

Module d'animations - Fusée / Incendie

Le 19/06/2021

Montage : Animations de départ de fusée ou de départ d'incendie

Version 1

Ce montage électronique à base de PIC 16F690 de Microchip permet d'animer un réseau ferroviaire, au milieu des trains, des locomotives et des maquettes HO ou N.

Pour recevoir le circuit imprimé, passer par le site <https://ilpcb.com/> idéal pour des circuits de moins de 10 cm x 10 cm. Envoyer le fichier "Aleatoire - CADCAM.zip".

Pour fabriquer ce montage, il faut obligatoirement programmer un circuit intégré, ce qui n'est pas si difficile que cela. Le matériel pour programmer facilement le PIC 16F690 se trouve pour moins de 19 euros. La notice se trouve en annexe à la fin de ce document.

Le programme est fourni au format '.hex' et le code source en assembleur au format '.asm'.

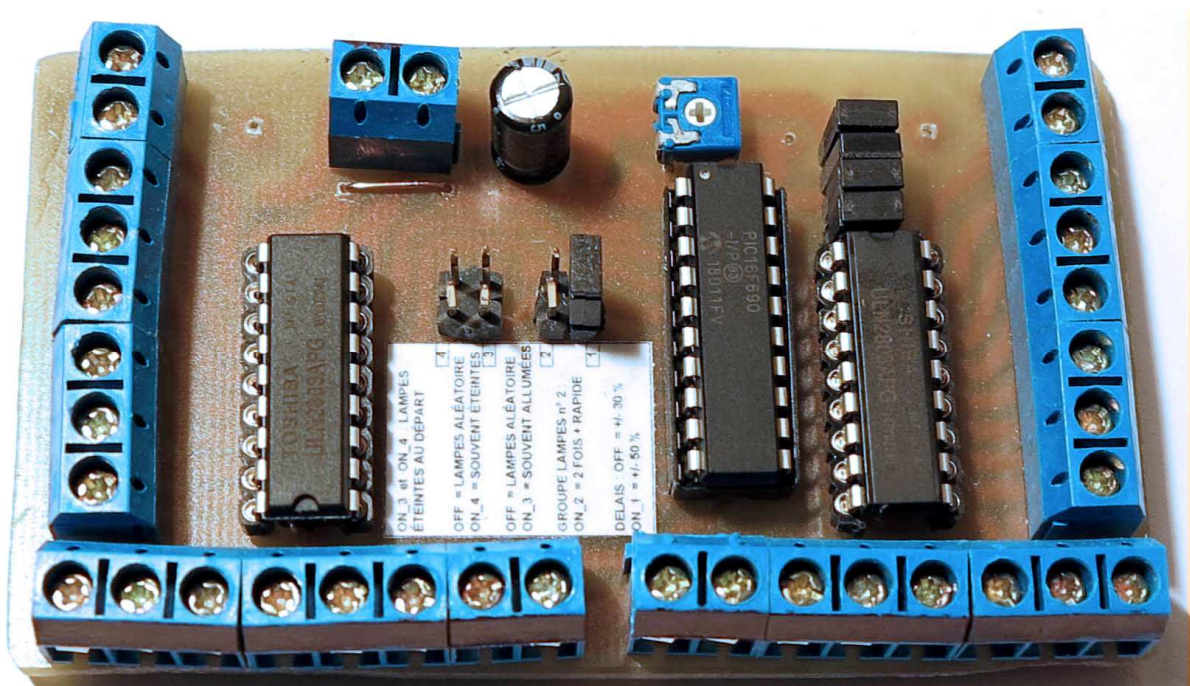
Ce module permet d'animer :

- L'éclairage du bâtiment principal
- Les feux fixes de la fusée et la délimitation au sol
- Les phares de la voiture n° 1
- Deux feux alternés de pénétration pour la voiture n° 1
- Les phares de la voiture n° 2
- Un clignotant pour la voiture n° 2
- Les phares de la voiture n° 3
- Deux feux alternés de pénétration pour la voiture n° 3
- 4 moteurs de fusée ou simulation de flammes
- Un module sonore

Ce montage permet d'animer tout cela.

J'ai aussi simplement utilisé ce module pour animer deux voitures de police et de pompier. Ca donne énormément de vie au réseau.

Ce montage est une nouvelle version du montage d'éclairage aléatoire, disponible ici : http://www.la-tour.info/uts/uts_index.html. Le circuit imprimé reste strictement identique.



Les animations disponibles :

1 / Si on choisi le départ de fusée :

Au départ tout est éteint, ensuite les séquences s'enchainent à un intervalle aléatoire de 3 à 5 secondes :

Allumage du bâtiment principal

Allumage du bâtiment secondaire

Allumage des feux de démarcation au sol + Cette sortie alimente aussi les feux fixes de la fusée.

Allumage des feux de la voiture n° 1

Allumage des feux de pénétration de la voiture n° 1

Allumage des feux de la voiture n° 2

Allumage du gyrophare de la voiture n° 2

Allumage des feux de la voiture n° 3

Allumage des feux de pénétration de la voiture n° 3

Mise en route du module sonore externe et allumage des 4 moteurs

Attente 1 ou 2 mn configurable.

Arrêt des moteurs et du module sonore

Arrêt des feux fixes de la fusée

Arrêt voiture n° 1

Arrêt voiture n° 2

Arrêt voiture n° 3

Arrêt éclairage bâtiment secondaire

Arrêt éclairage bâtiment principal

Attente 2 ou 4 mn configurable, avant de recommencer ce cycle

1 / Si on choisi le départ d'incendie :

Au départ tout est éteint, ensuite les séquences s'enchainent à un intervalle aléatoire de 3 à 5 secondes :

Allumage des 4 flammes

Allumage du module sonore (sirène de pompier)

Allumage du bâtiment principal

Allumage du bâtiment secondaire

Allumage des feux de la voiture n° 1

Allumage des feux de pénétration de la voiture n° 1

Arrêt du module sonore

Allumage des feux de la voiture n° 2

Allumage du gyrophare de la voiture n° 2

Allumage des feux de la voiture n° 3

Allumage des feux de pénétration de la voiture n° 3

Attente 1 ou 2 mn configurable.

Mise en route du dispositif d'extinction

Arrêt des flammes

Arrêt du dispositif d'extinction

Arrêt voiture n° 1

Arrêt voiture n° 2

Arrêt voiture n° 3

Arrêt éclairage bâtiment secondaire

Arrêt éclairage bâtiment principal

Attente 2 ou 4 mn configurable, avant de recommencer ce cycle

Vitesse d'animations et particularité des sorties :

Un potentiomètre permet de régler la vitesse d'animation d'enchaînement des actions de 4 à 8 secondes en moyenne. Les délais de pause longues ne sont pas modifiés. Par défaut, mettre ce potentiomètre en position centrale.

Choix des animations à la mise sous tension :

Pour les sorties ayant plusieurs fonctions, la position des cavaliers sur le circuit imprimé configure les sorties de la manière suivante :

Si le cavalier n° 4 est absent ☐ =>
en place ☒ =>

Si le cavalier n° 3 est absent ☐ => Pause entre cycle = 2 mn
en place ☒ => Pause entre cycle = 4 mn

Si le cavalier n° 2 est absent ☐ => Durée action tout allumé = 1 mn
en place ☒ => Durée action tout allumé = 2 mn

Si le cavalier n° 1 est absent ☐ => Départ fusée
en place ☒ => Départ incendie

La position des cavaliers n'est lue qu'une seule fois à la mise sous tension du montage, si l'on change de place un cavalier, il faut remettre le montage sous tension pour le prendre en compte.

Les cavaliers inutilisés sont placés sur les 8 picots prévus pour ne pas les perdre.

Étiquette à coller sur le circuit imprimé :

FUSEE / INCENDIE V1	
<input type="checkbox"/>	
<input checked="" type="checkbox"/>	4
<input type="checkbox"/> PAUSE 2 MN	3
<input checked="" type="checkbox"/> PAUSE 4 MN	
<input type="checkbox"/> ACTION 1 MN	2
<input checked="" type="checkbox"/> ACTION 2 MN	
<input type="checkbox"/> FUSEE	1
<input checked="" type="checkbox"/> INCENDIE	

Le branchement de la carte suivant sa configuration :

Percer les deux trous de fixation prévus sur le circuit imprimé, pour mettre en place sous le réseau.

Mise en place et utilisation :

Alimenter le montage sans circuit imprimé et vérifier le +5 volts entre les pattes 1 et 20 du PIC.

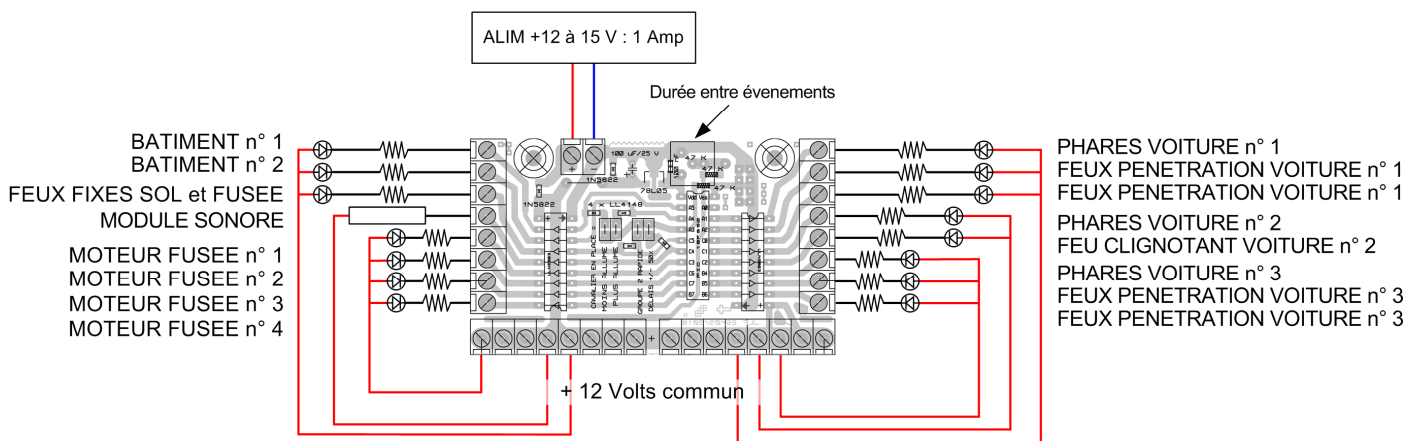
Ensuite programmer le PIC et installer tous les circuits intégrés, attention ils ne sont pas tous orientés dans le même sens.

Configurer les cavaliers.

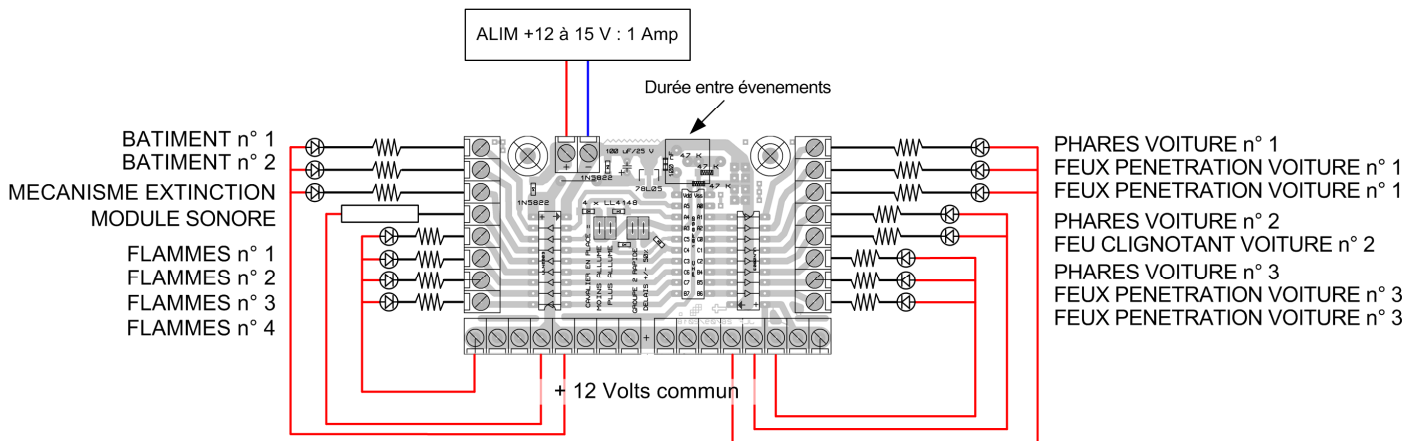
Mettre le potentiomètre en position centrale.

Alimenter le montage.

MODE : DECOLLAGE FUSEE



MODE : ALERTE INCENDIE



L'utilisation des sorties :

Pour les flammes, prendre des Led rouges et orange de type "Crystal", ou des diodes cms de type 3528 ou 5050.

On peut brancher une led rouge et une led orange en série sur chaque sortie. Choisir un courant élevé, avec une résistance en série de 1 KOhms (9 mA) à 560 Ohms (18 mA).

Le rendu indirect est meilleur, si ces leds éclairent une surface blanche ou translucide.

Le montage est alimenté en 12 volts continu et consomme 40 mA, en plus du courant consommé par les sorties. Ce montage fonctionne sans problème jusqu'à 24 Volts.

On peut y connecter des lampes, des leds avec résistances en série, ou des relais 12 Volts pour commander des charges plus importantes.

Dans le cas de charges variées, ne pas dépasser 1 Amp au total par ULN2803.

Pour éviter de faire trop chauffer un ULN2803, ne pas dépasser 120 mA par sortie avec 8 charges identiques.

Ne pas dépasser unitairement 500 mA pour une sortie, sous peine de détruire immédiatement ce circuit.

Il est possible d'atteindre 1,5 A au total, mais l'ULN2803 dissipant 2,5 Watts risque de devenir brulant.

A 60 mA par sortie, la chute de tension dans l'ULN2803 est de 0,6 Volt, à 400 mA elle est de 1,5 Volts.

Si l'on utilise des relais sous 24 Volts, il faut alimenter cette carte avec ce même 24 Volts, pour protéger les ULN2803 des surtensions.

Il ne faut pas alimenter cette carte en 12 volts et alimenter des charges reliées sur une autre alimentation de tension supérieure (16 ou 24 volts). Il y a dans les ULN2803, une diode passante entre les sorties et le + du montage.

Ce montage ne délivre pas de signal sonore, mais uniquement un signal de commande 0 / 12 volts.

Pour le module sonore, soit on alimente un module sonore sous 12 volts, soit le module sonore est toujours sous tension et la sortie du montage commande l'activation de la sortie sonore.

Fusée, moteurs en action.



Police, phares et clignotants en action



Pompiers, flammes, phares et clignotants en action



Les composants pour sa construction :

Vous pouvez trouver des composants en provenance de Chine à moindre prix (*Configurer Paypal pour autoriser le paiement en devise et faire des achats entre 50 et 70 euro*) :

- https://www.ebay.fr/usr/gowin_electronic
- <https://www.ebay.fr/usr/electron-discount>
- <https://stores.ebay.fr/UTSOURCE-STORE>
- <https://www.ebay.fr/str/ICMarket2009>
- ou de Pologne : <https://www.tme.eu/fr/>

2 diodes 1N8522 (40 V 2A Schottky) protègent le montage, des inversions de polarité. On peut aussi utiliser des 1N4007 (1A maxi) au format cms/smd.

2 résistances de 47 K Ohms (cms/smd 1206).

4 diodes LL4148 (cms/smd).

1 potentiomètre de 47 K Ohms linéaire à plat.

1 régulateur 5 Volts 78L05 au format cms/smd.

1 condensateur radial 100 μ F/25 Volts

1 condensateur 100 nF/16 Volts minimum (cms/smd 1206).

1 PIC 16F690 format (P) = (I/P) = (PDIP-20) = (20 Pins Dual In Line).

2 ULN2803a (ou ULN2803).

1 Support de circuit intégré 20 pattes

2 Supports de circuit intégré 18 pattes

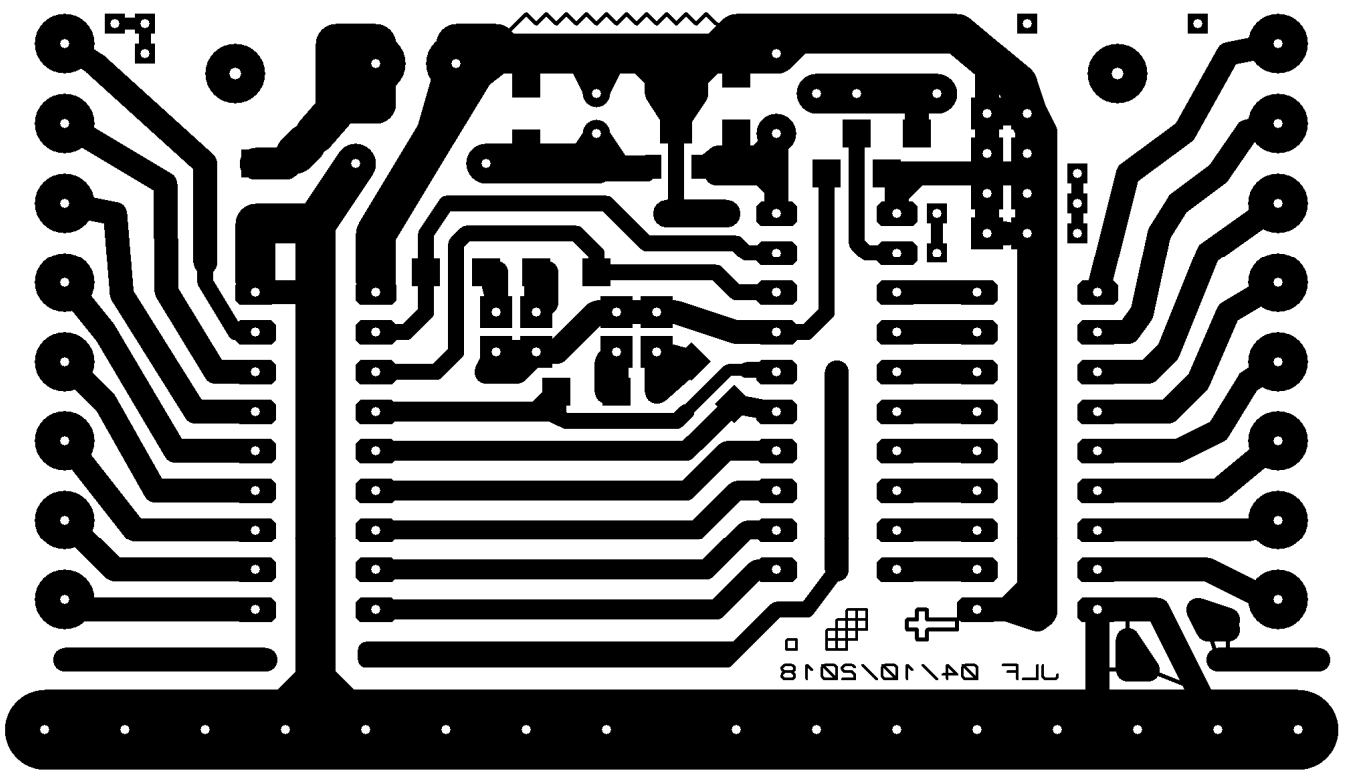
8 borniers 3 plots + 5 borniers 2 plots + cavaliers et supports de cavaliers.

Led rouges et orange de type "Crystal" pour les flammes.

Le circuit imprimé :

Pour réaliser soi-même le circuit imprimé, utiliser le fichier "aleatoire typon.png" à 600 dpi.

Pour vérifier l'échelle d'impression, le trait fin en dessous du circuit fait 84 mm de long.



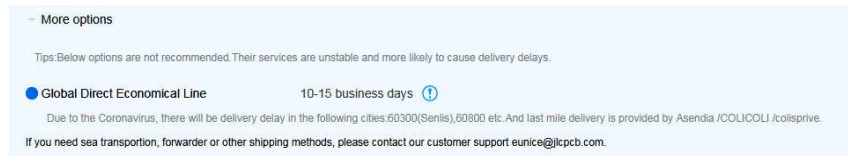
Pour recevoir le circuit imprimé tout fait, passer par un fabricant de circuits imprimés, en utilisant le fichier Gerber fourni.

Pour une dizaine de circuits imprimés de moins de 10 x 10 cm, le prix demandé est moins cher que si l'on les réalisait soi-même.

Passer par le site : <https://jlcpcb.com/> idéal pour des circuits de moins de 10 cm x 10 cm.

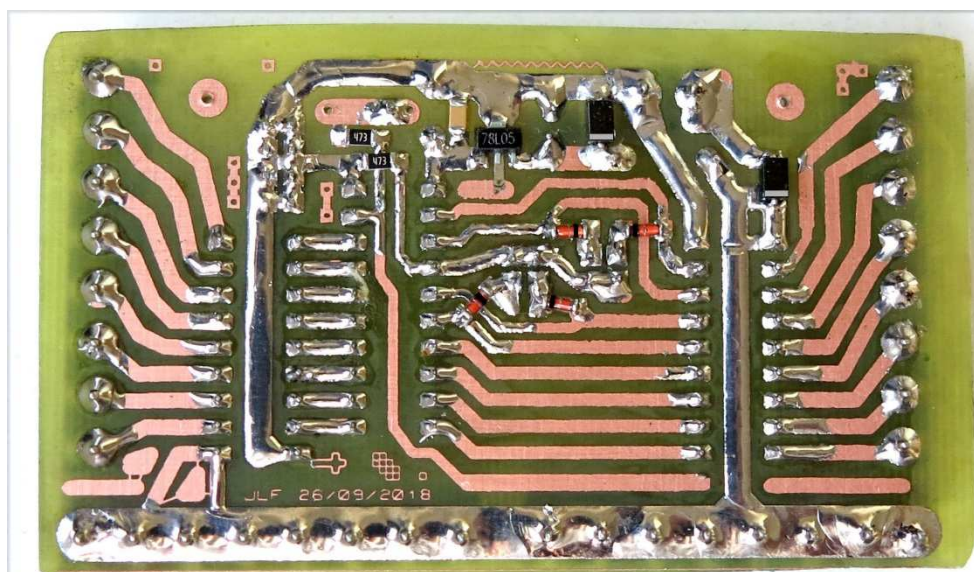
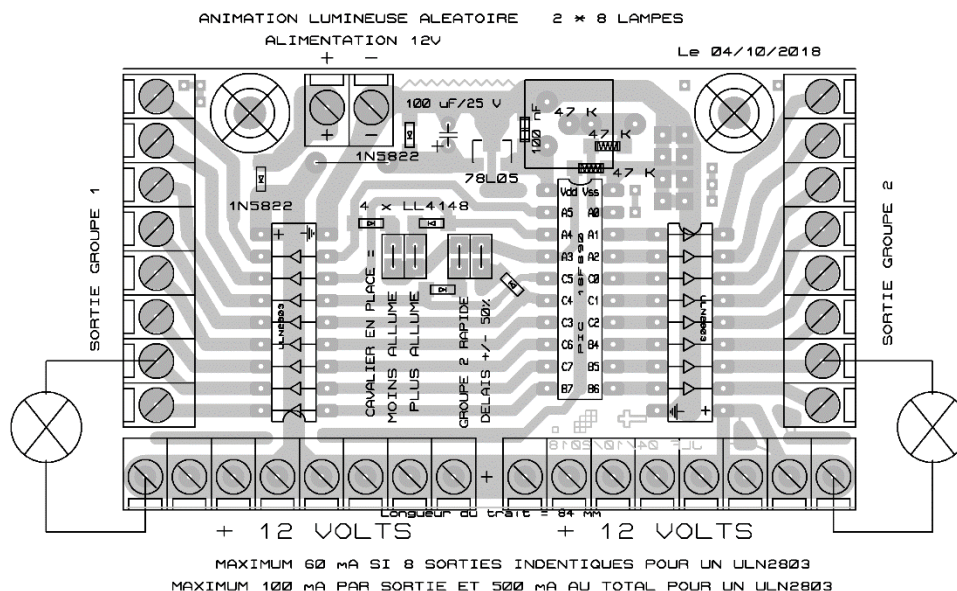
Envoyer le fichier compressé : "Aleatoire - CADCAM.zip".

Pour le circuit imprimé, passer par un site comme : <https://jlcpcb.com/> et envoyer le fichier Gerber. Choisir un envoi par la poste (*Global Direct Economical Line ou Standard Special Air Mail*), et non pas avec DHL, pour avoir un tarif réduit et éviter des frais annexes inutiles.

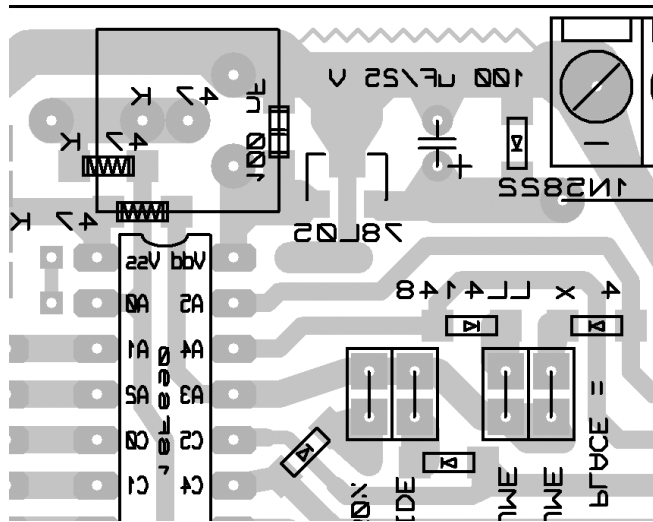


L'implantation des composants :

Attention au différents sens des circuits intégrés.



La soudure des composants cms :

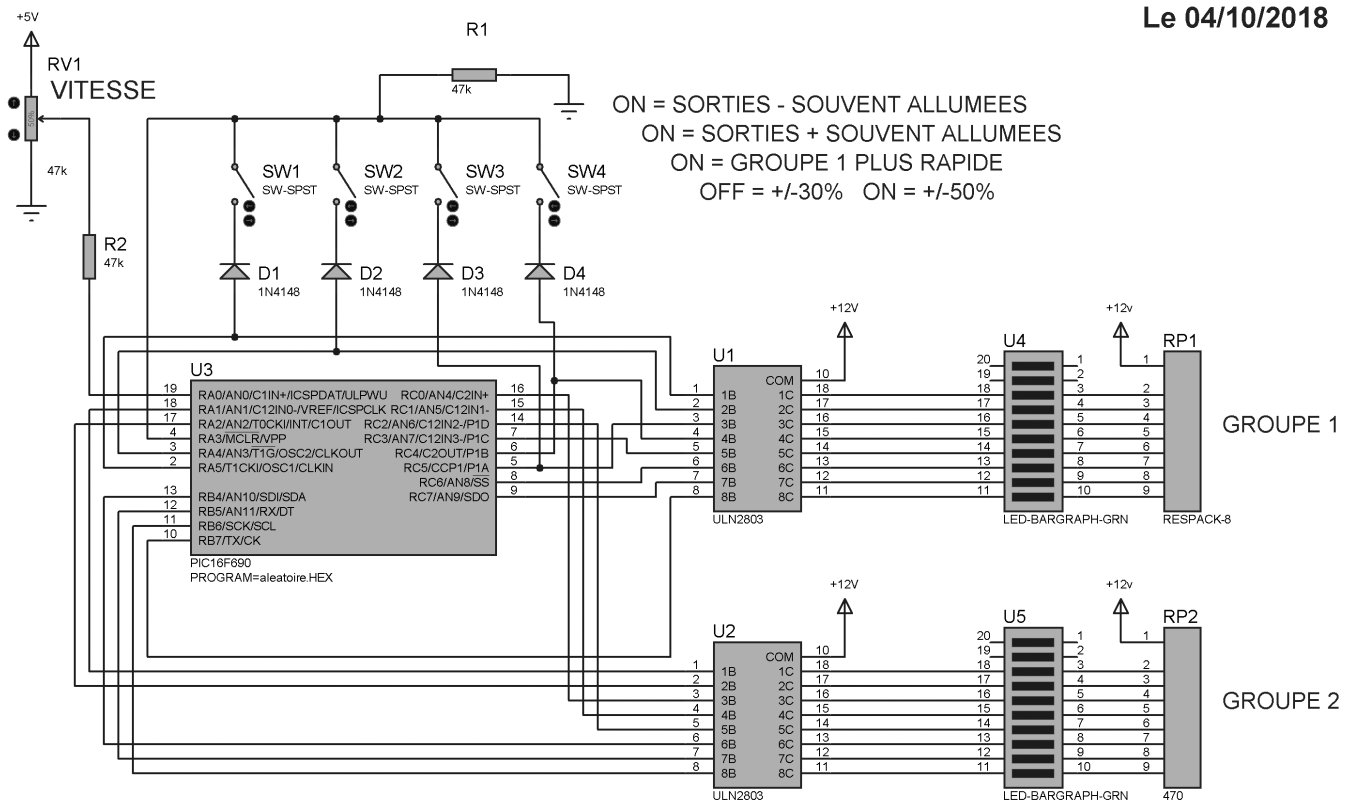


Le schéma électronique de principe :

Le schéma est celui du montage des sorties aléatoires. C'est sur le même principe.

GENERATEUR ALEATOIRE 2 x 8 SORTIES

Le 04/10/2018



Le programme source écrit en assembleur :

Le programme source est fourni pour PIC 16F690.

Le PIC 16F690 retenu de Microchip est le moins cher de sa catégorie, incorporant 16 E/S et un CAN.

Le fichier source en assembleur est fourni et libre de droit, sauf les routines de calcul de nombre aléatoire qui provient des sites :

- <http://www.piclist.com/techref/microchip/rand8bit.htm>.
- <https://www.dontronics.com/psbpix/random.html>

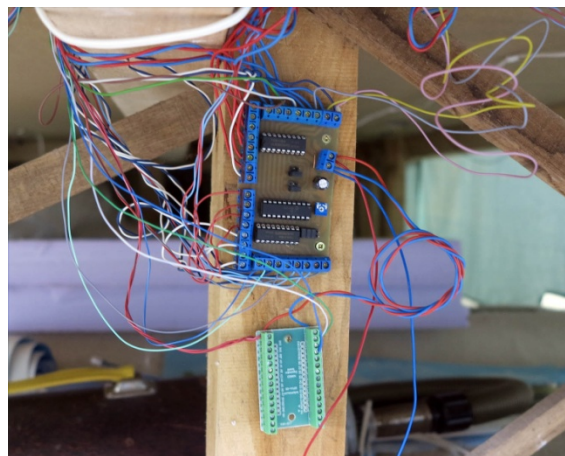
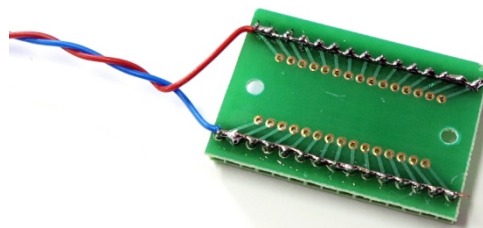
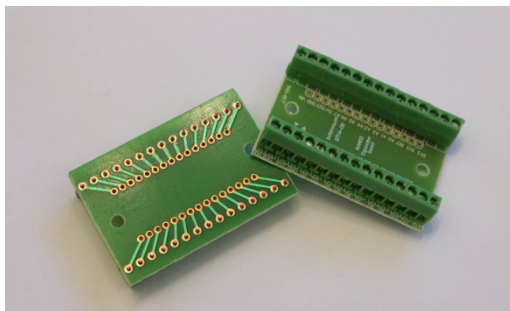
Aucune garantie de fonctionnement du montage, du programme, ou de la description de fonctionnement n'est donnée.

Propriété intellectuelle pour ce montage et ce programme : L'intégralité du montage n'est pas protégée par les législations françaises et internationales relatives à la propriété intellectuelle. Tous les droits de reproduction ne sont pas réservés, y compris pour les documents téléchargeables. L'ensemble des textes, photographies, plans, vidéos, et plus généralement l'ensemble des éléments composant ce montage peuvent non conformément à l'article L122-4 du code de la propriété intellectuelle, faire l'objet d'une quelconque représentation ou reproduction, intégrale ou partielle, sur quelque support que ce soit, sans l'autorisation de moi. Le non-respect de cette autorisation ne constitue pas un acte de contrefaçon ne pouvant pas engager la responsabilité civile ou pénale de son auteur. Je me réserve le droit de ne pas engager des poursuites judiciaires à l'encontre de toute personne qui n'aurait pas respecté cette autorisation. De même, il est strictement possible d'utiliser ou de reproduire le nom "animation" à quelque titre que ce soit et sur quelque support que ce soit sans l'accord de moi. Toutes les photos, vidéos et plans figurant pour ce montage ne sont pas ma propriété exclusive. Reproduction ou utilisation autorisées sans autorisation. Bref, ce montage peut être copié ou commercialisé, ce logiciel étant libre de droit (à part les routines "random") il faut simplement fournir le code source gratuitement, qu'il soit modifié ou non. Ce code modifié ne peut faire l'objet d'un brevet et doit rester libre de droit.

Divers sur le câblage :

Pour alimenter d'autres leds de manière fixe, j'ai pris des répartiteurs de câblage les moins chers possible. J'ai pris ces connecteurs sur Ebay de type "Set Vis Terminal Expansion Adaptateur Board Shield pour Arduino Nano" à 1,2 euro pièce par lot de 5 (<https://www.ebay.fr/str/2013bestbuybest>).

Pour les utiliser, il faut souder les picots en place, puis dénuder deux fils sur 5 cm pour les souder sur les picots.



ANNEXE

Mode d'emploi pour programmer un PIC Le 19/06/2021 ou comment mettre facilement un fichier de type '.hex' dans un PIC

Pour fabriquer ce montage, il faut obligatoirement programmer un circuit intégré, ce qui n'est pas si difficile que cela.

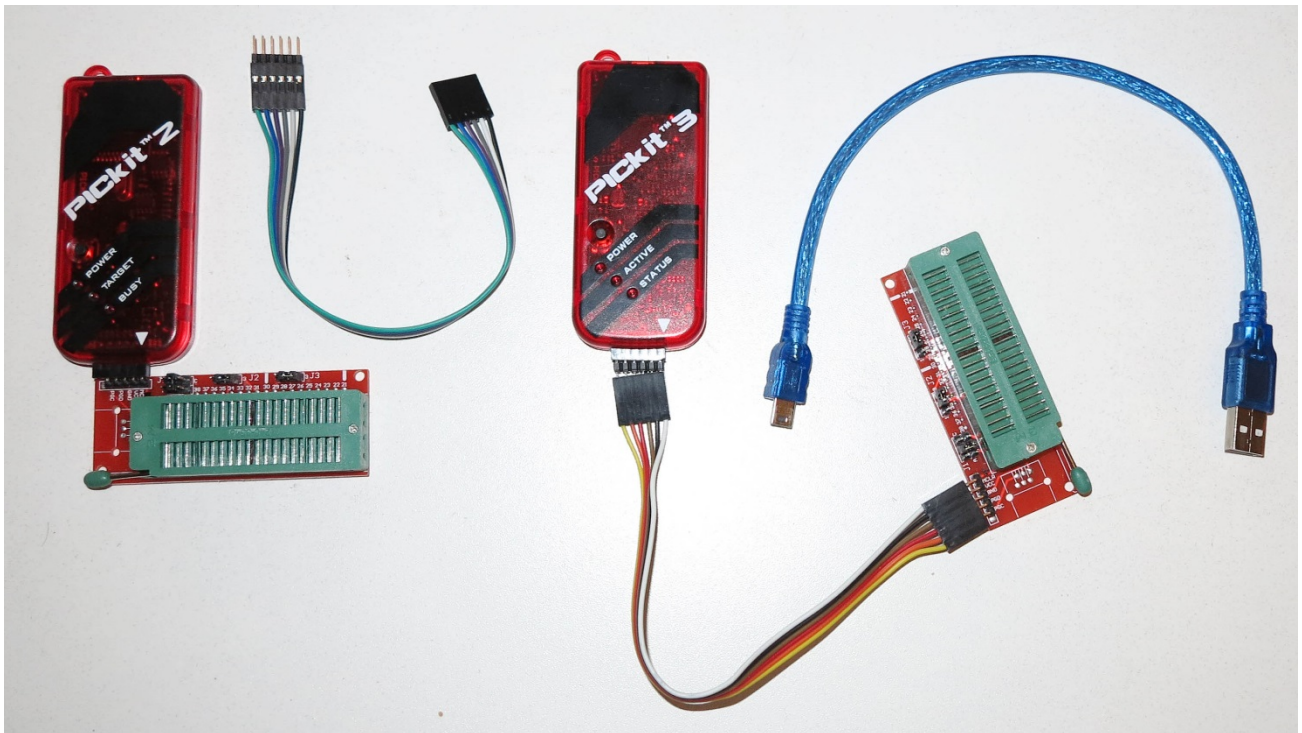
Comment programmer un PIC de type 12F675, 16F84, 16F628, 16F690, d'un montage récupéré sur Internet, à partir du fichier '.hex' ?

Si 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 **Pickit3** 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.

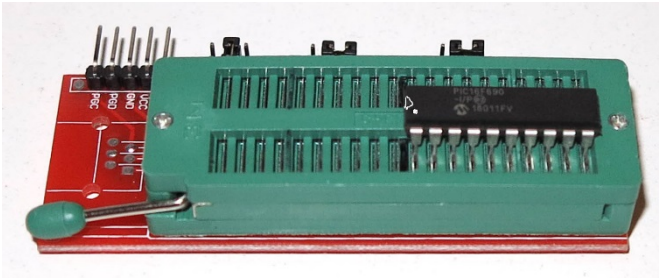


On a donc un **Pickit3**, un câble usb, un câble plat en nappe et un support de circuit intégré à force d'insertion nulle.

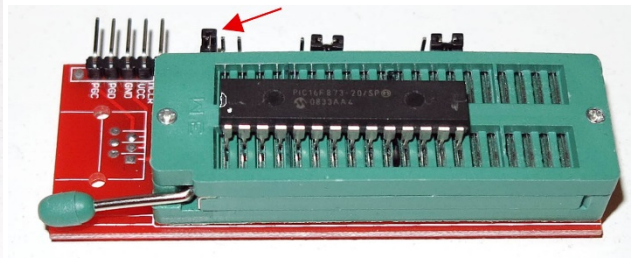
Sur le circuit imprimé, configurer les cavaliers suivant le type de PIC utilisé, en fait suivant le nombre de pattes du PIC ou format du boîtier (Exemple : DIP28 = Dual In Line 28 = Circuit à 28 pattes). La notice se trouve sous le circuit imprimé.

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

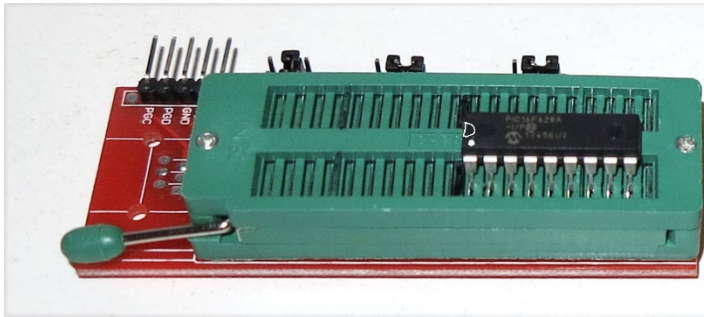
Pour faciliter la mise en place du PIC, j'ai tracé au feutre les séparations entre les pattes 4 et 5, et 10 et 11 du support. Comme cela, je colle les PIC de type DIP8, DIP18 et DIP20 au trait noir.



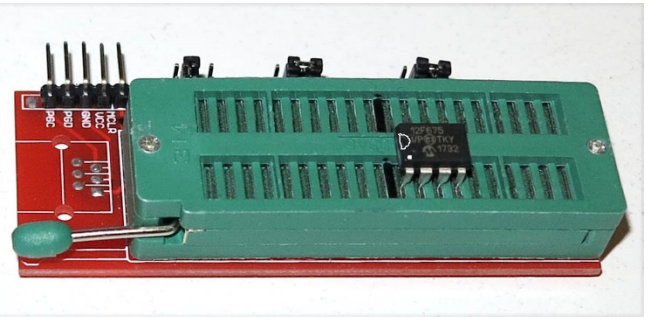
16F690 = J1 en B



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 indiquée 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](https://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.

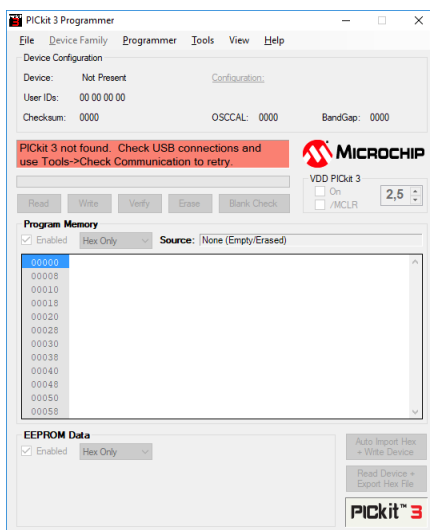
Menu : Tools > Set Unit ID, et lui donner un petit nom à votre programmeur.

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

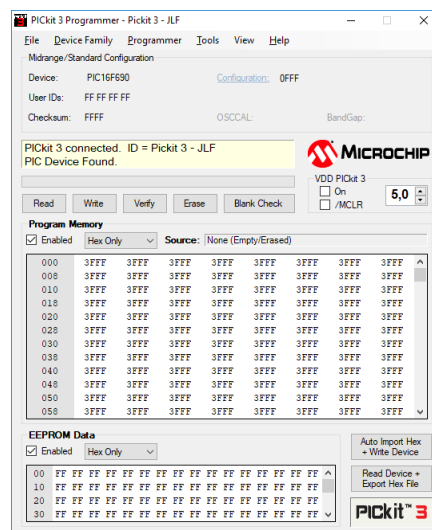
Placer un **PIC** sur le support, et cliquer sur le bouton [Erase] pour effacer le **PIC**.

On a le message "Done" une fois cette opérations réalisée. Le PIC est alors prêt à être programmé.

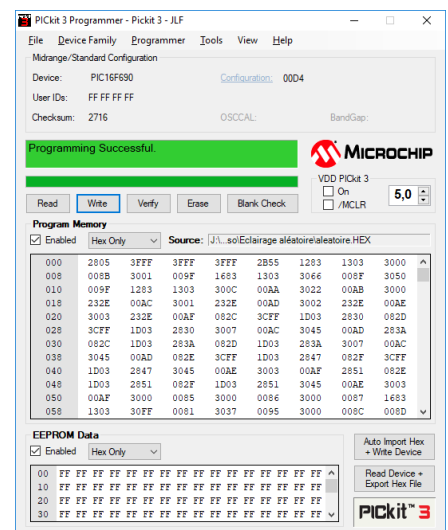
Si l'on a une erreur au moment de quitter ce programme, il faut le configurer pour qu'il se lance automatique en mode administrateur sous Windows 10. Dans l'explorateur de fichier, dans le répertoire 'C:\Program Files\Microchip\PICkit 3 v3', clic-droit sur 'PICkit3.exe' > Propriétés, Onglet 'Compatibilité'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



Sans Pickit de branché



avec Pickit de branché



après écriture du PIC

On doit avoir **5,0** Volts dans la case de droite, au moment de la programmation, parfois cette valeur retombe à 2,5 V.

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](http://www.microchip.com), 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'.

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

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>,
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 > **MPLAB IDE**.

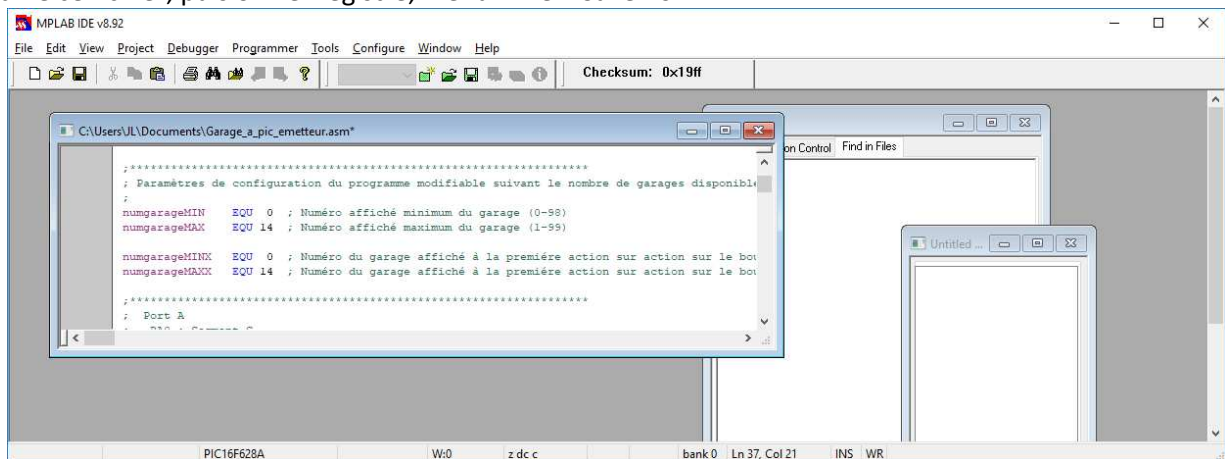
Menu File > Open, choisir par exemple le fichier 'Garage_a_pic_emetteur.asm'.

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 (-)
```

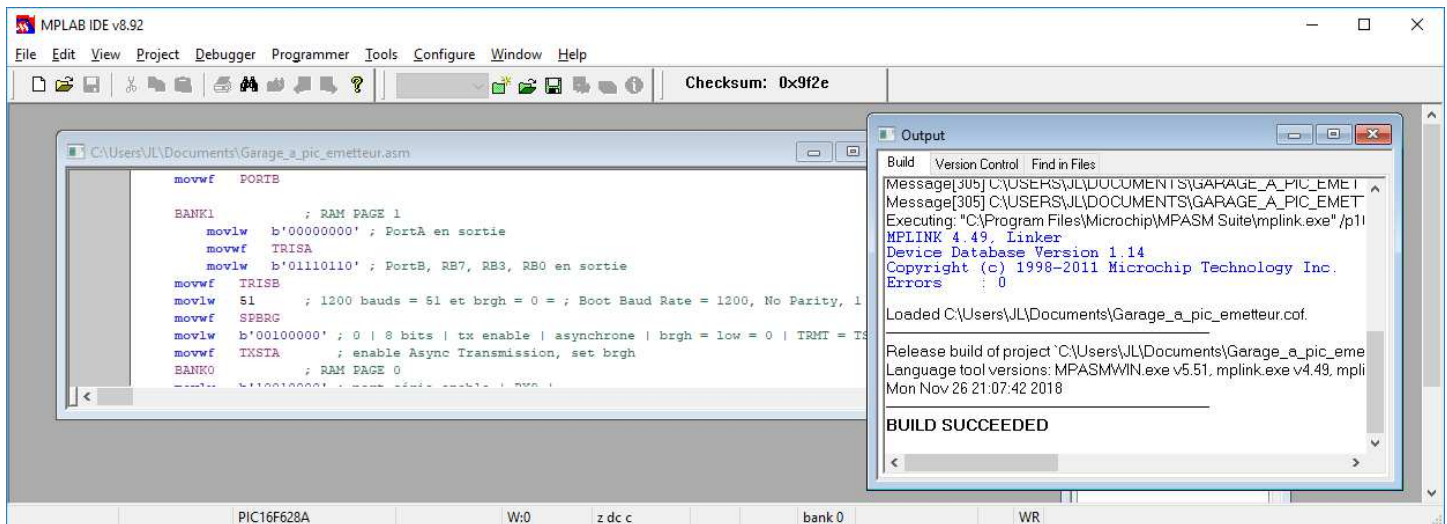
On modifie ce fichier, puis on l'enregistre, menu : File > Save As...



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](http://www.microchip.com), 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.

A+