

LogiPic, va vous permettre de programmer vos Pics (Microchip) de la très populaire série 16Fxxx et 12Fxxx, de manière très très simple. Encore plus simple que le Basic, car vous n'écrirez plus une ligne de code, tout se fera par la manipulation d'icônes. Il s'agit d'une programmation de type organigramme.

Version actuelle V1.50

## **Première prise en main**

Pour la première utilisation de LogiPic, Il est tout d'abord nécessaire d'indiquer où se trouvent les programmes MPASMWIN et IcProg. Le premier permet de transformer le code assemblé en code hexadécimal et le deuxième de transférer ce code dans la Pic via un programmeur. Si vous ne possédez pas ces deux programmes, vous pouvez les télécharger sur le site : <http://idmax.free.fr/>.

Options → Chemin d'accès → MPASMWIN

Indiquer l'emplacement du programme et cliquer sur ouvrir

Options → Chemin d'accès → ICPROG

Indiquer l'emplacement du programme et cliquer sur ouvrir

Voilà, maintenant LogiPic sera en mesure de travailler avec ces programmes.

## **Premier programme**

Fichier → Nouveau ou raccourci CTRL-N :

Cliquer sur la petite flèche de la boîte multi choix et indiquer le Micro contrôleur choisi.

Valider en cliquant sur Ok.

Une fenêtre s'ouvre alors, intitulée Start 16F628 par exemple.

Comme dans toute nouvelle fenêtre, vous y voyez deux modules.

L'un intitulé 'Start 16F628' et l'autre 'Saut module: N° 1'. Chacun porte un numéro d'ordre, d'apparition à l'écran.

La première, représente bien sûr le début du programme 'le START en assembleur', vous l'aviez compris.

Le second est bien entendu la fin du programme, c'est-à-dire 'le goto, suivi du end en assembleur'. Il s'agit donc d'un Saut faisant référence à un numéro d'ordre, vous en comprenez maintenant la raison d'être.

Vous aviez peut-être déjà remarqué un léger rectangle bleu situé sur le dernier module. En fait celui-ci vous indique le module qui est en cour de sélection. L'ajout d'un futur module se fera sous celui qui est sélectionné. Un simple clic, sur un module fera apparaître le rectangle bleu sur celui-ci.

Options → Options : (configuration du pic)

La fenêtre, qui apparaît contient de multiples onglets.

Cliquez sur l'onglet 'Directives', cet onglet vous permet de régler le mot de configuration du pic. Parfois appelé 'controlword' ou 'fusibles' etc.....voici mes réglages standard :

Oscillator Selection :            permet de configurer le type d'oscillateur, quartz, RC...  
Accès à la mémoire flash :    permet de protéger l'accès à la mémoire flash  
Code protection :                permet d'interdire la lecture d'une partie ou la totalité du code.  
WatchDog :                        mise en ou hors service  
Power-up timer :                mise en ou hors service du démarrage temporisé

Exemple de paramétrage pour un 16F628 :

Sans protection  
LVP\_OFF  
Watchdog hors service  
Reset tension hors service  
Démarrage temporisé (Vous n'êtes pas à une seconde près pour le démarrage.)  
MCLR\_OFF  
INTRC\_OSC\_CLKOUT (oscille tout seul sur 4Mhz)

Le 16F628 employé par l'auteur de cette documentation comme modèle, présente entre autres nombreux avantages, celui de pouvoir osciller ***tout seul*** sur une fréquence relativement stable de 4 Mhz. Ni quartz, ni condensateur. Sous réserve de laisser libre les pins RA5 et RA6. Cette valeur est suffisamment stable pour la majorité des programmes, excepté, bien sûr ceux du type chronomètre.

Cliquez maintenant sur l'onglet 'process' dans la case fréquence du quartz vous devriez voir 4.0000 Mhz, à titre indicatif, cette valeur ne fixe pas du tout la fréquence de l'oscillateur interne, mais permet, à LogiPic de savoir quelle est la base de temps, pour le calcul des tempos uniquement.

**Option → Option → Comparateurs :**

Dès maintenant, si vous n'utilisez pas les comparateurs, que se soit en toute connaissance de cause, ou par ignorance de la chose, il est impératif pour le 16F628 de positionner le registre CMCON sur la valeur 111 (7 en décimale), soit inactivation des comparateurs.

Cliquez sur Ok pour valider les valeurs.

Prenons dès maintenant les bonnes habitudes

Fichier → enregistrer sous, sélectionner à la manière windows votre répertoire puis Tapez un nom suivi de 1 et cliquez sur Ok. Votre travail est enregistré, pensez à le faire régulièrement, et de temps en temps incrémentez, le petit chiffre, de manière à avoir un historique de votre travail. Je ne reviendrais plus sur ce point de bon sens par la suite.

Cliquez sur le premier module 'Start 16F628'.

**Modules → Sortie → état d'une sortie**, choisissez par exemple le PortA, bit 7, sélectionnez 1 ou 0 pour indiquer l'état de la sortie, validez par OK.

Un nouveau module est apparu entourée de bleu. Numéro d'ordre 2.

Le bit 7 du port A est mis dans l'état indiqué.

**Modules → Sortie → inversion d'une sortie**, choisir le même Port que précédemment. Numéro d'ordre 3.

Le bit 7 du port A est inversé quel que soit son état précédent.

[Double cliquez sur le module](#) 'Saut Modules : N°1' cliquez dans la case blanche, et taper 3, maintenant, le saut se fait avant le module 3. (Vous pouvez aussi utiliser les petites flèches).

[Modules](#) → [Temporisation](#) → [Définir](#), ne rien toucher et cliquer sur Ok. Vous venez de créer une tempo de valeur 1000000µs soit 1S. Pour le vérifier :

Cliquez une fois sur le module 'inverse Port A7'.

[Modules](#) → [Temporisation](#) → [Insérer](#), puis cliquer sur Ok.

Vous devriez avoir entre le module 'inverse PortA7' et le module 'Saut Modules : N 3' un nouveau module 'Tempo T\_1000000µs'.

L'écriture est terminée.

[Compiler](#) → [Assembler](#)

[Affichage](#) → [Programme](#), pour voir le code source, que vous n'avez pas eu à écrire.

[Affichage](#) → [Logigramme](#), pour revenir au logigramme.

[Compiler](#) → [Compiler](#), compile le programme dans le répertoire du projet.

Un fichier commençant par votre nom de projet et portant l'extension 'hex' vient d'être créé.

[Compiler](#) → [Programmer](#), il est nécessaire d'avoir installé et paramétré IC-Prog, et d'avoir connecté votre programmeur de pic, pour ces deux derniers éléments, voir le net.

Voilà, vous aussi vous venez de faire clignoter votre première Led avec un Pic. Bienvenue au club.

Une dernière chose

[Option](#) → [commentaire](#) :

Cette page vous permet d'inscrire des explications générales sur le programme, celle-ci, si vous l'avez renseignée, s'ouvrira automatiquement, au chargement du programme.

N'hésitez pas à le faire, dans plusieurs mois quand vous aurez fait plusieurs autres programmes, ces explications vous seront bien utiles.

Ceci met un terme, à la prise en main de LogiPic.

## Les fonctions complémentaires

Un double clic sur un module permet de modifier ses paramètres.

### LES ROUTINES

Pour les débutant, une routine est une partie de programme pouvant se répéter plusieurs fois et comportant généralement plus d'une instruction. Mais ce peut être pour structurer un programme, par exemple une routine initialisation dans laquelle, on met toutes les valeurs de démarrage du début de programme, cela est plus propre.

#### **Routine → Définir Routine :**

Dans la petite fenêtre intitulée 'Etiquette' cliquez dans la case blanche, votre curseur, apparaît à droite du mot Routine appuyez sur underscore (la petite barre sous le chiffre 8 en haut du clavier) puis sans aucun espace complétez, le nom de votre routine. Ex : Routine\_test.

Vous venez de voir apparaître un deuxième onglet.

Le premier module 'Routine\_test' puis le deuxième 'Ret Routine' vous indiquant la fin de la routine. Chacune portant un numéro d'ordre propre à cet onglet.

Revenez sur l'onglet Start16F628, cliquez sur le module précédent l'endroit où vous voulez insérer la routine.

**Routine → Saut routine**, cliquez sur la routine choisie, puis validez, celle-ci sera insérée à l'endroit souhaité.

### LES VARIABLES

#### **Modules → variable → définir :**

Ici on commence par créer la variable nom, et paramètres.

#### **Modules → variable → supprimer :**

ATTENTION, il n'y a aucun contrôle pour la suppression d'une variable, si vous l'utilisez ailleurs, dans une autre page (ou onglet), la suppression sera quand même possible, les problèmes viendront, à la compilation suivante.

Voici ma solution : sauvegarde, suppression, compilation immédiate.

### LES AIDES

Vous disposez, pour chaque module créé d'une page d'aide à renseigner de 50 caractères, pour cela cliquez sur le module que vous souhaitez documenter, bien sûr ce n'est pas le module, mais son usage dans le programme, qu'il faut documenter, puis cliquez sur le module se trouvant dans la barre du haut avec un carré noir et un point d'interrogation noir.

Saisissez votre commentaire et validez. Le commentaire apparaîtra dans la barre du bas lorsque le pointeur de la souris sera positionné sur le module.

### LES SORTIES

Nous avons vu dans la prise en main une méthode pour initialiser les sorties

**Modules → Sortie → état d'une sortie.** Avec cette méthode, LogiPic gère automatiquement les directions des sorties (TRIS\*).

voici la seconde :

#### **Modules → bit → initialise :** (normalement destinée aux registres)

Nommer le registre (PORTx ou CMCON etc....), préciser le bit, indiquer l'état, valider.

Attention : si vous optez pour cette solution pour changer l'état d'une sortie, pensez à initialiser le registre TRIS\* en sortie.

### LES ENTREES (ou test d'un seul et unique bit)

Vous pouvez avoir besoin de connaître l'état d'un bit (une broche ou pin) donc de faire un test.

#### Modules → entrée :

Dans la petite fenêtre intitulée « Test sur une entrée » choisir le bit souhaité et préciser si le résultat vrai (donc égal à 1) est en bas ou à droite. Le test d'égalité se faisant toujours sur un état haut (1).

Personnellement, je préfère la solution suivante, plus générale.

#### Modules → Bit → Test : (normalement destinée aux registres)

Choisir le registre, Port x, choisir le bit x et indiquer si le test se fait sur l'égalité à 0 ou à 1.

Gros avantage de cette méthode, vous pouvez tester un bit, d'un registre interne, ce que ne permet pas le Modules entrée.

Inconvénient de cette méthode, vous ne pouvez pas choisir de positionner le résultat vrai ailleurs qu'en bas.

Faites votre choix en fonction de votre besoin.

Attention : si vous optez pour cette solution pour changer tester une entrée, pensez à initialiser le registre TRIS\* en entrée.

### LES SAUTS

#### Modules → Saut :

Permet tout simplement de faire un saut avant un autre module de la page en cours.

### LES BOUCLES

Bien utile pour exécuter un certain nombre de fois bien précis, une action donnée.

Mais attention il y a une restriction, pour le moment UNE SEULE et unique boucle par page (ou onglet, ou logigramme).

#### Modules → Boucle → Début boucle :

Positionne un module indiquant le début de la boucle tout simplement.

#### Modules → Boucle → fin boucle :

Permet de préciser soit un nombre indiquant exactement la quantité de boucle à exécuter, soit le nom d'une variable symbolisant, le nombre de boucle à exécuter.

## **Traitement sur un registre (ou variable) complet et plus un seul bit du registre.**

#### Modules → Registre/variable → Test :

Vous devez savoir avant de commencer si vous allez tester une variable, ou un registre.

Indiquez le type de test, égalité, supérieur, inférieur ou différent (inégalité)

Préciser à quelle entité vous comparez votre valeur d'origine, encore une fois, valeur, registre ou variable.

Enfin, indiquez si la sortie de la condition vraie doit se faire, en dessous, ou à droite.

#### Modules → Registre/variable → Opération à deux opérandes :

Permet de réaliser des opérations de type :

- addition
- soustraction
- multiplication
- division
- and, or , xor

Multiplication : Si le résultat de la multiplication dépasse 255, la valeur de l'octet de poids fort sera stockée dans la variable V\_tmp.

Division : Le reste de la division est stockée dans la variable V\_tmp.

Modules → Registre/variable → Opération à une opérande :

Permet de réaliser des opérations de type :

- égalité
- not
- incrémentation
- décrémentation
- rotation de bits
- inversion de quartet

## L'usage des afficheurs LCD

Option → Options → LCD :

Ici sera précisé, le câblage réel de votre LCD, si vous ne connaissez rien à la logique d'état des sorties, du micro contrôleur, je vous conseille très fortement de ne pas modifier les choix proposés, mais plutôt de câbler selon cet exemple.

Data	LCD	PIC
bit 0 data1	pin 11	PORTB,4(par défaut)
bit 1 data2	pin 12	PORTB,5(par défaut)
bit 2 data3	pin 13	PORTB,6(par défaut)
bit 3 data4	pin 14	PORTB,7(par défaut)
E	pin 6	PORTB,3 (par défaut)
RW	pin 5	Reliée à la masse
RS	pin 4	PORTB,0 (par défaut)

Modules → Gestion LCD → afficher texte :

Cette partie du menu traite l'intégralité du paramétrage de ces LCD standard, cela représente un soulagement de travail et un gain de temps énormes.

Il ne vous reste plus qu'à effacer, écrire, etc.....

## **Eeprom**

[Modules](#) → [Eeprom](#) :

Ce module permet de lire et d'écrire dans l'Eeprom de la Pic.

Deux méthodes sont possibles :

1. Soit on utilise un identifiant pour spécifier l'emplacement où l'on souhaite lire ou écrire, pour cela il faut définir un nom pour cet emplacement [Modules](#) → [Eeprom](#) → [Définir](#) puis saisissez le nom, insérer, ok.  
Ensuite, [Modules](#) → [Eeprom](#) → [Ecrire/Lire](#) Cliquez sur identifiant et sélectionnez le nom de l'identifiant. Le numéro de mémoire est automatiquement généré par l'application.
2. Soit on sélectionne l'adresse physique et l'on indique la variable qui contiendra la valeur de l'emplacement de la mémoire.

Indiquer si le module sera en lecture ou en écriture.

Dans le cas d'une lecture la valeur de l'emplacement de l'Eeprom sera copiée dans la variable, dans le cas d'une écriture c'est la valeur de la variable qui sera copiée dans l'emplacement de l'Eeprom.

## **Communication série**

[Modules](#) → [communication série](#)

Ce module permet d'envoyer ou de recevoir des données vers un pc ou une autre pic via les pin rx et tx. Il est nécessaire de mettre en service l'USART dans [Option](#) → [Options](#) → [Serial](#) et de paramétrer les options. Je vous recommande de vous rapprocher du datasheet.

Le module de réception bloque le programme jusqu'à ce qu'il reçoit une trame.

## **Eeprom externe I2C**

[Modules](#) → [Gestion I2C](#).

Ce module permet de gérer les Eeprom I2C. Il est nécessaire de renseigner l'adresse physique du composant 1010XXX, l'adresse haute et basse de la mémoire et le nom de la variable à lire ou à écrire.

## **Les interruptions**

Non, non, ne partez pas en courant, vous allez voir c'est simple.

[Modules](#) → [interruptions](#) → [RB0](#) :

La fenêtre option s'ouvre automatiquement, vous ne pouvez rien faire dans la sous fenêtre « priorité » car c'est votre première interruption, nous y reviendrons plus loin.

Pour la sous fenêtre « Options » indiquez votre paramétrage souhaité en fonction de vos besoins.

Validez.

Un nouvel onglet avec une page contenant deux modules vient d'être créé. Le premier indique le début de l'interruption, et le second indique le retour de l'interruption, à ne pas confondre avec le retour de routine, faites bien la nuance.

[Modules](#) ➔ [interruptions](#) ➔ [RB4 à RB7](#) :

La fenêtre option s'ouvre automatiquement, vous n'avez théoriquement rien à préciser pour cette interruption.

Validez.

Un nouvel onglet avec une page contenant deux modules vient d'être créé. Le premier indique le début de l'interruption, et le second indique le retour de l'interruption, à ne pas confondre avec le retour de routine, faite bien la nuance.

[Options](#) ➔ [Options](#) ➔ [interruptions: \(configuration du pic\)](#)

Dans la sous fenêtre priorité, vous avez deux interruptions, avec les boutons UP et Down, vous changez l'ordre d'exécution.

Il existe d'autre interruptions, en particulier sur les timers, et comme pour les LCD tout le travail est fait par LogiPic, il ne vous reste plus qu'à paramétrer, vos timer, à la conditions unique, que vous sachiez de quoi il s'agit, mais là, je vous conseille fortement, pour vous mettre au courant de voir l'excellent cours de BIGONOFF sur le net.

Voilà, vous en savez beaucoup sur l'usage de LogiPic, voué je le pense à un très grand avenir. Le reste, vous le découvrirez comme moi par l'usage, au fur et à mesure de l'évolution du produit.