

Installation de piq18-algocode sous Windows^(TM)

Version 1.1

I. Présentation :

piq18-algocode est un logiciel qui permet de créer le fichier ".hex" indispensable à la programmation des pics du constructeur **Microchip**^(TM), et ce, uniquement de façon graphique, à l'aide d'ordinogrammes.

Comme son nom l'indique, **piq18-algocode** se limite à la famille 18 de **Microchip**^(TM) et il a été développé en langage **CPP** à l'aide du framework **Qt4**^(TM) jusqu'à sa version 1.0. Il est dorénavant développé en langage **Python3** à l'aide du framework **PyQt4**. La version 1.1 introduit l'usage de greffons pour les composants. Les versions suivantes proposeront également des greffons pour étendre les possibilités de **piq18-algocode**.

I.1. Caractéristiques :

- Ouverture d'un projet de base structuré d'emblée en un corps principal et un sous-programme d'initialisation du microcontrôleur utilisé (l'initialisation des variables se fait à leur déclaration).
- Gestion graphique des bits de configuration des **pic18**.
- Gestion de la configuration initiale des ports.
- Visualisation de l'occupation des broches des ports.
- Chaque feuille de l'organigramme (appelée **algorigramme**) est un sous-programme en soit, qui peut être importé et exporté, afin par exemple de constituer une bibliothèque graphique. Ce sous-programme se traduit en langage **C** par une fonction. Les variables de retour y sont gérées. Un commentaire peut y être ajouté. Le prototype **C** de la fonction créée est affiché.
- En plus de l'importation et de l'exportation de sous-programmes, la copie dans le presse-papier et la restitution depuis le presse-papier permet l'échange d'un bout d'organigramme entre deux projets, et ce, sans perte de données, ce que ne font pas d'autres logiciels concurrents. Il faut néanmoins veiller à la gestion des variables utilisées dans les parties transférées pour garantir leur déclaration et l'unicité de leur noms.
- Nombreuses vérifications effectuées en cours de création des **algorigrammes** et avant compilation.
- Gestion de nombreux types de données (octets, mots de 16 ou 32 bits, signés ou non, voir plus bas les restrictions sur le type **float**). Les données numériques peuvent être écrites en décimal, hexadécimal (**0xValeur**) ou binaire (**0bValeur**). Elles peuvent être également un **caractère** ('a' par exemple) ou une **chaîne de caractères** (" ceci est une chaine de caracteres " par exemple).
- Gestion de la portée des variables : une variable peut être déclarée locale à un sous-programme ou globale.
- Gestion des variables simples, des tableaux et des pointeurs. Les tableaux de pointeurs ne sont pas gérés cependant.
- Édition et recherche avancée des variables, sous-programmes et registres utilisés.
- Suppression possible des variables inutilisées.
- Outil de vérification de la cohérence du projet.
- Nombreuses bibliothèques intégrées.
- Ajout aisé de bibliothèques personnelles (accessibles via le **pictogramme** " composants " qui donne accès aux prototypes des fonctions qu'elles intègrent).
- Processus de compilation très rapide.
- Entièrement libre.

I.2. Limitations :

Elles sont dues pour la plupart au fait que le compilateur utilisé (**cpik**, de Alain Gibaud) soit en cours de développement (mais très prometteur).

- Le type **float** n'est pas encore géré au niveau du compilateur (donc, cette option est invalidée pour l'instant).
- **piq18-algocode** encode de façon graphique à peu près toutes les structures connues des organigrammes et du langage **C**, à part : les énumérations, l'opérateur **switch case** (à remplacer facilement par des test successifs), l'instruction **goto**.
- Comme mentionnée plus haut, la gestion des variables peut poser problème. Bien que **piq18-algocode** gère leur visibilité, il suppose tout de même l'unicité des noms des variables (et des sous-programmes). Or, celle-ci peut être entachée lors de l'importation d'un sous-programme ou d'un morceau d'**algorigramme** par collage. Il faut donc veiller tout particulièrement à cela, notamment en évitant l'emploi de noms de variables (et de sous-programmes) génériques pour utiliser plutôt des noms spécifiques à l'application en cours de développement. C'est ce qui rend d'ailleurs la création de bibliothèques graphiques difficile. Néanmoins, vous disposez de nombreux outils : outil de gestion de ces variables (création, modification, suppression), outil de recherche de variables et pour chaque sous-programme, outil de visualisation des variables utilisables.
- **piq18-algocode** ne simule pas le fonctionnement de l'organigramme créé. Pour cela, il faut utiliser un logiciel tiers avec le fichier "**.hex**" obtenu, par exemple la suite **PROTEUS** de **Matrix Multimedia**^(TM) qui, dans sa version VSM, permet de simuler un pic.

II. Installation :

L'installation de **piq18-algocode** doit se faire selon le schéma suivant :

- 1) Installer **Python3** et **PyQt4**.
- 2) Installer **piq18-algocode** et les outils qu'il utilise (la suite **gputils** et **cpik**).
- 3) Associer l'extension "**.pac**" à l'application **piq18-algocode**.
- 4) Configurer **piq18-algocode**.

II.1. Installation de Python3 et PyQt4 :

Il suffit d'aller sur les sites officiels qui les maintiennent et de récupérer un installateur adapté à votre PC (des versions 32 bits et 64 bits existent).

Prenez soin de toujours prendre la dernière version en cours.

- **Python3** : <http://www.python.org/>.
 - **PyQt4** : <http://www.riverbankcomputing.com/software/pyqt/>.
- ➔ Installez d'abord **Python3** puis **PyQt4** (qui doit correspondre à la version de Python3 choisie).

II.2. Installation de piq18-alogocode et des outils qu'il utilise :

piq18-alogocode utilise des outils externes pour assurer la compilation de ses projets :

- **cpik** : compilateur C pour microcontrôleurs **Microchip**^(TM) **PIC18**. Auteur : Alain Gibaud (<http://pikdev.free.fr>). Outil qui lui-même utilise les outils suivants :
- **gpasm** (assembleur pour PIC, outil de la suite **gputils** : <http://gputils.sourceforge.net/>).
- **cpp** (préprocesseur C).

Avertissement sur la version de cpik:

La version **0.5.3** de **cpik** améliore énormément ce compilateur.

- Elle apporte notamment une nouvelle possibilité d'optimisation du code que je souhaite utilisée.
- Aussi, je limite l'usage de **cpik** avec **piq18-alogocode** : seule la version 0.5.3 et les suivantes peuvent être utilisées.

Pour simplifier leur installation, une archive fournit ces outils pour **Windows**^(TM) (**freepic.zip**).

- Après décompression de cette archive, copier simplement le répertoire **FreePic** obtenu à un emplacement simple.

Récupérez ensuite l'archive des sources de **piq18-alogocode** et

décompressez-la. Dans le répertoire obtenu, lancez le script **piq18_install.py** qui va y créer un sous-répertoire **/piq18-alogocode**.

- Ce répertoire est à copier en intégralité dans le répertoire racine de **freepic**. Vous devez arriver à la fin à une installation semblable à celle de la **figure 1**.

II.3. Association de l'extension ".pac" à l'application piq18-alogocode.

Les fichiers projet de **piq18-alogocode** ont une extension ".pac".

- Cette étape peut être passée. Vous pouvez attendre d'avoir créé un fichier ".pac" (voir plus loin).

- 1) Lancer l'explorateur en appuyant simultanément sur les touches **Windows** et **E**. La fenêtre de la **figure 2** apparaît.
- 2) Choisir l'onglet "**Outils**" et le menu "**Options des dossiers...**". La fenêtre de la **figure 2** apparaît, où on a déjà choisi l'onglet "**Types de fichiers**".
- 3) Cliquez sur le bouton "**Nouveau**". La fenêtre de la **figure 3** apparaît.
- 4) Entrez "**pac**" comme extension et vérifiez que le type associé est bien <**Nouveau**>. Valider.
- 5) Dans la fenêtre de la **figure 3**, rechercher la nouvelle extension créée.
- 6) La sélectionner et appuyer sur le bouton "**Modifier**".
- 7) Rechercher alors le fichier **piq18-alogocode.bat** dans le sous-répertoire **.../freepic/piq18_alogocode/**.
- 8) Valider.

Figure 1.

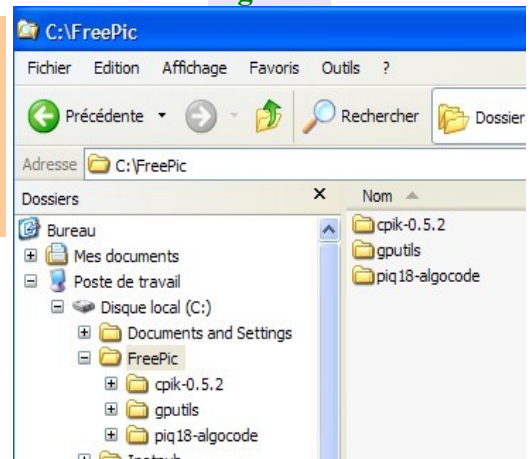


Figure 2.

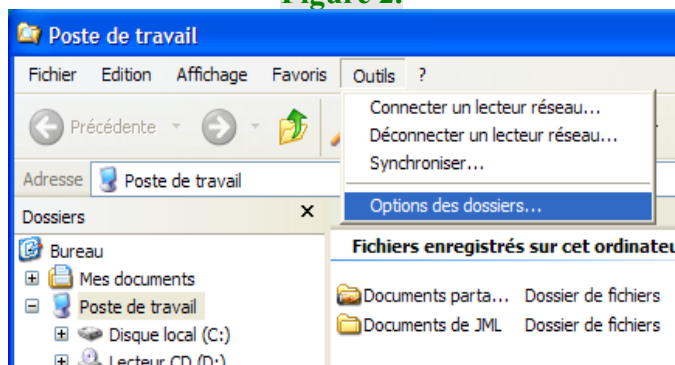


Figure 3.

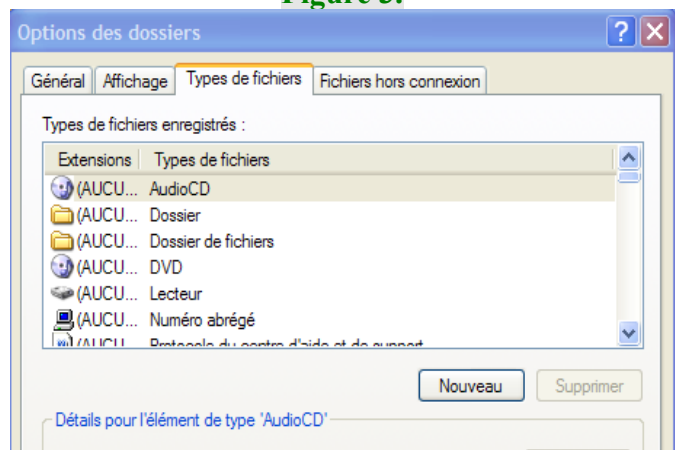
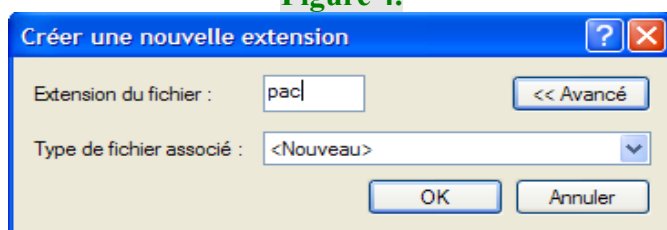


Figure 4.



II.4. Configurer piq18-algocode :

- 1) Lancer **piq18-algocode** en double-cliquant sur le fichier `.../freepic/piq18_algocode.bat`.
→ Le mieux, pour la suite, est de créer un raccourci sur cet exécutable, ou de double-cliquer directement sur un projet **piq18-algocode** (fichier **piq18-algocode** d'extension " **.pac** ") pour le lancer.
- 2) Renseigner les informations qui sont demandées à l'ouverture de la fenêtre de configuration qui s'ouvre alors (**figure 5**). Valider. La fenêtre se referme alors. Relancer **piq18-algocode**.
- 3) Le programme doit alors s'ouvrir sur une fenêtre de gestion de projets (**figure 6**). La renseigner en mode " **Créer un nouveau projet** ". Valider.
- 4) La fenêtre principale de **piq18-algocode** s'ouvre alors avec une fenêtre de configuration du pic (**figure 7**). La renseigner maintenant ou plus tard (onglet " **Projet** " (**figure 8**), menu " **Configurer...** ").
- 5) Dans l'onglet " **Projet** " (**figure 8**), choisir le menu " **Cartographie des registres spéciaux** ". Vérifier que vous avez bien une fenêtre semblable à celle de la **figure 9**, ce qui indique que l'installation a été correctement effectuée.
- 6) Enregistrer éventuellement ce projet (onglet " **Fichier** " et menu " **Enregistrer** ").

Si vous avez créé un fichier ".pac" :

- Vous pouvez maintenant associer l'extension ".pac" à **piq18-algocode** si vous ne l'avez pas fait précédemment. Pour cela :
- Cliquer-droit sur ce fichier et choisir " **Ouvrir avec...** ". Sélectionner alors le programme `.../freepic/piq18-algocode.bat`. Cocher la case qui permet d'obtenir une association définitive (**figure 10**).

Figure 5.

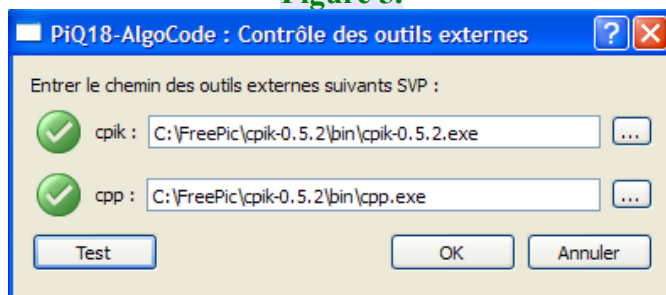


Figure 6.

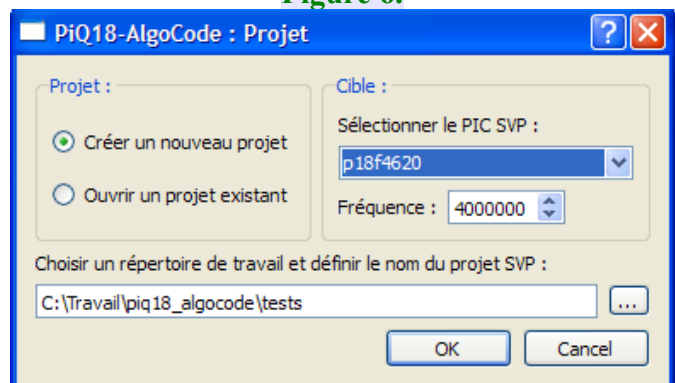


Figure 7.

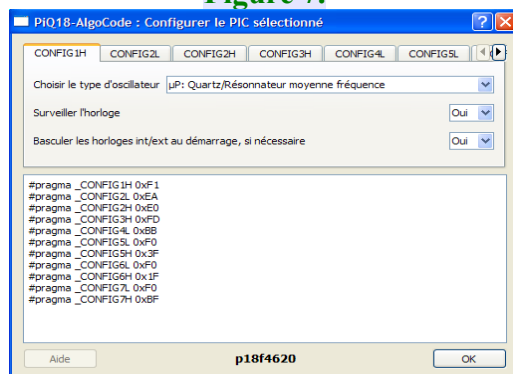


Figure 8.

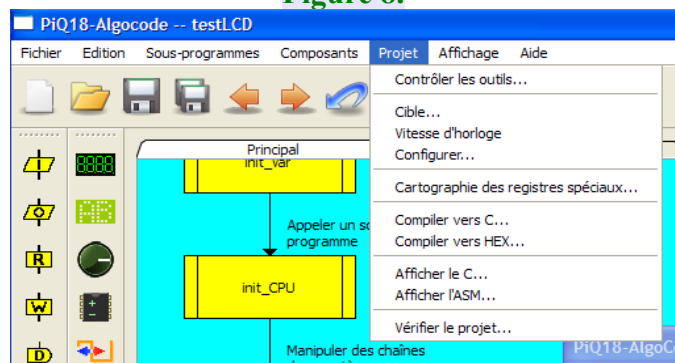


Figure 9.

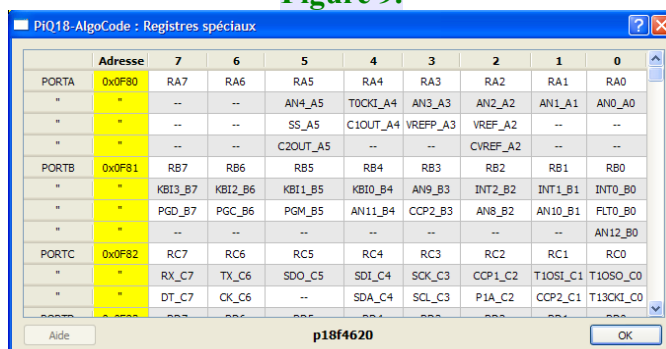


Figure 10.

