

Module d'animations Enseigne - Feux - Croix

Le 10/03/2019

Montage : Animations - Enseigne, Feux Tricolores, Croix

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://jlcpcb.com/> idéal pour des circuits de moins de 10 cm x 10 cm. Envoyer le fichier "Typon-Aleatoire-G - CADCAM.zip". C'est le meilleur rapport qualité/prix. Ne pas choisir DHL comme moyen de livraison, sous peine de payer 20 euros pour des frais de dossier.

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.

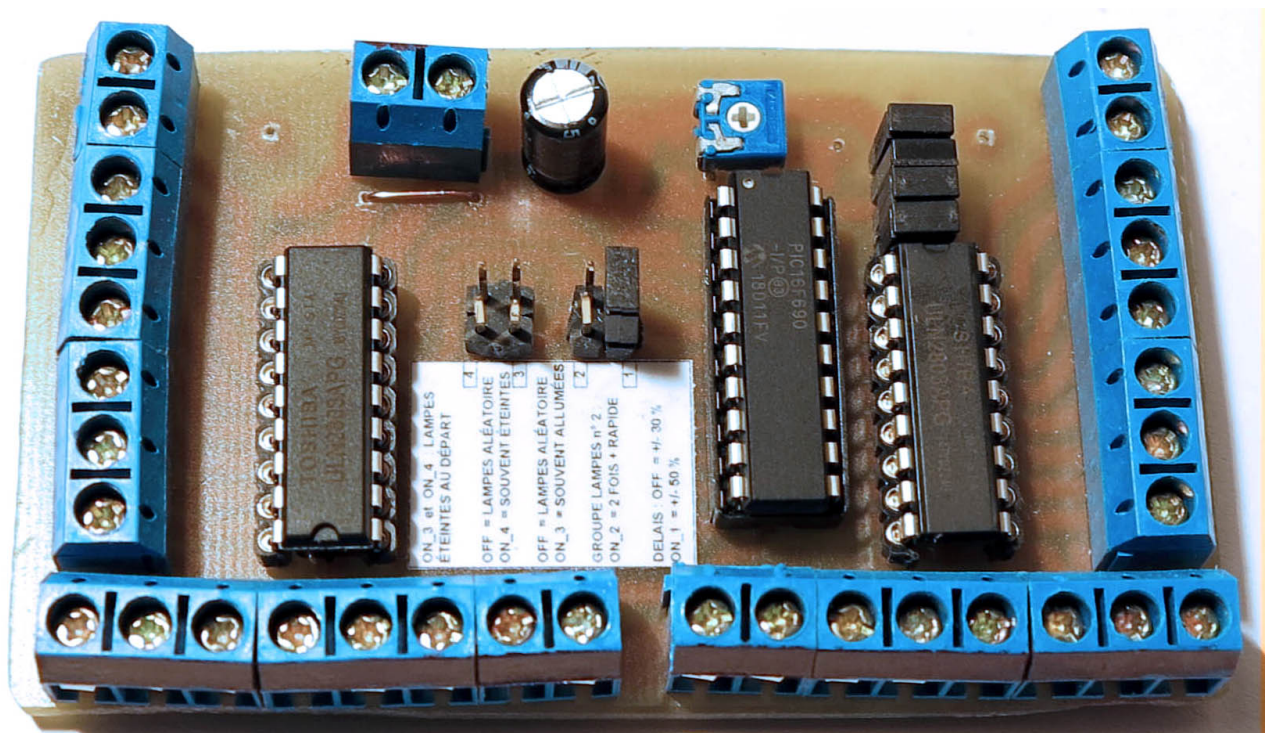
Le programme est fourni au format '.hex' et le code source en assembleur au format '.asm'.

Ce module permet d'animer :

- Une enseigne de publicité ou de magasin, verticale ou horizontale à 6 lampes,
- Une croix de pharmacie à 5 lampes,
- Des feux tricolores de carrefour routiers à 2 fois 3 lampes,
- Un chenillard pour zone de travaux à 6 lampes,
- Trois tubes fluorescents en panne, qui clignotent,
- Un feu de camp,
- Deux simulations de poste de soudure à l'arc.

Ce montage permet d'animer tout cela, mais il faudra choisir une partie des animations pour les 16 sorties disponibles.

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 :

Le feu de camp :

- Le feu de camp alterne différentes séquences de fonctionnement des flammes. On passe d'un feu continu, à un feu balayé par le vent, un feu en extinction ou un feu qui se ravive. On peut configurer un mode utilisant 2 leds, ou un mode plus réaliste utilisant 3 leds rouge ou orange. Il faut éclairer une matière translucide, ou par réflexion sur une zone blanche.

Une enseigne de magasin à 6 leds:

- Huit séquences différentes se suivent, avec un clignotement de l'enseigne entre chaque séquence.

La croix de pharmacie à 5 leds:

- Six séquences différentes se suivent, avec un clignotement de la croix entre chaque séquence.

Les feux tricolores de carrefour routiers :

- Séquence classique de feux routiers à 3 lampes, rouge orange et vert.

Un chenillard à 6 leds installé en bordure de route pour signaler des travaux :

- Séquence classique de lampes allumées l'une après l'autre.

Les tubes fluorescents :

- Tube Fluo **1** = Cinq tentatives d'allumage puis reste allumé un temps relativement long. Ensuite, reste éteint pendant 40 secondes.
- Tube Fluo **2** = Nombreuses tentatives d'allumage, mais ne reste allumé qu'un court moment. Ensuite, reste éteint pendant 30 secondes.
- Tube Fluo **3** = Tentatives d'allumage en continu, mais ne parvient pas à s'allumer. Ensuite, reste éteint pendant 20 secondes.

Les postes de soudure à l'arc :

La durée élémentaire d'un flash est de 10 à 104 msec, et la pause entre flashes est de 10 à 32 msec. Le premier flash de la série dure 50 msec.

- Soudure à l'arc **1** = Fonctionne pendant 40 secs, suivi d'une pause de 30 secs. De 5 à 21 flashes de suite, et de 0,3 à 0,8 secs de pauses courtes.
- Soudure à l'arc **2** = Fonctionne pendant 30 secs, suivi d'une pause de 40 secs. De 10 à 41 flashes de suite, et de 0,5 à 2 secs de pauses courtes.

Vitesse d'animations et particularité des sorties :

Un potentiomètre permet de régler la vitesse d'animation de :

- l'enseigne publicitaire,
- la croix de pharmacie,
- les feux routiers,
- le chenillard de zone de travaux.

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 [] => Soudure à l'Arc n° 2 ou
en place [■] => Tube Fluorescent n° 2

Si le cavalier n° 3 est absent [] => Chenillard ou
en place [■] => Feux Tricolores

Si le cavalier n° 2 est absent [] => Feux de camp à 2 lampes + (Tube Fluo n° 1 ou Arc n° 1)
en place [■] => Feux de camp à 3 lampes

Si le cavalier n° 1 est absent [] => Enseigne de magasin ou
en place [■] => Croix de pharmacie

Si le cavalier n° 1 est en place [■] et
+ n° 2 est absent [] => Tube Fluorescent n° 1 ou
+ n° 2 est en place [■] => Soudure à l'Arc n° 1

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 :

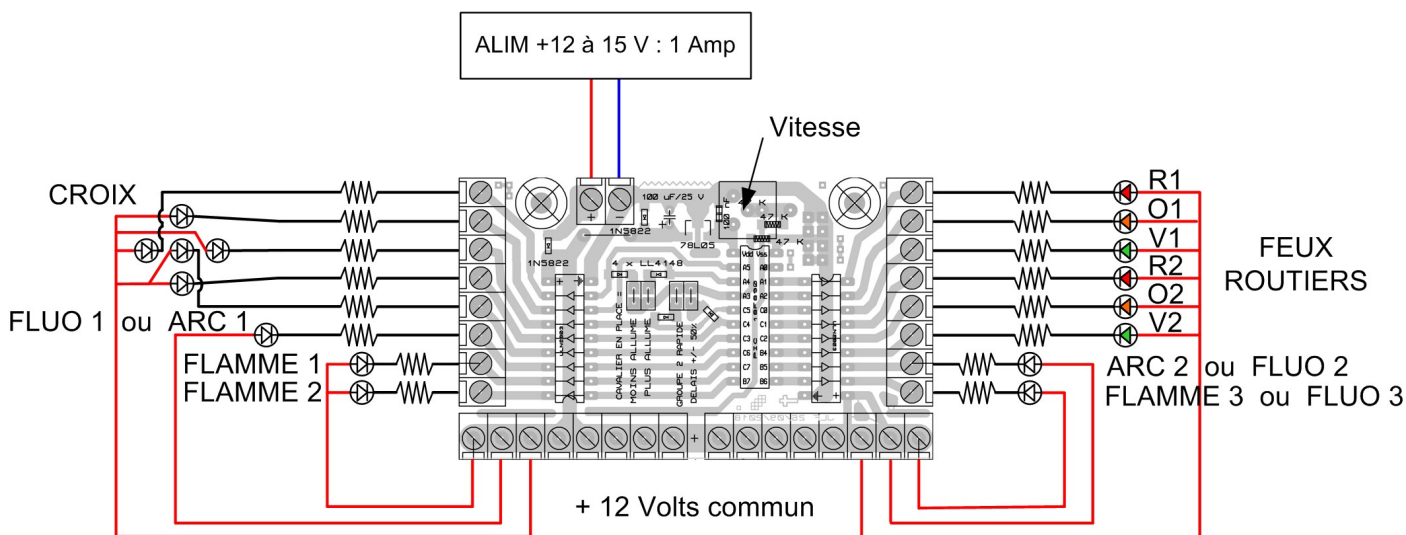
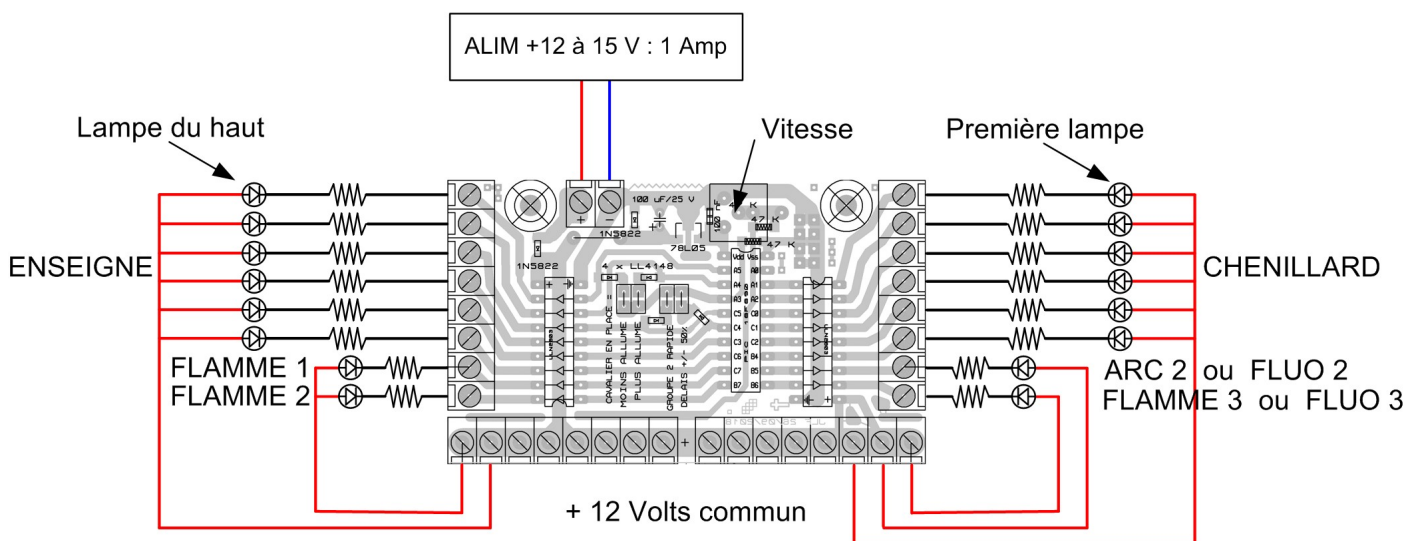
1[=]+2[]FLUO1, 1[=]+2[=]ARC1	
[] ARC 2	
[=] FLUO 2	4
[] CHENILLARD	
[=] FEUX ROUTIER	3
[] 2 FLAMMES+FL3	
[=] 3 FLAMMES	2
[] ENSEIGNE	
[=] CROIX+FL1 ou AC1	1

Étiquette pour le montage :

1[=]+2[]FLUO1, 1[=]+2[=]ARC1	
[] ARC 2	
[=] FLUO 2	4
[] CHENILLARD	
[=] FEUX ROUTIER	3
[] 2 FLAMMES+FL3	
[=] 3 FLAMMES	2
[] ENSEIGNE	
[=] CROIX+FL1 ou AC1	1

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.

L'utilisation des sorties :

Le montage est alimenté en 12 volts continu et consomme 40 mA, plus le courant consommé par les sorties. Ce montage fonctionne sans problème jusqu'à 24 Volts. Il faut toujours que les charges en sortie soient branchées entre les ULN2803 et les bornes du + commun de cette carte.

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

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

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

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 brûlant.

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.

L'utilisation des sorties (suite) :

Pour l'enseigne, la croix, le chenillard ou les feux routiers, un fil commun aux leds, relié au +12 Volts commun à la carte, simplifiera le montage.

Il faut alors relier le (+) des leds ensemble et le relier au bornier (+), et prévoir une résistance par led vers les borniers de sortie des ULN2803.

Pour les postes de soudure à l'arc, utiliser des leds blanches à ton froid (cold). On peut aussi ajouter une diode bleue + résistance sur la même sortie.

Pour les postes de soudure à l'arc, pour ajouter le bruit de crépitements de l'arc, relier une pastille piézoélectrique à 1 euro, entre le +12Volts et la sortie arc1 ou arc2.

Exemples de transducteur piézoélectrique :



Les feux routiers et de zone de travaux à chenillard orange ou plots de chantier.



Pour le feu de camp, la meilleure restitution se réalise avec 3 leds, 2 leds est un minimum. Prendre des modèles translucides ou cristal, en orange/rouge/jaune.

Des enseignes lumineuses pour des hôtels, Hilton, Casino, Auchan, Cinéma, ...



Pour les flammes du feu de camp, il est conseillé de prendre si possible la configuration à 3 leds.

Plusieurs effets s'enchainent à des intervalles compris entre 6 et 10 secondes, comme la descente et la montée des flammes en mode rapide ou lent, les flammes vacillantes, les flammes puissantes ou encore les flammes en mode classique.

Il faut prendre des leds jaune, orange, rouge de type transparent (cristal).

Il faut éclairer une matière translucide comme du papier sulfurisé, pour avoir un meilleur effet.

On peut aussi éclairer une surface blanche et visualiser le feu par réflexion.

Pour simuler un brasero, prendre une led rouge opaque et placer autour un grillage très fin peint en noir. Planter directement la led dans le décor.



Pour alimenter qu'une partie des animations, il suffit d'intercaler un interrupteur entre le bornier +12 Volts et les leds. On peut ainsi alimenter qu'une partie des animations, selon les besoins du moment.

Si par exemple, on ne veut alimenter l'enseigne "HOTEL" que la nuit, il faut que l'éclairage des bâtiments en + 12 volts soit relié directement au +12 Volts de l'enseigne, ou que la commande d'éclairage active un relais dont les contacts relient le +12 Volts aux leds de l'enseigne.

Cette option est aussi intéressante pour les tubes fluorescents clignotants, pour alimenter un seul lampadaire de la rue. Quand on allume tous les lampadaires de la rue, un seul va clignoter. Si l'on arrête l'éclairage, tous les lampadaires resteront éteints.

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

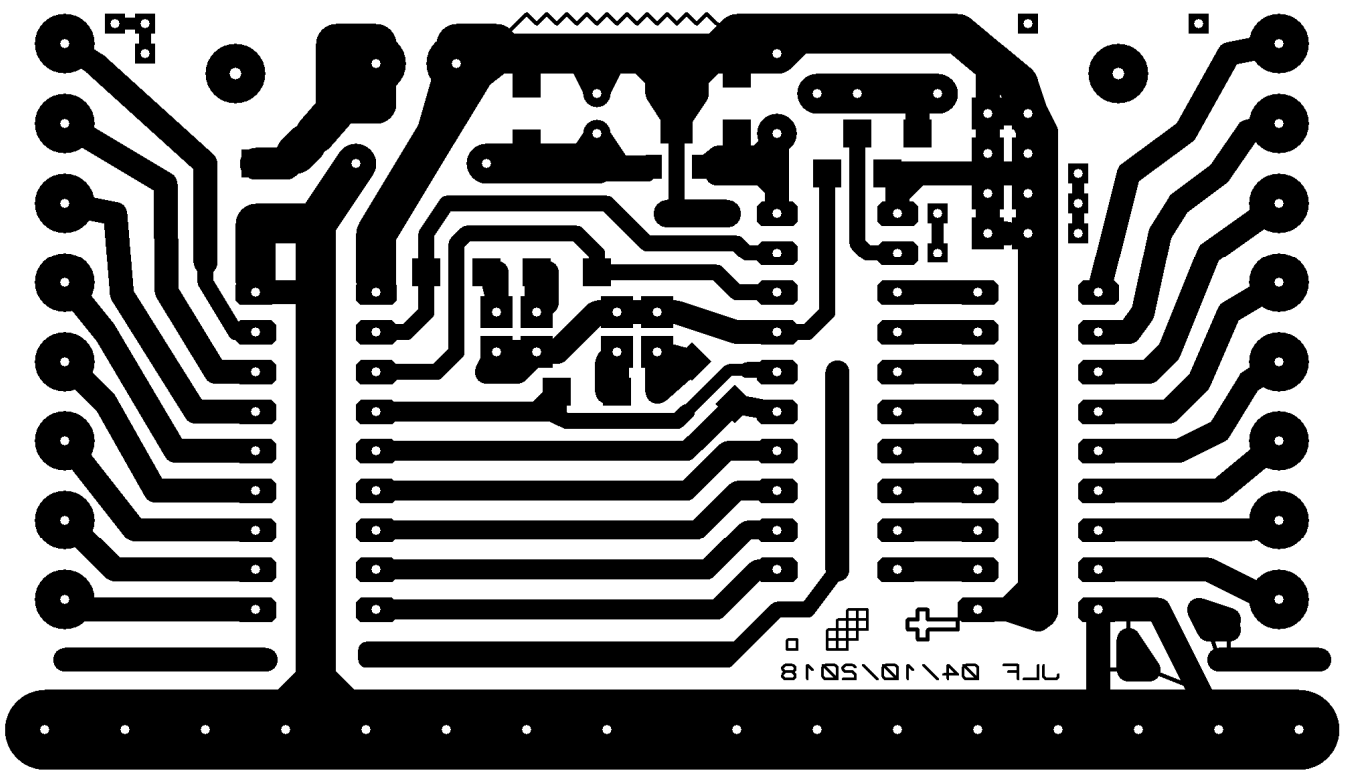
2 Supports de circuit intégré 18 pattes

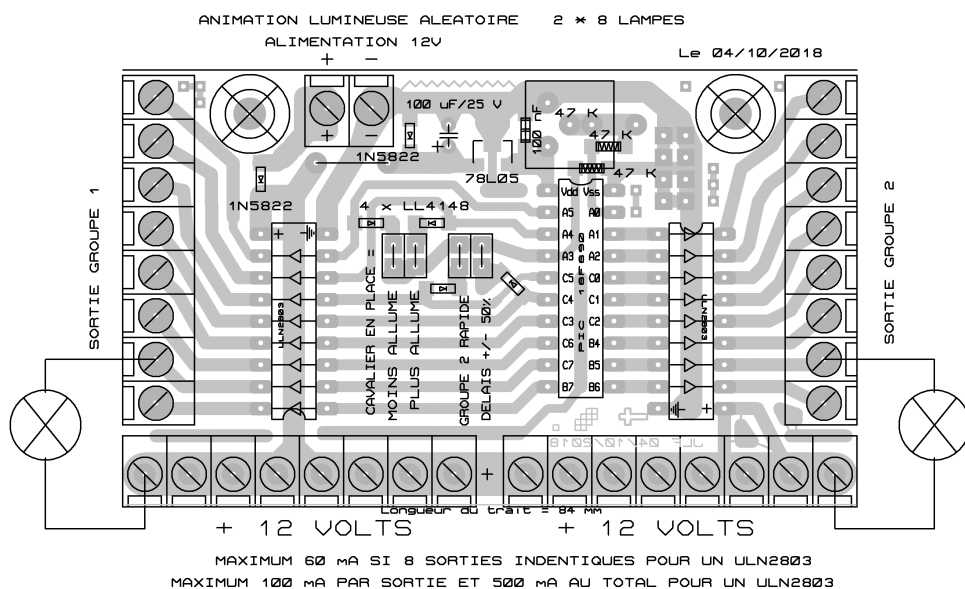
8 borniers 3 plots + 5 borniers 2 plots + cavaliers et supports de cavaliers.

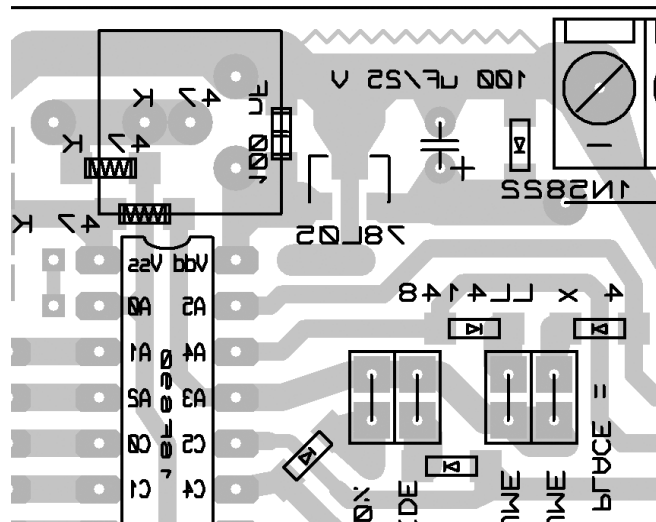
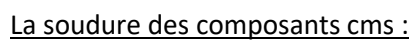
Le circuit imprimé :

Pour réaliser le circuit imprimé soi-même, 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.







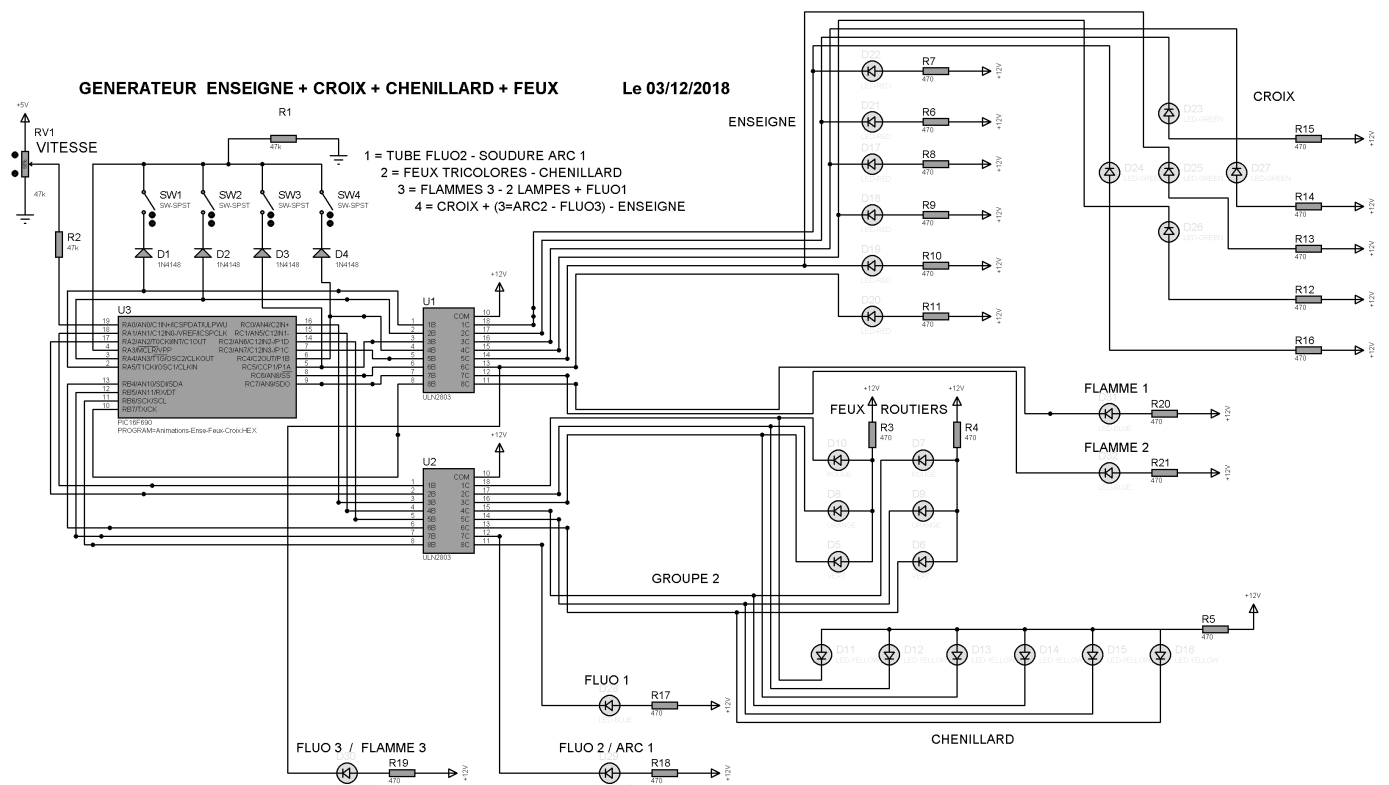
Présentation couleur de la configuration des cavaliers pour le choix des animations :

Configuration des cavaliers n° 1 à 4 :

Cavalier = Cavalier en place.

4		cavalier		cavalier		cavalier		cavalier		cavalier		cavalier		cavalier		cavalier
3			cavalier	cavalier			cavalier	cavalier			cavalier	cavalier			cavalier	cavalier
2					cavalier	cavalier	cavalier	cavalier					cavalier	cavalier	cavalier	cavalier
1									cavalier	cavalier	cavalier	cavalier	cavalier	cavalier	cavalier	cavalier
LED	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Enseigne	Enseigne	Enseigne	Enseigne	Enseigne	Enseigne	Enseigne	Enseigne	Croix	Croix	Croix	Croix	Croix	Croix	Croix	Croix
2	6 lampes								5 lampes							
3																
4																
5																
6	6 lampes	6 lampes	6 lampes	6 lampes	6 lampes	6 lampes	6 lampes	6 lampes	Tube fluo 1	Tube fluo 1	Tube fluo 1	Tube fluo 1	Soudure arc 1	Soudure arc 1	Soudure arc 1	Soudure arc 1
7	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes	Flammes
8	2 lampes	2 lampes	2 lampes	2 lampes	3 lampes	3 lampes	3 lampes	3 lampes	2 lampes	2 lampes	2 lampes	2 lampes	3 lampes	3 lampes	3 lampes	3 lampes
LED	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Chenillard		Feux tricolore	Feux tricolore	Chenillard		Feux tricolore	Feux tricolore	Chenillard		Feux tricolore	Feux tricolore	Chenillard		Feux tricolore	Feux tricolore
2			O 1	O 1			O 1	O 1			O 1	O 1			O 1	O 1
3			V 1	V 1			V 1	V 1			V 1	V 1			V 1	V 1
4			R 2	R 2			R 2	R 2			R 2	R 2			R 2	R 2
5			O 2	O 2			O 2	O 2			O 2	O 2			O 2	O 2
6	6 lampes	6 lampes	V 2	V 2	6 lampes	6 lampes	V 2	V 2	6 lampes	6 lampes	V 2	V 2	6 lampes	6 lampes	V 2	V 2
7	Soudure arc 2	Tube fluo 2	Soudure arc 2	Tube fluo 2	Soudure arc 2	Tube fluo 2	Soudure arc 2	Tube fluo 2	Soudure arc 2	Tube fluo 2	Soudure arc 2	Tube fluo 2	Soudure arc 2	Tube fluo 2	Soudure arc 2	Tube fluo 2
8	Tube fluo 3	Tube fluo 3	Tube fluo 3	Tube fluo 3	3 lampes	3 lampes	3 lampes	3 lampes	Tube fluo 3	Tube fluo 3	Tube fluo 3	Tube fluo 3	3 lampes	3 lampes	3 lampes	3 lampes

Le schéma électronique de principe :



Le programme source écrit en assembleur :

Le programme source est " Animation-Ense-Feux-Croix.asm" pour PIC 16F690.

Le fichier " Animation-Ense-Feux-Croix.hex" est directement utilisable pour programmer le PIC 16F690, avec par exemple un kit Pickit2.

Il est possible de modifier les temporisations suivantes par défaut, décrites dans le programme '.asm' :

```
; Définitions des constantes internes modifiables par l'utilisateur.

VIT_FEU EQU 153 ; Vitesse de base pour les feux tricolores (1 = rapide, 153 = moyen, 255 = lent).
VIT_CHE EQU 9 ; Vitesse de base pour le chenillard (1 = rapide, 9 = moyen, 255 = lent).

VIT_ENS EQU 31 ; Vitesse de base pour l'enseigne (1 = rapide, 31 = moyen, 255 = lent).
VIT_CRO EQU 17 ; Vitesse de base pour la croix (1 = rapide, 17 = moyen, 255 = lent).

; Tube Fluo 1 = Cinq tentatives d'allumage puis reste allumé un temps relativement long.
VIT_FL1 EQU 7 ; Vitesse de base pour le tube fluo 1 (1 = rapide, 7 = moyen, 255 = lent).
POS_FL1 EQU 8 ; Durée de pause longue durée du tube fluo 1, par pas de 5 secondes (1=5sec rapide, 8=40sec moyen, 255=20mn lent).

; Tube Fluo 2 = Nombreuses tentatives d'allumage, mais ne reste allumé qu'un court moment.
VIT_FL2 EQU 8 ; Vitesse de base pour le tube fluo 2 (1 = rapide, 8 = moyen, 255 = lent).
POS_FL2 EQU 6 ; Durée de pause longue durée du tube fluo 2, par pas de 5 secondes (1=5sec rapide, 6=30sec moyen, 255=20mn lent).

; Tube Fluo 3 = Tentatives d'allumage en continu, mais ne parvient pas à s'allumer.
VIT_FL3 EQU 9 ; Vitesse de base pour le tube fluo 3 (1 = rapide, 9 = moyen, 255 = lent).
POS_FL3 EQU 4 ; Durée de pause longue durée du tube fluo 3, par pas de 5 secondes (1=5sec rapide, 4=20sec moyen, 255=20mn lent).

; Soudure à l'arc 1 = Fonctionne 40 secs suivi d'une pause de 30 secs. De 0,3 à 0,8 secs de pauses courtes, de 5 à 21 flashes de suite.
ACT_AR1 EQU 4 ; Durée de fonctionnement de l'arc 1, par pas de 10 secondes (1=10sec rapide, 4=40sec moyen, 255=40mn lent).
POS_AR1 EQU 3 ; Durée de pause longue durée de l'arc 1, par pas de 10 secondes (1=10sec rapide, 3=30sec moyen, 255=40mn lent).

; Soudure à l'arc 2 = Fonctionne 30 secs suivi d'une pause de 40 secs. De 0,5 à 2 secs de pauses courtes, de 10 à 41 flashes de suite.
ACT_AR2 EQU 3 ; Durée de fonctionnement de l'arc 2, par pas de 10 secondes (1=10sec rapide, 3=30sec moyen, 255=40mn lent).
POS_AR2 EQU 4 ; Durée de pause longue durée de l'arc 2, par pas de 10 secondes (1=10sec rapide, 4=40sec moyen, 255=40mn lent).
```

Le programme fait tourner le pic à 8 Mhz.

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>

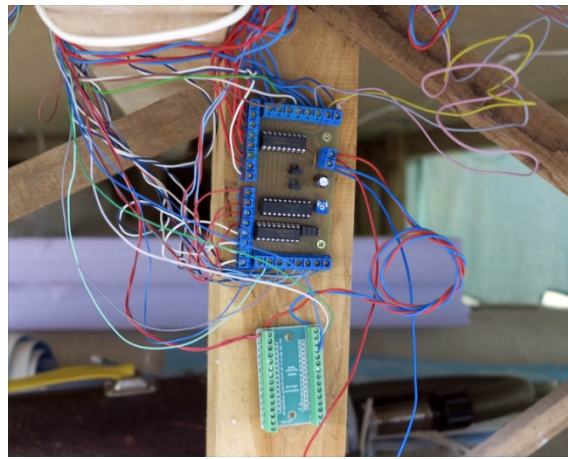
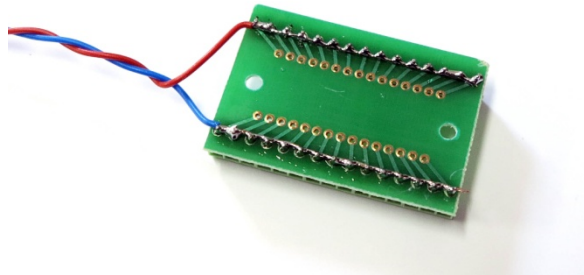
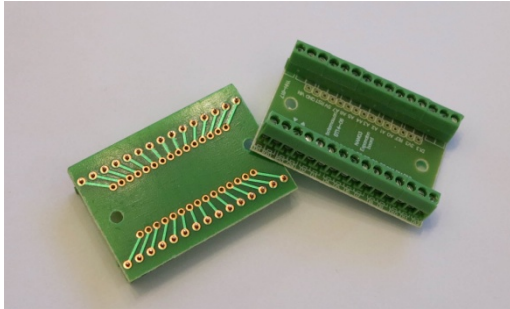
Aucune garantie de fonctionnement du montage, du programme, ou de la description de fonctionnement n'est donnée.

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 10/03/2019 ou comment mettre facilement un fichier de type '.hex' dans un PIC

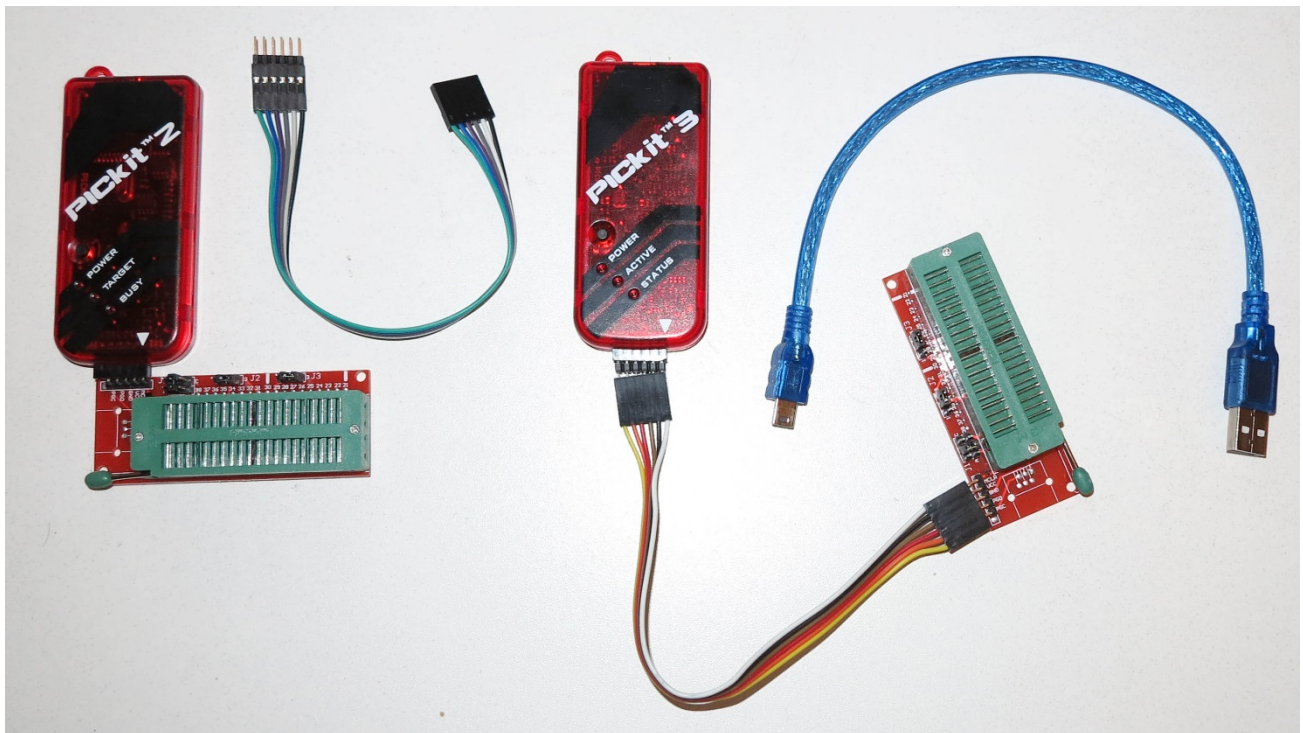
Comment programmer un PIC de type 16F84, 16F628, 16F690, 12F975 d'un montage récupéré sur Internet, à partir du fichier '.hex' ?

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 **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.

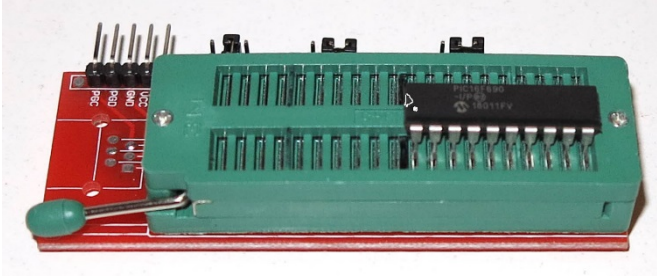


On a donc un **Pickit3**, un câble usb, un câble plat en nappe et un support de circuit intégré à force d'insertion nulle.

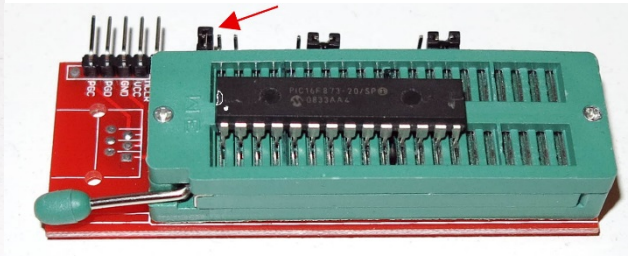
Sur le circuit imprimé, configurer les cavaliers suivant le type de PIC utilisé, en fait suivant le nombre de pattes du PIC ou format du boîtier (*Exemple : DIP28 = Dual In Line 28 = Circuit à 28 pattes*). La notice se trouve sous le circuit imprimé.

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

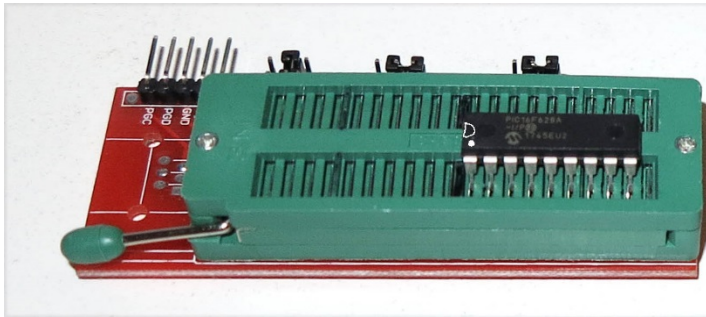
Pour faciliter la mise en place du PIC, j'ai tracé au feutre les séparations entre les pattes 4 et 5, et 10 et 11 du support. Comme cela, je colle les PIC de type DIP8, DIP18 et DIP20 au trait noir.



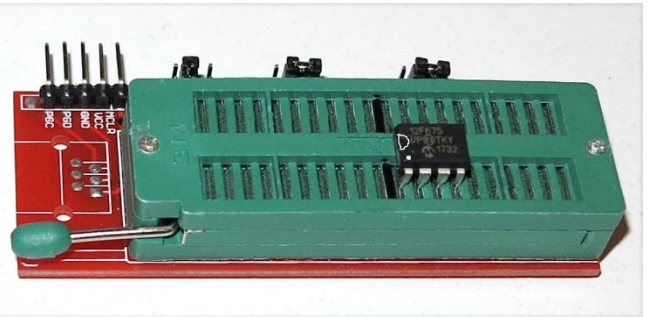
16F690 = J1 en B



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.

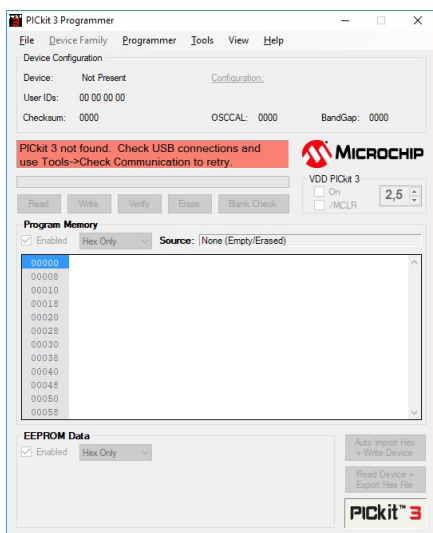
Menu : Tools > Set Unit ID, et lui donner un petit nom à votre programmeur.

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

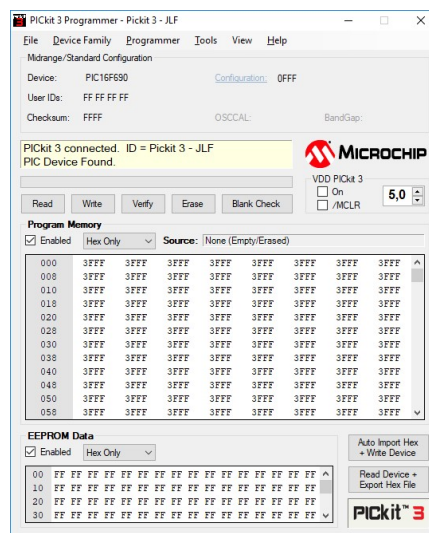
Placer un **PIC** sur le support, et cliquer sur le bouton [Erase] pour effacer le **PIC**.

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

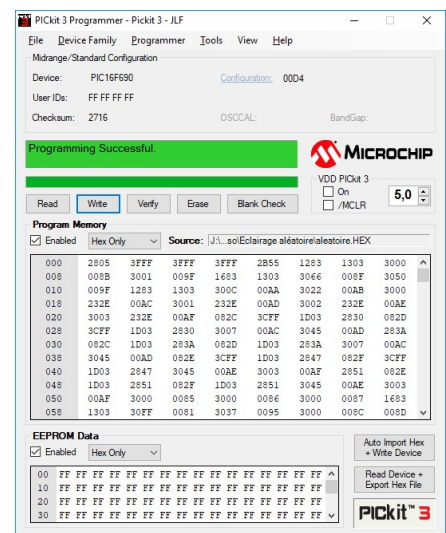
Si l'on a une erreur au moment de quitter ce programme, il faut le configurer pour qu'il se lance automatique en mode administrateur sous Windows 10. Dans l'explorateur de fichier, dans le répertoire 'C:\Program Files\Microchip\PICkit 3 v3', clic-droit sur 'PICkit3.exe' > Propriétés, Onglet 'Compatibilité'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

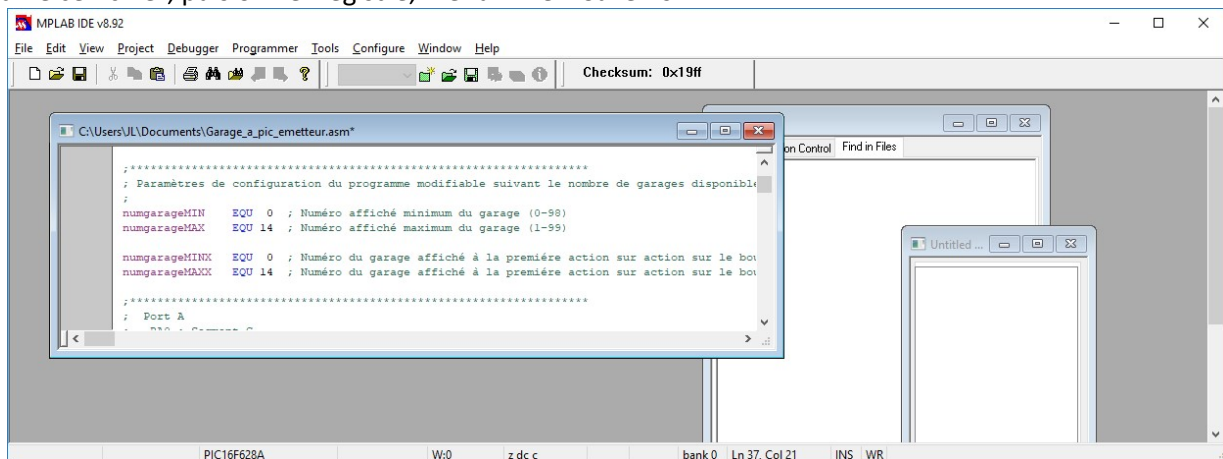
Menu File > Open, choisir par exemple le fichier 'Garage_a_pic_emetteur.asm'.

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 (-)
```

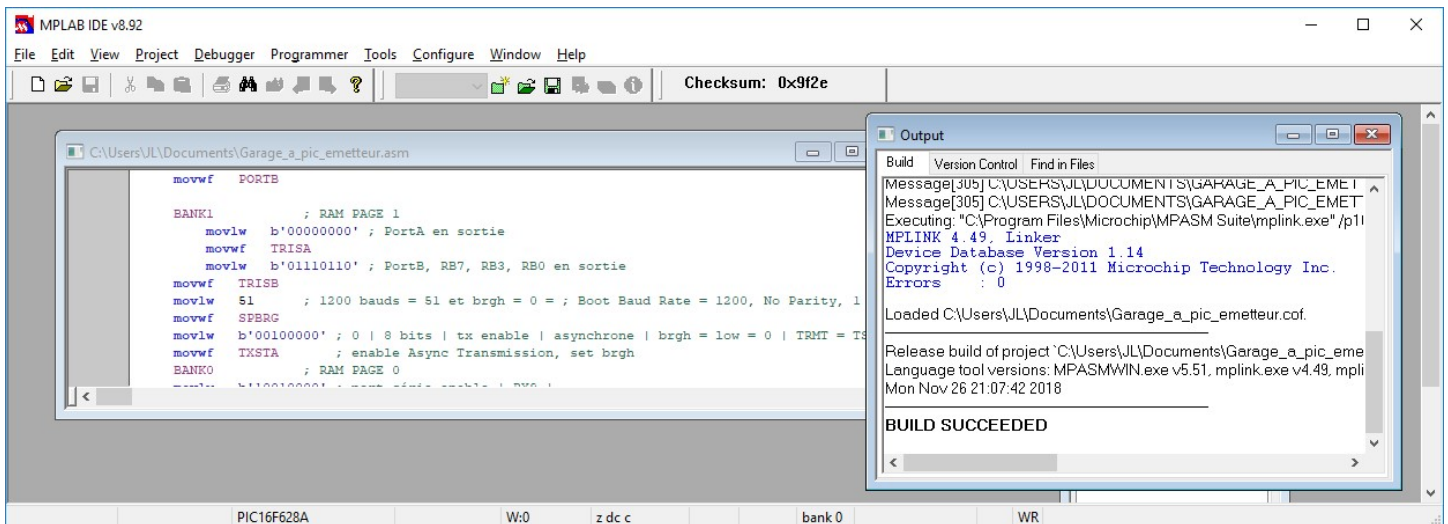
On modifie ce fichier, puis on l'enregistre, menu : File > Save As...



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**, 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://kudelsko.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+