

MODULE DÉTECTEUR UNIVERSEL A RÉFLEXION INFRAROUGE V1.2

Le 15/02/2023

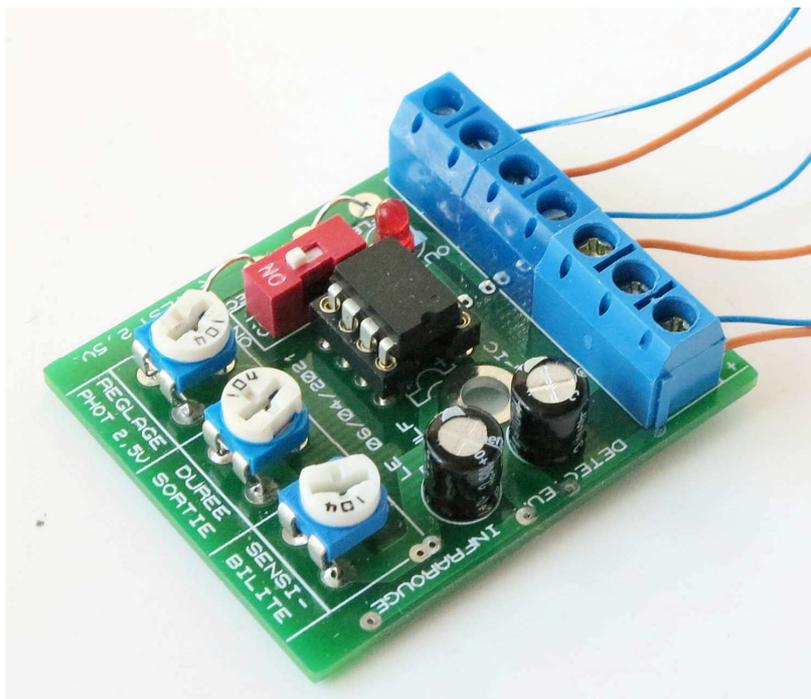
J'ai réalisé ce petit module électronique pour détecter facilement le passage d'un train, sur un réseau ferroviaire. Il est conçu pour être intégrable sur un réseau existant, en faisant simplement deux trous au milieu des traverses, ou en positionnant les éléments infrarouges sur le coté de la voie.

Ce document est sur le site UTS 2000 : http://www.la-tour.info/uts/uts_index.html .

Version V1.1 = Programme et documentation améliorés.

Version V1.2 = Un nouveau montage et une amélioration d'utilisation en BAL fictif.

C'est un montage à base de **PIC 12F675** de Microchip, qui permet d'avoir un montage particulièrement performant.



Ce montage est alimenté en 12 volts et consomme seulement 10 mA.

La sortie est à collecteur ouvert, 100 mA / 30 V max.

L'impulsion de sortie est réglable entre 0,2 et 25 secondes.

Il y a quatre réglages :

- Le choix de fonctionnement en mode réflexion sur un obstacle, ou en mode barrière infrarouge.
- Le réglage optimum pour la tension de polarisation du phototransistor.
- La sensibilité du montage, pour éliminer les fausses détections, tout en détectant bien les trains.
- La durée de l'impulsion en sortie réglable de 0,2 à 25 secondes.

Sur un réseau existant, la temporisation réglable en sortie de module permet d'animer les feux au passage des trains, sans avoir à tout recâbler. Le schéma détaillé de cette fonctionnalité est proposé à la fin du document.

La performance de ce montage le destine à fonctionner en mode réflexion. Il fonctionnera bien sûr très bien en mode direct, pour réaliser une barrière infrarouge. Ce montage ne peut pas servir de capteur pour calculer la vitesse entre deux barrières infrarouges, car la détection se fait avec un délai variable.

La portée de ce montage est de 30 cm maximum en mode réflexion, mais dépend énormément de la couleur du support. Ce montage est fait pour de la courte portée, car l'objectif est qu'il consomme peu. En utilisation normale, il faut plutôt compter sur 10 cm si l'obstacle est noir, et 50 cm si il est blanc.

Variantes au 04/01/2023

Le montage original fonctionne très bien. Je n'ai rien modifié, et le programme fonctionne parfaitement dans toutes les conditions. Il faut quand même éviter que le soleil tape sur le phototransistor, et que les rayons de la diode infrarouge atteignent indirectement (*Exemple : Par réflexion sur la tranche du rail*) le phototransistor.

Le choix des éléments infrarouge est aussi important. Le modèle "TCRT5000" fonctionne très bien.

Au montage de base, j'ai ajouté un nouveau montage pour faire clignoter directement une led rouge.

Ce deuxième montage permet d'animer un les feux d'un passage à niveau sans barrière, sans autre composant additionnel.

Le clignotement sera de type progressif, pour simuler le clignotement des anciennes lampes à filament. Voir la description complète plus loin.

Le troisième montage, possède deux sorties temporisées. La seconde sortie donne un signal deux fois plus long que la première sortie. Il sert à simuler le feu rouge et le feu orange pour un système de cantonnement fictif. Voir la description complète plus loin.

LE MONTAGE ORIGINAL - Principe de fonctionnement du programme du PIC :

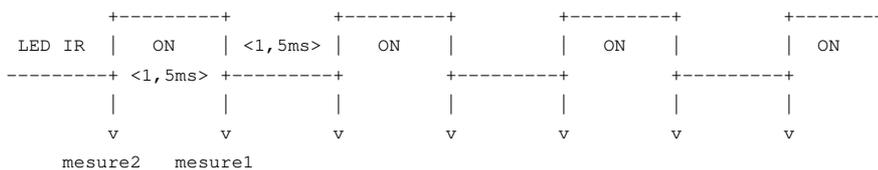
En mode barrière infrarouge, il faut absolument utiliser le programme en version V1.1.

La version V1.1 est aussi conseillée pour améliorer le mode "Réflexion infrarouge".

Détecter un obstacle par réflexion est beaucoup plus compliqué, que de détecter la coupure d'un faisceau infrarouge. La quantité de lumière reçue est plus faible, et de plus très variable suivant la couleur de l'obstacle.

Le PIC éteint la led. Il attend 1,5 msec que la tension se stabilise sur le phototransistor. Il mesure la tension V2.

Ensuite, le PIC allume la led. Il attend 1,5 msec que la tension se stabilise sur le phototransistor. Il mesure la tension V1.



La tension V1 est plus basse, quand le phototransistor est éclairé par la led.

La différence V2 - V1 augmente d'autant plus, que le phototransistor est bien éclairé par la diode led.

Le PIC compte les fois où le phototransistor est éclairé, et au bout de 20 fois active la sortie.

On a un retard de 60 msec (20 x 1,5 x 2) plus ou moins variable à la détection. **Ce montage ne peut pas servir de capteur pour calculer la vitesse entre deux barrières infrarouges.**

Quand le phototransistor est bien éclairé, le compteur interne est incrémenté de 1.

Quand le phototransistor n'est pas éclairé, le compteur est décrémenté de 3.

La sortie est activée quand le compteur est supérieur ou égal à 20.

Pour avoir un peu d'hystérésis, le compteur peut compter jusqu'à 25.

La sortie est active pendant un certain temps, réglable (0,2 à 25 sec).

Chaque détection d'obstacle réactive le délai en sortie.

Le programme est conçu pour être réactif. Un wagon roulant à 400 Km/h de 10 cm de longueur, met 78 msec à passer devant le capteur. Normalement, il doit être détecté en 60 msec.

Le programme est conçu pour être insensible aux lampes à 50 Hz. Il compte sur 60 msec = 3 demi-période à 50 Hz.

Le montage consomme que 10 mA sous 12 Volts. Pour réduire sa consommation en absence d'obstacle, ou en présence continue d'obstacle, le PIC économise l'activité de la LED en l'alimentant qu'une fois sur trois. Dès qu'un obstacle passe devant, la led sera de nouveau alimenté toutes les 3 msec, pour une détection plus rapide.



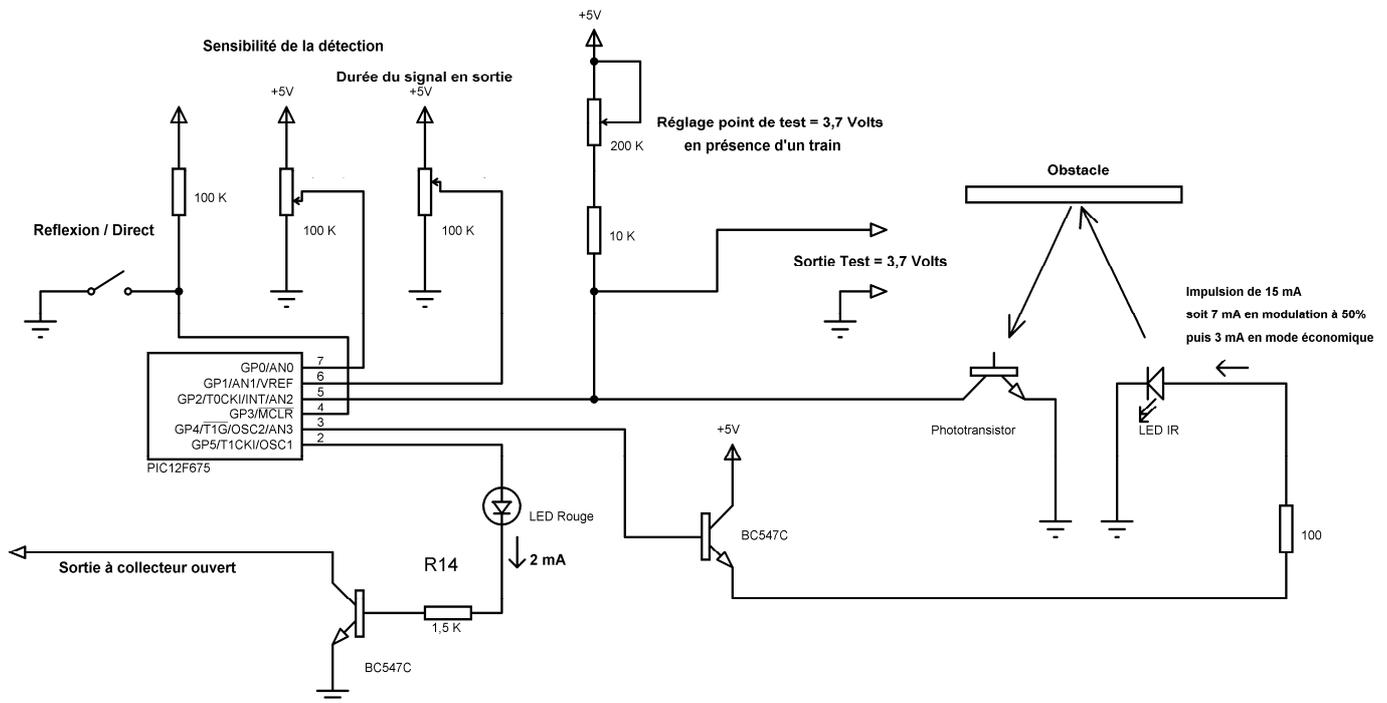
Il faut que le compteur interne reste à 0 ou à 25, 10 fois de suite, pour passer en mode économique.

La sortie est à collecteur ouvert, 100 mA / 30 V max. Si l'on commande un relais dans la limite des 100 mA, il faut impérativement placer une diode de protection en parallèle de la bobine.

Le montage peut être utilisé en mode réflexion ou en mode barrière infrarouge directe, par simple inversion de l'interrupteur.

La diode led rouge s'allume quand la sortie est activée.

Schéma électronique :



Le fichier ".hex" et le fichier source du programme assembleur est fourni. Le programme assemble ".asm" est libre de droit. Si il est utilisé dans des montages personnels ou commerciaux, il faut que le fichier source original ".asm" et les sources modifiés soient livrés avec le montage (ou au minimum disponibles librement sur le net).

Fichiers source "Detecteur IR a PIC 12F675.asm" et code pour le PIC "Detecteur IR a PIC 12F675.hex".

En mode barrière infrarouge, il faut utiliser le programme corrigé en version V1.1 du 27/05/2021.

Composants :

PIC = 12F675 de Microchip au format DIL. Lead Plastic Dual In-line (P) - PDIP.

Programme à mettre dans le PIC = "Detecteur IR à PIC 12F675.HEX".

Support CI 8 pattes = 1.

Régulateur = 7805 au format cms.

Transistors = NPN = 2 x BC 847C (Boitier marqué [1G]) *ou sinon* (BC846C, BC847B, BC847B).

Diodes = 1N4004 ou 1N4007 au format CMS DO214 ou SMA = [M4] ou [M7].

Diode et phototransistor infrarouge.

Led rouge 3 mm (*Obligatoirement rouge*).

Condensateurs = 100 nF cms 1206 + 100µF/25V radial + 220µF/16V radial.

Résistances = 100 + 1,5K + 10K + 100K cms format 1206.

Potentiomètres horizontaux = 1 x 200K + 2 x 100K linéaire. (*Attention, il existe plusieurs tailles*).

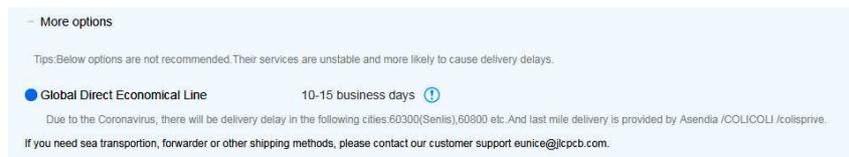
Mini-interrupteurs DIL 1 contact (*Optionnel **).

Borniers = 1 x 3 broches + 2 x 2 broches.

Circuit imprimé = Fichier à envoyer à jlpcb : "Detecteur IR a PIC 12F675 - CADCAM.ZIP".

(*) Si l'on utilise ce montage qu'en réflexion, pas besoin de souder le mini interrupteur sur le circuit imprimé.

Pour le circuit imprimé, passer par un site comme : <https://jlpcb.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.

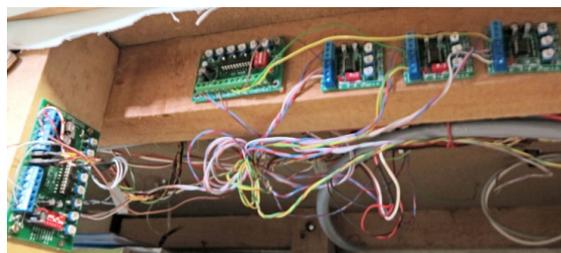


Pour fabriquer un circuit imprimé soit même: <http://letransfertpellicule.free.fr/index.php/tutoriels/les-circuits-imprimes.html> ou http://www.la-tour.info/uts/uts_page09.html

Installation des capteurs IR entre les traverses :



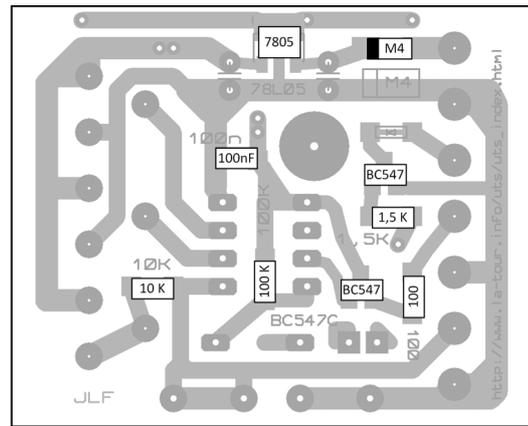
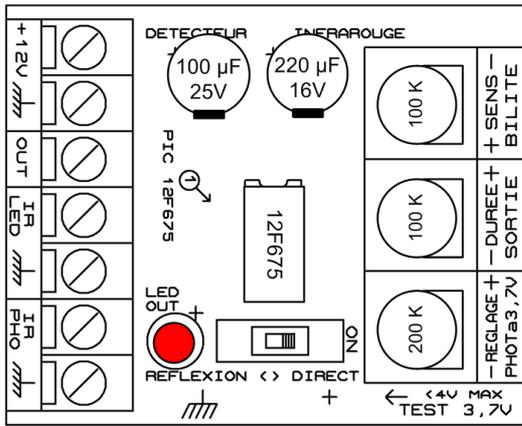
Installation des 3 modules IR + module de gestion du PN + module de commande de PN à servomoteur ([Voir le site UTS](#)) :



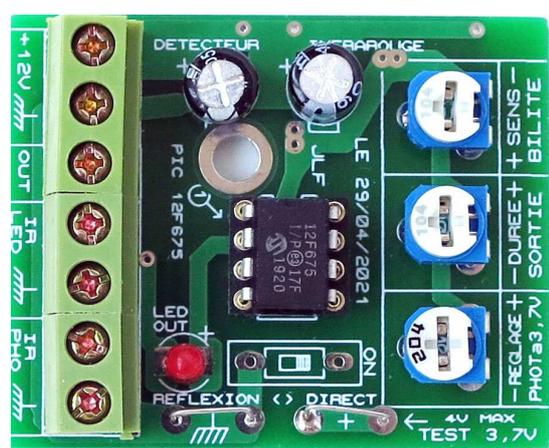
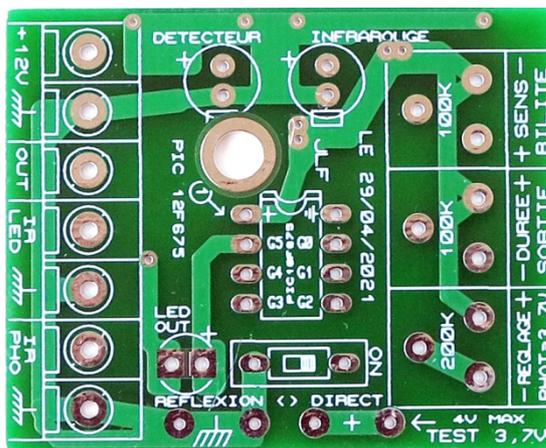
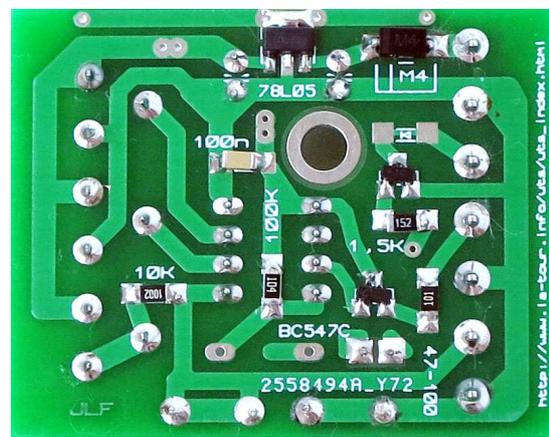
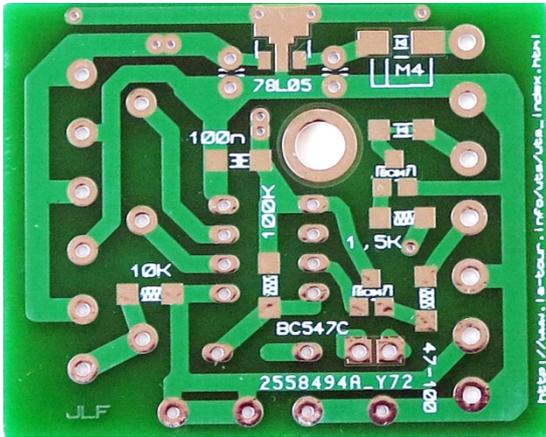
Implantation des composants :

Si l'on utilise ce montage qu'en mode réflexion, pas besoin d'installer le mini-interrupteur.

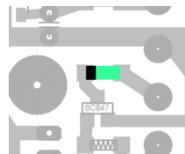
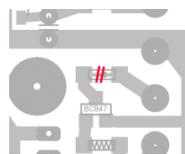
On utilisera les chutes des pattes des condensateurs, pour confectionner les plots de test.



Photos du circuit imprimé :



Pour des besoins d'interconnexions de modules, on peut installer en sortie une diode ou une résistance, on coupe la piste avec un trait de cutter avant de souder le composant.

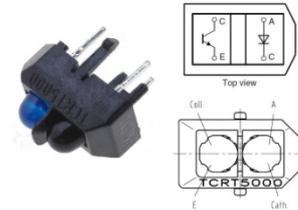


Installation et mise au point :

ATTENTION DE NE PAS FORCER SUR LA VIS EN FIXANT LE MODULE SUR LE RESEAU ! SI L'ON VISSE TROP FORT, LE CIRCUIT IMPRIME VA SE COURBER ET LES COMPOSANTS CMS RISQUENT DE SE BRISER.

Pour les éléments infrarouges, prendre des "TCRT5000".

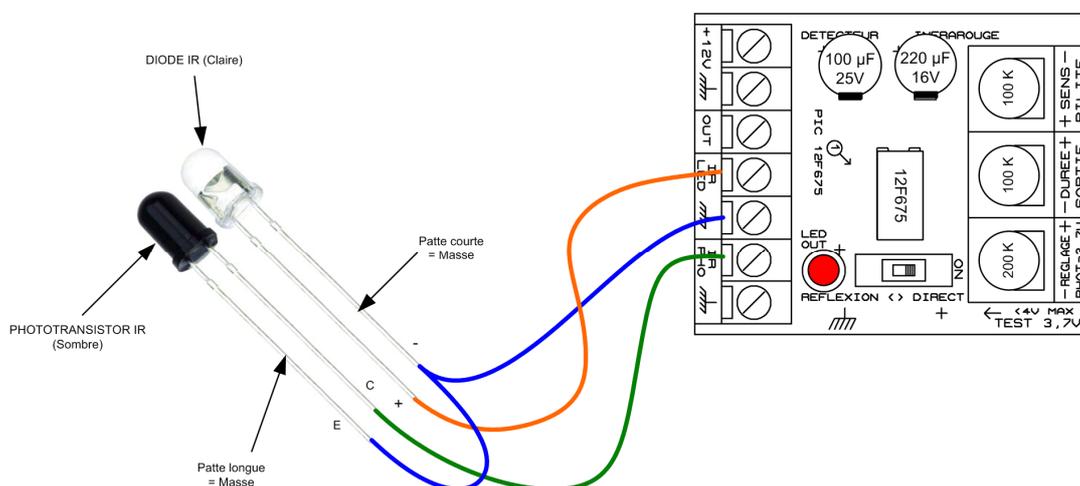
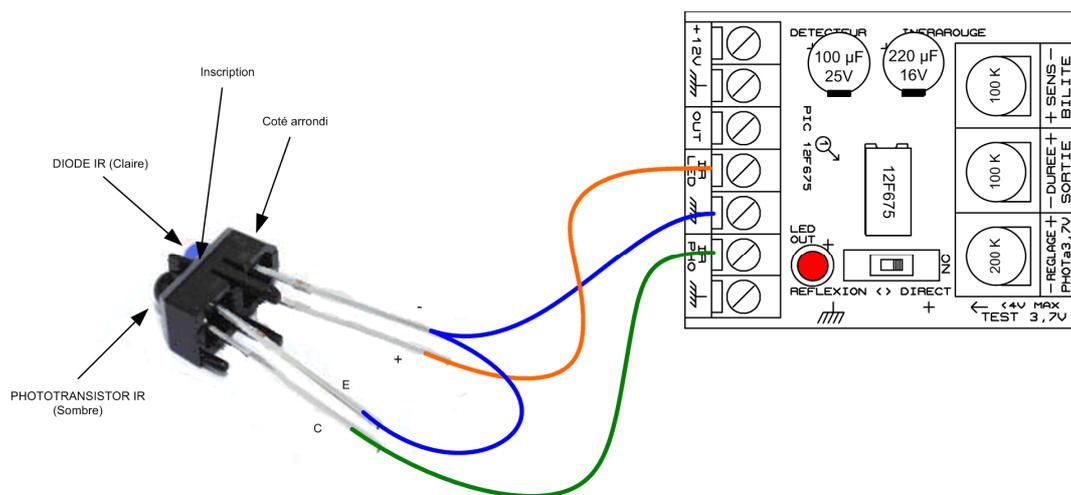
En général, la diode infrarouge est transparente. Le phototransistor lui, est noir, car il incorpore un filtre infrarouge.



La patte "E" et la patte "Cathode" sont à relier à la masse. On peut avoir un fil de masse commun, et une longueur de 2 mètres entre les éléments infrarouges et le circuit imprimé.

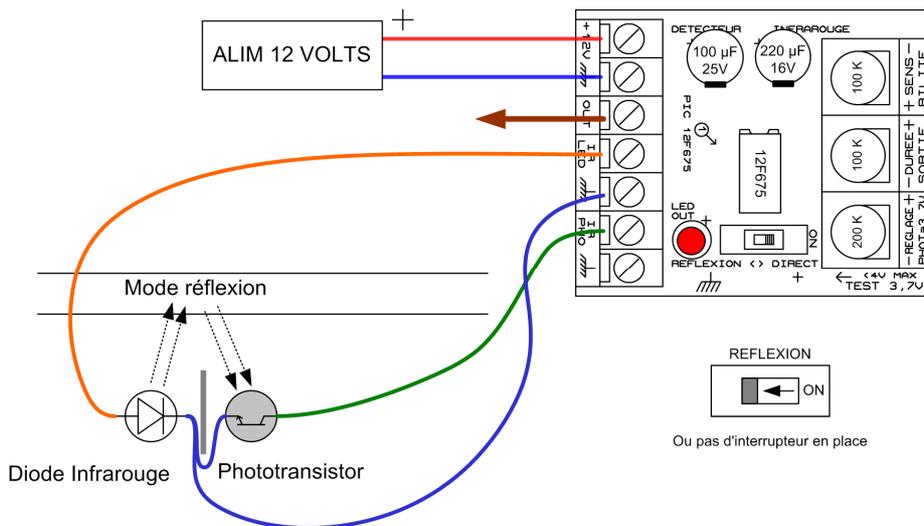
En cas d'éléments séparés, le (-) du phototransistor est la patte la plus longue. Avec un multimètre, si l'on vérifie ce composant non éclairé, c'est un circuit ouvert. Sous un éclairage direct d'une lampe halogène ou du soleil, on peut vérifier ce composant comme on le ferait avec une résistance inférieure à 10 KOhms.

Le (-) de la diode infrarouge est la patte la plus courte. Avec un multimètre, on peut vérifier ce composant comme on le ferait avec une diode ordinaire.



Alimenter le montage et vérifier le +5 Volts entre les pattes 1 et 8. Mettre hors tension, puis placer le PIC sur son support.

1 / Schéma de câblage en mode réflexion :

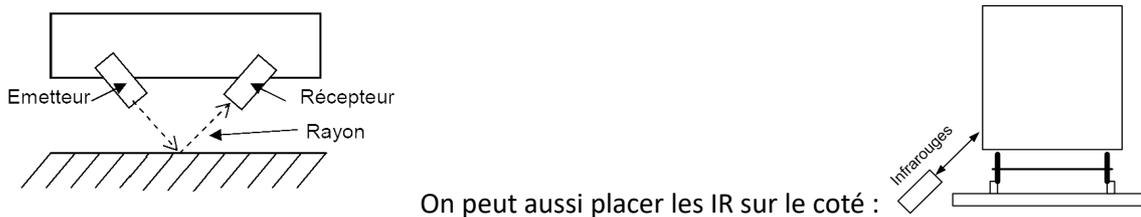


Bien protéger le phototransistor d'une émission directe de la led infrarouge. C'est très important !!!

Dans ce mode Réflexion, on détecte un objet blanc à 50 cm ou un objet noir à 10 cm. Ne pas placer le capteur en face d'une surface blanche, même éloignée.

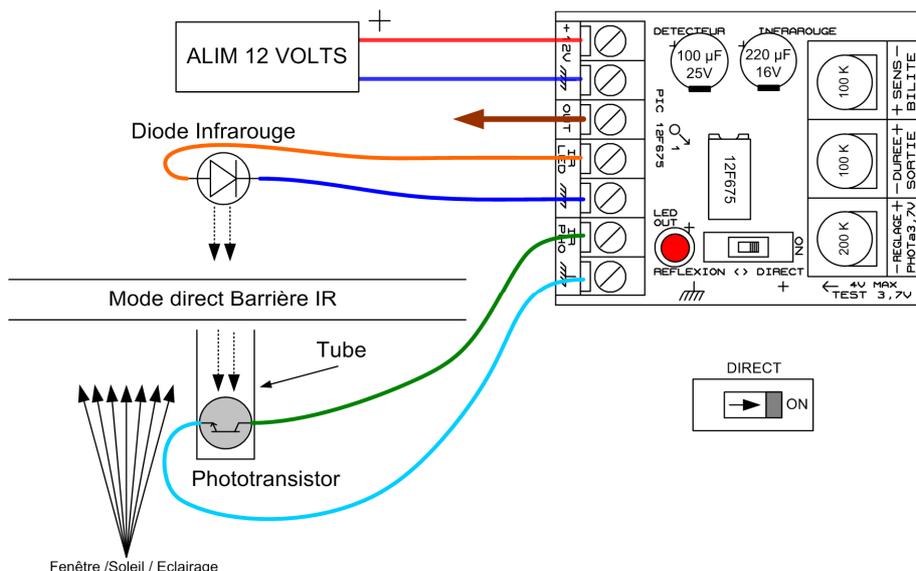
Le phototransistor peut être perturbé par un rayon IR renvoyé par le côté d'un rail brillant !

Si la diode et le phototransistor ont un faisceau IR concentré, la distance où les faisceaux se croisent doit être la distance à laquelle passe le train.



On peut aussi placer les IR sur le côté :

2 / Schéma de câblage en mode direct de type barrière infrarouge :



Pour régler ce montage en vue directe, aligner les éléments infrarouges pour avoir le minimum de tension sur les plots de mesure du circuit imprimé.

Une fois le montage en place, sur le réseau,

- Mettre l'interrupteur en bonne position.
- Régler la durée de sortie au minimum (0,2 seconde).

Mettre les réglages par défaut dans ces positions :

En mode réflexion infrarouge :

TENSION AU MINIMUM ↘	0,2 SECONDE	MOYENNEMENT SENSIBLE
- RÉGLAGE PHOTO +	- DURÉE SORTIE +	+ SENSIBILITÉ -

En mode barrière infrarouge directe :

TENSION MOYENNE	0,2 SECONDE	MOYENNEMENT SENSIBLE
- RÉGLAGE PHOTO +	- DURÉE SORTIE +	+ SENSIBILITÉ -

Placer un wagon avec le dessous noir, centré au-dessus du capteur.

1 / Régler le potentiomètre pour avoir entre 3,5 et 3,7 Volts sur le point de mesure, sinon laisser le réglage par défaut.

Si besoin, vérifier l'éclairage de la diode Infrarouge, avec un Smartphone en photographiant la led.

2 / Le réglage de sensibilité doit se faire avec un wagon à fond noir.

La sensibilité dépend du wagon, un dessous noir étant moins réfléchissant, il est plus difficile à détecter.

Régler ce potentiomètre pour avoir une bonne sensibilité, quitte à la réduire plus tard.

Le wagon doit être centré, car les essieux brillants réfléchissent énormément les IR, alors que l'on doit régler le module sur le fond noir du wagon. **Si le montage fonctionne mal, tester en inversant les connexions du phototransistor. Si le montage fonctionne pas du tout, tester en inversant les connexions de la diode IR.**

On peut aussi visualiser sur un oscilloscope, la tension du point de test, pour avoir le maximum d'amplitude du signal.

Une fois bien réglé, le module détecte aussi les attelages, ce qui fait qu'un train est détecté de manière continue, sans coupure entre wagons.

3 / Au final, régler la temporisation de sortie entre 0,2 à 25 secondes (12 secondes au point milieu).

TENSION ↘	TENSION ↗	0,2 SECONDE	25 SECONDES	TRÈS SENSIBLE	PEU SENSIBLE
- RÉGLAGE PHOTO +		- DURÉE SORTIE +		+ SENSIBILITÉ -	

Le module a un trou de fixation, pour permettre de le visser facilement sous le réseau. Ne pas visser trop fort.

Led et phototransistor infrarouge.

Il est probable de devoir reprendre le réglage de la sensibilité, quand l'environnement change. Par exemple, on retire un pont ou un bâtiment pour installer ce module, mais ensuite quand on repositionne le tout, le réglage risque de devoir être repris.

Ce montage est très sensible. Si par exemple, il est placé dans un tunnel, il va détecter le haut du tunnel. Si un bâtiment blanc est au bord des voies il sera aussi détecté.

En cas de problème, il faut orienter différemment les éléments infrarouges ou les placer en biais. Sinon, placer le phototransistor au fond d'un tube noir, ou d'un bout de gaine thermo rétractable, pour diminuer les rayons parasites.

Le phototransistor peut être perturbé par un rayon IR, renvoyé par le coté d'un rail !

J'ai fait des essais avec ce type d'élément qui fonctionne très bien :



Eviter les rayons de soleil directs sur le phototransistor.

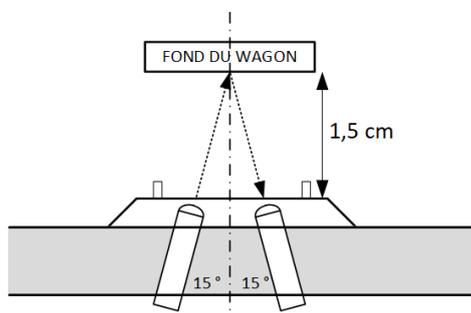
Par défaut, on utilisera des éléments infrarouges de 3 mm de diamètre, ce qui donne déjà de bons résultats. Pour un montage encore plus performant, utiliser des éléments infrarouges de 5 mm.

On peut placer le capteur de réflexion infrarouge sous la voie entre deux traverses, sur le coté du train, ou à coté de la voie.

Pour une détection de plus grande portée en mode barrière infrarouge, utiliser des éléments infrarouges de 5 mm de diamètre à faible cône de lumière, soit un angle d'environ $20^\circ (\pm 10^\circ)$. Dans ce cas l'alignement des éléments infrarouges sera plus délicat à faire.

Pour la pose des derniers détecteurs, j'ai opté finalement pour une pose plus facile, et qui abime moins la voie. Je perce deux trous avec un foret de 3,5 mm.

Il faut gainer une des deux pattes des éléments infrarouges, pour éviter des courts-circuits entre ces deux pattes. Et au final, le montage est plus discret sur la voie.



J'ai du gainer les deux éléments infrarouges, sinon on a trop de lumière parasite sur les cotés de la diode infrarouge. C'est aussi indispensable pour diminuer la sensibilité à la lumière ambiante. On place juste deux bouts de gaine thermo-rétractable, sans les chauffer.



Deux détecteurs en place.



Pour régler ce module, le mieux est d'utiliser un petit oscilloscope de poche.

On en trouve à moins de 50 euros sur Ebay, voir moins de 30 euros sans boîtier.

Chercher : Oscilloscope 2.4 5 mhz avec l'option ☉ "Monde entier" de coché.

[J'ai choisi ce modèle fonctionnel, autonome et simple d'emploi](#) : Ecran 2,4 pouces couleur, bande passante 5 Mhz.



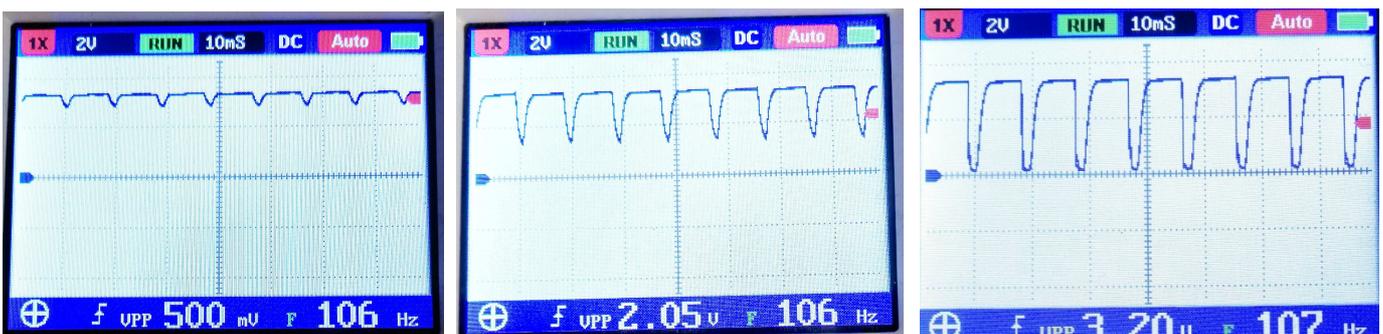
Le brancher sur les plots de test. Après la mise sous tension, un clic sur \uparrow pour passer à 2 volts/carreau et 4 ou 5 clics sur \leftarrow pour passer à 10 msec par carreau.

On pose un wagon à fond noir à l'aplomb du capteur et l'on règle le potentiomètre de 200 KOhms, pour que les pics descendent jusqu'à 0 volt.

Cela permet de vérifier que le phototransistor n'est pas perturbé par une autre source de lumière.

Il faut au moins des pics de 3 volts d'amplitude.

Si le montage est bien fait, on obtient un signal comme sur l'écran de droite. C'est parfait, car le signal descend bien jusqu'à 0 volts et monte à plus de 3 Volts.



Si le signal descend sous 3 volts, sans modulation, c'est que le phototransistor est trop exposé à la lumière ambiante.

La modification des paramètres du programme source en assembleur est réalisable en modifiant ces lignes, on modifie le comportement du programme :

```
; Déclaration des constantes :
cpt_obs_max    equ .25    ; Valeur maximum du comptage de top IR. On bloque à cette valeur. (cpt_obs_det < cpt_obs_max < 255).
cpt_obs_inc    equ .1     ; En cas de détection IR, on incrémente le compteur de top IR de cette valeur. (1 à 5).
cpt_obs_dec    equ .3     ; En cas de non détection IR, on décrémente le compteur de top IR de cette valeur. (1 à 10).
cpt_obs_det    equ .20    ; Seuil de détection pour activer la sortie. Si le compteur de top IR dépasse cette valeur, on active la sortie.
                    ; cpt_obs_det < cpt_obs_max.
cpt_eco_max    equ .5     ; Nombre de cycle sans activation de la led IR en mode économe 5 = 1/3 de cycle.
max_eco        equ .10    ; Pour passer en mode économique, nb de fois ou le compteur = 0 ou 25.
```

Ce montage peut alimenter directement mes autres montages :

- Le module de commande de barrières de passage a niveau à servomoteur : http://www.la-tour.info/uts/uts_page13.html#pn_srv
- Le module de gestion du passage des trains pour commander un passage à niveau : http://www.la-tour.info/uts/uts_page13.html#pn_srv
- Un montage pour animer les feux SNCF : http://www.la-tour.info/uts/uts_page13.html#feu_sncf

Utilisation de la sortie pour animer un Bal fictif à moindre coût :

Utilisation avec mon module "Montage pour animer les feux SNCF" : http://www.la-tour.info/uts/uts_page13.html#feu_sncf

Avec la sortie temporisée de ce détecteur à réflexion, il est possible de reproduire le fonctionnement d'un BAL au passage des trains. Si l'on commande un relais dans la limite des 100 mA, il faut impérativement placer une diode de protection en parallèle de la bobine.

Sur un réseau déjà câblé, c'est extrêmement pratique pour ajouter une animation dynamique des feux, au passage des trains. Le câblage est réduit à quelque fils d'interconnexion, avec les feux animés par la position des aiguillages et le mouvement des trains.

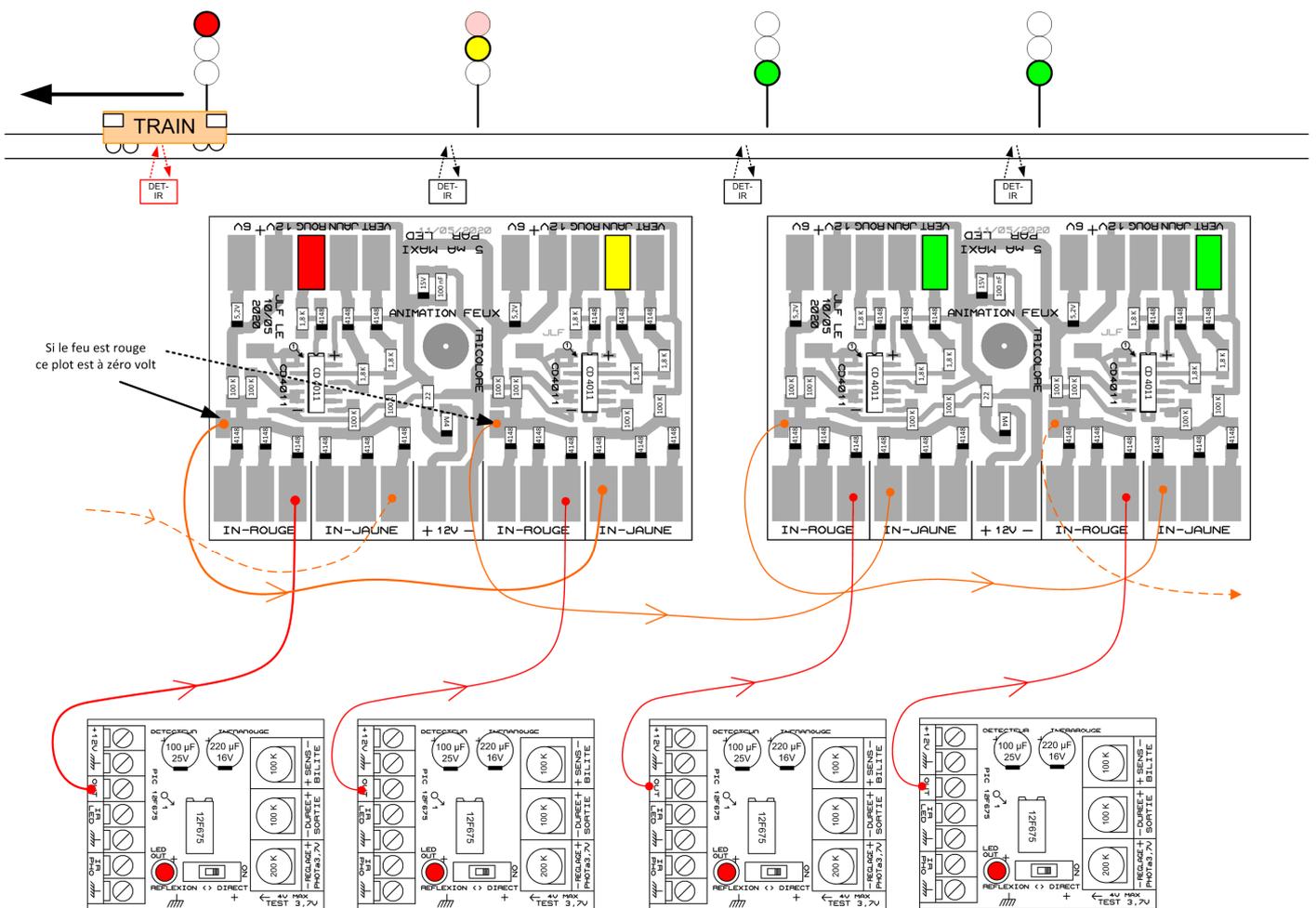
On ajuste la temporisation sur chaque détecteur (5 à 20 sec), pour avoir un BAL réaliste.

Quand le train passe devant un détecteur, le signal passe au rouge et le signal précédent au jaune. Quand le train quitte ce canton, les feux restent dans cette position pendant la temporisation réglable du détecteur.

Il faut simplement placer un détecteur infrarouge après chaque feu.

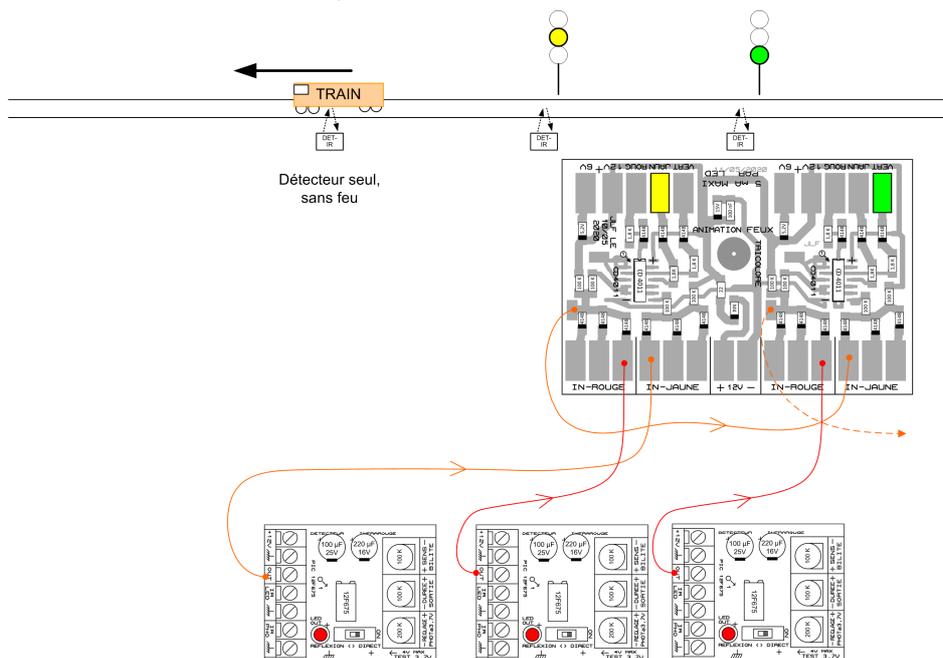
Avec la temporisation intégrée au 12F675, on fait l'économie d'utilisation de mes modules BAL-FICTIF.

Exemple d'implantation pour un BAL-FICTIF animé au passage du train :



AJOUT V1.2. Quand on ajoute ce système de BAL fictif sur une portion d'un réseau, le premier feu SNCF va repasser directement du rouge au vert, sans passer par le jaune, lors du passage d'un train.

Pour que ce premier signal passe aussi par le jaune, il faut ajouter plus loin sur la voie un détecteur de passage qui sera relié à la commande "Jaune" du circuit imprimé.



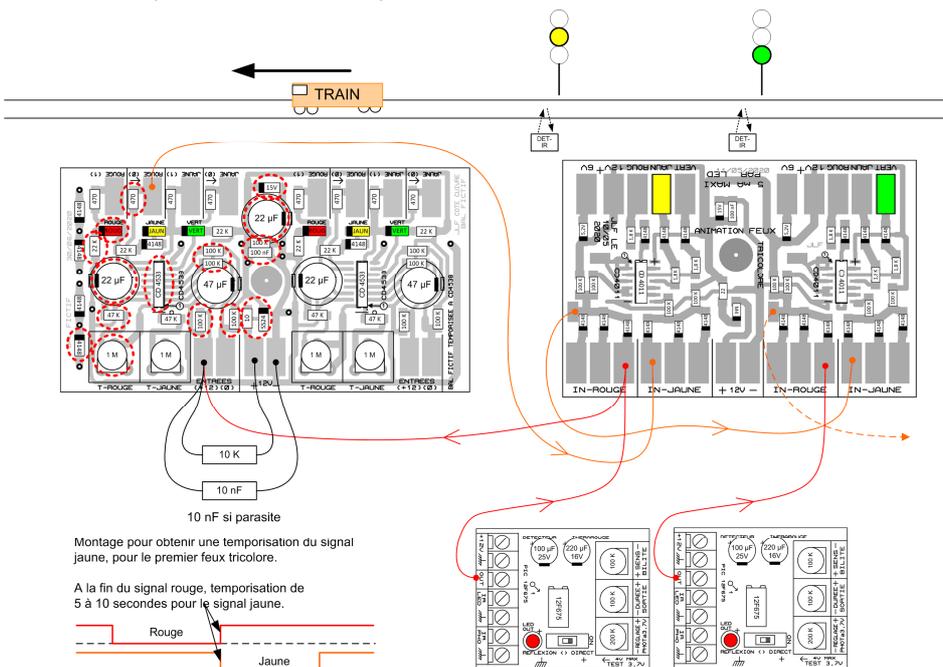
S'il n'est pas facile d'installer ce détecteur de passage, on peut aussi simuler ce signal jaune par une temporisation.

Dans ce cas, on utilise une partie du montage "Un montage BAL-FICTIF temporisé", disponible ici : http://www.la-tour.info/uts/uts_page13.html#feu_sncf

A la fin de la temporisation du premier module détecteur, ce circuit de temporisation va ensuite forcer le signal jaune pendant quelques secondes.

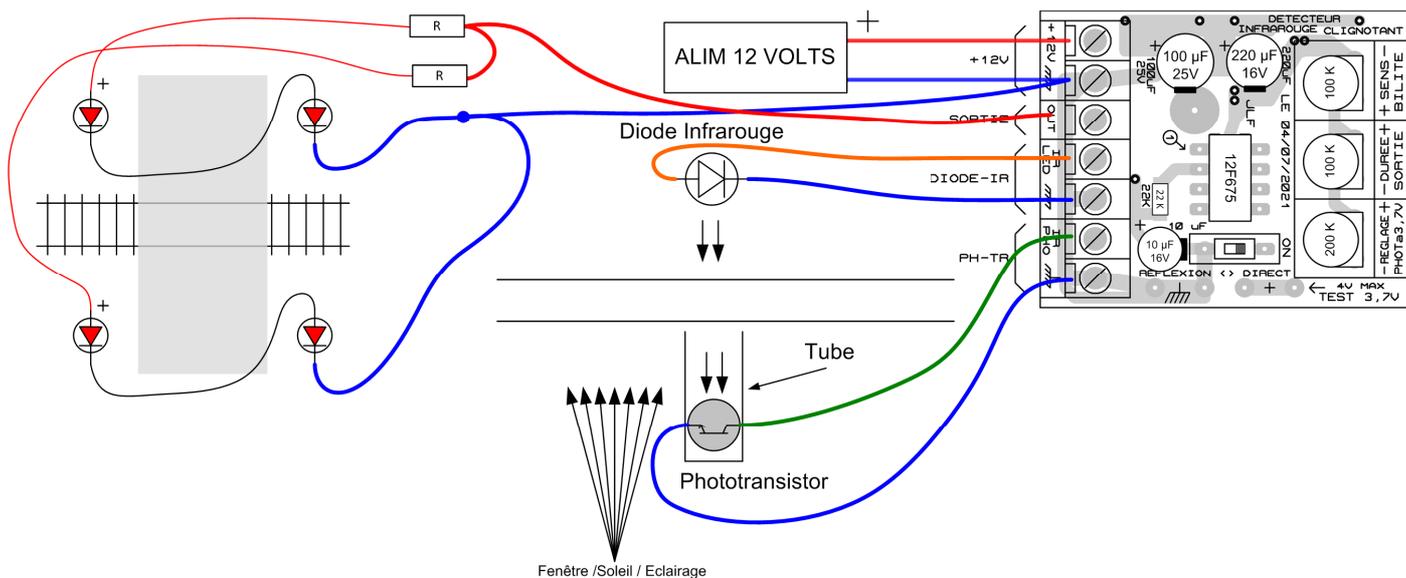
On n'aura besoin de souder que les composants entourés de rouge. La durée de temporisation du signal jaune est réglable par le potentiomètre. La résistance additionnel de 10 KOms est obligatoire pour forcer le signal d'entrée au + 12 Volts.

Le condensateur de 10 nF est optionnel, et sera rajouté si besoin en cas de mauvais fonctionnement



Le branchement du module clignotant :

R = Résistance optionnelle pour réduire le courant.



Le circuit imprimé au format GERBER est le fichier : **Detecteur IR a PIC 12F675 - Clignotant - CADCAM.ZIP**.

Le fichier ".hex" à utiliser est :

1 / Plutôt pour les feux modernes à led à allumage franc. Pour avoir le signal en sortie vers la led, à 50% allumé et 50% éteint. **Detecteur IR a PIC 12F675_Clignote_55.HEX**

2 / Plutôt pour les feux anciens à lampe à filament. Pour avoir le signal en sortie vers la led, à 40% allumé et 60% éteint. **Detecteur IR a PIC 12F675_Clignote_46.HEX**

Pour la mise au point lors de l'installation, se reporter à la notice du premier module.

A+

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.

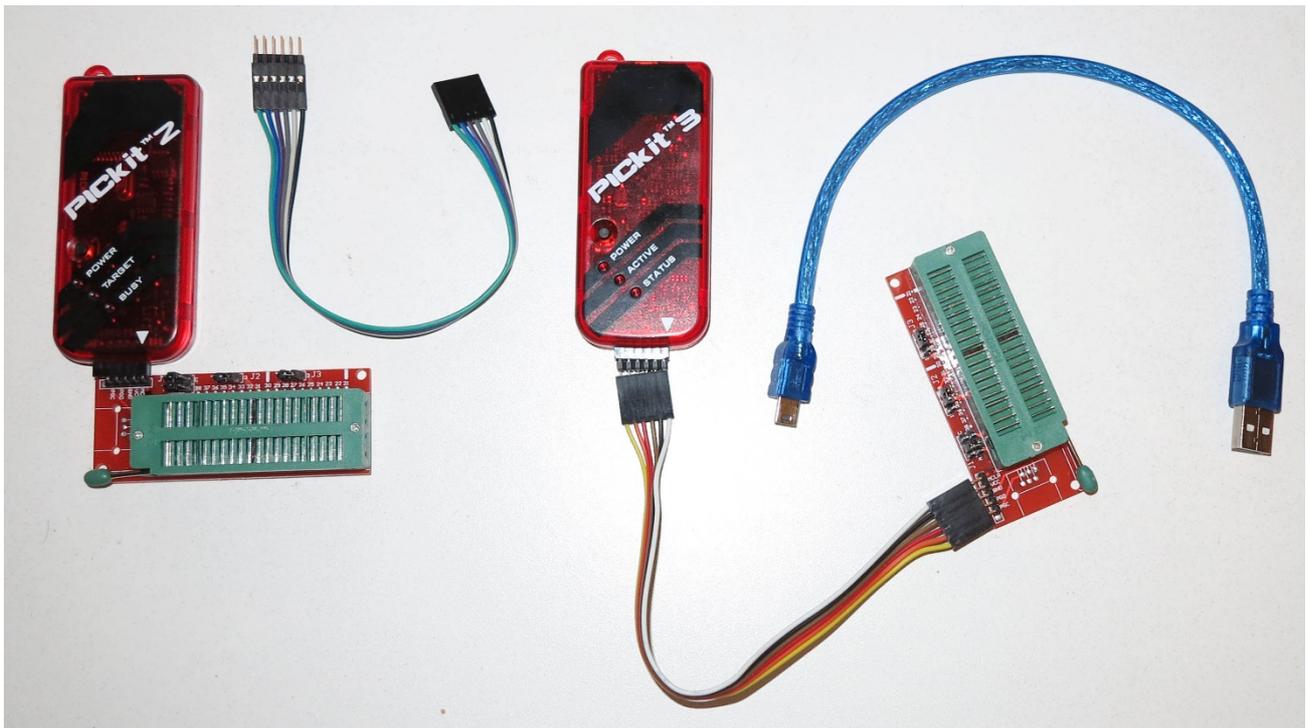
Comment programmer un PIC de type 12F675, 16F84, 16F628, 16F690, d'un montage récupéré sur Internet, à partir du fichier '.hex' ?

Si un montage à PIC du site UTS2000 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.

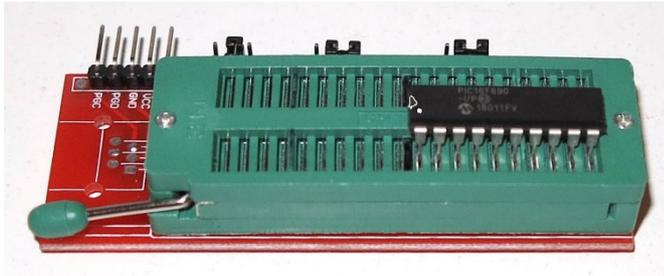


On a donc un **Pickit3**, un câble usb, un câble plat en nappe et un support de circuit intégré à force d'insertion nulle.

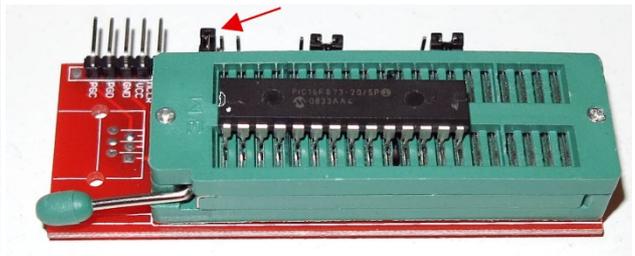
Sur le circuit imprimé, configurer les cavaliers suivant le type de PIC utilisé, en fait suivant le nombre de pattes du PIC ou format du boîtier (Exemple : DIP28 = Dual In Line 28 = Circuit à 28 pattes). La notice se trouve sous le circuit imprimé.

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

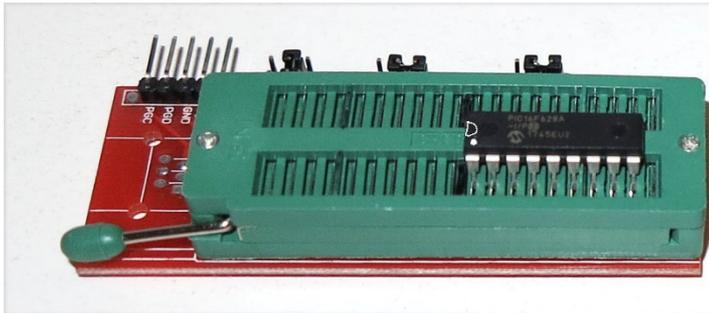
Pour faciliter la mise en place du PIC, j'ai tracé au feutre les séparations entre les pattes 4 et 5, et 10 et 11 du support. Comme cela, je colle les PIC de type DIP8, DIP18 et DIP20 au trait noir.



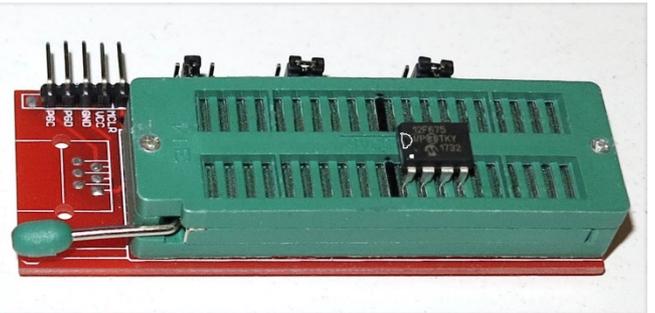
16F690 = J1 en B



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.

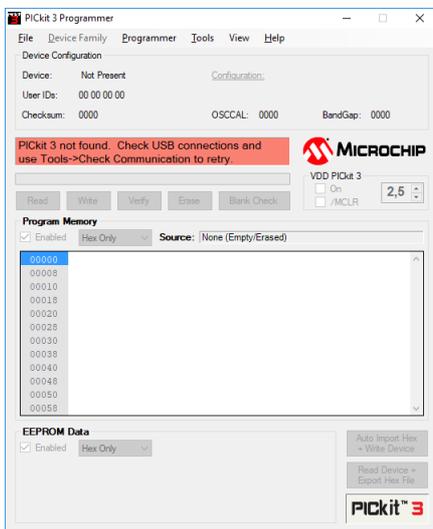
Menu : Tools > Set Unit ID, et lui donner un petit nom à votre programmeur.

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

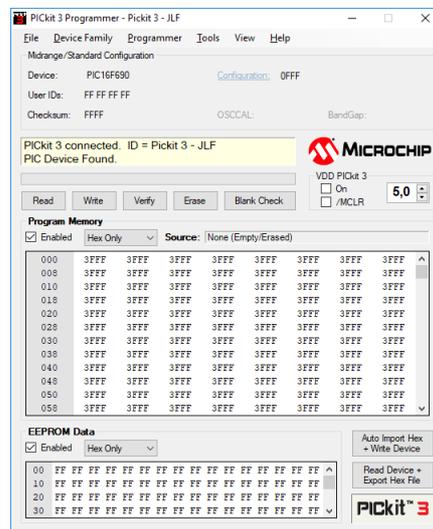
Placer un **PIC** sur le support, et cliquer sur le bouton [Erase] pour effacer le **PIC**.

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

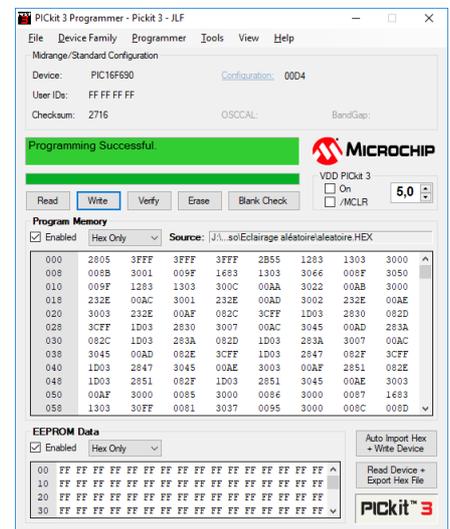
Si l'on a une erreur au moment de quitter ce programme, il faut le configurer pour qu'il se lance automatique en mode administrateur sous Windows 10. Dans l'explorateur de fichier, dans le répertoire 'C:\Program Files\Microchip\PICkit 3 v3', clic-droit sur 'PICkit3.exe' > Propriétés, Onglet 'Compatibilité'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



Sans PickKit 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 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

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**.

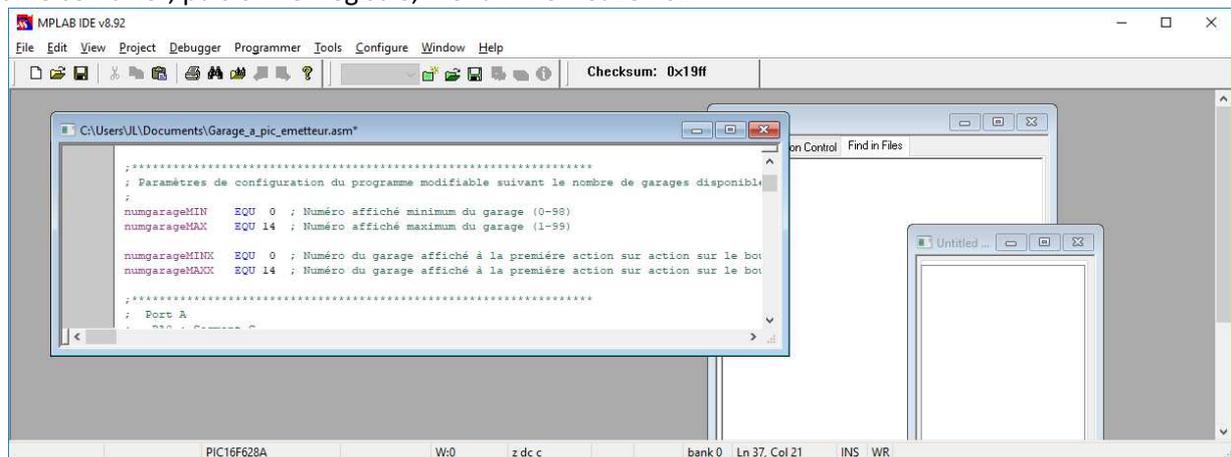
Menu File > Open, choisir par exemple le fichier 'Garage_a_pic_emetteur.asm'.

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 (-)
```

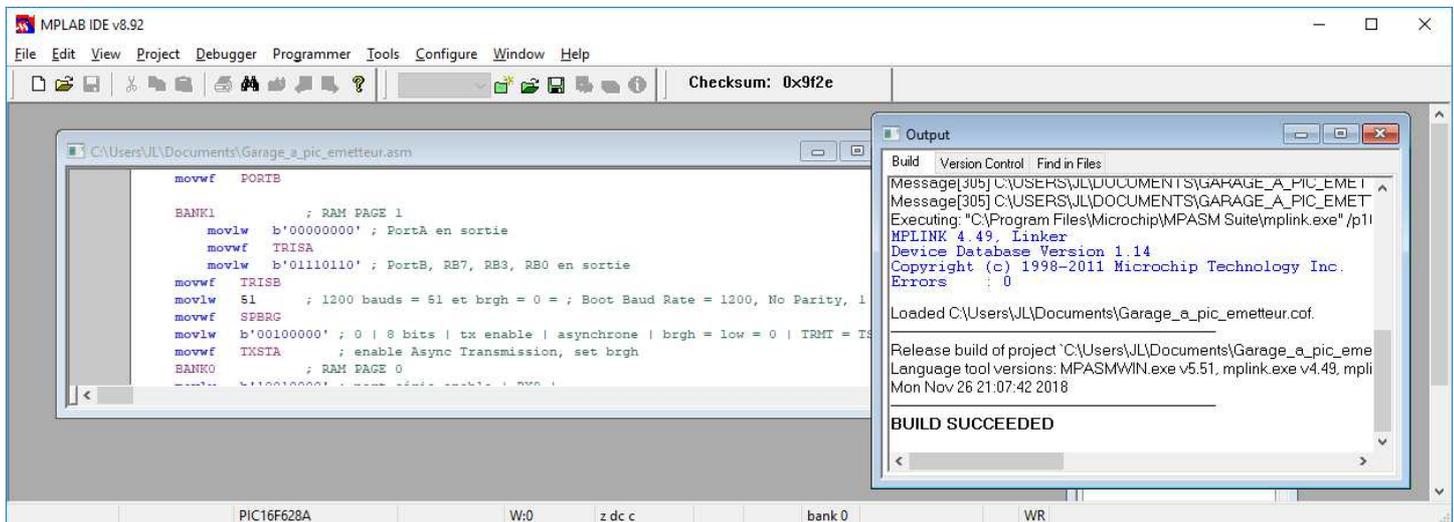
On modifie ce fichier, puis on l'enregistre, menu : File > Save As...



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+