

REALISATION D'UN PROGRAMMATEUR ET D'UNE PLATINE DE TEST POUR PIC 16F84 (16F628) MISE EN PLACE ET UTILISATION DE LOGIPIC

Vous voulez apprendre à programmer les pics seulement vous ne voulez pas passer des heures avec l'assembleur. Je vous propose de réaliser une platine très simple capable de programmer et de tester immédiatement votre programme en utilisant un logiciel de programmation visuel : Logipic.

Le programmeur est basé sur l'utilisation d'un programmeur série qui permettra de programmer le pic en mode icsp (In Circuit Serial Programmer).

La seule contrainte pour cette réalisation est l'utilisation d'une prise RS232 de votre ordinateur. L'utilisation d'une prise USB sera abordée dans un prochain cours.

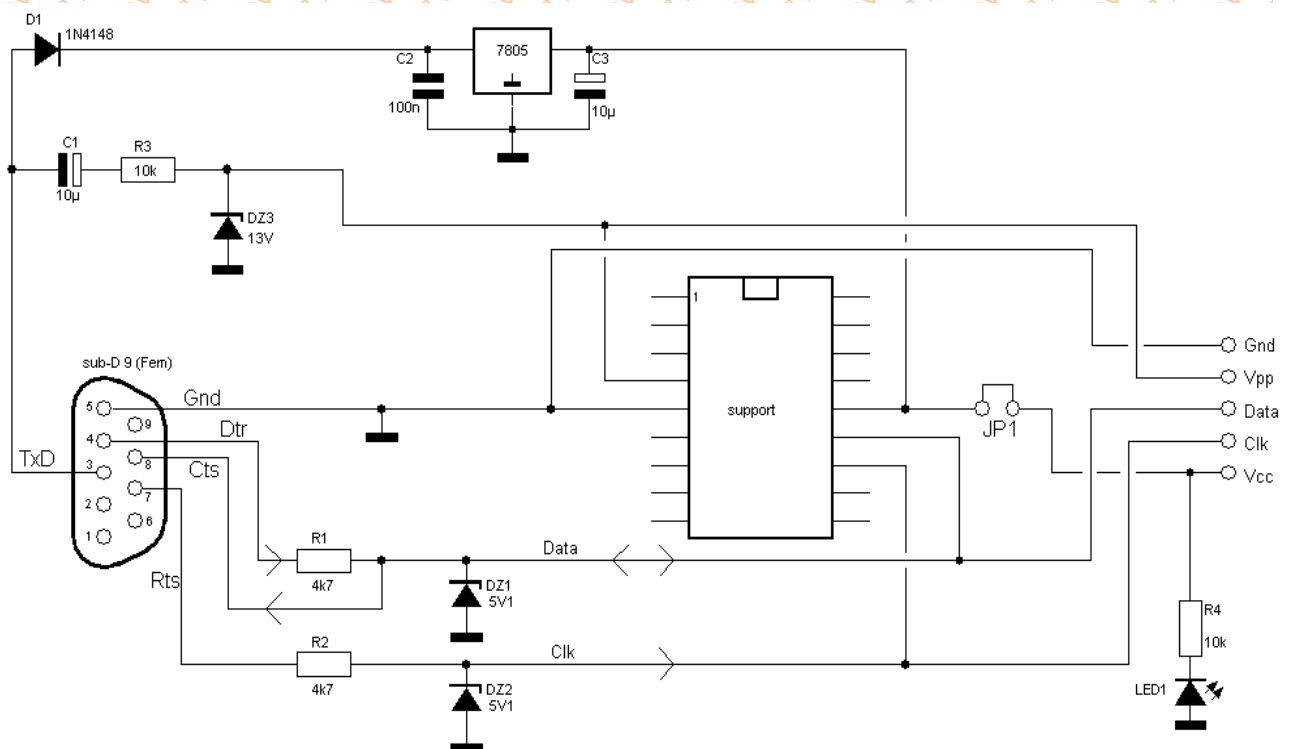
Réalisation du programmeur

Le programmeur est capable de recevoir un pic 18 broches (type 16F84 ou 16F628). Cependant, il est prévu un câble d'extension pour fonctionner en mode icsp.

Ce programmeur est compatible avec ICprog ou Winpic.

La programmation d'un autre pic est possible à condition d'adapter le câblage du circuit pour être compatible avec la série du pic (par exemple 12Cxxx ou 18Fxxx). Il faut aussi que le soft supporte le pic (par exemple ICProg ne supporte pas le pic 18F2550).

Schéma du programmeur

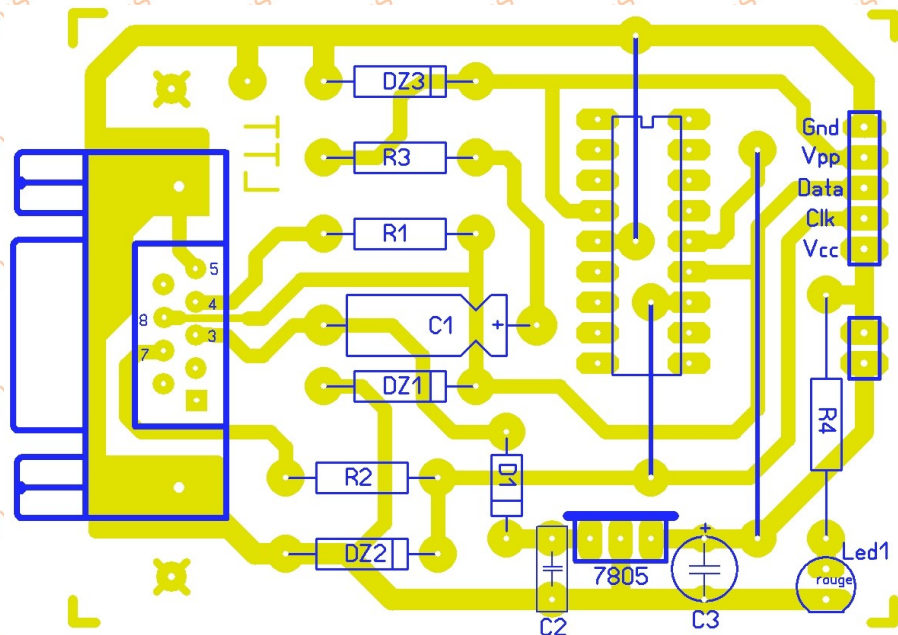


Le Jumper JP1 permet de déconnecter l'alimentation +5V de la prise RS232. Le cavalier JP1 permettant de déconnecter Vcc du programmeur par rapport au +5V de la platine de test. JP1 devra être présent uniquement pour la programmation d'un pic sur le support du programmeur.

Si vous voulez utiliser le programmeur avec ma platine de test, il est possible d'enlever les composants suivants : D1, C2, 7805 et C3.

R4 et Led1 ne sont pas indispensables, ils permettent uniquement de vérifier la présence du +5V.

Implantation des composants du programmeur



Liste des composants :

R1, R2 : 4,7 k Ω

R3, R4 : 10 k Ω

C1 : de 10 μ F à 100 μ F mini 25V axial

C2 : 100 nF plaquette

C3 : 22 μ F mini 16V radial

D1 : 1N4148

DZ1, DZ2 : 5V1

DZ3 : 13V (ou 2 zeners de 6V2)

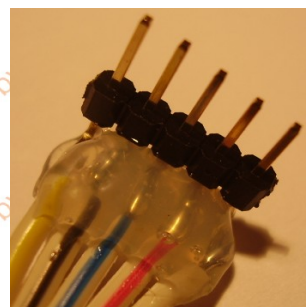
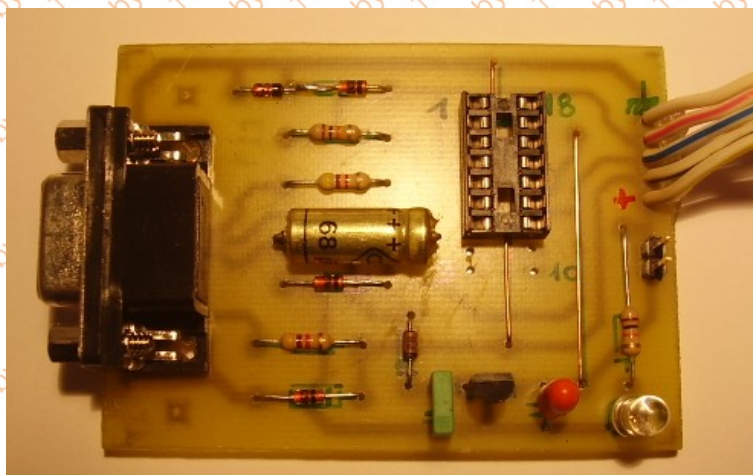
Support 18 broches

Régulateur 78L05 ou 7805

Led1 : led rouge haute luminosité

Connecteur SUB-D circuit imprimé coudé femelle DE9S

Pour pouvoir réaliser le jumper JP1 et le connecteur icsp : Barette 2,54mm sécable



Aspect visuel du programmeur et du connecteur icsp

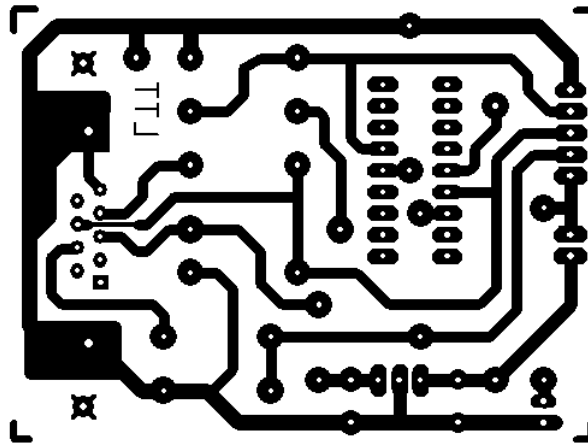
Remarque : le support du circuit intégré utilisé sur mon circuit est un support 14 broches.

La ligne Vcc est connectée cependant elle n'est pas indispensable, elle permet uniquement de visualiser la présence du +5V (en provenance de la platine de test).

Circuit imprimé du programmeur

L'ensemble tient sur une plaque d'époxy de 70 x 50.

Le fichier LTT.LAY réalisé avec le logiciel Sprint Layout s'ouvre avec le viewer gratuit [ici](#) et permet d'imprimer de façon impeccable le circuit.



Réalisation de la platine de test

Le mode icsp permet de programmer le pic celui-ci étant en position sur la platine d'utilisation, ce mode évite donc les nombreuses manipulations du pic entre le programmeur et la platine de test. Ce mode est particulièrement utile pour la mise au point d'un programme.

L'utilisation du mode icsp nécessite quelques composants complémentaires et quelques précautions de mise en œuvre (Voir Databook microchip : How to implement ICSP).

L'élément principal du mode icsp est sans aucun doute la diode D1, cette diode BAT85 permet de limiter la chute de tension (cependant une diode 1N4148 fonctionne la plupart du temps).

En mode normal (fonctionnement de la platine de test), cette diode permet à la broche Mclr de se retrouver à l'état haut, le pic peut ainsi fonctionner.

Lors de la programmation, Vpp de 13 V est bloqué par D1 et évite de se retrouver sur Vcc.

L'ensemble R1, D1, C1 permet une Raz à la mise en route du pic tout en autorisant une Raz manuelle via le Bouton poussoir Raz et R2.

Une autre précaution à prendre lors de la programmation en mode icsp concerne les lignes Clk et Data (B6 et B7). Ces lignes doivent être découplées du reste du montage par l'intermédiaire des switchs. L'utilisation de switchs n'est pas indispensables si la charge est supérieure à 10k.

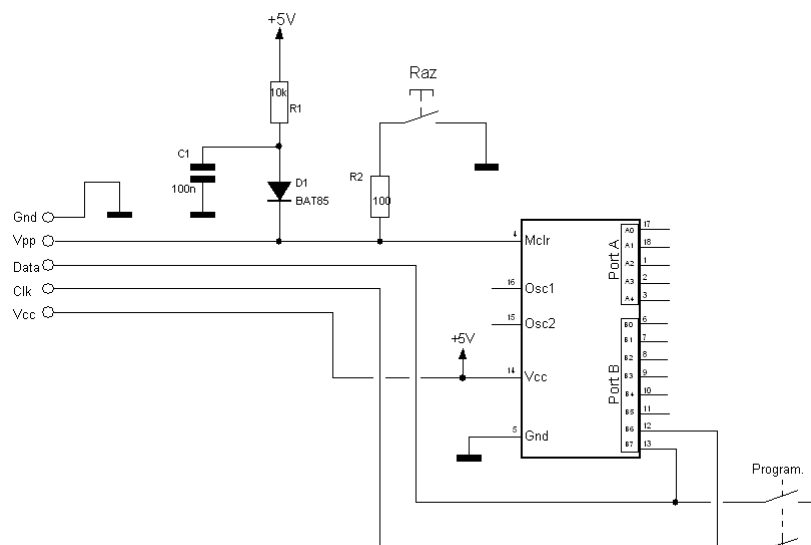
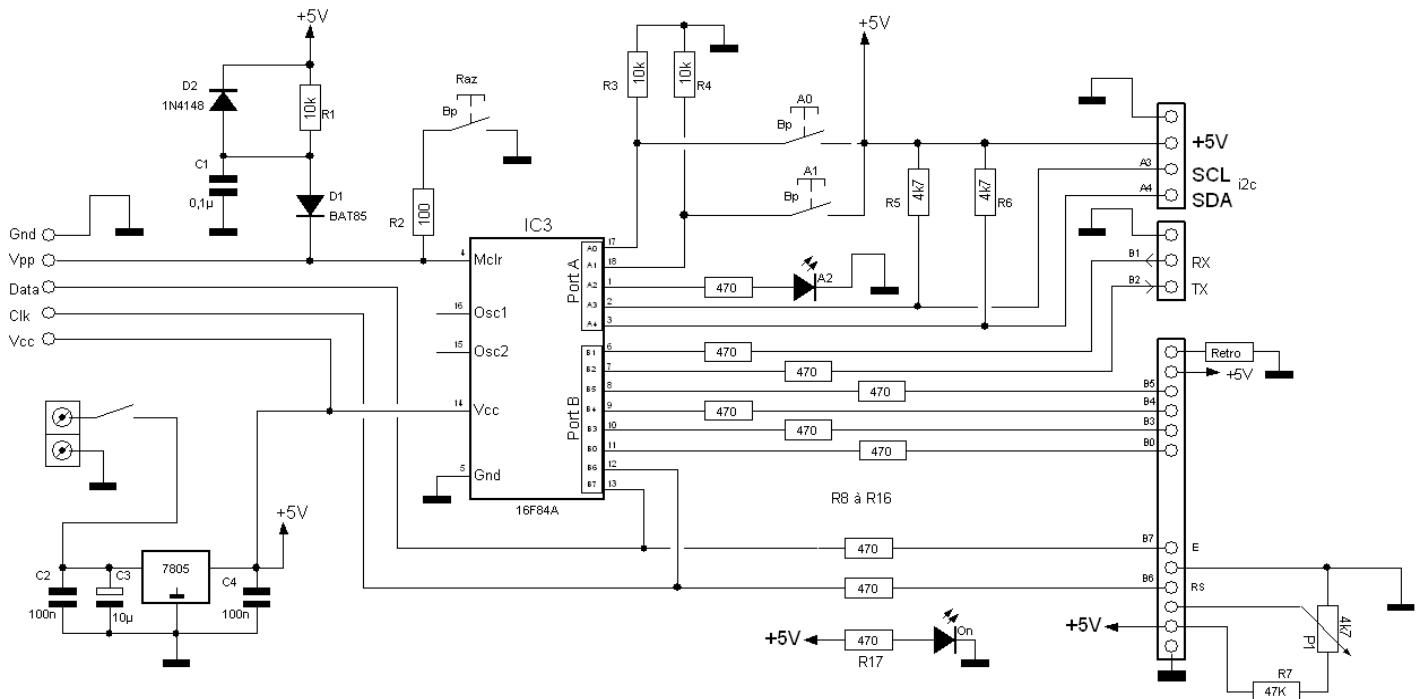


Schéma de la platine de test



L'alimentation +5v est réalisée par l'ensemble C2, C3, 7805 et C4. L'alimentation +5V se visualise avec la led On et permet d'alimenter le pic et les équipements entourant le pic.

Il est facile de reconnaître l'adaptation icsp composé de R1, D1, D2, C1 et R2.

A0 et A1 sont utilisés comme entrée. A0 et A1 sont à l'état logique 0 grâce à R3 et R4 les boutons poussoirs permettent de changer l'état logique lors de chaque appui.

A2 est utilisé comme sortie, ce port est connecté sur la diode led A2.

A3 et A4 sont utilisés pour la liaison i2c, les résistances R5 et R6 sont donc les résistances de pull-up indispensables pour le bon fonctionnement du bus i2c. Contrairement aux autres lignes (du port A ou B), le port A4 permet de fonctionner en mode collecteur ouvert : il est donc le seul possible pour la liaison SDA.

Les connexions du port B sont dictées par 2 impératifs :

- un impératif technique, ainsi B1 et B2 sont laissés libres pour pouvoir exploiter la liaison RX, TX pour des tests d'envoi et de réception de données,
- un impératif de réalisation, afin de simplifier au maximum le circuit imprimé.

L'afficheur LCD en mode 4 bits nécessite 6 liaisons et exploite les lignes restantes : B0, B3, B4, B5, B6, B7. La charge des lignes B6 et B7 est faible, c'est pourquoi il n'y a pas besoin de switchs sur les lignes Data et Clock. P1 permet de régler le contraste et doit être correctement ajusté pour pouvoir lire des informations sur l'afficheur.

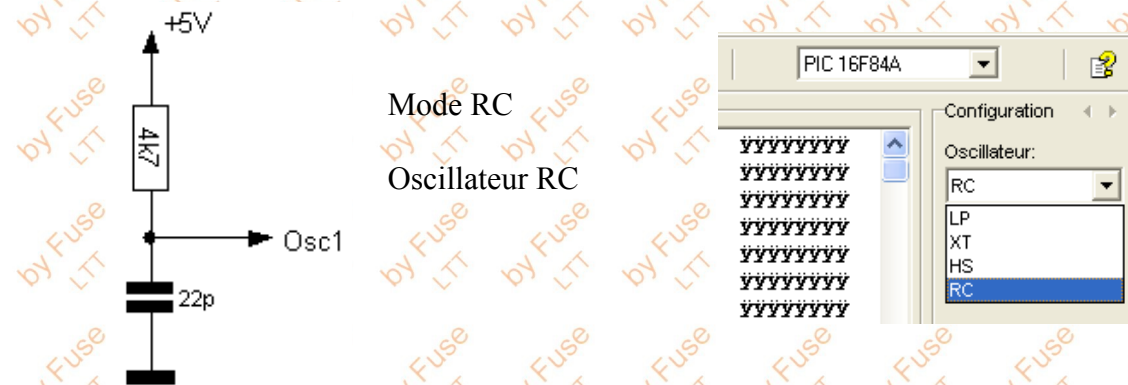
L'afficheur n'est pas indispensable pour commencer à prendre en main la platine de test, c'est pourquoi j'ai positionné des résistances (R8 à R16) qui permettent de relier des diodes led sur les sorties B. La valeur de 470 Ω permet à la fois de relier sans soucis une diode led sur les sorties et de commander les étages d'entrée de l'afficheur.

Cette valeur permet aussi toute erreur de manipulation et évitera la destruction de l'étage de sortie du pic si la sortie est reliée par mégarde sur la masse.

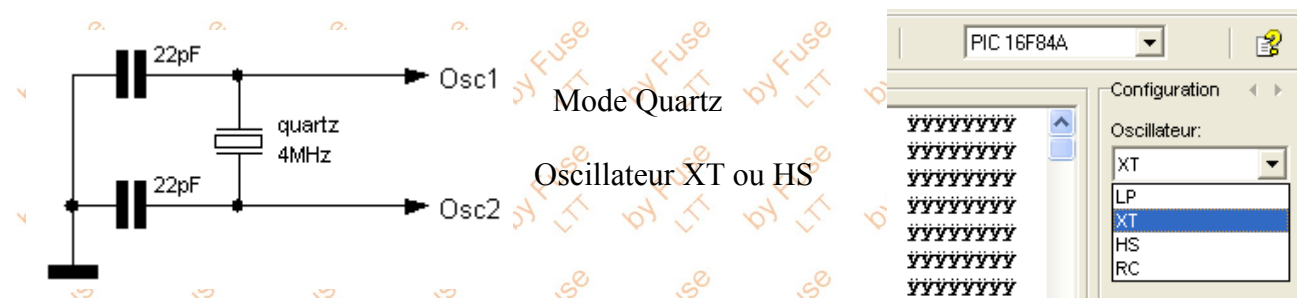
L'oscillateur du picaxe

L'utilisation d'un 16F84 nécessite la présence **impérative** d'un oscillateur externe, oscillateur RC ou oscillateur à quartz. Le choix du type d'oscillateur est un compromis cout/utilisation, en effet, si il s'agit de tester et de faire clignoter des leds, un oscillateur RC est économique et largement suffisant. S'il s'agit de réaliser une horloge, la présence d'un quartz est nécessaire à une bonne précision. Le mode de fonctionnement est indiqué au pic par la Configuration de l'oscillateur dans IC-Prog.

Le circuit imprimé autorise le fonctionnement dans les différents modes :



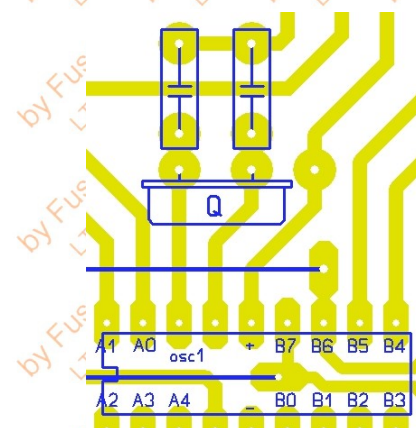
avec les valeurs indiquées, la fréquence est d'environ 4 MHz.



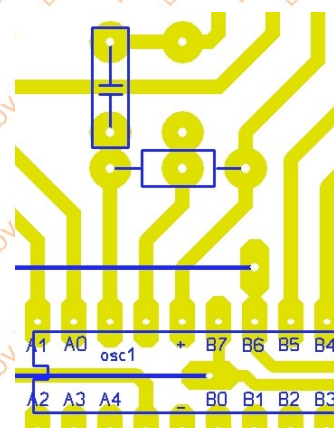
Le mode XT est utilisé pour un quartz jusqu'à 4MHz

Le mode HS est utilisé pour un quartz supérieur à 4MHz et jusqu'à 20MHz

Le mode LP est utilisé pour limiter la consommation, le quartz maxi est de 200KHz

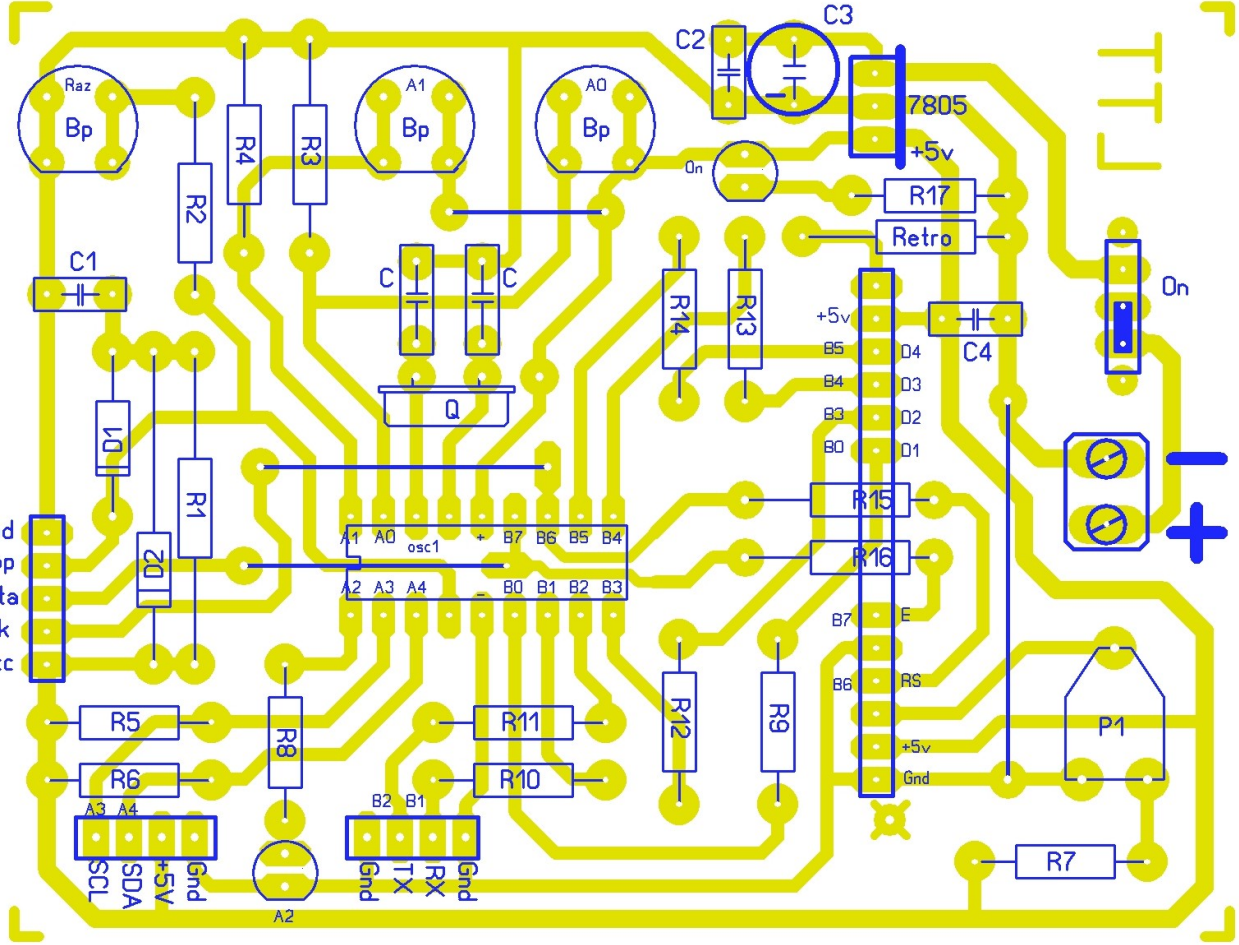


Mode Quartz



Mode RC

Implantation des composants de la platine de test

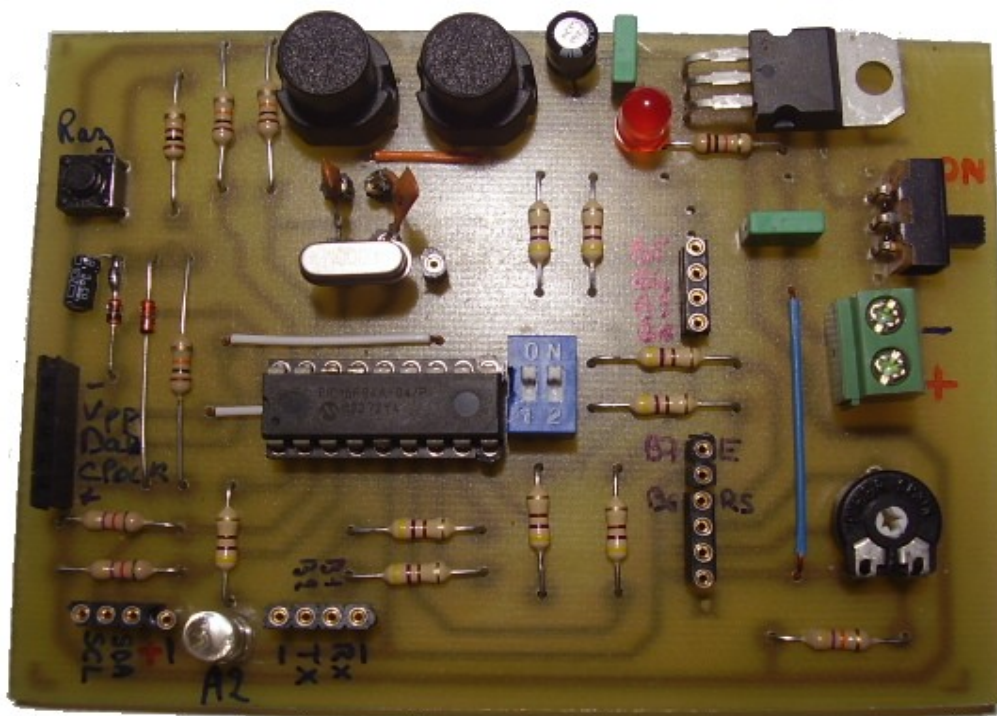
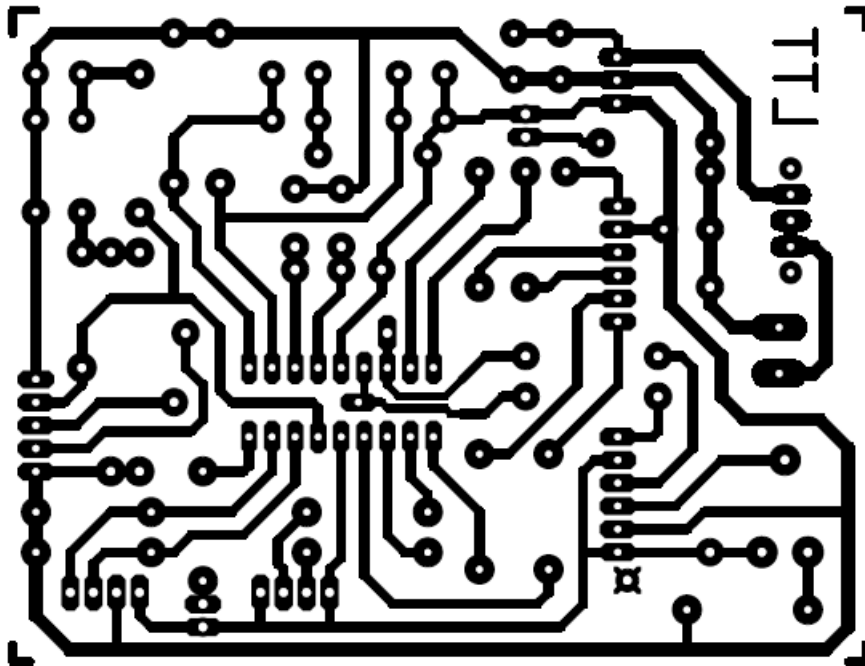


Interrupteur à glissière

Circuit imprimé de la platine de test

L'ensemble tient sur une plaque d'époxy de 100 x 70.

Le fichier LTT.LAY réalisé avec le logiciel Sprint Layout s'ouvre avec le viewer gratuit [ici](#).



Aspect de la platine de test. Cette platine de test prototype est équipée d'un switch. Celui ci non indispensable n'est pas présent sur la platine finale.

La platine est équipée d'un 16F84A et d'un quartz 4 MHz.

Mise en œuvre du logiciel Logipic


Logipic est un formidable logiciel graphique qui permet de programmer facilement les principaux pics. Le concepteur IdMax propose ce formidable logiciel qu'il met gracieusement en ligne sur son [site](#).

Téléchargement des logiciels

Sur le site D'IdMax, dans la rubrique Téléchargement, télécharger :

Logipic16 v2.14

MPASMWIN 5.2

	<p>LogiPic16 V2.14d (Installation complète)</p> <ul style="list-style-type: none">• Pour que LogiPic puisse être polyvalent avec l'utilisation de nombreux Pic, il est possible de créer des fichiers ".pic ". Ces fichiers contiennent les caractéristiques propres aux Pics. <p>Une aide sur la constitution des fichiers est disponible plus sur cette page. Les fichiers ".pic" peuvent être ouverts avec un éditeur de texte tel que notepad.</p> <ul style="list-style-type: none">• Modification du module d'import/export de routines, il possible d'exporter plusieurs routines dans un seul fichier.• Affichage des commentaires lorsque la souris pointe sur un module.• Correctifs généraux.• Gestion de la mémoire flash
---	---

Et sur le site IC-PROG, dans la rubrique Download, télécharger :

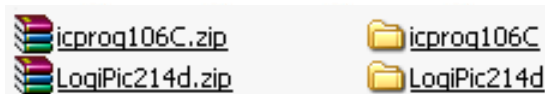
IC-Prog Software 1.06C Multi-Lingual

Download IC-Prog:

 [IC-Prog Software 1.06C Multi-Lingual \(updated 5th April 2011\)](#)

Installation des logiciels

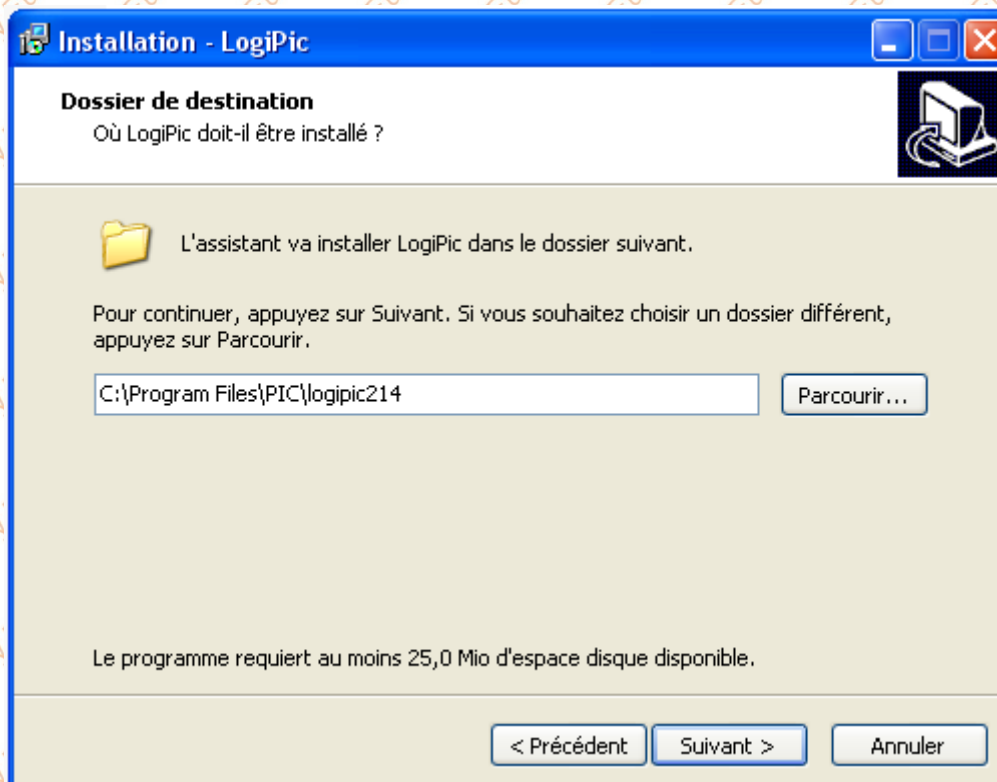
Décompressez les fichiers dans des dossiers séparés (ce n'est pas indispensable mais un peu de rigueur n'est pas plus mal...)



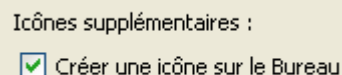
Installez Logipic214d, pour cela cliquez sur setup.exe :



Je vous conseille d'installer Logipic dans un dossier spécifique ; C:\Program Files\PIC\Logipic214



Je vous conseille de valider la création d'icônes sur le bureau :



Positionnez les logiciels ICProg106C dans le dossier PIC.

Pour cela faire un Copier/Coller de ses logiciels dans C:\Program Files\PIC)

Dans le dossier PIC, vous devez obtenir ceci :

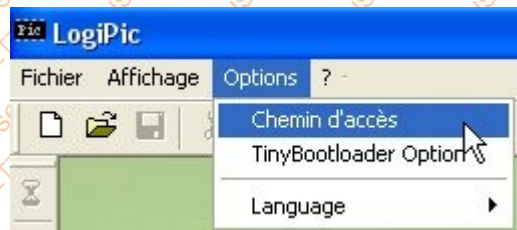


Configuration des logiciels

Lancer Logipic214

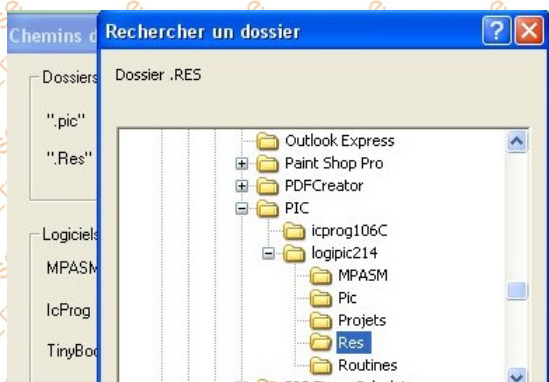


Demander Options > Chemin d'accès

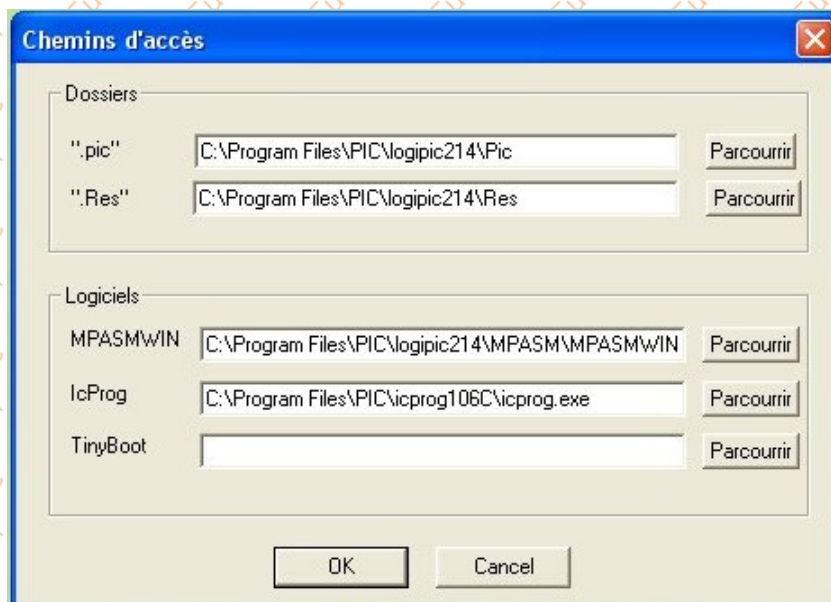


Précisez les chemins d'accès qui correspondent à votre installation :
pour le dossier pic

et pour le dossier Res



Et enfin, précisez les chemins d'accès de MPASMWIN et IcProg



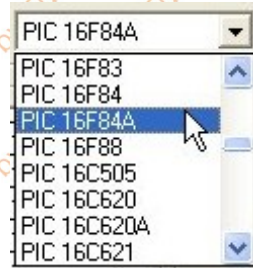
Pour le moment TinyBoot n'est pas utilisé...

Validez par OK

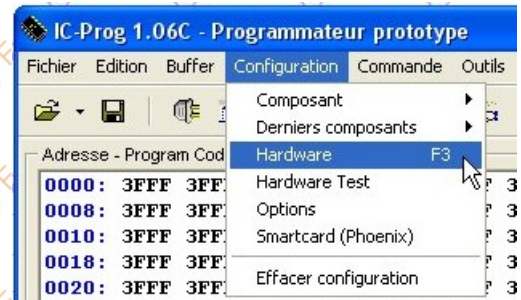
Il reste à configurer IcProg106C. Lancer IC Prog :



Sélectionner le pic utilisé :

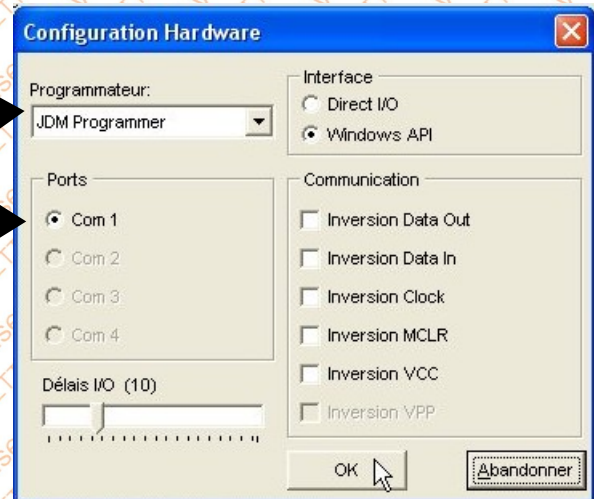


Demander Configuration > Hardware :



Choisir le JDM Programmer
et Windows API (sous XP)

(Sélectionner votre Ports Com) :



J'ai eu de nombreux soucis de lecture du pic avec l'ensemble Programmeur + Platine de test.

Peut être en raison de la longueur des lignes icsp

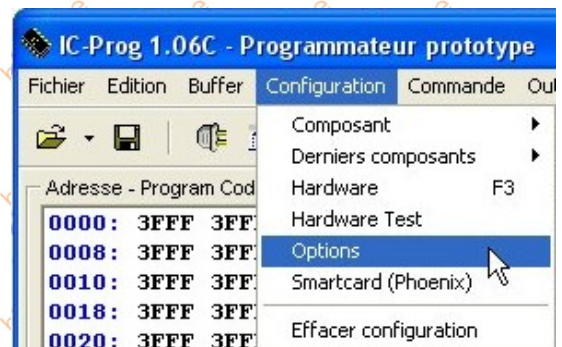
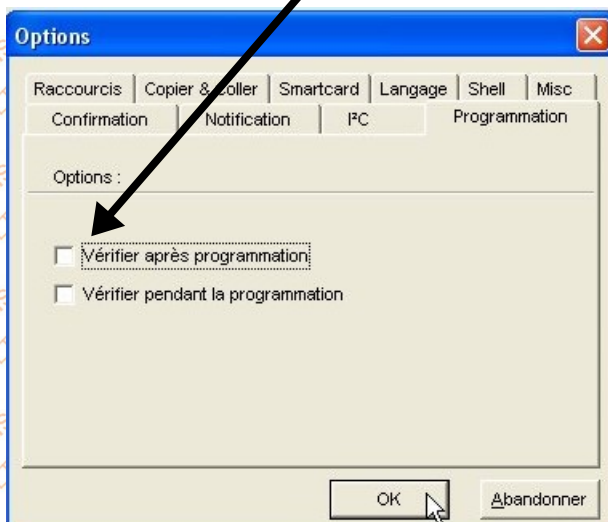
Peut-être par le fait que l'alimentation 5V est produite par la platine de test.?

Pour éviter un message d'erreur de ICProg, je conseille de décocher la vérification après la programmation...

Pour cela Demander Configuration > Options :

Dans l'onglet Programmation

Décocher 'Vérification après Programmation'



Vous pouvez fermer ICProg,

Le prochain appel d'IcProg
s'effectuera directement sous Logicpic.

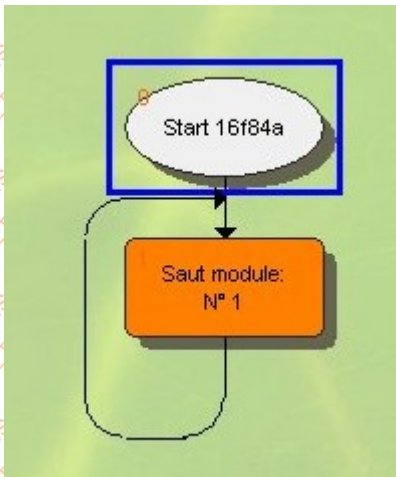
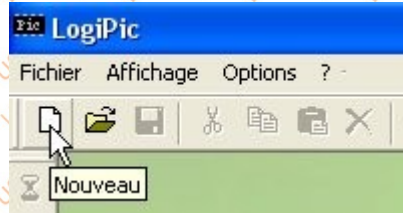
Premier programme avec Logipic

Le premier programme consistera à faire clignoter la led A2 et utilise un pic 16F84A.

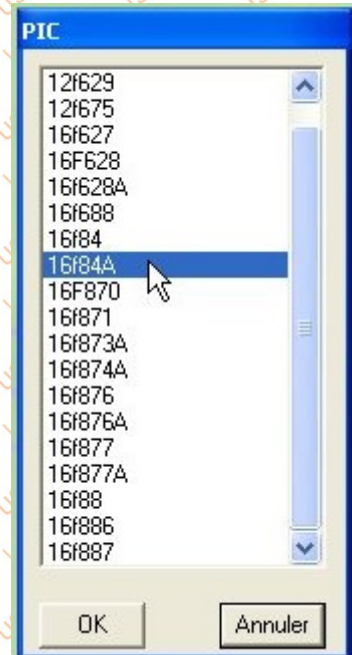
Lancer Logipic214



Demander Nouveau



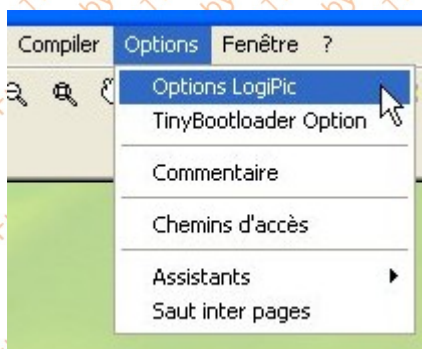
Choisir le pic



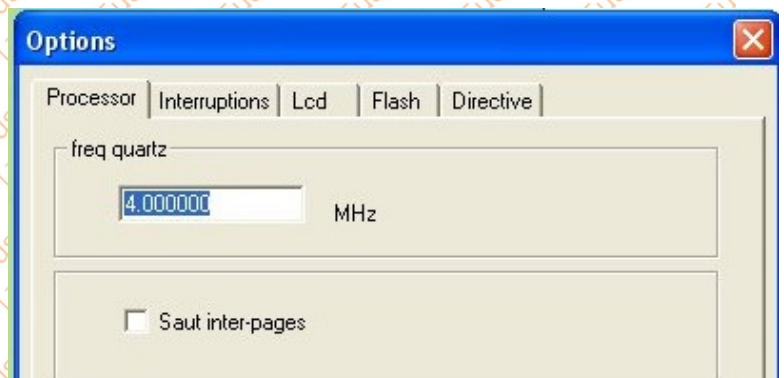
Je vous conseille d'enregistrer votre programme dans un dossier caractéristique, par exemple 'soft pic' nom du premier fichier : prog1.prj
Attention longueur maximum de nom : 8 caractères

Dans un premier temps, il est très important de configurer les options de Logipic.
Sans cette phase , le pic ne fonctionnera pas

Demander Options > Options Logipic :



choisir la fréquence du quartz
(Cela permet le calcul de temporisation sur Logipic).

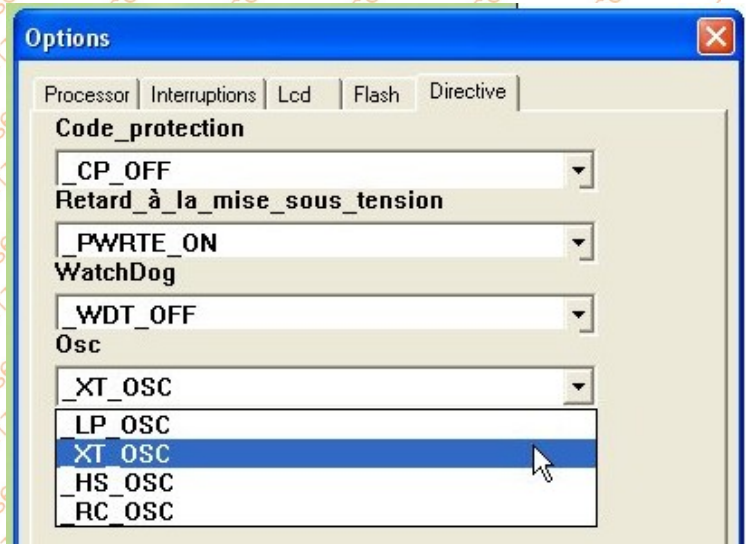


Pour le moment inutile de s'attarder sur les onglets Interruptions Lcd et Flash

L'onglet Directive est très important

Il permet de configurer l'oscillateur du pic.
La platine de test utilise un quartz,
on choisira donc XT_OSC

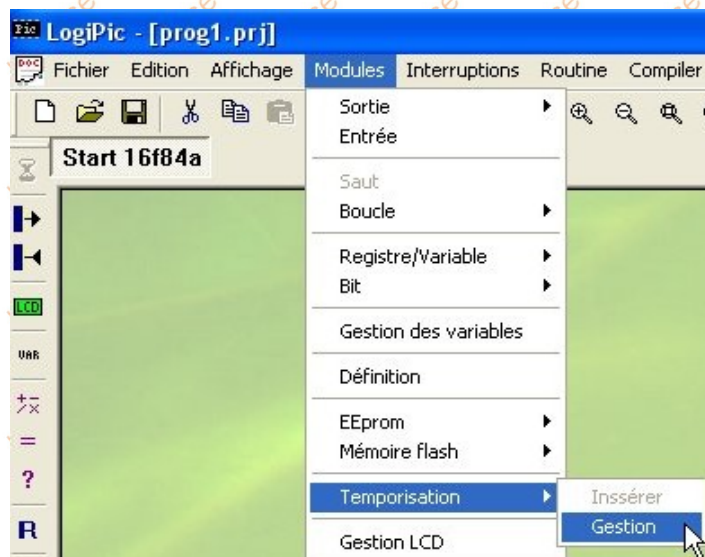
Si vous utilisez un oscillateur RC, il
faudra préciser RC_OSC



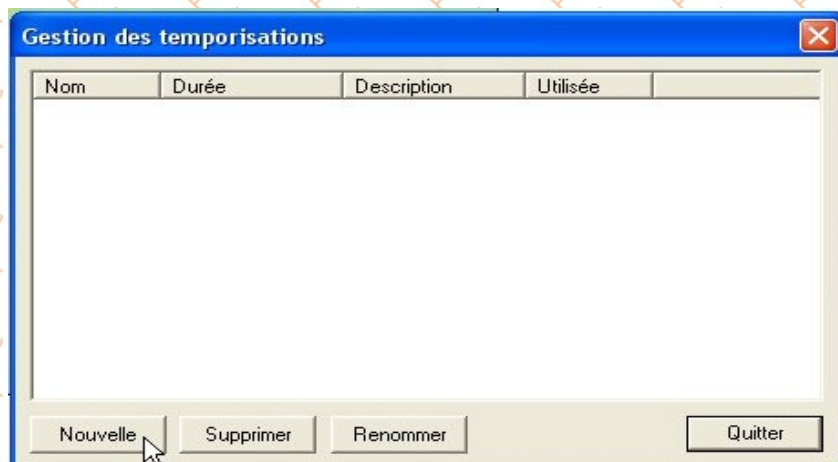
Faire clignoter une led

Pour faire clignoter une led, il est nécessaire de définir une temporisation, par exemple 1seconde.

Demander Modules > Temporisation > Gestion



Demander Nouvelle

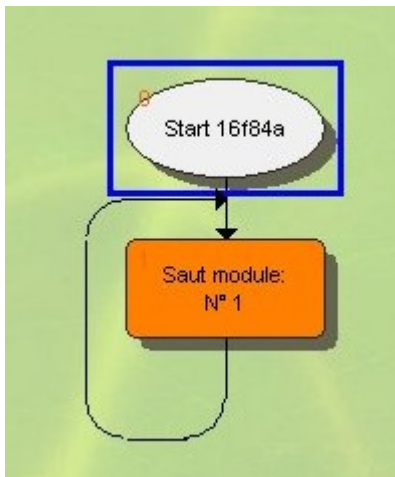


On garde la temporisation de base qui est de 1 seconde (1000000µs)

Je vous conseille de mettre un nom facile à retenir
par exemple 1sec



Revenons à notre programme, vous devez démarrer sur
l'instruction entourée en bleue (Start 16f84a)

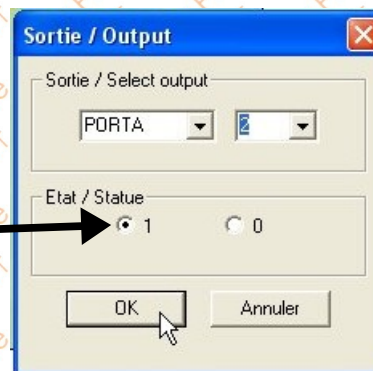


cliquer sur Sortie

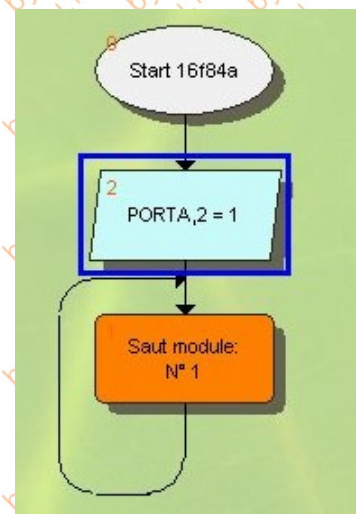


Etant donné que la diode de la platine de test est sur le Port A2, on sélectionnera le PortA2

On demandera l'Etat 1
ce qui permettra d'allumer la diode A2



Le programme commence :

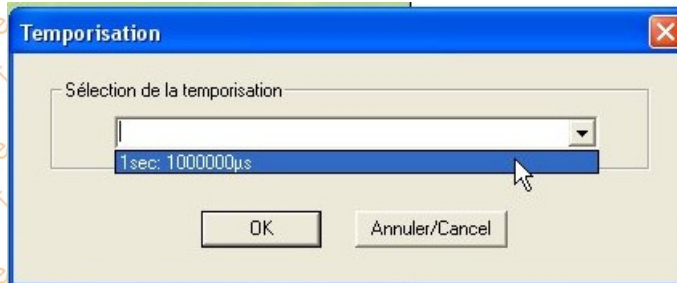


Pour que la diode reste allumer pendant 1 seconde, il va falloir mettre une temporisation et repositionner la sortie à A2 à 0.

Sélectionner Tempo



Choisir la tempo 1sec



Reprendre une sortie

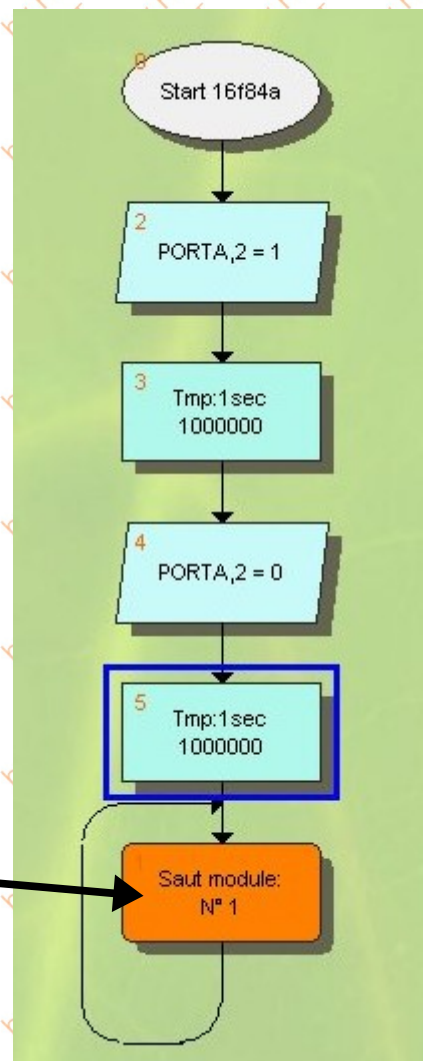


Sélectionner la sortie PORTA2
la positionnée à 0



Le programme est pratiquement complet :

Double-cliquez sur 'Saut Module N°1'



Mettre le curseur en début de programme
Valider par un clic, le programme est terminé.

Le programme est facilement compréhensible :

PORTA,2=1 allume la diode A2
Tmp : 1sec pendant 1 seconde
PORTA,2=0 éteint la diode A2
Tmp : 1sec pendant 1 seconde
et rebouclage du programme.

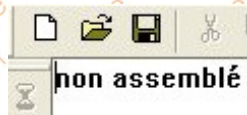
La diode A2 clignote

Il est temps de tester le programme pour vérifier que la pratique rejoint la théorie.

Demander le code du programme



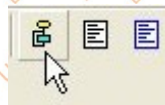
Vous obtenez



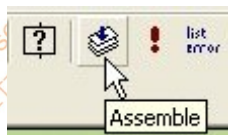
Il n'y a rien ; normal le programme n'est pas assemblé

L'assemblage consiste à transformer le logigramme en code assembleur....**Cette étape est indispensable.**

Reprendre le logigramme



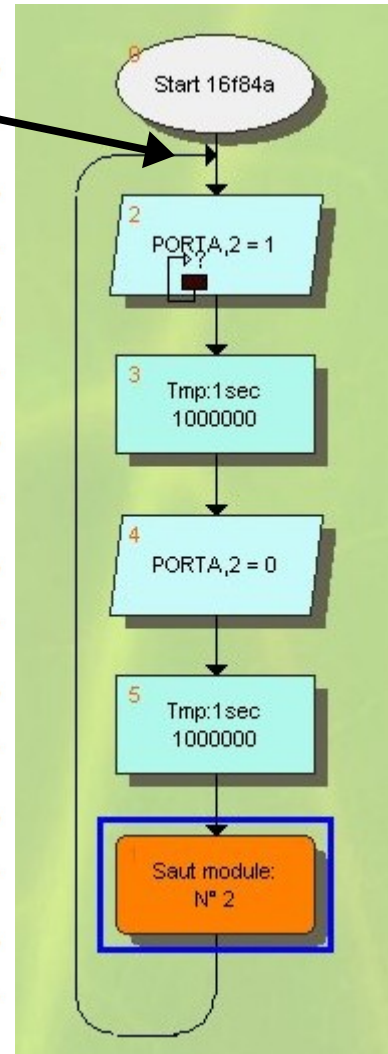
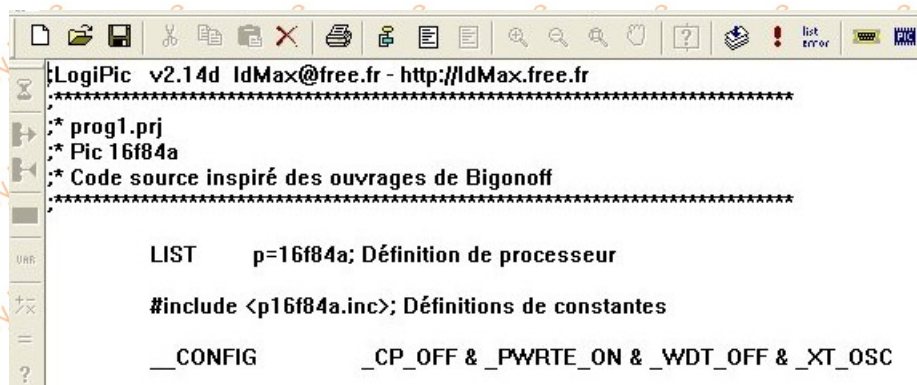
Effectuer l'assemblage



Redemander le code du programme



L'assemblage permet la création des lignes de code



Le code assembleur ne peut pas être envoyé tel quel dans le pic : il faut le compiler.
La compilation consiste à transformer le code assembleur en code compréhensible (du .hex)

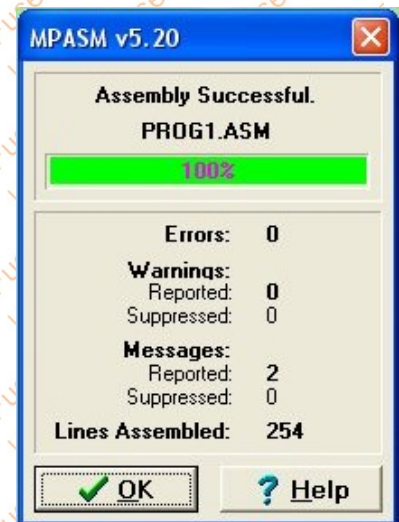
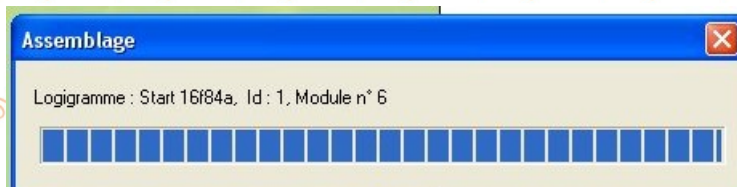
Cette étape est encore une fois indispensable.

Il faut utiliser le compilateur de microchip MPLAB qui s'appelle MPASMWIN et qui est installé dans un dossier de Logipic (voir page 10).

La compilation s'effectue très simplement en cliquant sur



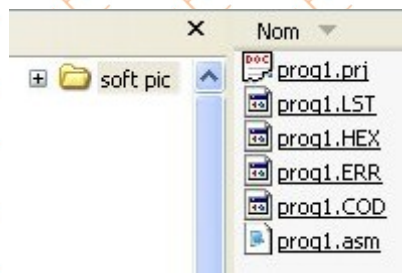
Si tout se passe bien l'assemblage indique un Assembly Succesful



Si vous êtes curieux, regarder dans votre dossier soft pic :

Il y a
le projet logipic (prog1.prj)
le programme assembleur (prog1.asm)
le programme à envoyer au pic (prog1.HEX)

les fichiers .LST, .ERR et .COD
sont générés en même temps.



Toute erreur d'assemblage est indiquée par une barre rouge, il y a de nombreuses raisons d'erreurs, voici les plus communes :

nom de tempo ou de variables non autorisé
problème de directives dans le fichier .pic

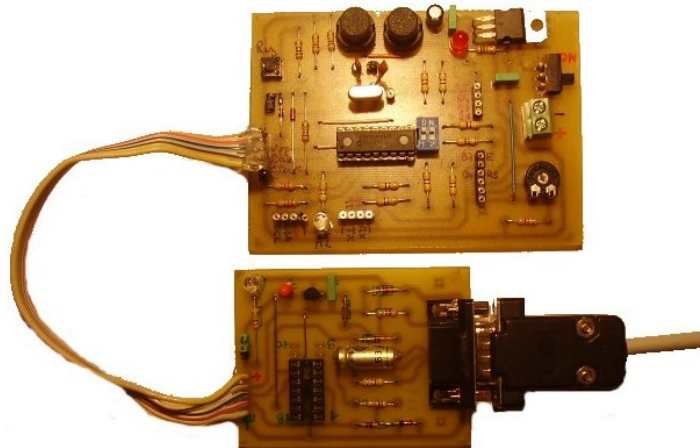
La dernière étape consiste à télécharger le fichier prog1.HEX dans la cible c'est à dire la platine de test.

Pour cela brancher le programmeur
sur la prise RS232 du PC

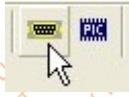
Relier la ligne icsp sur la platine de test :

N'oubliez pas d'enlever le cavalier
sur le programmeur.

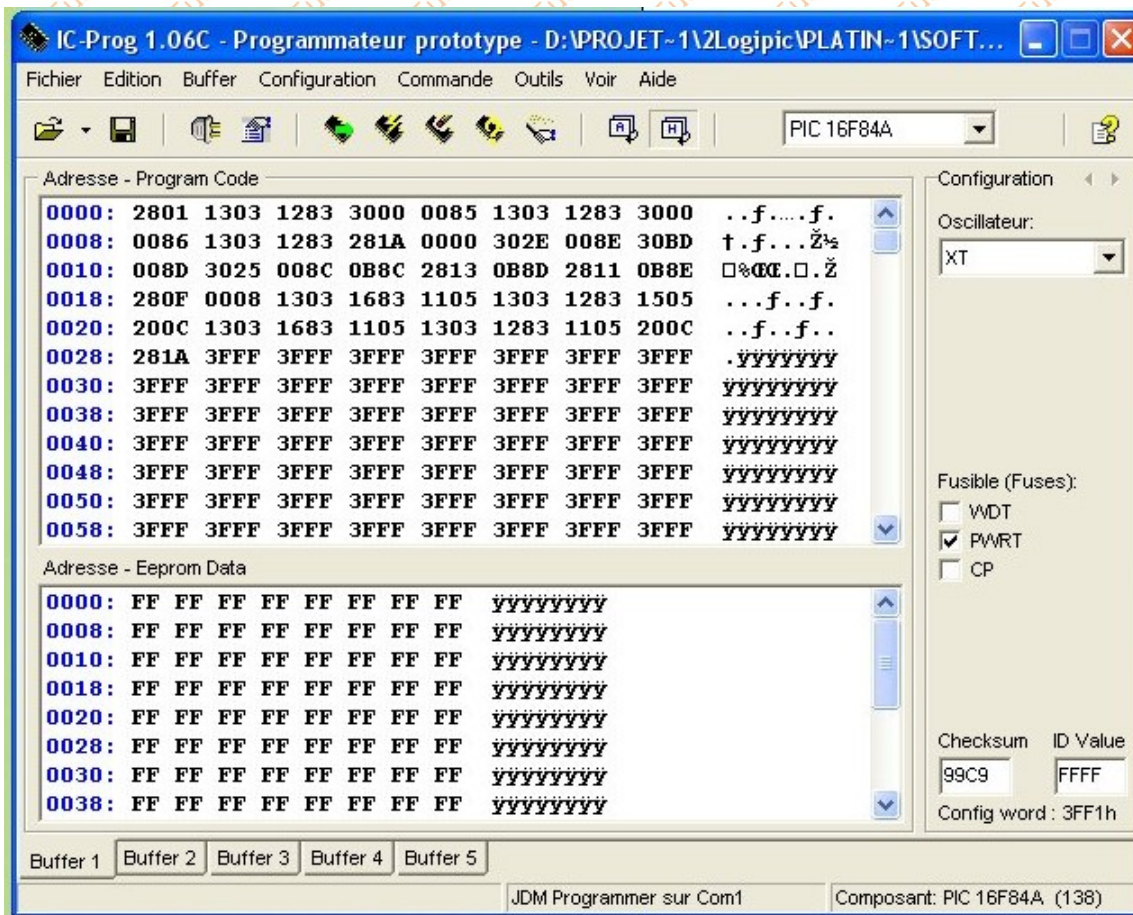
Alimenter la platine de test



cliquer sur

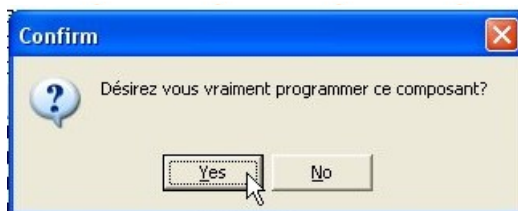


Automatiquement IC-Prog s'ouvre avec le fichier pro1.HEX prêt à être téléchargé.
Si vous avez suivi le dossier, vous devez obtenir l'affichage suivant (Checksum 99C9)



Télécharger le programme en cliquant sur Tout programmer

Confirmer la programmation



La programmation s'effectue.



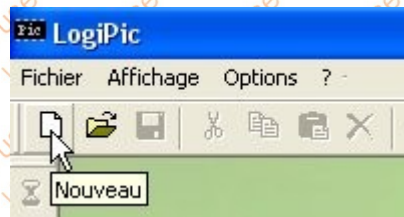
Dès la fin du téléchargement,
la platine de test s'active et la diode A2 clignote.

Vous savez dorénavant programmer un pic.

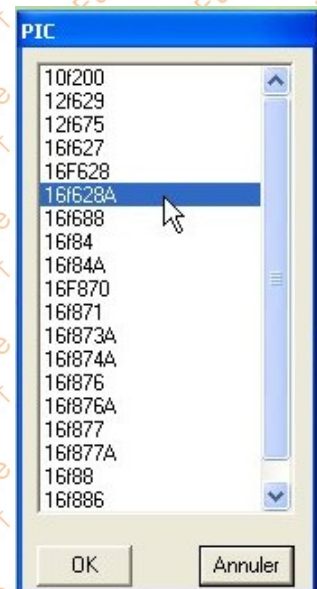
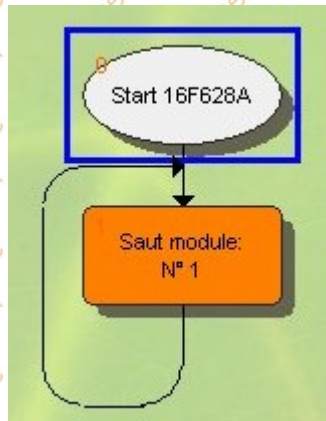
Utilisation d'un pic 16F628A

L'utilisation d'un pic 16F628A n'est pas plus compliquée, il faut cependant repartir d'une nouvelle feuille, il n'est pas possible de reprendre un programme écrit avec le pic 16F84 et de changer le pic.

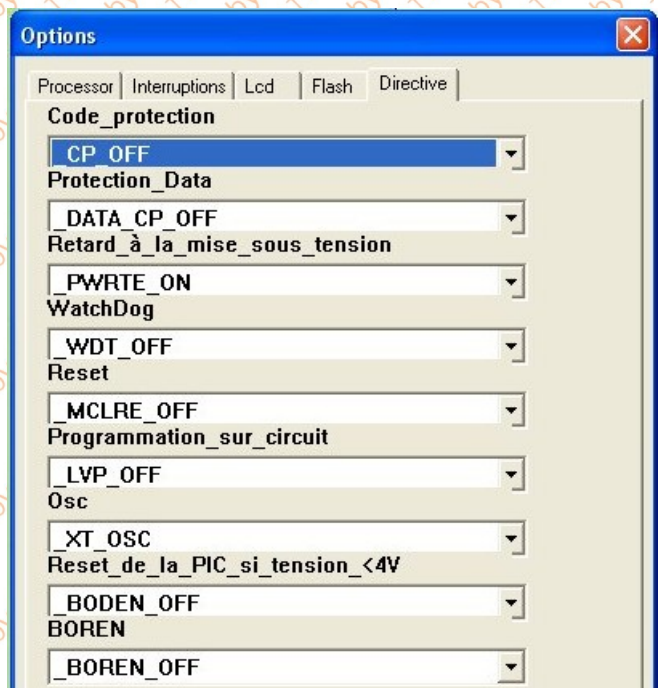
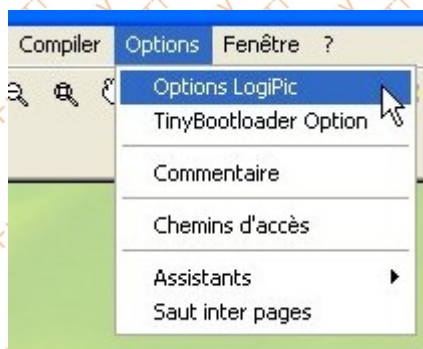
Demander Nouveau



Choisir le pic :

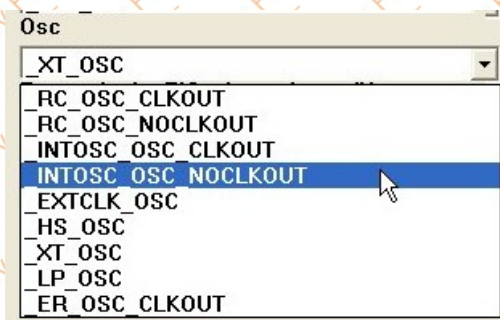


Demander Options > Options Logipic :



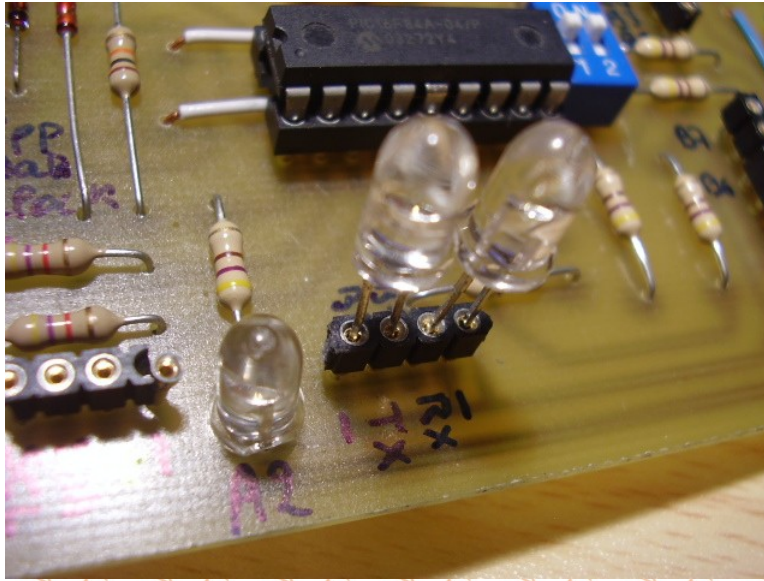
Avec un quartz,
Les directives sont les suivantes :

Sans oscillateur, il suffira de passer en
INTOSC_OSC_NOCLKOUT



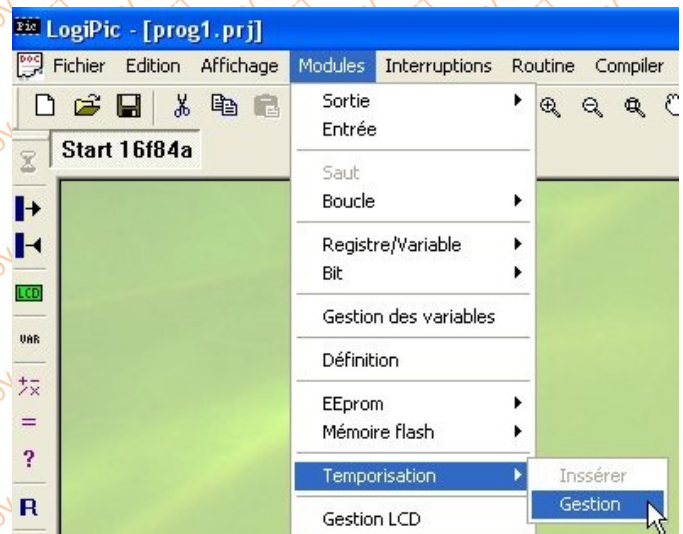
Chenillard à 3 diodes leds

Continuons notre progression avec l'utilisation de plusieurs sorties du pic. Pour cela , disposer 2 autres leds sur les sorties B1 et B2. Le méplat doit être sur la masse (le -)



Nous allons réaliser un programme permettant d'effectuer un chenillard sur les 3 leds A2 , B2 et B1 avec une temporisation de 500ms

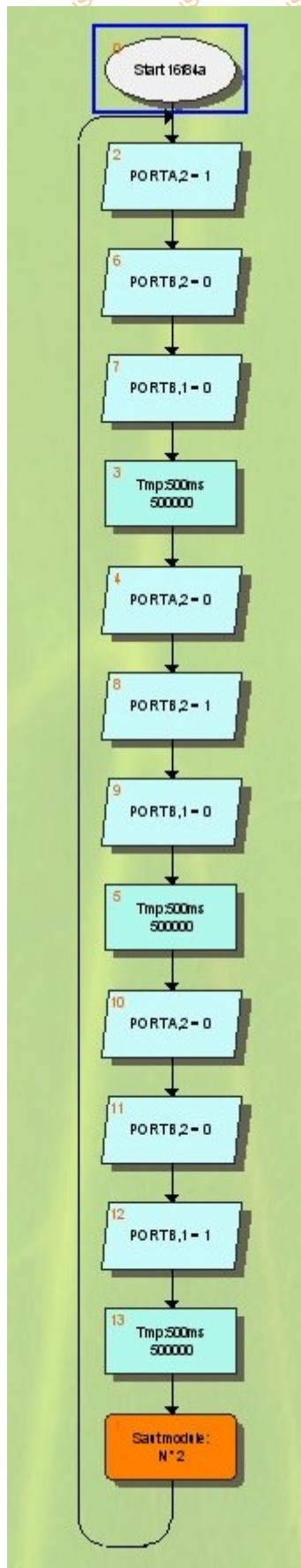
Ouvrir une nouvelle feuille, choisir le pic
Demander Modules > Temporisation > Gestion :



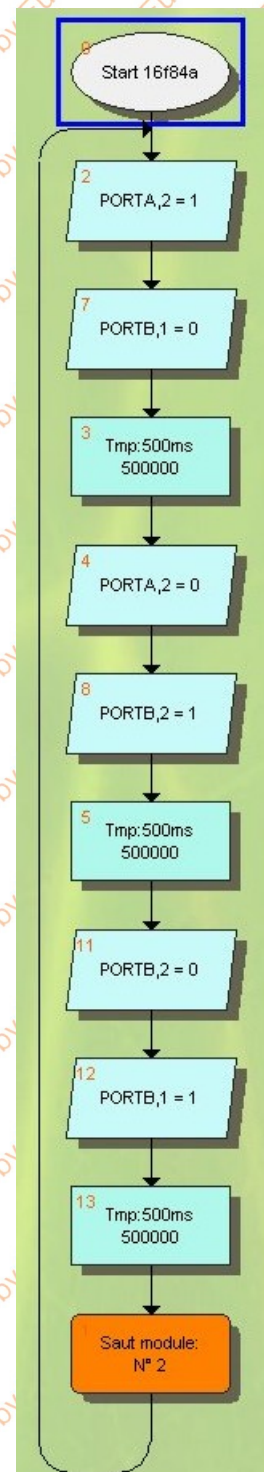
Définir une temporisation de 500ms



La première solution est d'effectuer toutes les commandes de contrôle des portA et portB :



Cependant, l'examen du logigramme permet de se rendre compte qu'il est inutile d'écrire tous les changements de portA et portB le logigramme peut donc être simplifié :



N'oubliez pas de modifier les directives en fonction de votre cible :

rappel pour un fonctionnement avec un quartz 4MHz

Ecrire le logigramme
Assembler, compiler
puis télécharger le programmer
pour vérifier le bon fonctionnement.



Réalisation d'un test

Le 3° programme permettra d'effectuer un test. On utilisera le bouton poussoir de la platine de test connectés sur A0. Ce programme permettra d'allumer la led A2 lorsque l'on appui sur le Bp A0.

Les résistances R3 et R4 connectées à la masse permettent de définir un niveau logique bas (0), l'appui du bouton poussoir relié au +5v permet de définir un niveau logique haut (1).

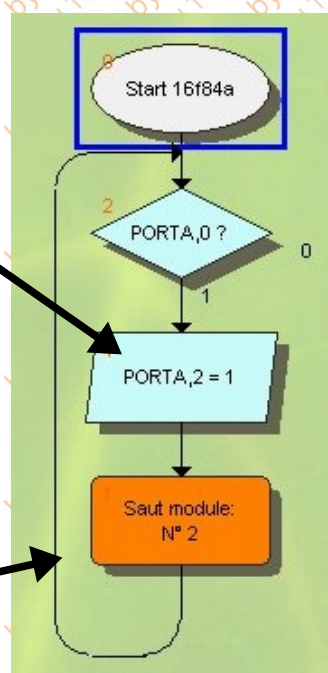
Ouvrir une nouvelle feuille, sélectionner le pic, définir les options et directive.

L'utilisation d'un test s'effectue avec l'icône Entrée



Positionner la sortie PORTA,2 = 1

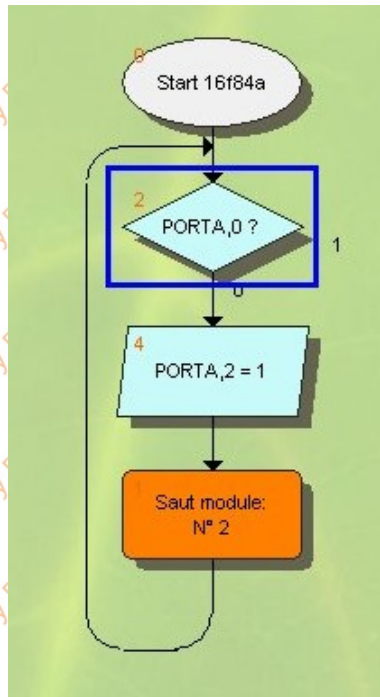
en dessous du test :



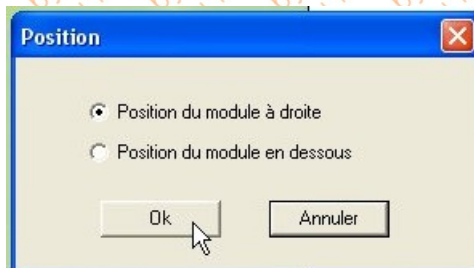
Reboucler l'organigramme

Sélectionner le Test (il s'entoure en bleu)

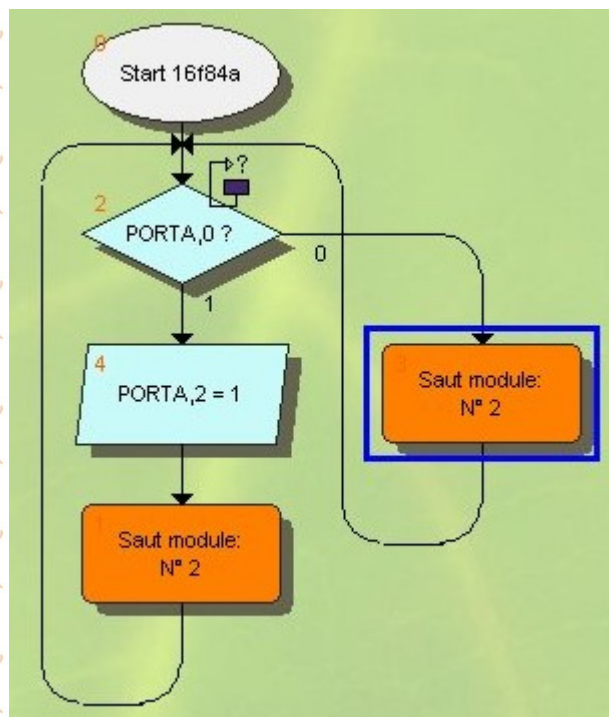
cliquer sur Saut



Demander Position du module à droite



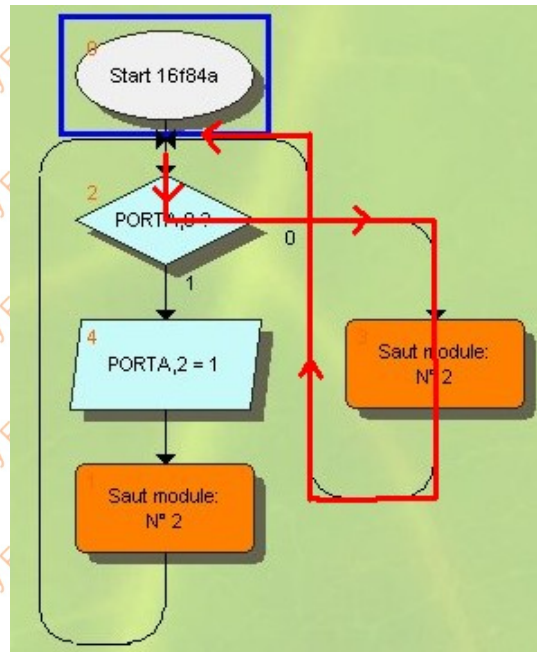
Et reboucler le programme :



Le fonctionnement du programme est très simple :

Sans appui sur le bouton poussoir, le niveau est à 0. Le parcours est le suivant :

Comme il n'y a pas d'instruction, le programme se reboucle en permanence sans rien faire....



Si l'on appui sur le bouton poussoir, le niveau est à 1

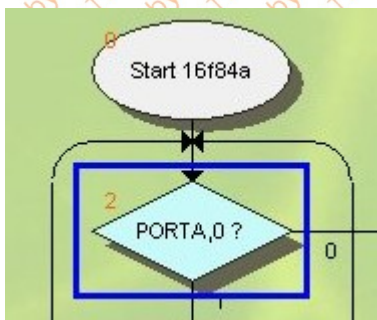
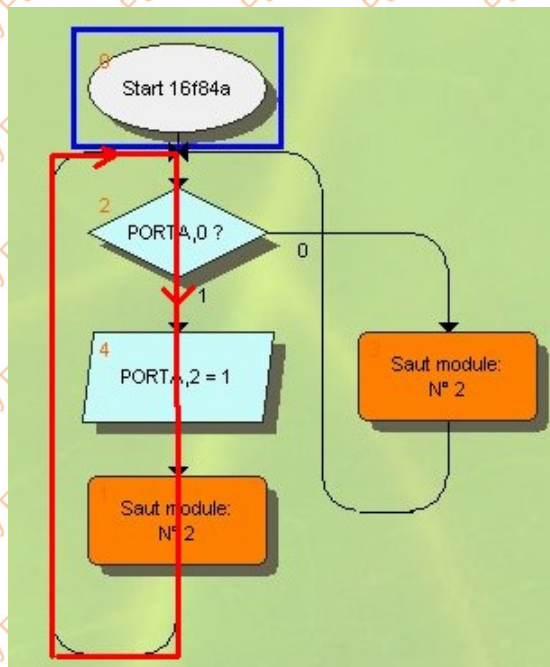
Le parcours est modifié, PORTA,2 = 1

La diode Led A2 s'allume.

Le relachement du bouton poussoir ne repositionne pas PORTA,2 à 0

Si l'on veut éteindre A2, il suffira de positionner une instruction PORTA,2 = 0 dans la boucle 0

Rien de plus simple, se positionner sur le test



Demander Sortie



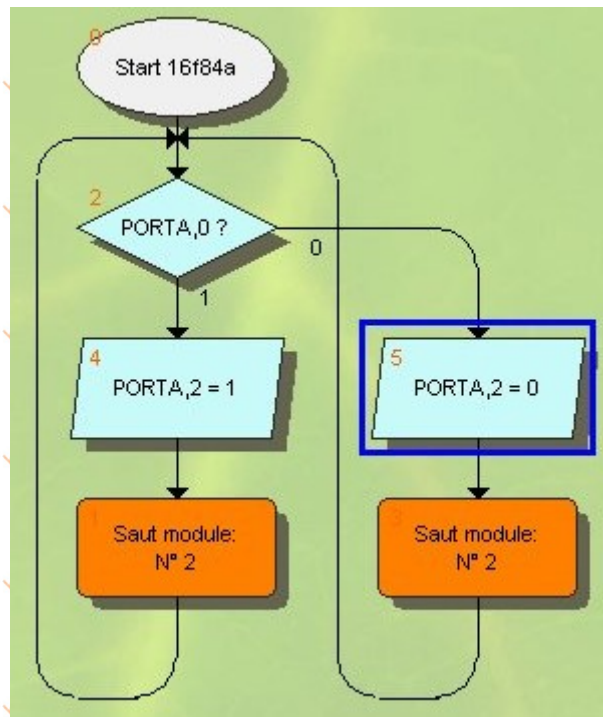
Demander la position du module à droite



Le programme complet est donc le suivant :

Effectuer l'assemblage,
la compilation,

Et télécharger le programme
sur la platine de test.



Prendre de bonnes habitudes

Pour rendre le programme facile à exploiter après plusieurs jours (voire des mois ou des années) mais aussi pour que quelqu'un d'autre puisse comprendre votre programme, je vous conseille de positionner des commentaires.

Les commentaires se positionnent sur l'instruction sélectionnée.

Ensuite le fait de passer la souris sur la commande affiche le commentaire.

