



On peut simplifier l'affichage en ne positionnant pas les deux derniers afficheurs sur leurs supports.



### **Utilisation :**

#### **A / Interrupteur "Tour/Tronçon" fermé = Mesure sur un tour de réseau.**

On est mode "TOUR". La led jaune "Tour" est allumée.

On attend le passage du train devant le capteur 1. Le capteur 2 n'est pas utilisé.

Le train passe devant le capteur 1, le comptage commence et la led tour clignote rapidement.

On temporise pendant 3 secondes la fin de détection du premier passage du train complet, pour être sûr que tous les wagons sont passés. Si au bout de 3 secondes plus aucun wagon ne passe devant le capteur, on considère que le train est entièrement passé, et on attend les passages suivants.

Le temps de passage est au maximum de 10 minutes, sinon la led erreur s'allumera et l'on affichera "0.00".

Aux passages suivants, on affiche la vitesse.

On temporise la fin de détection du passage du train pendant 3 secondes sans coupure de la barrière, pour être sûr que tous les wagons sont passés.

La led tour clignote ensuite lentement pour les passages suivants.

Il faut donc au moins 3 secondes de délai entre le passage du dernier wagon, et le passage de la loco devant le capteur.

#### **B / Interrupteur "Tour/Tronçon" ouvert = Mesure sur un tronçon de voie.**

On est mode "100 cm, 50 cm, 20 cm ou 10 cm". La led verte "Tronçon" est allumée.

Le train passe devant le premier capteur 1 ou le capteur 2, le comptage commence et la led Tronçon clignote rapidement.

Le temps de passage est au maximum de 1 minute, sinon la led erreur s'allumera et l'on affichera "0.00".

Au passage devant le second capteur, on affiche la vitesse.

La led Tronçon ensuite clignote lentement, et l'on temporise la fin de détection du passage du train pendant 3 secondes sans coupure de la barrière, pour être sûr que tous les wagons sont passés.

La led Tronçon redevient fixe et le compteur est prêt à un nouveau cycle de comptage.

#### **C / Interrupteur "Tour/Tronçon" est modifié.**

Le changement de position de cet interrupteur réinitialise le programme, comme à la mise sous tension.

La led Tronçon ou Tour s'allume et le compteur est prêt à un nouveau cycle de comptage.

## **Fabrication :**

La consommation du montage est de 50 mA sous 12 volts, avec un mini module DC/DC. Il faut utiliser un mini module DC/DC à la place d'un régulateur 7805, comme sur la photo, sinon avec un régulateur 7805 la consommation est de 100 mA. Le régulateur 7805 même avec un radiateur chauffera beaucoup trop.

Commencer par régler le module à 5 volts, avant de placer les circuits intégrés.

La carte principale dispose d'une sortie +5 Volts pour alimenter les capteurs de passage à détection rapide.

### Liste des composants

- 1 x mini module DC/DC réglable ou sortie fixe 5 volts
- 1 x PIC 16F628A
- 1 x Quartz 4Mhz low profil
- 1 x 74HCT138 (Ne pas prendre un LS, HC, S, F qui ne permet pas d'éteindre complétement les afficheurs)
- 1 x 74LS47 (Eviter de prendre un 7447 qui consomme 2 fois plus)
- 7 x BC557C ou BC556C (pnp)
- 2 x 1N4007
- 1 x 1N4148
- 2 x supports 16 broches
- 1 x support 18 broches
- 2 x supports 40 broches en ligne pour surélever les afficheurs
  
- 6 x afficheurs 13 ou 14 mm 7 segments Anode Commune, exemple : 5161BS
- 1 x diode led jaune diffusante (ne pas prendre des leds translucides ou crystal)
- 1 x diode led verte diffusante
- 1 x diode led rouge diffusante
  
- 2 x 22 pF
- 5 x 100 nF
- 1 x 220  $\mu$ F 25 Volts
- 1 x 330  $\mu$ F 16 Volts
  
- 8 x 180 Ohms (150 à 220 Ohms suivant luminosité)
- 1 x 470 Ohms
- 7 x 2,2 kOhms
- 4 x 10 kOhms
- 5 x 33 kOhms
  
- 2 x borniers 2 plots
- 2 x borniers 3 plots
- 
- 1 x LM339
- 2 x BC547C ou BC 546C (npn)
- 1 x support 14 broches
  
- 1 x 1N4007
- 2 x Diodes IR 950nm (diamètre 3 ou 5 mm) Exemple : SFH415 = 17°, 950 nm, SFH485 = 880 nm
- 2 x Phototransistors IR 950nm (diamètre 3 ou 5 mm) Exemple : SFH205 = 950 nm, SFH 313 FA = 880 nm
- 2 x Diodes led rouge 3 mm
- 1 x Bouton poussoir pour CI
  
- 1 x 100 nF
- 3 x 1  $\mu$ F 63 Volts
  
- 2 x 220 Ohms
- 2 x 1 kOhms
- 2 x 3,3 kOhms
- 4 x 4,7 kOhms
- 1 x 47 kOhms
- 1 x 150 kOhms
- 1 x 220 kOhms
- 2 x 1 MOhms

2 x 1 kOhms Potentiomètres ajustables horizontaux

3 borniers 2 plots

3 borniers 3 plots

Les leds et les condensateurs ont leurs pattes (+) plus longue que leurs pattes (-).

Pour les composants : <http://www.stquentin-radio.com/> ou <https://www.tme.eu/fr/> ou <https://www.ebay.fr/>.

Pour le circuit imprimé, passer par un site comme : <https://jlcpcb.com/> et envoyer les fichiers Gerber. Choisir un envoi par la poste, et non pas avec DHL, pour avoir un tarif réduit.



Pour fabriquer un circuit imprimé soit même: <http://letransfertpellicule.free.fr/index.php/tutoriels/les-circuits-imprimes.html>

### **Programmation du PIC :**

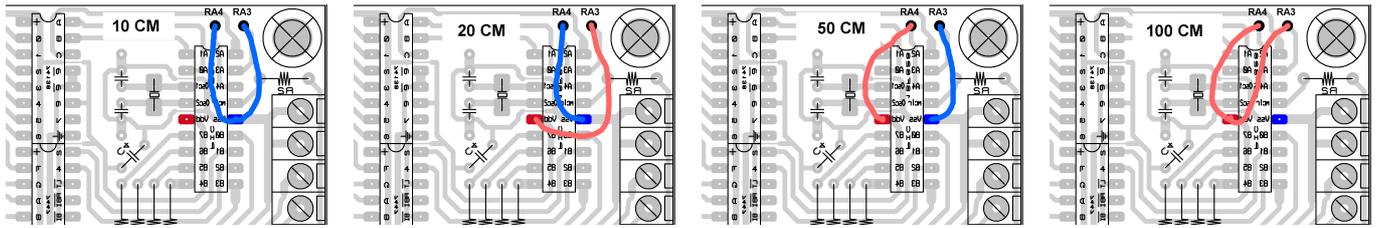
La programmation du PIC 16F628 se fait facilement. Voir en annexe ou sur le site : [http://www.la-tour.info/uts/uts\\_page14.html](http://www.la-tour.info/uts/uts_page14.html)

### **Configuration de la longueur du tour du réseau :**

Pour calculer la vitesse au tour de circuit, il faut rentrer la longueur totale du circuit dans le PIC. Voir plus loin la procédure.

## Configuration de la longueur du tronçon de 10 à 100 cm :

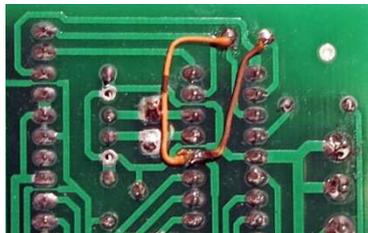
Avant utilisation, il faut souder deux fils sous le circuit imprimé, suivant la longueur du tronçon de mesure.



Deux fils à souder de RA3 et RA4 ver la borne + 5 Volts ou vers la borne à la masse.

- RA4 = masse et RA3 = masse => 10 cm
- RA4 = masse et RA3 = 5 Volts => 20 cm
- RA4 = 5 volts et RA3 = masse => 50 cm
- RA4 = 5 volts et RA3 = 5 Volts => 100 cm (Comme sur la photo)

Exemple de fils soudés pour un tronçon de 100 cm



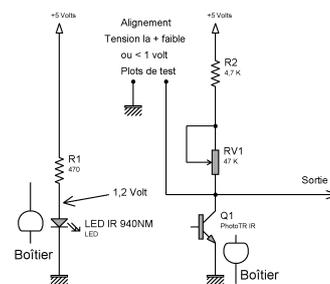
## Détecteurs de passage et les circuits de barrières infrarouge

Le circuit proposé est construit pour être très rapide, mais reste complexe à réaliser.

Il est possible d'utiliser simplement une led infrarouge en face d'un phototransistor, et de connecter celui-ci à l'entrée du compteur. L'entrée du compteur est active quand elle vaut 0 volt au passage du train.

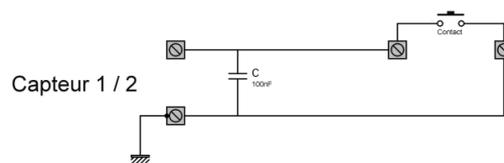
Utiliser de préférence des diodes et photo transistors de 5 mm de diamètre, beaucoup plus sensible que les 3 mm.

DETECTEUR DE PASSAGE SIMPLE



Il est aussi possible d'utiliser les rails de contact Jouef. Dans ce cas, pour éviter trop de faux contacts qui feraient clignoter l'afficheur, on ajoutera un condensateur de 100 nF aux bornes des contacts.

Dans le cas d'une distance particulière (Exemple : 47 cm), on peut modifier la définition de la longueur dans le code .asm, à la place des 100 cm, pour afficher la bonne vitesse.



Le montage de barrières infrarouge s'alimente en +5 Volts. La consommation du montage est de 10 à 30 mA.

Quand on coupe le faisceau IR, la led s'allume et la sortie passe à 0 volt.

La distante entre émetteur et récepteur IR est comprise entre 5 à 20 cm.

Une fois le circuit entièrement monté, câbler les leds et les transistors Infra Rouge.

Le phototransistor doit être monté dans un tube opaque de 2 à 3 cm de long, pour le protéger des lumières parasites.

Si une lumière importante est présente dans la pièce, il faut qu'elle illumine la diode IR plutôt que le phototransistor. C'est la diode IR qui doit être orientée vers une fenêtre.

En infrarouge, il faut que la diode et le phototransistor fonctionnent sur la même longueur d'onde 880 nm, 930 nm, 940 nm ou 950 nm. Pour notre usage, la diode infrarouge sera avec un faisceau étroit (< 20°).

Ce circuit réagit en moins de 5  $\mu$  secondes à la coupure du faisceau lumineux.

Le faisceau n'est pas modulé pour que la détection soit plus rapide. Ce montage sensible peut être perturbé par une lumière ambiante trop forte.

Le méplat des diodes et des transistors est indiqué sur le schéma et le plan d'implantation des composants.



Alimenter le montage en +5 volts.

Si les leds ne sont pas montées en vis-à-vis des phototransistors, les leds rouges sont allumées en permanence, quelle que soit la position des potentiomètres. Pour que les leds rouges s'éteignent, il faut placer une source infrarouge devant les phototransistors. (Soleil, lampe halogène, led IR).

Chaque voie doit être réglée séparément.

**Au départ, tourner les potentiomètres au maximum du sens anti-horaire, pour avoir le maximum de courant dans les leds infrarouge.**

Ces potentiomètres permettent de faire varier le courant entre 3 et 17 mA dans chaque diode IR.

Réglage 1 / Alignement des phototransistors et des Led infrarouge.

Placer un voltmètre sur les plots de test.

Le meilleur alignement est donné pour une tension la plus faible possible.

Régler donc l'alignement pour avoir le minimum de tension sur les plots de test.

Cette tension doit être inférieure à 1 volt.

Pour affiner ce réglage, on peut mettre le potentiomètre à mi-course, pour reprendre ce réglage.

Réglage 2 / Une fois cet alignement fait, appuyer sur le bouton poussoir, attendre 5 secondes et régler doucement le potentiomètre à la limite extinction - allumage de la led.

Ce réglage doit être fait lentement, à cause de la temporisation de la led rouge.

Si ce n'est pas possible et que la led reste éteinte, laisser le potentiomètre au maximum du sens horaire, cela signifie que le montage est très sensible aux 3 mA de la diode led IR, ce qui est parfait.

Quand on coupe le faisceau lumineux, la led s'allume au pendant 1 seconde et la sortie passe à l'état bas (0 volt).

Si le détecteur ne fonctionne pas, vérifier :

- les sens des composants, LM339, BC547, Condensateurs, Led.
- que l'on a 1,2 volt aux bornes de la diode infrarouge.
- que l'on a 3 volts sur la patte n° 5 du LM339.
- que l'on a 0,9 volt sur les bases des BC547C.
- que l'on a sur le plot de test < 1 volt sans obstacle.
- que l'on a sur le plot de test > 4 volts avec obstacle.
- que le bouton poussoir ne fait pas tout le temps court-circuit.

Si le détecteur ne fonctionne pas, il faut essayer en inversant le sens du phototransistor.

A l'ohmmètre, la diode infrarouge se comporte comme une vraie diode, et le phototransistor non éclairée comme une résistance de grande valeur > 100 KOhms.

**En cas de difficulté de réglage, tourner les potentiomètres au maximum du sens anti-horaire, pour avoir le maximum de courant dans les leds infrarouge.**

#### **Améliorations :**

Si de multiples impulsions sont générées en sortie du détecteur de passage, il faudrait placer une résistance de 10 K dans le circuit d'entrée (entre C3-R9 et la patte n°7[et 14]) du comparateur de sortie et mettre une résistance de 10 MOhms entre cette entrée (+) et la sortie. Ceci permet de rajouter de l'hystérésis sur ces comparateurs.

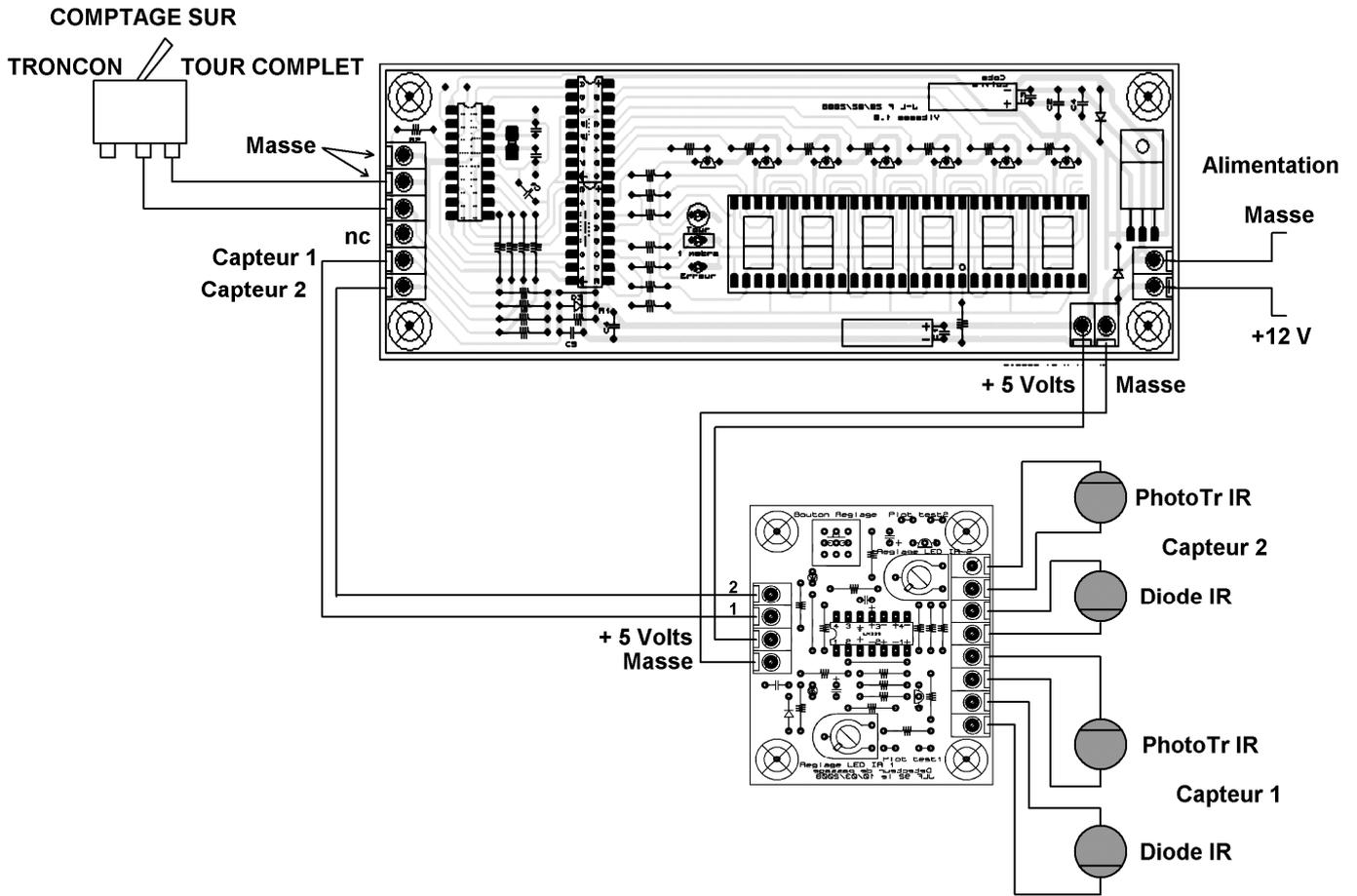
#### **Remarques :**

Comme tout montage et programme, à fortiori gratuit, celui-ci est susceptible de comporter des erreurs, connues ou inconnues. Ce montage est présenté comme une réalisation libre, sans garantie d'aucune sorte.

Théoriquement les entrées n° 3 et n° 5 du 74LS47 devraient être reliées au + 5 Volts par une résistance de 2,2K Ohms, et non pas directement.

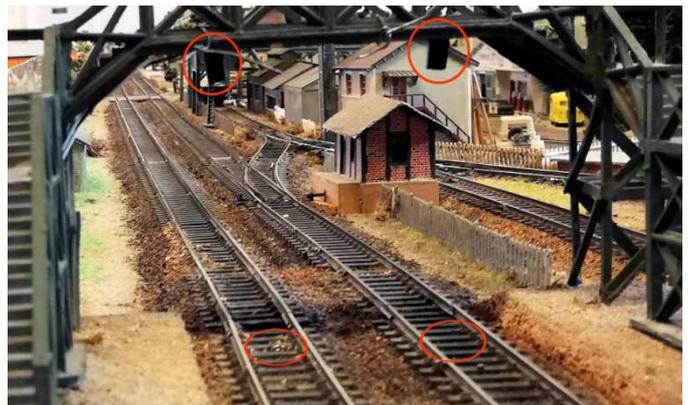
Voir le site : [http://www.la-tour.info/uts/uts\\_index.html](http://www.la-tour.info/uts/uts_index.html)

A+

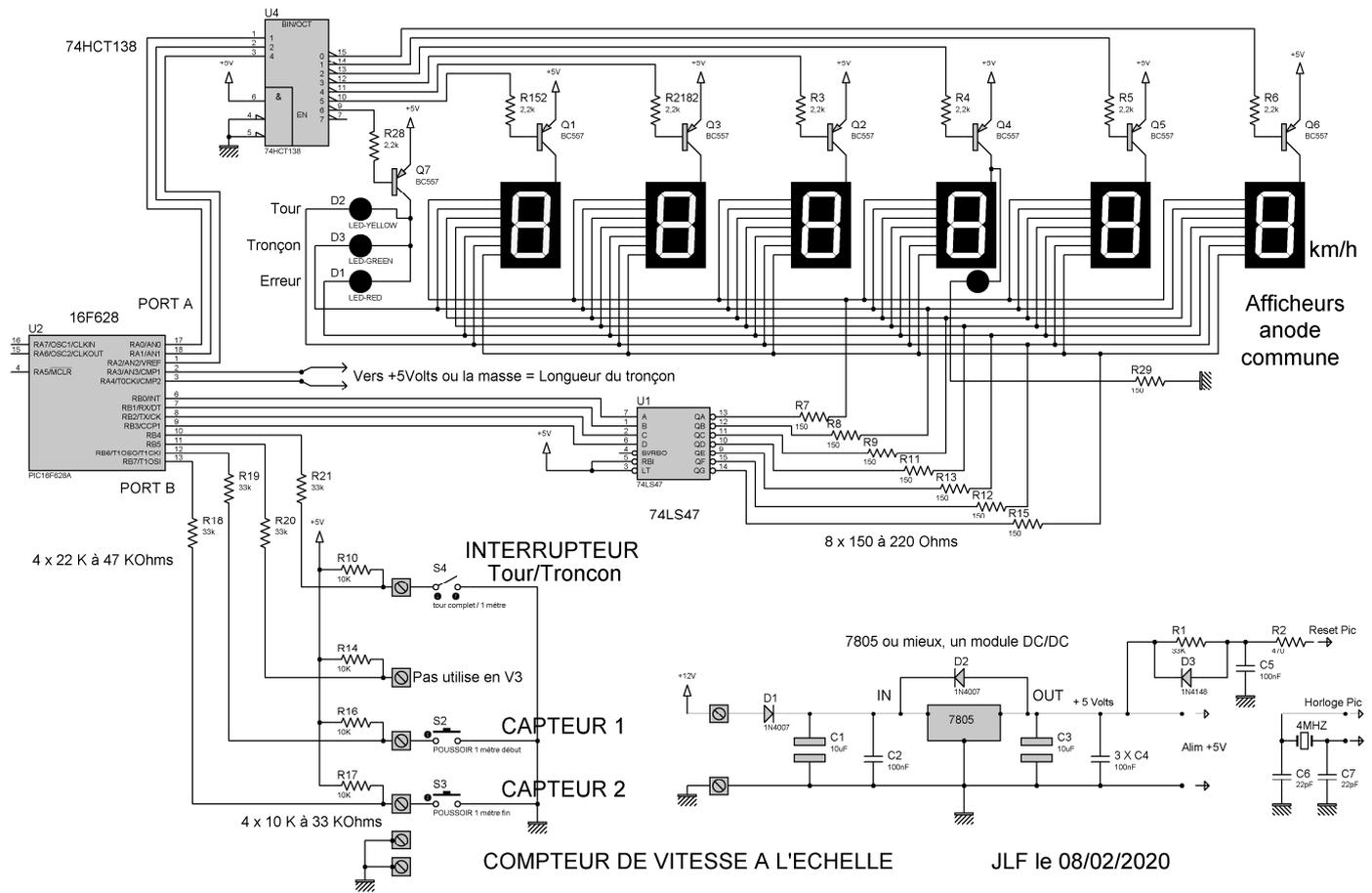


Exemples d'implantation de barrière infrarouge rapide :

Les phototransistor sont en haut, pour être protégés des rayons du soleil et de l'éclairage de la pièce.



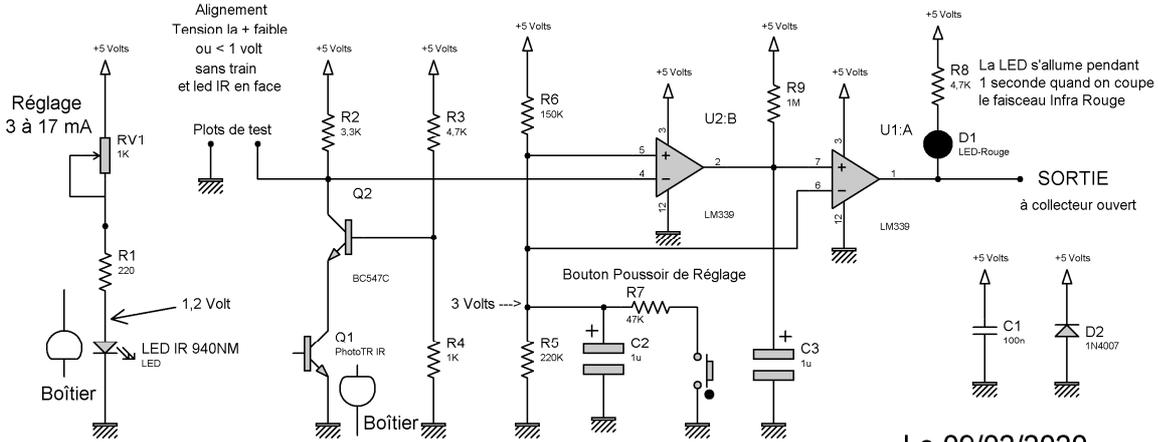
# Schéma V3 :



COMPTEUR DE VITESSE A L'ECHELLE

JLF le 08/02/2020

## DETECTEUR DE PASSAGE RAPIDE



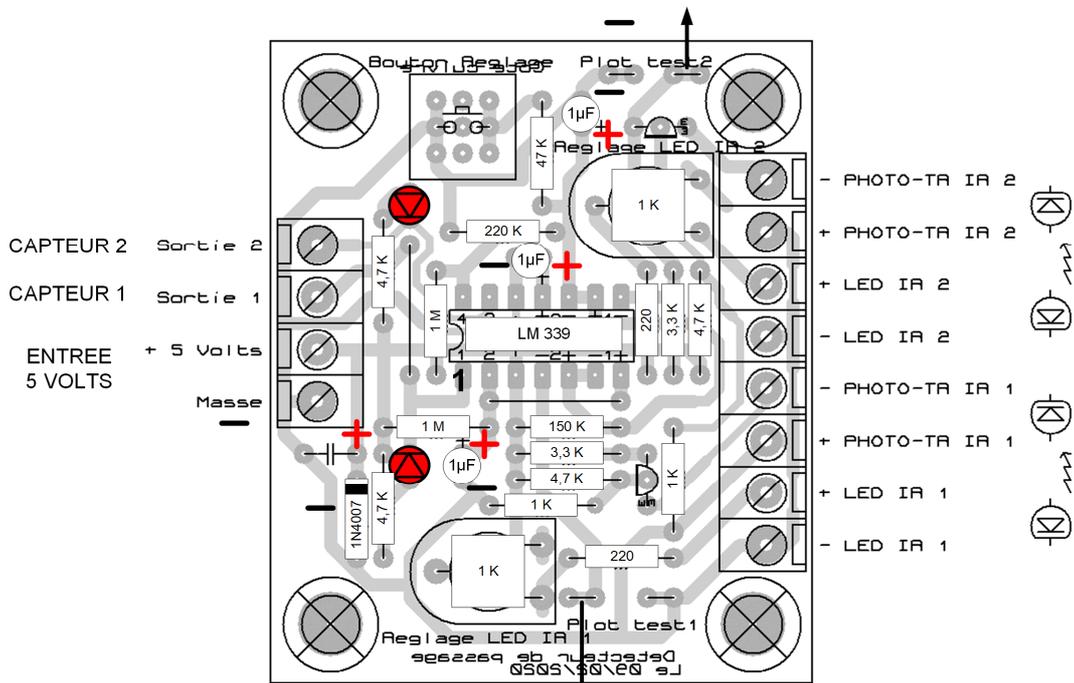
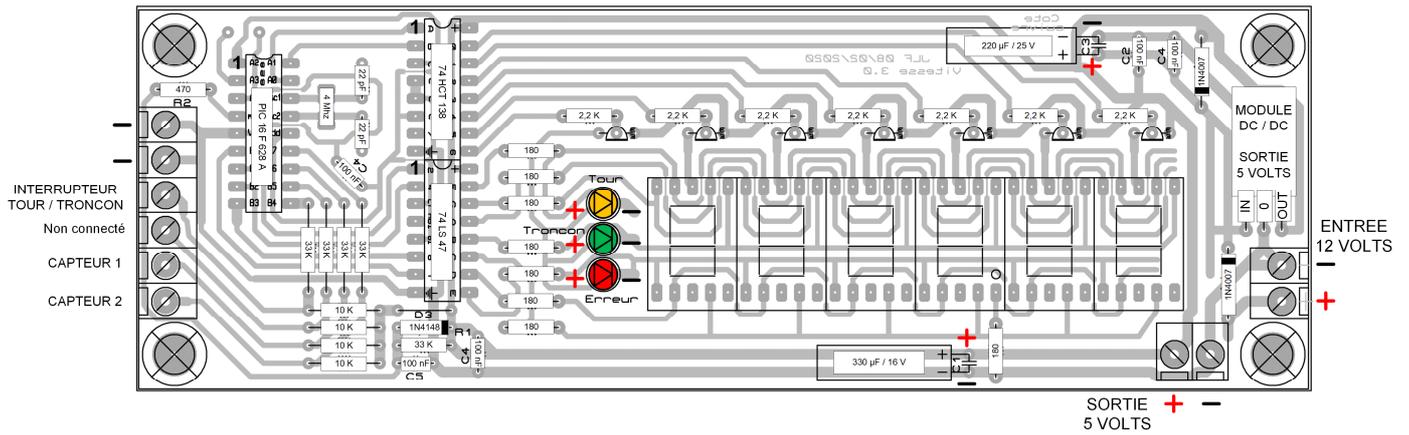
Le 09/02/2020

### Réglage 1 - Alignement :

- Placer un voltmètre sur les plots de test.
- Le meilleur alignement est donné pour une tension la plus faible possible.
- Pour faciliter ce réglage, on peut modifier le réglage du potentiomètre.
- Led éteinte = Barrière active sans train
- Led allumée = Phototransistor pas éclairé, ou passage du train

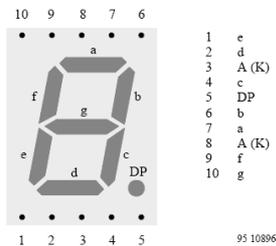
### Réglage 2 - Sensibilité :

- Appuyer sur le bouton poussoir et attendre 5 secondes
- Régler le potentiomètre à la limite extinction / allumage de la led.
- Ce réglage doit être fait lentement, à cause de la temporisation de la led rouge.
- Si réglage impossible, mettre au maximum de la position anti-horaire



Les led s'allument au passage du train.

Pin connections



Anode commune (+)

## Modification du programme pour inscrire sa longueur du tour de réseau.

Si l'on utilise que le comptage de vitesse sur un tronçon, pas besoin d'installer l'interrupteur "Tronçon/Tour de réseau".

Si l'on veut aussi disposer de la vitesse au tour, pour que l'affichage de la vitesse sur un tour de réseau soit correct, il faut donner au programme la longueur de son réseau.

On peut directement modifier le fichier "CPT\_V3.HEX", ou sinon, modifier le fichier " CPT\_V3.ASM", puis le recompiler pour obtenir un fichier "CPT\_V3.HEX" personnalisé.

Il faut d'abord mettre cette distance à l'échelle, puis la convertir en hexadécimal sur plusieurs octets.

On va utiliser la calculatrice Windows, qui est capable de faire des conversions Décimal vers Hexadécimal.

Si un tour de réseau mesure "LONG" cm de long (Exemple : LONG = 3000 cm = 30 mètres).

Ouvrir cette calculatrice en mode "Standard", et faire l'opération : LONG (en cm) \* 3,6 \* 87 \* 31250.

Exemple : 30000 \* 3,6 \* 87 \* 31250 = 29 362 500 000.

Copier le résultat (Ctrl-C).

Passer cette calculatrice en mode "Programmeur" et choisir le mode "DECimal".

Coller le résultat (Ctrl-V), on a les valeurs en DEC.( Exemple 29362500000 et en HEX = 6D62431A0).

Sélectionner le mode "HEX". Ouvrir un nouveau fichier dans Notepad et copier le résultat (Ctrl-C).

Dans Notepad, séparer ce nombre sur 6 octets. Exemple : 00 06 D6 24 31 A0. (En rouge complété par des 0 sur 6 octets).

1 / La méthode directe : Ouvrir le fichier "CPT\_V3.HEX" dans Notepad, et remplacer les 6 chiffres en rouge par votre résultat de calcul.

```
:100190004422271A6128A71D4C29A71126183B299C
```

```
:1001A0002508031DE0280030BB000630BC00D63017
```

```
:1001B000BD002430BE003130BF00A030C000232974
```

```
:1001C0002508FC3E031DF1280030BB000130BC00B7
```

2 / Ou modifier le programme assembleur " CPT\_V3.ASM ".

Dans ce cas, répartir les 6 octets sur ses 6 lignes :

```
#define divi_t1 0xA0 ; Octet de poids faible est ici en premier.
```

```
#define divi_t2 0x31
```

```
#define divi_t3 0x24
```

```
#define divi_t4 0xD6
```

```
#define divi_t5 0x06
```

```
#define divi_t6 0x00 ; Octet de poids fort est ici en dernier.
```

Et recompiler le programme comme indiqué en annexe, pour produire un fichier "CPT\_V3.HEX" personnalisé.

Pour vérifier cette configuration, passer la main pour couper la barrière infrarouge, puis après XX secondes recouper la barrière infrarouge. XX = 3,1 \* Distance en mètre du tour de réseau (Exemple : 10 mètres = 31 secondes).

On doit afficher une vitesse proche de 100 km/h.

# ANNEXE

## Mode d'emploi pour programmer un PIC Le 19/05/2021 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.

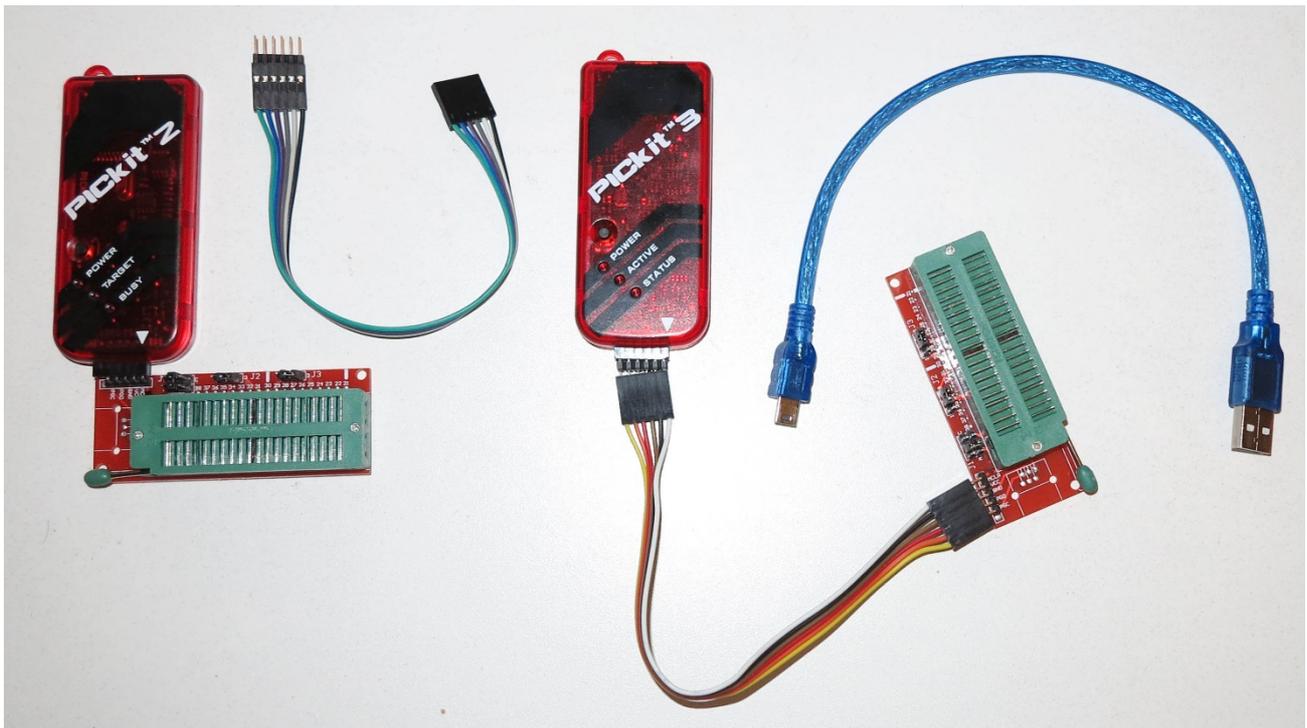
Comment programmer un PIC de type 12F675, 16F84, 16F628, 16F690, d'un montage récupéré sur Internet, à partir du fichier '.hex' ?

Si un montage à PIC du site [UTS2000](http://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](http://UTS2000). J'utilise indifféremment les **Pickit2** et **Pickit3** pour mes montages.

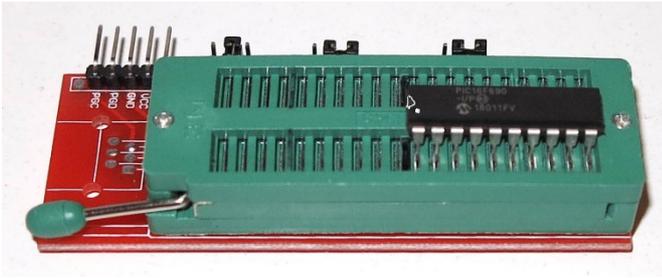


On a donc un **Pickit3**, un câble usb, un câble plat en nappe et un support de circuit intégré à force d'insertion nulle.

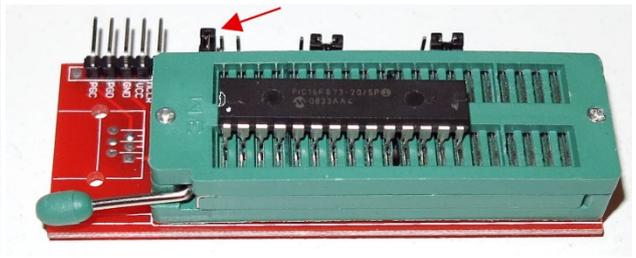
Sur le circuit imprimé, configurer les cavaliers suivant le type de PIC utilisé, en fait suivant le nombre de pattes du PIC ou format du boîtier (*Exemple : DIP28 = Dual In Line 28 = Circuit à 28 pattes*). La notice se trouve sous le circuit imprimé.

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

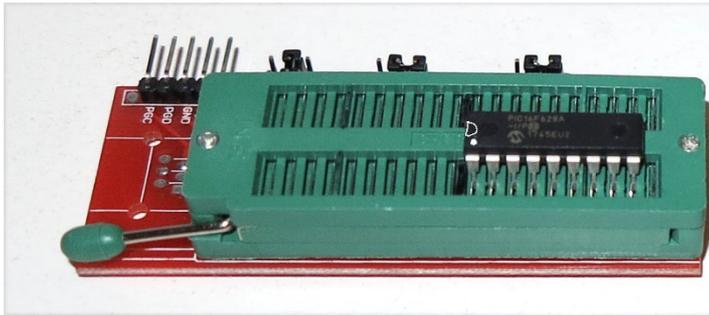
Pour faciliter la mise en place du PIC, j'ai tracé au feutre les séparations entre les pattes 4 et 5, et 10 et 11 du support. Comme cela, je colle les PIC de type DIP8, DIP18 et DIP20 au trait noir.



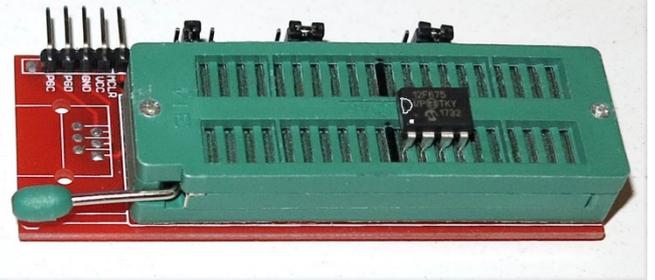
16F690 = J1 en B



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](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.

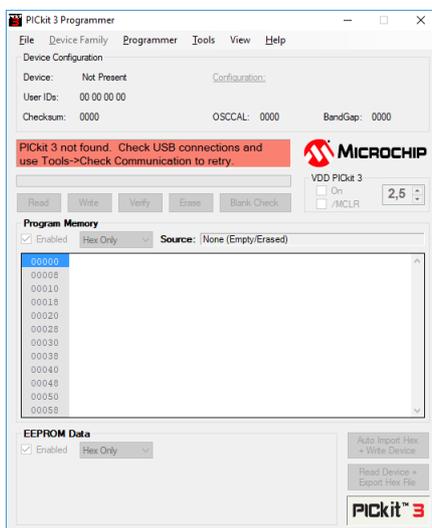
Menu : Tools > Set Unit ID, et lui donner un petit nom à votre programmeur.

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

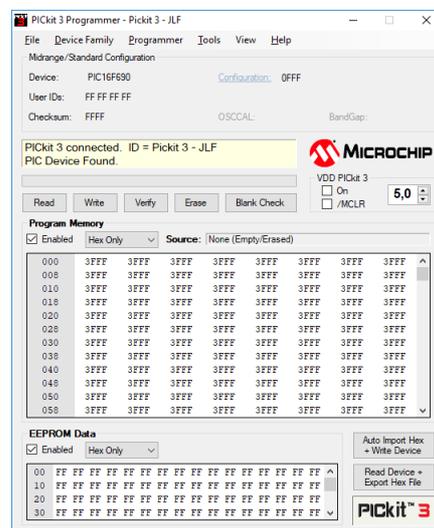
Placer un **PIC** sur le support, et cliquer sur le bouton [Erase] pour effacer le **PIC**.

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

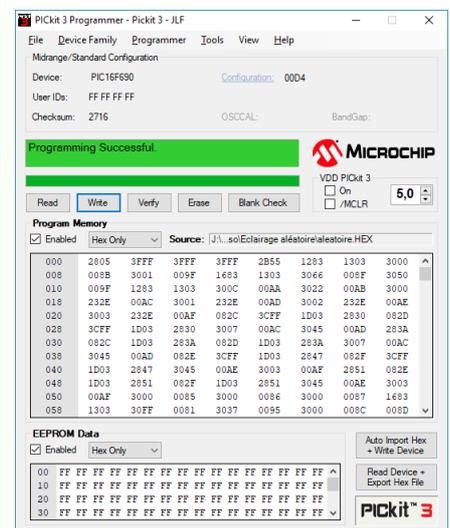
Si l'on a une erreur au moment de quitter ce programme, il faut le configurer pour qu'il se lance automatique en mode administrateur sous Windows 10. Dans l'explorateur de fichier, dans le répertoire 'C:\Program Files\Microchip\PICkit 3 v3', clic-droit sur 'PICkit3.exe' > Propriétés, Onglet 'Compatibilité'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

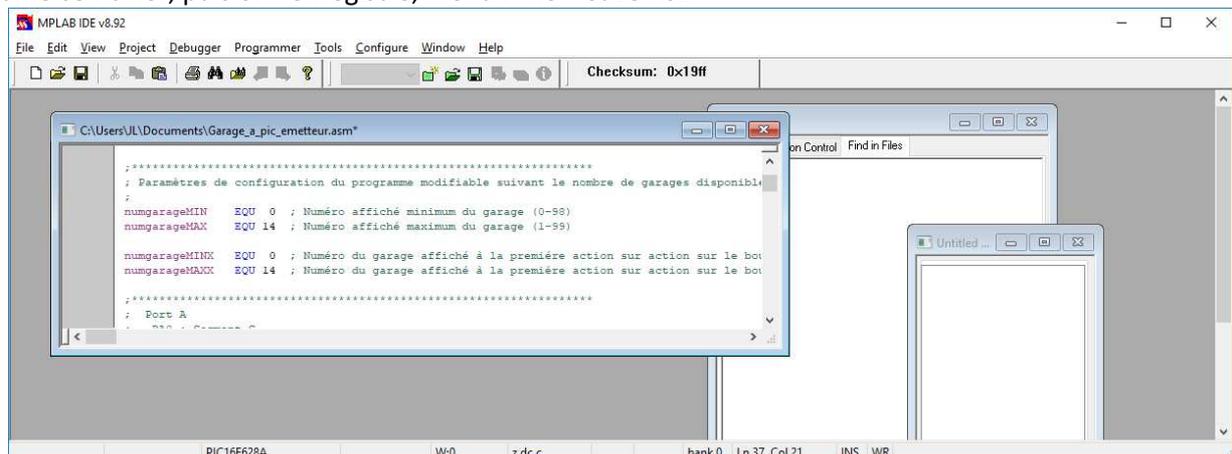
Menu File > Open, choisir par exemple le fichier 'Garage\_a\_pic\_emetteur.asm'.

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 (-)
```

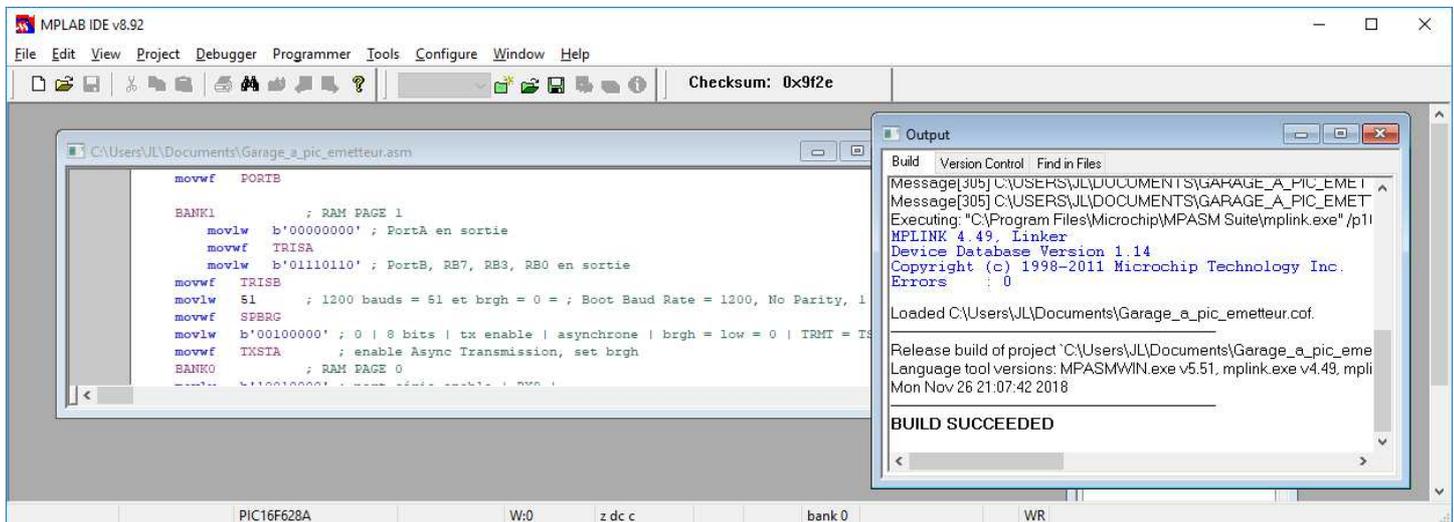
On modifie ce fichier, puis on l'enregistre, menu : File > Save As...



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](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+