

# LOGIPIC

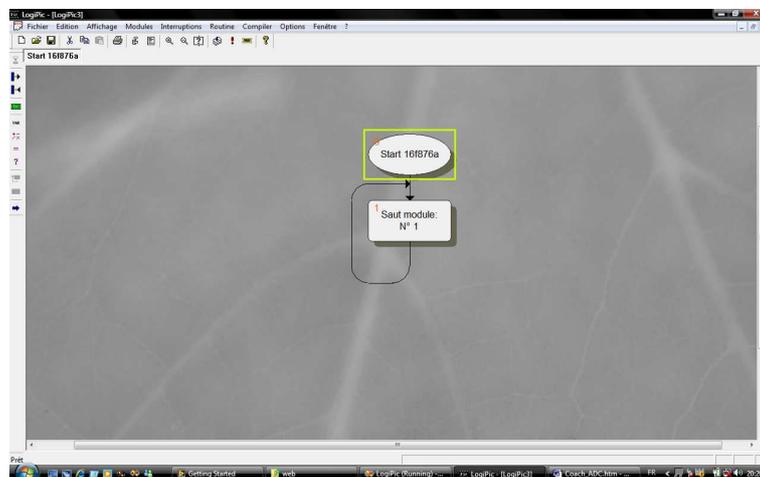
## Logiciel de programmation graphique des microcontrôleurs PIC

### Conversion analogique/numérique.

Certains PIC tels que le 16f876 peuvent gérer la conversion analogique / numérique. Pour faciliter l'usage de cette fonction, j'ai créé des routines qui permettent d'initialiser les registres concernés.

Nous allons créer un programme qui permet de faire varier la fréquence de clignotement d'une led en fonction de la tension présente à l'entrée de la pin d'acquisition, on fera varier la tension avec un potentiomètre.

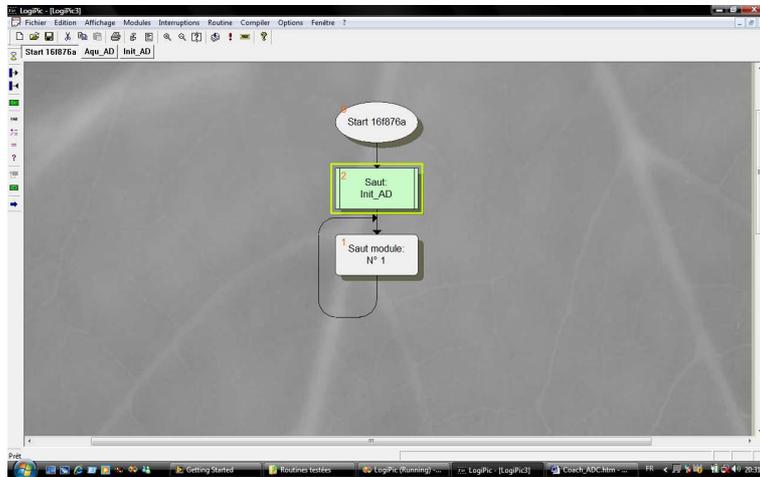
- Lancer LogiPic
- Créez un nouveau programme en choisissant un 16f876a.



Nous allons commencer par insérer les routines de gestion de l'acquisition analogique, celles-ci sont disponibles sur le site dans la rubrique téléchargement sous le nom « AcqAN.rtn » .

- Dans le menu routine, sélectionnez « importer routine », sélectionnez le fichier AcqAN.rtn dans le répertoire où vous l'avez téléchargé.
- Le programme vous fait remarquer que les routines importées comportent des temporisations et (ou) des variables non définies, cliquer sur le bouton « oui » pour les créer.

- Vous pouvez constater que deux routines ont été importées « Aqu\_AD » et « Init\_AD », la première gère l'acquisition à l'instant t, la deuxième initialise les registres au démarrage du programme.
- Dans la boucle principale, nous allons faire un appel vers la routine Init\_AD



La routine « Init\_AD » comporte 3 modules :

Le premier initialise le bit 0 du registre TRISA à 1 pour positionner la pin PORTA,0 (AN0) en entrée. Vous pouvez modifier le bit si vous souhaitez travailler avec une autre entrée analogique, le 16f876 en possède 5 (AN0, AN1, AN2, AN3, AN5).

Le deuxième module initialise le registre ADCON1 :

- b7 : ADFM : 0 justification du résultat à gauche
- b6 : ADCS2 : 0 clock conversion
- b5 : Inutilisé
- b4 : Inutilisé
- b3 : PCFG3 : 0
- b2 : PCFG2 : 0
- b1 : PCFG1 : 0
- b0 : PCFG0 : 0 toutes les entrées configurées en mode analogique

Le résultat est codée sur 10 bits (de 0 à 1023), c'est pour cette raison qu'il est mémorisé sur 2 registres : ADRESL et ADRESH.

Exemple : 111111111 (1023 déc)

- Avec justification à gauche : ADRESH = 11111111      ADRESL = 11000000
- Avec justification à droite : ADRESH = 0000011      ADRESL = 11111111.

Dans notre programme, nous utiliserons la valeur codée sur 8 bits de ADRESH justifié à gauche. Cette valeur sera copiée dans la variable V\_ADH, ce qui signifie que les deux bits de poids faible seront perdus. En négligent ces deux bits, pour une mesure de 0 à 5v, la valeur convertie sera de 0,02v à 4,98v.

Le troisième module initialise le registre ADCON0 :

- b7 : ADCS1 : 0
- b6 : ADCS0 : 1 Fosc/8 (diviseur en fonction de la fréquence du quartz ici 5MHz maxi voir tableau)
- b5 : CHS2 : 0

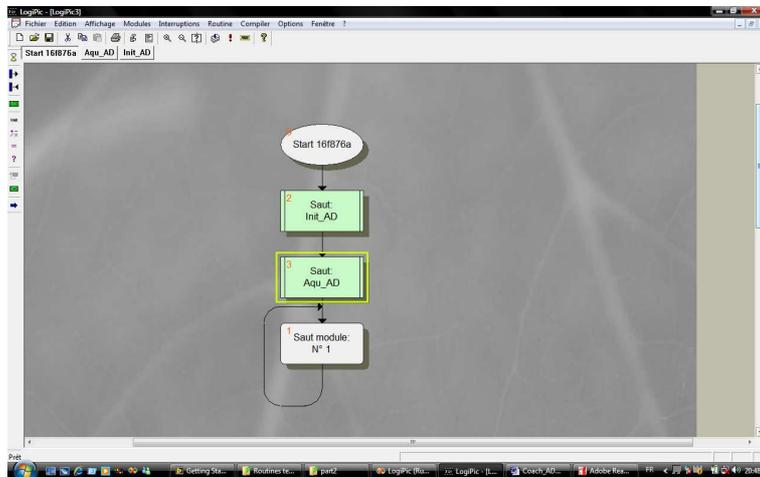
- b4 : CHS1 : 0
- b3 : CHS0 : 0 sélection du canal AN0. (à modifier si entrée analogique différente)
- b2 : GO/DONE : 0
- b1 : Inutilisé
- b0 : ADON : 0 Mise en service du convertisseur, pour le moment à 0.

Tableau de paramétrage des bits ADCS\* en fonction de la fréquence (Datasheet).

**TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (F))**

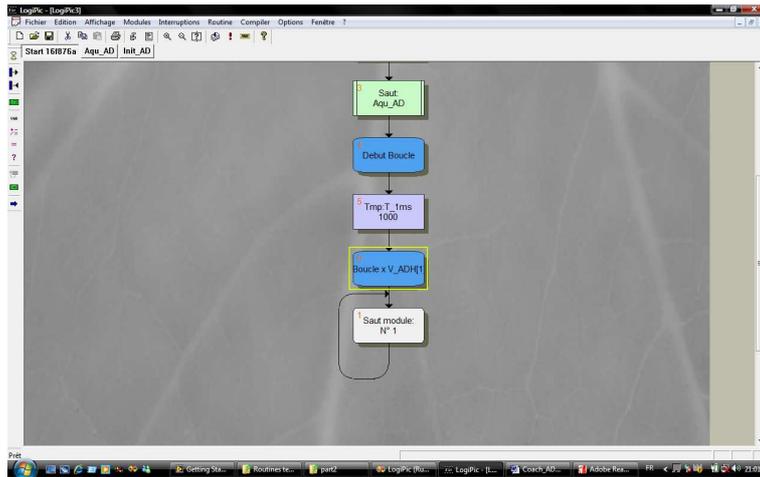
AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS2:ADCS1:ADCS0	
2 Tosc	000	1.25 MHz
4 Tosc	100	2.5 MHz
8 Tosc	001	5 MHz
16 Tosc	101	10 MHz
32 Tosc	010	20 MHz
64 Tosc	110	20 MHz
RC <sup>(1, 2, 3)</sup>	x11	<b>(Note 1)</b>

- Dans la boucle principale, nous allons faire un appel vers la routine Aqu\_AD pour lancer une 1ere acquisition

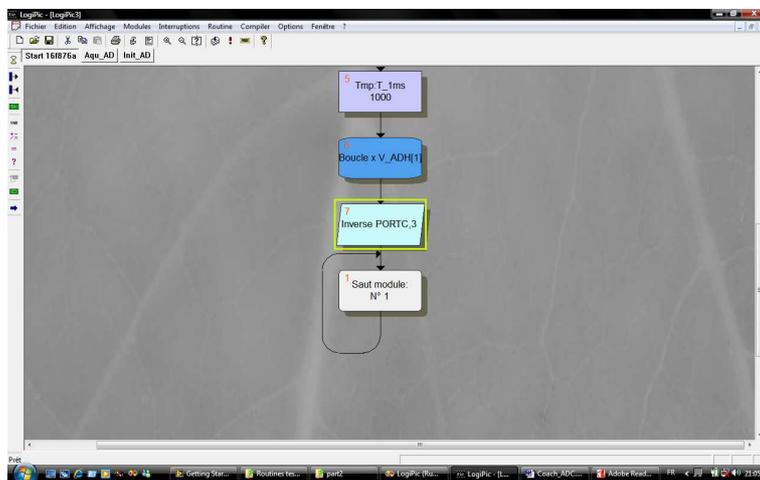


Le but étant de faire varier la durée de clignotement, nous allons donc placer une boucle avec une temporisation, celle-ci sera multipliée par la valeur mesurée.

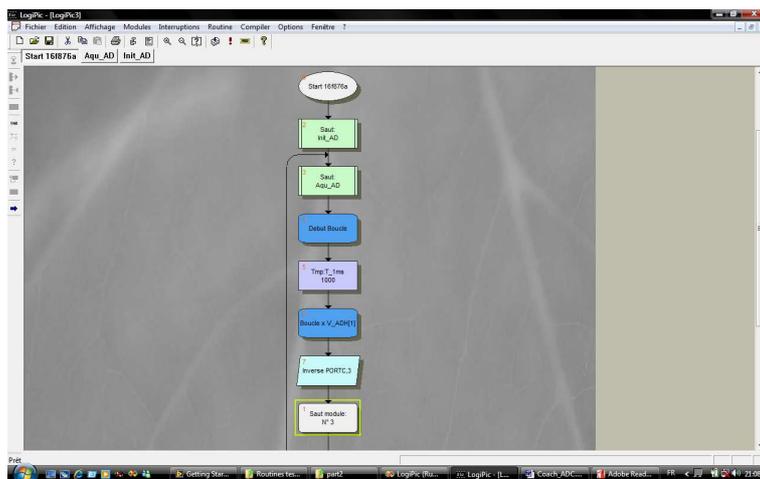
- Insérer un module de début de boucle, une temporisation d' 1ms (1000µs), et un module de fin de boucle de N boucles correspondant à V\_ADH (V\_ADH étant égale au registre ADRESH)



- Insérez un module d'inversion de sortie sur la pin 3 du port C, correspondant à la sortie où sera reliée la résistance de la led.



- Modifiez le retour de saut vers le module 3 (Saut routine Aqu\_AD)



Le programme est maintenant terminé, vous pouvez le compiler avec MPASMWIN et le transférer dans la PIC.

Pour la mise en pratique, il est nécessaire de raccorder un potentiomètre (par exemple 10K) en 0 et 5v avec le curseur relié sur la pin AN0 (PORTA,0).

Lors de la mise sous tension, la led devra clignoter plus ou moins vite en fonction de la position du potentiomètre.