

Mode d'emploi pour programmer un PIC Le 26/11/2018

ou comment mettre facilement un fichier de type '.hex' dans un PIC

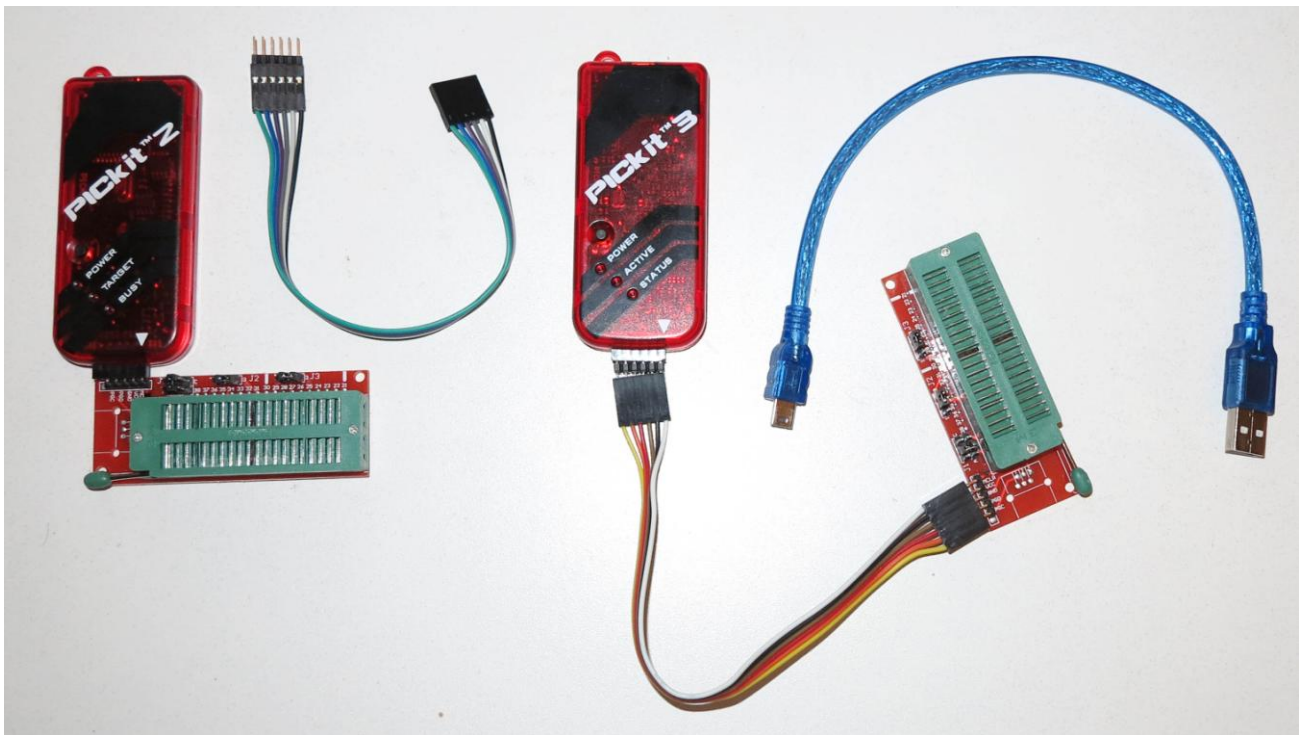
Comment programmer un PIC de type 16F84, 16F628, 16F690, 12F975 d'un montage récupéré sur Internet, à partir du fichier '.hex' ?

Un montage à **PIC** du site UTS2000 vous intéresse, mais la programmation d'un PIC vous semble compliqué, voici comment procéder simplement, en utilisant un PC sous Windows. Les PIC (16F84, 16F628, 16F690, 12F675...) sont fabriqués par **Microchip**. Ce fabricant propose plusieurs produits pour leurs programmations.

1 / Acheter le matériel, le Pickit3

Acheter un boîtier USB **Pickit3** de programmation de PIC. On en trouve sur Ebay.fr, Amazon.fr, Aliexpress.com entre 15 et 25 euros. Ne pas oublier de prendre la version avec le support d'insertion fourni, généralement de couleur verte sur un circuit imprimé rouge.

Le **Pickit4** est mieux, mais nettement plus cher. Le **Pickit 2** est très bien, sauf qu'il ne pourra pas programmer les PIC très récents, mais il pourra quand même programmer tous les PIC utilisés sur le site UTS2000. J'utilise indifféremment les **Pickit2** et **Pickit3** pour mes montages.

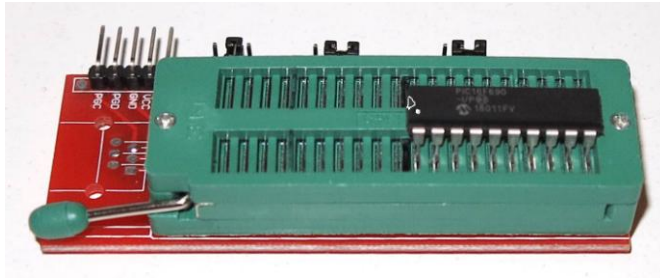


On a donc un **Pickit3**, un câble usb, un câble plat en nappe et un support de circuit intégré à force d'insertion nulle.

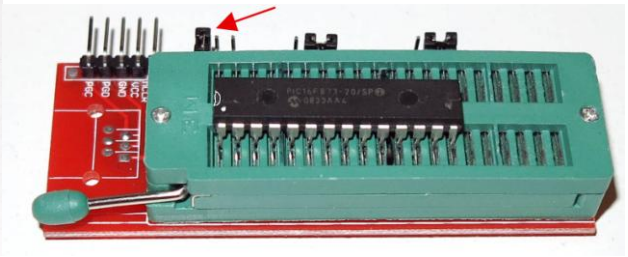
Sur le circuit imprimé, configurer les cavaliers suivant le type de PIC utilisé, en fait suivant le nombre de pattes du PIC ou format du boîtier (*Exemple : DIP28 = Dual In Line 28 = Circuit à 28 pattes*). La notice se trouve sous le circuit imprimé.

- DIP8 = 12F675
- DIP18 = 16F84, 16F88, 16F628
- DIP20 = 16F690
- DIP28 = 16F873 (

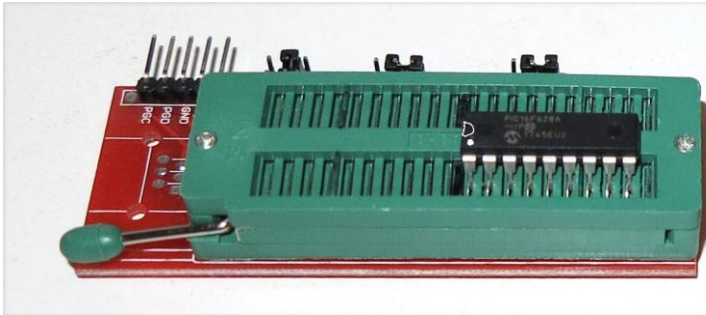
Pour faciliter la mise en place du PIC, j'ai tracé au feutre les séparations entre les pattes 4 et 5, et 10 et 11 du support. Comme cela, je colle les PIC de type DIP8, DIP18 et DIP20 au trait noir.



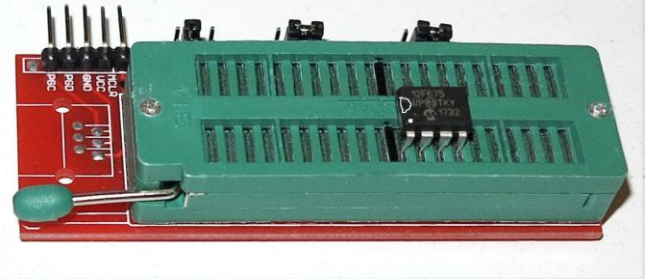
16F690 = J1 en B



16F873 = J1 en A



16F628 = J1 en B



12F675 = J1 en B

Relier le câble plat au **Pickit3**, ou le boîtier directement sur le connecteur du circuit imprimé. La broche n° 1 du boîtier est indiqué par un repère ▼ et correspond au signal 'vpp/MCLR'. La broche n° 6 du **Pickit3** n'est pas utilisée.

Suivre la couleur du câble pour cette broche n°1 et connecter la nappe avec ce fil en face de la broche 'vpp/MCLR' du circuit imprimé.



Sur cette photo, le fil jaune n'est pas relié coté circuit imprimé. Pour me repérer, j'ai dessiné un triangle noir ▼ en face de la broche 'vpp/MCLR' du circuit imprimé.

Ne pas encore placer de PIC sur le support. Brancher le **Pickit3** sur une prise USB de l'ordinateur.

2 / Installer le logiciel 'Pickit3 V3.01' pour envoyer un fichier '.hex' dans un PIC

Ce logiciel est gratuit. Sur le site de [Microchip](http://www.microchip.com), récupérer le fichier :

Sur le site : <https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive>,

dans le paragraphe 'PICkit Archives > Pickit3' situé en base de page, prendre 'Pickit3 Programmer App and Scripting Tool v3.10'.

Ou directement, 'Pickit3 Programmer App and Scripting Tool v3.10' (10 Mo) :

[http://ww1.microchip.com/downloads/en/DeviceDoc/PICkit3 Programmer Application v3.10.zip](http://ww1.microchip.com/downloads/en/DeviceDoc/PICkit3%20Programmer%20Application%20v3.10.zip)

Décompresser le fichier : 'PICkit3 Programmer Application v3.10.zip', puis 'PICkit3 Programmer Application Setup v3.10.zip' et exécuter le fichier d'installation 'Setup.exe'.

Une fois le logiciel installé, brancher le programmeur **Pickit3** sur un port usb, sans mettre de **PIC** dessus. Utiliser un câble usb de bonne qualité, de préférence court sans rallonge usb.

Démarrer le programme : Démarrer > Microchip > **Pickit3 V3.01**.

Mettre à jour le microprogramme du **Pickit3**. Menu : Tools > Download PICkit Operating System, choisir le fichier "**PK3OSV020005.hex**". Attendre la fin de la mise à jour du programmeur.

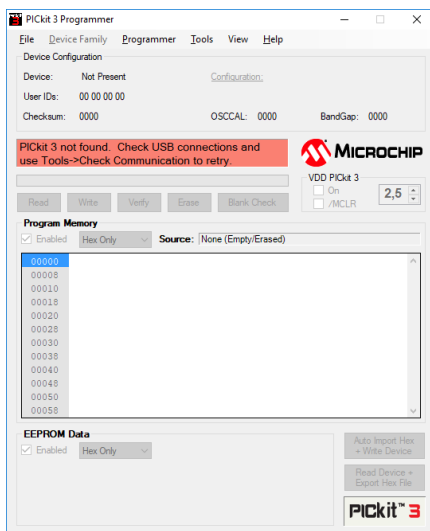
Menu : Tools > Set Unit ID, et lui donner un petit nom à votre programmeur.

Configurer les cavaliers du circuit imprimé suivant le nombre de pattes du **PIC**, comme indique la notice sous le circuit imprimé.

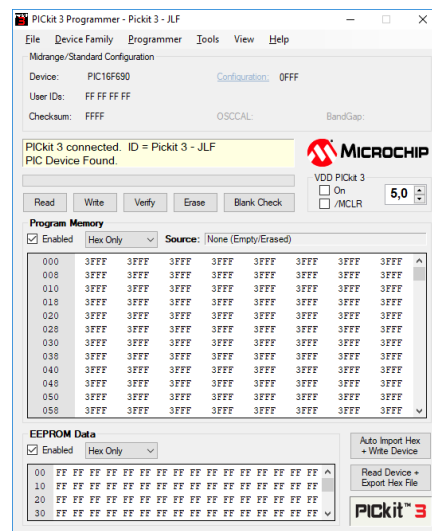
Placer un **PIC** sur le support, et cliquer sur les boutons [Erase], puis [Read].

On a le message "**Done**" une fois ces opérations réalisées. Le PIC est alors prêt à être programmé.

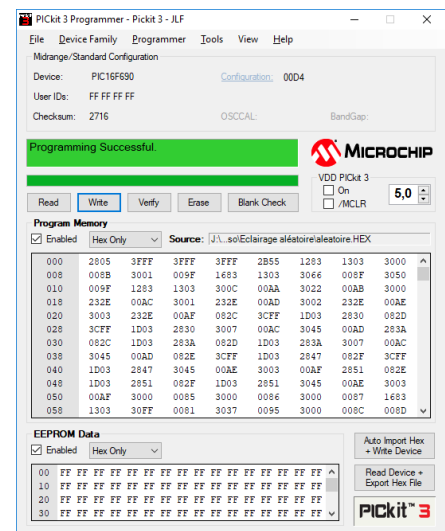
Si l'on a une erreur au moment de quitter ce programme, il faut le configurer pour qu'il se lance automatiquement en mode administrateur sous Windows 10. Dans l'explorateur de fichier, dans le répertoire 'C:\Program Files\Microchip\PICkit 3 v3', clic-droit sur 'PICkit3.exe' > Propriétés, Onglet 'Compatibilité' > Cocher la case : ☒ Exécuter ce programme en tant qu'administrateur.



Sans Pickit de branché



avec Pickit de branché



après écriture du PIC

3 / Programmer le PIC à partir du fichier '.hex'

Configurer les cavaliers du circuit imprimé suivant le nombre de pattes du **PIC**, comme indique la notice sous le circuit imprimé.

Relier le câble plat au **Pickit3**. La broche n° 1 du boîtier est indiquée par un repère ▼ et correspond au signal 'vpp/MCLR'.

Brancher le programmeur **Pickit3** sur un port usb.

Soulever le levier, mettre le **PIC** à la bonne place sur le support et rabaisser le levier

Démarrer le programme : Démarrer > Microchip > **Pickit3 V3.01**.

Cliquer sur les boutons [**Erase**] pour effacer le **PIC**.

Chercher le fichier '.hex', menu : File > Import Hex, parcourir l'arborescence et sélectionner le fichier '.hex' désiré.

Cliquer sur le bouton [**Write**] pour écrire dans le **PIC**.

Au bout de quelques secondes, le PIC est programmé, le retirer du support.

Il faudra à chaque fois effacer le PIC avant d'écrire dedans. On peut faire cette opération des centaines de fois.

4 / Si l'on dispose d'un ancien Pickit2

Le programme est alors différent. Il faudra utiliser le programme **PICKit 2 v2.61** adapté au **Pickit2**.

Sur le site de **Microchip**, récupérer les fichiers :

Sur le site : <https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive>,

en bas de page dans le paragraphe 'PICKit Archives > Pickit2', '**PICKit 2 Software for Windows v2.61**' + '**PICKit 2 Device Firmware v2.32**' (Soumis à la création d'un compte utilisateur).

Ou directement :

PICKit 2 Programmer Application v2.61 (4 Mo) : http://ww1.microchip.com/downloads/en/DeviceDoc/PICKit_2_v2.61.00_Setup_A.zip

PICKit 2 Programmer Application v2.61 avec dotNet A (31 Mo) :
http://ww1.microchip.com/downloads/en/DeviceDoc/PICKit_2_v2.61.00_Setup_dotNET_A.zip

PICKit 2 Device Firmware v2.32 (1 Mo) : <http://ww1.microchip.com/downloads/en/DeviceDoc/FirmwareV2-32-00.zip>

Décompresser le fichier : 'PICKit 2 v2.61.00 Setup A.zip' et exécuter le fichier d'installation 'setup.exe'.

Brancher le programmeur **Pickit2** sur un port usb.

Lancer le programme : Démarrer > Microchip > **PICKit 2 v2.61**.

Si besoin, mettre à jour le microprogramme du **Pickit2**. Menu : Tools > Download PICKit2 Operating System, choisir le fichier "**PK2V023200.hex**". Attendre la fin de la mise à jour du programmeur.

Menu : Tools > Calibrate VDD & Set Unit ID ... Pour calibrer si besoin le programmeur et lui donner un petit nom.

Pour calibrer le programmeur, il faut un voltmètre numérique correct pour mesurer une tension de 5 volts avec au moins 2 chiffres après la virgule (Mesuré entre VDD et GDN en sortie du **Pickit2**). Ensuite, on peut lui donner son petit nom.

Le reste ressemble à l'utilisation du **Pickit3**.

5 / Pour modifier le fichier '.hex' avec MPLAB Tools avant écriture dans le PIC

On a récupéré un fichier source contenant le code en clair, de type `'asm'`. Ce fichier contient le code assembleur en clair, modifiable facilement.

Par exemple, on peut modifier ce fichier pour ajuster des temporisations, ou configurer la table des aiguillages dans les fichiers 'Garage_a_pic_emetteur.asm' et 'Garage_a_pic_recepteur.asm'.

Pour compiler ce fichier assembleur de type '.asm', pour les pic 8 bits, comme les 16F84, 16F628, 16F67, 16F690, 12F675, utiliser le programme gratuit **MPLAB IDE 8.92** de **Microchip**.

Sur le site de [Microchip](https://www.microchip.com), récupérer le fichier :

Sur le site : <https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive>,
en milieu de page dans la partie : MPLAB IDE Archives > 32-bit Windows **MPLAB IDE 8.92** (*Soumis à la création d'un compte utilisateur*).

Ou directement :

MPLAB IDE 8.91 (110 Mo) : http://ww1.microchip.com/downloads/en/DeviceDoc/mplab_ide_8_92.zip

Décompresser le fichier : 'mplab_ide_8_92.zip' et exécuter le fichier d'installation '**Setup.exe**'.

Attendre la fin de l'installation et quitter ce programme.

Démarrer le programme : Démarrer > Microchip > **PMLAB IDE.**

Menu File > Open, choisir par exemple le fichier '**Garage_a_pic_emetteur.asm**'.

Dans la fenêtre d'édition, modifier le texte du fichier.

Par exemple, pour le code '**Garage_a_pic_emetteur.asm**', on peut modifier les nombres '19' sur les lignes suivante:

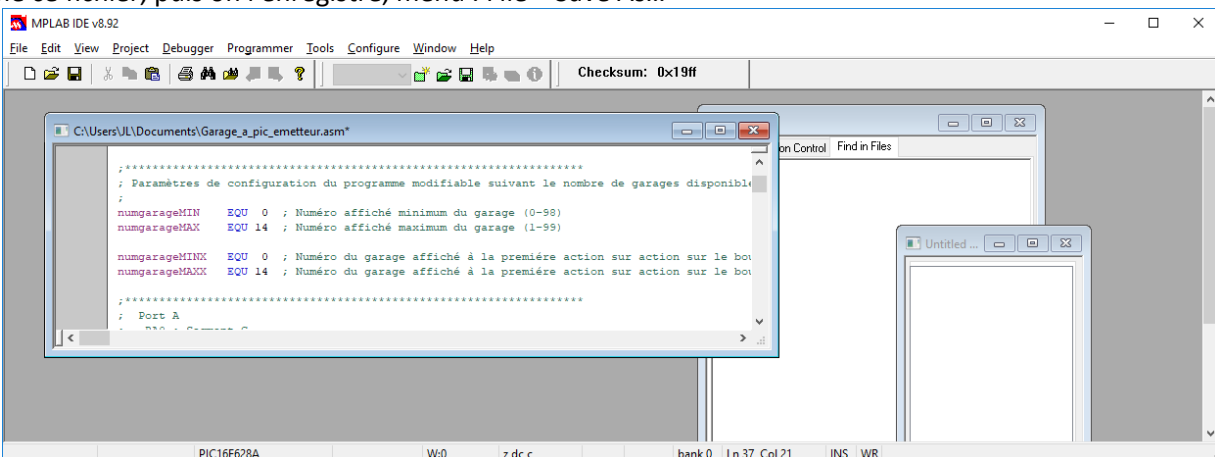
```

; *****
; Paramètres de configuration du programme modifiable suivant le nombre de garages disponibles.
;
numgarageMIN EQU 0 ; Numéro affiché minimum du garage (0-98)
numgarageMAX EQU 19 ; Numéro affiché maximum du garage (1-99)

numgarageMINX EQU 0 ; Numéro du garage affiché à la première action sur action sur le bouton (+)
numgarageMAXX EQU 19 ; Numéro du garage affiché à la première action sur action sur le bouton (-)

```

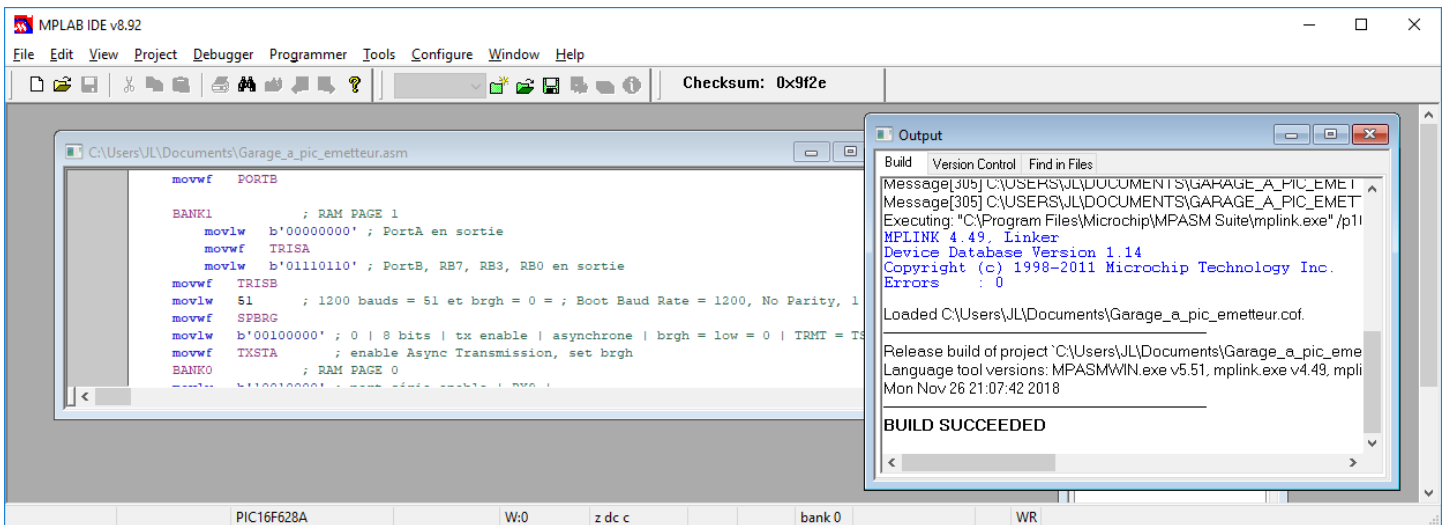
On modifie ce fichier, puis on l'enregistre, menu : File > Save As...



Une fois le fichier modifié, sélectionner la fenêtre d'édition du fichier '**Garage a pic emetteur.asm**'

Pour produire le nouveau fichier '.hex', menu : Project > Quickbuild Garage a pic emetteur.asm.

Si il n'y a pas d'erreur dans le programme, on la le message 'BUILD SUCCEEDED'.



Dans le même répertoire que le '.asm', le programme vient de créer le fichier '**Garage_a_pic_emetteur.hex**'
Il ne reste plus qu'à programmer le **PIC** avec ce fichier.

6 / Installer le logiciel de développement et mise au point gratuit de Microchip

Si nécessaire, pour modifier de façon plus conséquente ou pour créer un nouveau programme, utiliser un environnement de développement plus complexe, mais plus pratique.

Il est conseillé d'utiliser un langage plus évolué comme le Basic ou le C pour écrire de nouveaux programmes. C'est plus facile à écrire et à maintenir, et donc utiliser **MPLAB X IDE**.

Sur le site de **Microchip**, récupérer le fichier **MPLAB® X IDE v5.10** (850 Mo) :
<https://www.microchip.com/mplab/mplab-x-ide>

Son installation et son utilisation sort du cadre de cette petite note d'introduction au PIC. Se référer alors à d'autres sites plus conséquents sur la programmation des PIC.

7 / Divers

D'autres sites sur la programmation des **PIC** :

- UTS 2000 : - http://www.la-tour.info/uts/uts_page13.html
- KUDELSCO.FREE : http://kudelsco.free.fr/prog_pic_usb_V2/logiciel_microchip.htm
- FANTASPIC : <http://www.fantaspic.fr/index.php>
- BIGONOFF : <https://www.abcelectronique.com/bigonoff> (Cours sur les PIC...)

A+