

Installation de piq18-algocode sous Linux

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**.

→ **Remarque** : La suite de ce document suppose que vous ayez une expérience de la compilation de sources en Linux. Si ce n'est pas le cas, les sites cités indiquent toujours comment compiler les sources qu'ils proposent. Il suffit de chercher au bon endroit et de prendre son courage à deux mains !

II.1. Installation de Python3 et PyQt4 :

Il s'agit du point le plus délicat car, à ce jour, peu de plateformes **Linux** proposent des paquets " tout fait " : **Python** est livré généralement en version 2.x et il n'y a pas, à ma connaissance, de paquet pour **PyQt4**. De toute façon, il vaut mieux utiliser les dernières versions de ces logiciels.

→ Il faut donc récupérer les sources et les compiler soi-même (cycle "**configure**, **make**, **make install**" en ligne de commande).

Ces sources se trouvent sur les sites officiels qui les maintiennent (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).

Remarques importantes :

- Une fois **Python3** installé, créer un lien en **usr/bin** vers le binaire **/usr/local/bin/python3**.
- Pour installer **PyQt4**, il faut déjà installer **Qt4** (<http://qt.nokia.com/>) puisqu'il va en compiler les sources. Le mieux est peut être de récupérer le kit de développement (sdk) qui intègre les sources de **Qt4** et l'IDE **QtCreator**.
- Faites attention lors de l'installation du sdk **Qt4** : le répertoire par défaut est le répertoire **/home** du compte sur lequel vous vous êtes logué. **Qt4** et **QtCreator** ne pourront alors être utilisés que sur ce compte ! Pour régler ce problème, je l'ai installé chez moi en **/usr/local/QtSdk**.
- Pour installer **PyQt4**, il faut également installer **Sip** auparavant (sources à la même adresse que **PyQt4**).
- Une fois **Python3** et **PyQt4** installés, le sdk **Qt4** peut être supprimé normalement si vous n'en voulez pas.

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

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

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

Pour l'installation de cpik :

- ➔ Télécharger la dernière version de **cpik** sur le site de Alain Gibaud, son créateur : <http://pikdev.free.fr>.
S'il n'y a pas de paquet **Debian** satisfaisant à vos attentes (pas de version 64bits proposée), il faudra compiler les sources.

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-alcocode** : seule la version 0.5.3 et les suivantes peuvent être utilisées.

Pour l'installation de gpasm:

Normalement des paquets sont proposés pour la suite **gputils** qui l'intègre.

Pour l'installation de c++ :

c++ est intégré normalement d'office dans toute plateforme **Linux**.

L'installation de piq18-alcocode est quand à elle fort simple :

Récupérer l'archive de ses sources et la décompiler. A la racine du répertoire obtenu, se trouve un fichier Python3 nommé **piq18_install.py**. Il suffit de le lancer en ligne de commande et en mode administrateur :

En ligne de commandes, taper :

- A) `cd <chemin des sources piq18-alcocode>`
- B) `sudo python3 piq18_install.py` (demande du mot de passe administrateur après validation)

L'installation se fait entièrement en **/usr/share/piq18-alcocode**. Un raccourci est créé dans le menu de lancement, normalement sous la rubrique "programmation".

III. Association de l'extension ".pac" à l'application piq18-algocode.

A l'installation de **piq18-algocode**, un script **sh** qui porte son nom a été créé dans son répertoire d'installation (**/usr/share/piq18-algocode/**).

- ➔ Associer l'extension « .pac » qui définit les projets **piq18-algocode** à ce script. Peut-être est-il plus facile pour cela d'attendre la création de l'un de ces projets par l'étape suivante.

IV. Configuration de piq18-algocode :

- 1) Lancer **piq18-algocode** en utilisant son raccourci (menu "programmation") ou en double-cliquant sur son script de lancement (chemin **/usr/share/piq18_algocode**).
- 2) Si tous les outils ont été correctement configurés, il ne doit pas y avoir de message d'avertissement qui s'affiche. Une fenêtre de gestion de projets s'affiche (cf. **figure 1**).
- 3) Vérifier que la liste des pics proposée n'est pas vide, ce qui indique sinon une mauvaise installation en **/usr/share/piq18-algocode**.
- 4) Vérifier également et pour les mêmes raisons que la fenêtre qui s'ouvre sur sélection du menu «**Projet**» + «**Cartographie des registres spéciaux**» est correctement remplie (cf. **figure 2**).
- 5) Choisir un répertoire de travail et y enregistrer le fichier ouvert ("**Fichier**" + "**Enregistrer**" ou raccourci "Ctrl + S").

Figure 1.

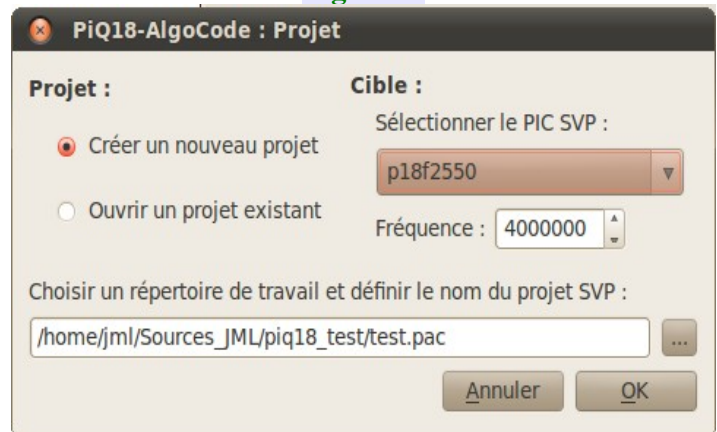


Figure 2.

