vous pouvez mesurer la tension sur la broche 4 du support tulipe 18 broches durant une phase de lecture ou d'écriture. Lorsque la LED verte est allumée on doit trouver une tension de 13V. Lorsque la LED verte est éteinte on doit trouver une tension de 0V. Si vous n'avez pas ceci changer le transistor PNP par un BC557 ou un 2N2907A.

Egalement lorsque vous insérez un composant à l'envers sur le support la LED rouge s'éteint car l'alimentation se met en protection.

## **III) Utilisation du logiciel Icprog :**

### **Principe de fonctionnement du buffer:**

Les logiciels du programmeur utilisent un buffer, c'est à dire une mémoire intermédiaire entre les fichiers sur disques et les mémoires programmables des composants, tableau hexadécimal visualisé à l'écran.

Pour programmer un composant à partir d'un fichier vous devez d'abord charger le contenu du fichier dans le buffer à l'aide de la commande « Fichier/Ouvrir fichier », puis transférer le contenu du buffer vers le composant avec le menu

« Commande\Tout programmer ».

Pour sauvegarder le contenu d'un composant sur disque vous devez d'abord charger le contenu de la mémoire du composant dans le buffer à l'aide du menu « Commande\Tout lire », puis transférer le contenu du buffer sur disque à l'aide du menu

« Fichier\Enregistrer sous ».

Lorsque le buffer est chargé à partir d'un fichier sur disque ou à partir de la mémoire d'un composant il est alors possible de modifier son contenu à l'écran en double cliquant sur la valeur à modifier.

Attention, si vous effacez un composant, cela n'effacera pas le contenu du buffer à l'écran pour autant. Vous devez alors refaire une lecture du composant pour voir apparaître les modifications à l'écran.

Le contenu de la fenêtre « Check Sum » (en bas à droite) résulte de l'addition de tous les octets que constitue le programme lu dans le composant. Cela permet par exemple d'identifier un programme par rapport à un autre pour voir s'ils sont identiques. Cette valeur se modifie automatiquement lorsque l'on modifie le programme à l'écran.

### **« Fichier/Ouvrir fichier » :**

Ouvre un fichier pour le charger dans la fenêtre « Adresse-Programme Code » et « Adresse-Eeprom data ». Avant de charger un fichier vous devez avoir sélectionné le type de composant à programmer dans le menu "Configuration/Composant". Vous pouvez lire différents formats de fichiers : .hex (IHX8) pour les fichiers au format Intel 8 bits, .hex (IHX16) pour les fichiers au format Intel 16 bits, .bin pour les fichiers binaires, .mot pour les fichiers au format Motorola-S, .obj pour les fichiers projets et .e2p pour les fichiers au format PonyProg.

**« Fichier\Enregistrer sous » :**

Permet de sauvegarder le contenu de la fenêtre courante avec modification possible du nom de fichier et du format de sauvegarde. Si vous sauvegardez avec le format .bin, le contenu du fichier reflétera exactement le contenu de la mémoire du composant. Si vous sauvegardez avec le format .hex le fichier sera codé au format Intel (IHX8).

**« Fichier\Ouvrir fichier Data » :**

Normalement, le fichier que vous chargez à l'écran dans le buffer se scinde automatiquement dans la partie haute et basse du buffer. Dans certain cas le fichier pour la partie Eeprom Data du microcontrôleur PIC se trouve dans un fichier séparé. Dans ce cas vous utiliserez ce menu pour charger le contenu "Adresse-Eeprom data" du buffer. Vous pouvez lire différents formats de fichiers : .hex (IHX8) pour les fichiers au format Intel 8 bits, .hex (IHX16) pour les fichiers au format Intel 16 bits, .bin pour les fichiers binaires.

**« Fichier\Fichier récents » :**

Permet de recharger directement les derniers fichiers utilisés par le programme.

**« Fichier\Imprimer » :**

Permet d'imprimer le contenu du buffer en cours.

**« Fichier\Exit » :**

Permet de quitter le logiciel. Si des changements ont été apportés le logiciel vous demandera si vous souhaitez enregistrer les modifications.

**« Edition\Remplir Buffer » :**

Permet d'initialiser avec la même valeur partout le contenu complet du buffer « Adresse-Programme Code ». Cette valeur peut être écrite en décimale, hexadécimale ou ASCII.

**« Edition\Remplir buffer depuis » :**

Permet d'initialiser avec la même valeur une portion du buffer « Adresse-Programme Code ». Dans le champ « Data » inscrire en décimale, hexadécimal ou ASCII la valeur qui doit être écrite dans le buffer. Dans le Champ « Adresse » inscrire en hexadécimal l'adresse de départ et l'adresse de fin de la portion à modifier.

**« Edition\Copier Depuis le Buffer et Edition\Coller dans le buffer » :**

Permet de copier une partie du buffer « Adresse-Programme Code » dans une autre partie du buffer. Sélectionner d'abord avec la souris les codes que vous voulez copier puis appeler le menu « Edition\Copier Depuis le Buffer ». Ensuite, placer la souris à l'endroit à partir duquel vous voulez replacer la portion de code puis appeler le menu « Edition\Coller dans le Buffer ».

**« Edition\Remplir Data Buffer » :**

Permet d'initialiser avec la même valeur partout le contenu complet du buffer « Adresse-Eeprom Data ». Cette valeur peut être écrite en décimale, hexadécimale ou ASCII.

**« Edition\Remplir buffer DATA depuis » :**

Permet d'initialiser avec la même valeur une portion du buffer « Adresse-Eeprom Data ». Dans le champ « Data » inscrire en décimale, hexadécimal ou ASCII la valeur qui doit être écrite dans le buffer. Dans le Champ « Adresse » inscrire en hexadécimal l'adresse de départ et l'adresse de fin de la portion à modifier.

**« Edition\Copier Depuis le Buffer Data » et « Edition\Coller buffer Data » :**

Permet de copier une partie du buffer « Adresse-Eeprom Data » dans une autre partie du buffer. Sélectionner d'abord avec la souris les codes que vous voulez copier puis appeler le menu « Edition\Copier Depuis le Buffer ». Ensuite, placer la souris à l'endroit à partir duquel vous voulez replacer la portion de code puis appeler le menu « Edition\Coller dans le Buffer ».

**« Buffer\Active Buffer » :**

Icprog contient plusieurs buffers que l'on peut activer à l'aide de ce menu.

**« Buffer\Comparer » :**

Permet de comparer le contenu de deux buffers. ICprog comparera le buffer actif avec le buffer sélectionné dans le menu. Un message d'erreur apparaîtra en cas de différence entre les deux buffers.

**« Configuration\Composant\I<sup>2</sup>C EEprom » :**

Permet de sélectionner une EEprom du type 24Cxx. S'il s'agit d'une EEprom du type 24LCxx, sélectionner également le composant comme s'il s'agissait d'une 24Cxx. Fonctionne avec les programmeurs PIC-01, ATM-01 et SER-01.

#### **« Configuration\Composant\Microchip PIC » :**

Permet de sélectionner un microcontrôleur PIC du type 12Cxxx, 12Fxxx, 16Cxxx, 16Fxxx, 18Fxxx pour une utilisation avec le programmeur PIC-01. Pour les composants de la série 16C54/55/56/57/58, le mode de programmation est différent et il faut utiliser le programmeur PIC-02.

Différentes options apparaîtront également dans le cadre « Configuration » permettant de modifier les registres de configurations. Pour connaître l'utilisation de ces registres veuillez consulter le « datasheet » du fabricant concerné. Dans la plupart des cas ces options sont initialisées automatiquement lors du chargement du fichier .hex dans le Buffer.

#### **« Configuration\Composant\Autres » :**

Les menus « 4-Wire Eeprom, IM-Bus Eeprom, Microwire Eeprom, Serial Eeprom, Scenix, SPI Eeprom » sont utilisables avec le programmeur SER-01 uniquement.

Le menu « SPI µC » permet de sélectionner un microcontrôleur AVR du type AT90Sxxxxx pour une utilisation avec le programmeur ATM-01.

Le menu « Flash » est supportée par le programmeur ATM-02.

Le menu « Serial µC » n'est pas supportée par les programmeurs SEEIT.

#### **« Configuration\Derniers Composants » :**

Permet de resélectionner facilement l'un des derniers composants utilisé.

#### **« Configuration\Hardware F3 » :**

Permet de configurer l'interface de programmation entre le logiciel et la carte de programmation.

##### **« Programmeur » :**

JDM programmer pour le PIC-01 / SER-01 / ATM-01 / CAR-01/02/03/04/07.

TAIT parallel programmer pour le PIC-02

##### **« Ports » :**

COM1 ou COM2 / LPT1 ou LPT2 suivant la connexion de votre programmeur sur votre PC.

Dans tous les cas la LED verte de votre programmeur doit s'allumer lorsque vous effectuez une opération de lecture ou d'écriture. Si ce n'est pas le cas changez de port sélectionné.

##### **« Délais I/O » :**

Ce réglage dépend du PC utilisé, essayez sur 1 à 20 en cas de problème de programmation. Ce réglage n'a pas d'effet sur le port parallèle. (PIC-02).

##### **« Interface » :**

Sélectionner Windows API ou Direct I/O.

##### **« Communication » :**

Permet d'inverser les signaux envoyés ou reçu sur le port série. En général aucune case n'est cochée.

Pour la configuration exact, se référer à la première partie de ce document.

#### **« Configuration\Hardware Test » :**

Permet de tester les différentes broches du port série. En cliquant sur une des cases vous activez un signal haut sur la broche correspondante du port série. Ceci permettant de vérifier le bon fonctionnement d'un programmeur ou de suivre un signal stable sur les différents composants du programmeur.

#### **« Configuration\Options\Confirmation » :**

Permet de faire apparaître une fenêtre de demande de confirmation à chaque fois que l'on doit effectuer une programmation, un effacement, une protection en lecture ou une réécriture.

#### **« Configuration\Options\Notification » :**

Permet de faire apparaître un message lorsque qu'un fichier est chargé à l'écran et qu'il ne contient pas d' ID-WORD ou de CONFIG-WORD.

#### **« Configuration\Options\I<sup>2</sup>C » :**

Cette option agit sur la programmation des composants 24Cxx (I<sup>2</sup>C)

##### **« Active MCLR comme VCC » :**

Permet d'activer la tension de programmation +13V lors de la programmation d'un composant 24Cxx ou d'une carte à bus I<sup>2</sup>C. Ne cochez pas cette case car vous risquez de griller le composant ou la carte à puce. Cette option n'est à utiliser qu'en connaissance de cause pour faire des tests éventuellement.

##### **« Active page write » :**

Permet d'activer le mode « Page write » permettant de programmer plus rapidement une EEprom 24Cxx. Certaines marques d'EEPROMs 24Cxx ne supportent pas ce mode de fonctionnement. Dans ce cas décocher cette case mais la programmation sera moins rapide.

##### **« Adresse hardware » :**

Normalement sur 0. Avec le programmeur SER-01, suivant la configuration des cavaliers A0, A1, A2 il est possible de sélectionner une autre valeur de cette adresse pour la programmation du composant 24Cxx.

**« Configuration\Options\Programmation » :**

Permet d'effectuer la vérification de la bonne programmation d'un composant après l'avoir entièrement programmé ou durant la programmation du composant. Si aucune case n'est cochée, le composant ne sera pas vérifié et dans ce cas il n'y aura jamais de message d'erreur, que le composant soit bien ou mal programmé.

**« Configuration\Options\Raccourcis » :**

Permet de sélectionner 4 raccourcis (CTRL+F1, CTRL+F2, CTRL+F3, CTRL+F4) pour accéder plus facilement à une fonction.

**« Configuration\Options\Copier&coller » :**

NC

**« Configuration\Options\Smartcard » :**

Permet de configurer la programmation des cartes à puces WaferCard, GoldCard et SilverCardII.

**« Port » :**

Sélectionner le port série où est branchée votre interface Phoenix/smartmouse. Si vous utilisez les programmeurs CAR-03 ou le CAR-04 il faut sélectionner le même port série que celui sélectionné dans le menu « Configuration\Hardware ».

**« Inverse le Reset (smartmouse) » :**

Si votre programmeur CAR-04 ou CAR-07 est configuré en mode « Phoenix » ne cochez pas cette case.

Si votre programmeur CAR-04 ou CAR-07 est configuré en mode « Smartmouse » cochez cette case.

**« Protocol » :**

Icprog supporte 2 types de Loader, Multimac et Secanix. Sélectionner le mode Multimac pour les cartes à puces.

**« Composant » :**

Sélectionner le composant qui se trouve sur votre carte à puce. Pour les PIC16F876 ou les PIC16F877 sélectionner 16F84.

**« Fréquence » :**

Sélectionner la fréquence sur laquelle fonctionne votre programmeur : 3,579MHz ou 6,000MHz.

**« Enable DTR as Vcc » :**

Cette case est cochée normalement.

**« Configuration\Options\Language » :**

Permet de sélectionner votre langue favorite pour l'utilisation du logiciel.

**« Configuration\Options\Shell » :**

NC

**« Configuration\Options\Misc » :**

**« Priorité » :**

Permet de définir la priorité du logiciel par rapport aux autres logiciels fonctionnant en multitâches sous Windows. En général utiliser le mode « normal ». Utiliser le mode « haute » pour que ICprog soit prioritaire par rapport aux autres logiciels.

**« Active Driver NT/2000/XP » :**

Sous Windows 95/98/ME cette option n'est pas accessible. Sous Windows NT/2000/XP cocher cette case. Vérifier dans ce cas que le fichier « ICprog.sys » se trouve bien dans le même répertoire que ICprog.exe.

Pour WindowsXP il faut également sélectionner le fichier ICprog.exe et cliquer avec la touche droite de la souris. Lorsque le cadre apparaît, allez dans le menu « Propriétés ». Sélectionner l'onglet « Compatibilité » et cocher la case située dans le cadre « Mode de compatibilité ». Dans ce même cadre sélectionner « Windows 2000 » dans le menu déroulant.

**« Active Vcc Control pour JDM » :**

Ne pas cocher cette case.

**« Mettre toutes les sorties au niveau haut » :**

Cette fonction permet de mettre toutes les sorties du port parallèle au niveau haut lorsque le port série est utilisé et de mettre toutes les sorties du port série au niveau haut lorsque le port parallèle est utilisé. Cette fonction sert uniquement lorsque l'on utilise un programmeur spécial ayant à la fois le port série et le port parallèle de connecter sur le PC.

**« Configuration\Smartcard(Phoenix) » :**

Cocher cette option si vous utilisez les programmeurs CAR-02, CAR-03, CAR-04 ou CAR-07, permettant ainsi d'accéder aux modes Smartmouse/Phoenix pour la programmation de l'EEPROM 24Cxx sur votre carte à puce. Attention, si vous programmer des EEPROMs 24Cxx avec le programmeur PIC-01, ATM-01 ou SER-01 il faut décocher cette option, sinon un message d'erreur apparaîtra.

**« Configuration\Effacer Configuration » :**

Permet de réinitialiser la configuration et les options du logiciel comme lors de la première utilisation.

**« Commande\Tout lire » :**

Permet de lire le contenu d'un composant dans le buffer. La lecture portera aussi bien sur les codes de programme, que sur les données de l'EEPROM, que sur le contenu des registres de configurations.

Si le composant est protégé en lecture, le buffer n'indiquera pas les bonnes valeurs à l'écran et affichera des 0. Dans ce cas s'il s'agit d'une mémoire Flash le composant pourra être effacé et reprogrammé normalement.

Si vous voulez sauvegarder le contenu d'un composant sur disque, vous devez d'abord utiliser cette fonction, puis utiliser la fonction "Enregistrer Sous".

Si vous voulez faire une copie d'un composant, vous devez d'abord utiliser cette fonction pour charger le buffer, changer le composant, puis utiliser la fonction "Command\Tout Programmer".

**« Commande\Tout Programmer » :**

Permet de programmer le contenu d'un composant à partir du buffer.

La programmation portera aussi bien sur les codes de programme, que sur les données de l'EEPROM, que sur le contenu des registres de configurations. Dans ce cas assurez vous bien que les registres de configuration sont bien positionnés avant de lancer cette fonction.

**« Commande\Programmer Configuration » :**

Permet de programmer uniquement les registres de configuration du composant.

**« Commande\Tout Effacer » :**

Permet d'effacer le contenu du composant.

Tous les bits dans ce cas sont remis au niveau logique 1, (hFF). Cette fonction n'est pas toujours disponible car certains composants ne sont programmables qu'une fois.

**« Commande\Test de virginité » :**

Permet de faire un test de virginité pour vérifier que le composant est vide.

Si le composant est vierge ou effacé tous les bits de la mémoire seront au niveau logique 1 (FF...). Cette fonction est à utiliser avant toute programmation car il n'est pas possible de programmer un composant correctement si celui-ci n'est pas vierge ou n'a pas été effacé préalablement. En effet une programmation met certains bits au niveau logique 0, l'inverse n'étant possible que si le composant peut être effacé à l'aide de la fonction "Command\Tout Effacer".

**« Commande\Vérifier » :**

Vérifie le contenu du composant avec celui du buffer. Un message d'erreur apparaît en cas de différence.

**« Commande\Assistant Smartcard » :**

Assistant permettant la programmation d'une carte à puce, pas à pas, suivant la procédure suivante.

Avec le CAR-04 et le CAR-07, lorsque le logiciel demande de déconnecter le programmeur de PIC et de connecter le programmeur Phoenix, il vous suffit de déplacer le cavalier se trouvant sur le programmeur, de la position JDMprog à la position Phoenix et vice versa.

**Procédure :**

Vérifier que le cavalier situé sur le CAR-04 ou CAR-07 est bien sur la position « JDMprog ».

Insérer la carte à puce, contact vers le bas, dans le lecteur puis mettre le programmeur CAR-04 ou CAR-07 sous tension.

Cliquer sur « Continuer ». Le Loader permettant l'accès à l'EEPROM se charge automatiquement dans le PIC16F84.

Déplacer le cavalier situé sur le programmeur CAR-04 ou CAR-07, sur la position « Phoenix ».

Cliquer sur « continuer ». Sélectionner le fichier pour l'EEPROM 24LC16B. Cliquer sur Ouvrir.

Cliquer sur « Continuer ». Le programme se charge dans l'EEPROM 24LC16B.

Remettre le cavalier sur la position « JDMprog ».

Cliquer sur « continuer ». Sélectionner le fichier pour le PIC16F84. Cliquer sur Ouvrir.

Cliquer sur « Continuer ». Le programme se charge dans le PIC16F84.

**« Outil\CRD upload » :**

Permet de convertir le programme afficher dans le buffer en code assembleur.

**« Voir\Assembleur » :**

Permet de convertir le programme afficher dans le buffer en code assembleur.

**« Voir\Voir en Hex » :**

Permet de revenir à l'affichage en hexadécimal du buffer.

**« Voir\Emplacement composant » :**

Permet de voir le positionnement du composant sélectionné sur le support.

Ce menu n'est pas utile avec les programmeurs SEIT car tous les supports tulipes sont existant sur le programmeur et il n'est donc pas nécessaire d'implanter de façon particulière ou avec un adaptateur un composant sur les supports.

**« Voir\Information composant » :**

Permet d'obtenir quelques renseignements sur le composant sélectionné :

Nom, type, taille de la mémoire, architecture mémoire, nombre de pattes.

**« Voir\Rafrachir » :**

Permet de réinitialiser l'écran. A utiliser l'été en cas de forte chaleur uniquement !