

MULTI-PROG PIC

Multi-prog existe aussi pour les microcontrôleurs Atmel, avec la même interface graphique et des fonctionnalités identiques ou très proches.

1 INTRODUCTION	4
1.1 Utilité de Multi-prog.....	4
1.2 Multi-prog et chaîne de développement.....	5
1.3 Décomposition en algorigrammes	11
1.4 Mise à jour automatique du logiciel	12
2 PRINCIPALES CARACTERISTIQUES DE MULTI-PROG PIC.....	12
2.1 Particularités et limitation du compilateur CC5X	14
2.2 Particularités et limitation du compilateur SDCC.....	14
3 CREATION ET CONFIGURATION D'UN PROJET.....	15
3.1 Boîte de configuration du projet	15
3.1.1 Définition du µC cible et de la carte de programmation	15
3.1.2 Définition des entrées / sorties.....	16
3.1.3 Définition (déclaration) des données.....	16
3.1.4 déclaration des fonctions.....	17
3.1.5 Noms pour les données, les procédures et les fonctions	18
3.1.6 Interruptions et périphériques	19
3.2 Choix du compilateur.....	19
4 ELABORATION D'UN ALGORIGRAMME	19
4.1 Démarche	19
4.2 Constitution d'un programme	20
4.3 Structures de base	20
4.4 Placement et édition des Symboles	23
4.4.1 Traitement (calcul).....	24
4.4.2 Affectation d'une valeur à une sortie (Action).....	25
4.4.3 Procédure (SP).....	25
4.4.4 Structure if / do ... while / while.....	26
4.4.5 Boucle for	27
4.5 Edition des symboles après placement.....	27
4.6 Placement des liaisons	27
4.7 Manipulation de symboles déjà placés	27
4.7.1 Sélection de symboles.....	27
4.7.2 Déplacement de symboles / copier / Coller.....	28
4.8 Insertion de lignes ou colonne entre les symboles déjà placés	28
4.9 Insertion de symboles	28
5 PRODUCTION DU FICHIER EXECUTABLE / PROGRAMMATION	29
5.1 Vue d'ensemble	29
5.2 Traduction automatique en langage C et compilation	30

5.3 Programmation du PIC cible (transfert).....	31
5.3.1 Programmation avec une carte CASTOR ou une carte quelconque avec un connecteur compatible ICD microchip	32
5.3.2 Programmation avec une carte compatible IC-prog	33
6 OBSERVATION DU DEROULEMENT DU PROGRAMME.....	33
7 GESTION DE PERIPHERIQUES & INTERRUPTIONS	34
7.1 Vue d'ensemble	34
7.2 CAN.....	35
7.3 Afficheur LCD	35
7.4 Liaison RS232	36
7.5 Liaison I2C	36
7.6 Génération d'un signal rectangulaire à rapport cyclique variable.....	37
7.7 Gestion du clavier	37
7.8 EEPROM.....	38
7.9 Interruption par débordement du Timer 0	38
7.9.1 Interruption non périodique.....	39
7.9.2 Interruption périodique	39
7.10 Interruption sur front de l'entrée RB0/INT.....	39
8 FONCTIONS ET PROCEDURES	40
8.1 Introduction	40
8.2 Fonction pour CAN	41
8.3 Procédures pour afficheur LCD.....	42
8.4 Liaison RS232	42
8.5 Liaison I2C	43
8.6 Génération d'un signal rectangulaire à rapport cyclique variable.....	43
8.7 Fonction pour Clavier	43
8.8 EEPROM.....	44
8.9 Procédure pour interruption non périodique avec le timer 0	44
8.10 Procédure ou fonction utilisateur	44
8.10.1 Procédure ou fonction définie par un texte.....	45
8.10.2 Procédure ou fonction définie graphiquement.....	46
9 ALGORIGRAMMES POUR LES INTERRUPTIONS	46
10 UTILISATION D'UNE BIBLIOTHEQUE UTILISATEUR	47
11 CONFIGURATION & UTILISATION D'IC-PROG.....	47
11.1 Configuration pour Windows XP	47
11.2 Options d'IC-prog	50
11.3 Programmation du PIC cible	50
12 UN EXEMPLE DE A A Z : THERMOMETRE NUMERIQUE	52
12.1 Introduction.....	52

12.2 Schéma simplifié de la carte mère	53
12.3 Valeurs en différents points de la chaîne de traitement	53
12.4 Configuration du μ C	54
12.5 Le programme complet.....	56
13 FONCTIONNEMENT EN MODE MISE AU POINT	59

ANNEXES

MUTI-PROG PIC

1 INTRODUCTION

1.1 UTILITE DE MULTI-PROG

Multi-prog est un logiciel de Développement Rapide d'Applications simples pour μ C PIC. Il permet de créer rapidement des programmes exécutables sans nécessiter la lecture de la documentation technique du PIC cible et sans apprentissage d'un langage de programmation.

C'est la **programmation graphique**, associée à de nombreuses boîtes de dialogues avec des options, des listes déroulantes, des aides intégrées, qui permet une utilisation particulièrement facile.

Pour utiliser Multi-prog, il suffit d'avoir quelques notions de logique. Voir par exemple mon document « Introduction au microcontrôleur et à sa programmation » (Intro μ C & prog graphique.doc).

Multi-prog est conçu pour être utilisé dans le cadre de manipulations ou dans le cadre de projet et de mini-projets.

Multi-prog est essentiellement destiné à des applications pédagogiques. Il sert introduire les notions élémentaires de la programmation : organisation d'un programme, algorithme, algorithme.

Il ne permet pas d'introduire les notions de structures de données, telles que structures, tableaux de structures, etc. (difficile à assimiler par les élèves). Il accepte cependant les tableaux de constantes et variables de types simples.

Mutliti-prog permet aussi de réaliser de petites applications qui peuvent fonctionner de façon autonome. Ces petites applications peuvent correspondre à des projets ou mini-projets en Bac STI, en BTS, etc.

Le fonctionnement autonome peut être précédé d'une phase d'exécution du programme sur le PIC cible avec une liaison à un ordinateur, pour « voir » le déroulement du programme.

Le fonctionnement autonome est possible avec toutes les cartes.

Multi-prog permet aussi l'apprentissage du C, en confrontant l'algorithme de départ et sa traduction en algorithme et langage C. *Voir détail plus loin.*

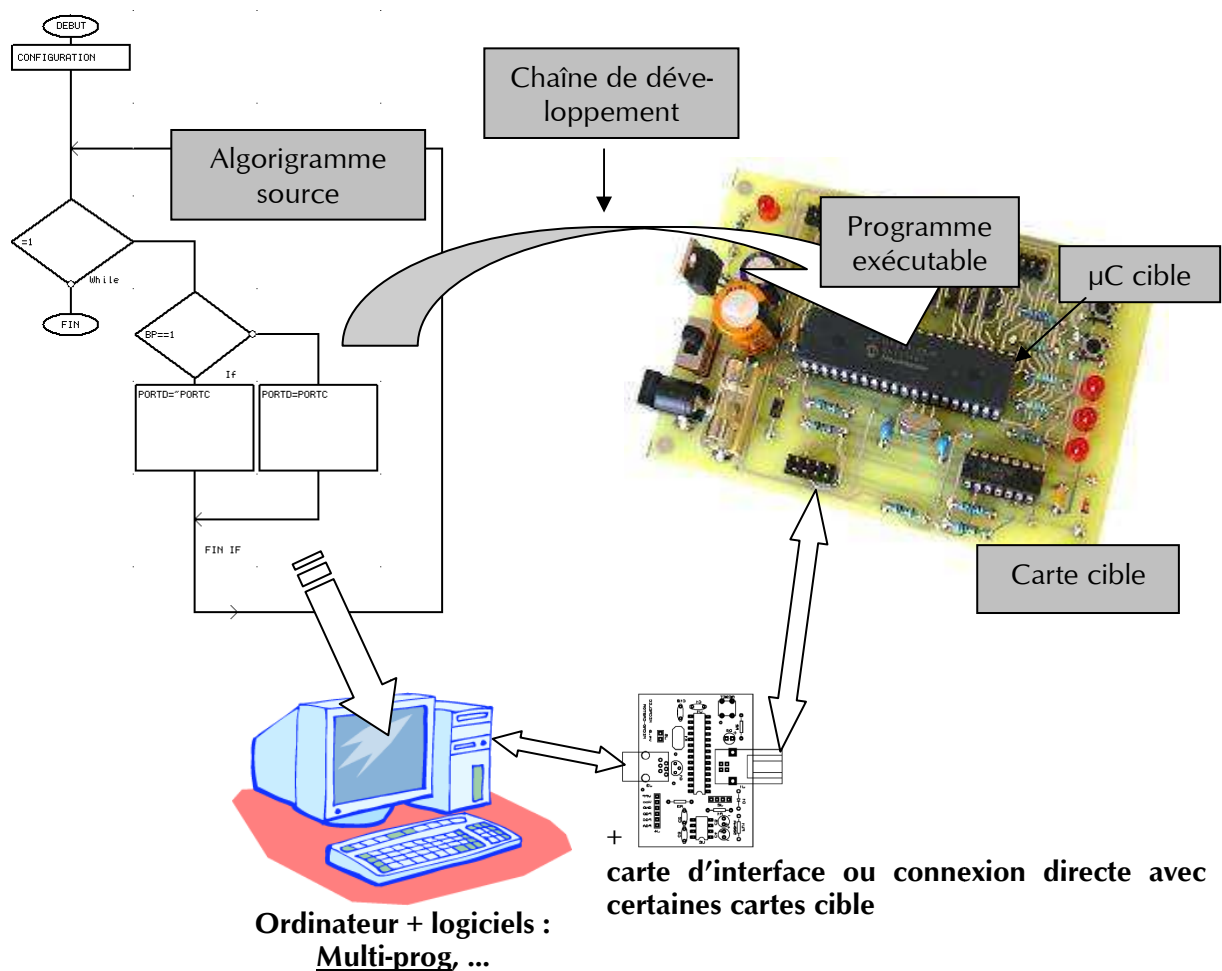
Voici quelques exemples d'utilisation de Multi-prog selon le niveau d'enseignement.

Niveau d'enseignement	Cadre	Remarque
2 nd e option MPI	Partie thématique. Extrait du programme officiel : La partie thématique (laissée à la liberté de l'enseignant en fonction des spécificités locales et de l'intérêt des élèves) pourra être l'occasion [...] d'utiliser quelques notions de programmation (algorithme élémentaire : analyse d'un problème simple, instruction conditionnelle alternative...)	Il est possible par exemple de réaliser un thermomètre autonome très simplement avec une petite maquette comprenant un PIC + un afficheur à cristaux liquides (commande prise en charge par Multi-prog) + un capteur + un conditionneur.
Bac STI Electronique		
Bac S option SI		

STS Systèmes Electroniques	Manipulations pour préparation à l'épreuve écrite. Compétence terminale A3 : Expliciter une structure logicielle commentée. Le candidat dispose [...] de l'algorithme ou de l'algorithme de fonctionnement [...]	Les dernières épreuves du BTS contiennent des questions sur des algorithmes.
STS Systèmes Electroniques	Projet. Compétence terminale C2 : Adapter le logiciel à un nouveau cahier des charges. <i>(remarque personnelle : il est souvent nécessaire de créer un nouveau logiciel, souvent très simple pour créer une procédure de test, etc.)</i>	Extrait des recommandations officielles : On cherche à privilégier l'utilisation d'un langage de haut niveau tel que le langage C ou la programmation graphique en évitant tant que possible l'utilisation de l'assembleur.

1.2 MULTI-PROG ET CHAÎNE DE DEVELOPPEMENT

Multi-prog est au cœur d'une chaîne de développement très simple d'emploi pour μ C PIC.



Multi-prog PIC est un logiciel qui permet :

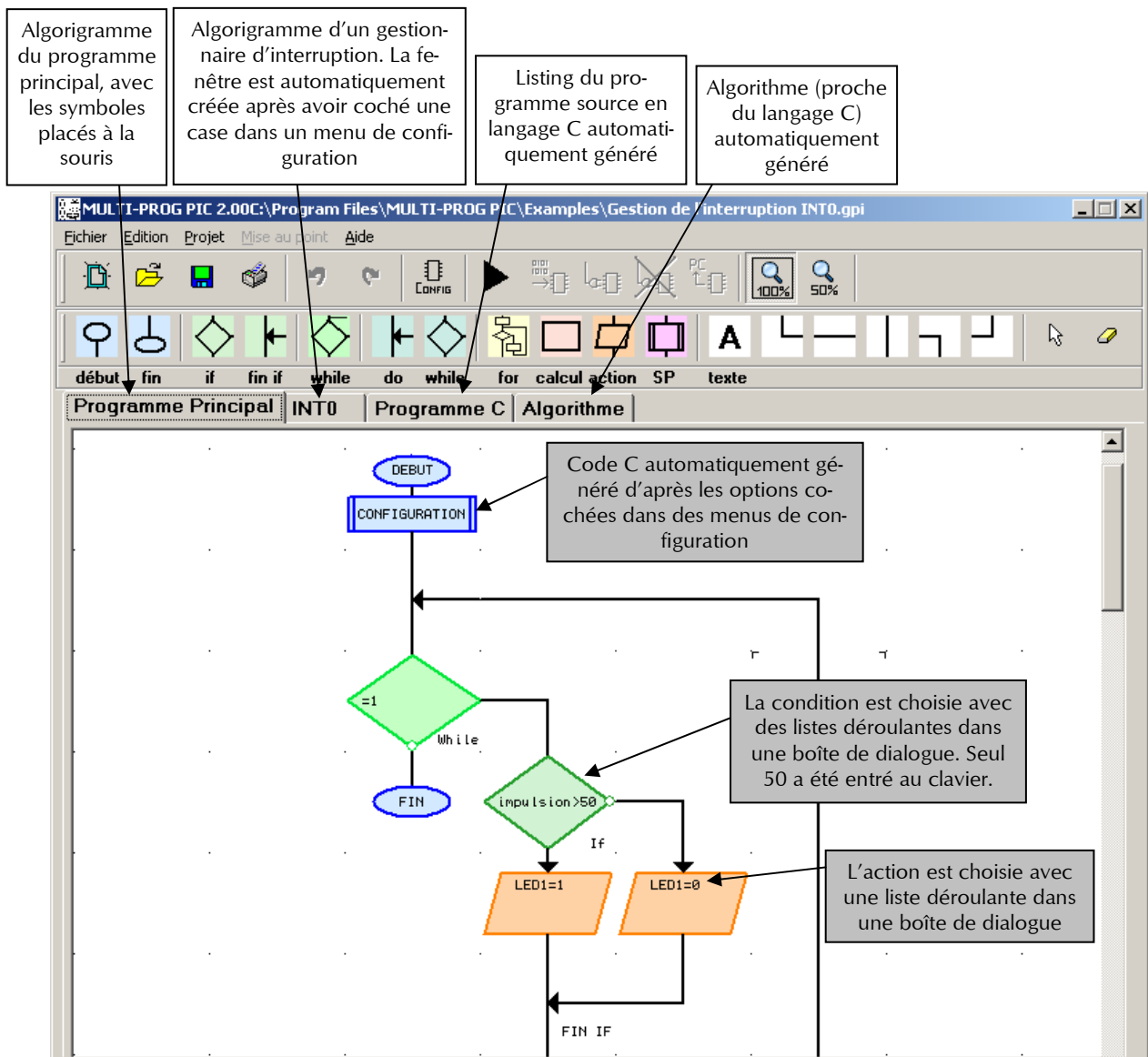
- la **saisie** essentiellement **graphique du programme source** pour une application devant s'exécuter sur un µC **PIC** (avec de nombreuses boîtes de dialogue très intuitives d'utilisation)
- la génération automatique du fichier source en langage C correspondant
- le lancement d'une compilation en utilisant automatiquement de façon transparente à l'utilisateur le compilateur externe CC5X ou SDCC
- la **programmation du PIC cible** en utilisant automatiquement de façon transparente à l'utilisateur un logiciel de programmation Castor-prog (ou IC-prog ou PICFLASH de mikroElektronika). Après programmation, la carte cible peut fonctionner seule.
- l'**exécution du programme avec visualisation du déroulement sur l'algorithme + observation des variables** si on a choisi cette option

L'utilisation de Multi-prog est très intuitive. Il n'est presque jamais nécessaire de se référer au manuel. De nombreuses boîtes de dialogues avec listes déroulantes, aide intégrée, rendent la saisie très simple.

La programmation du PIC cible et l'exécution du programme depuis Multi-prog n'est possible qu'avec certaines cartes.

Le programme est dessiné sous forme d'algorithme ou de Grafcet. Seul le dessin avec un algorithme est développé ici. Le présent document est basé sur les versions 2 de Multi-prog (depuis février 2006) et 1.10 à 1.12 (janvier 2006). Les indications sur ces dernières versions sont le plus souvent en annexe, en fin de ce document.

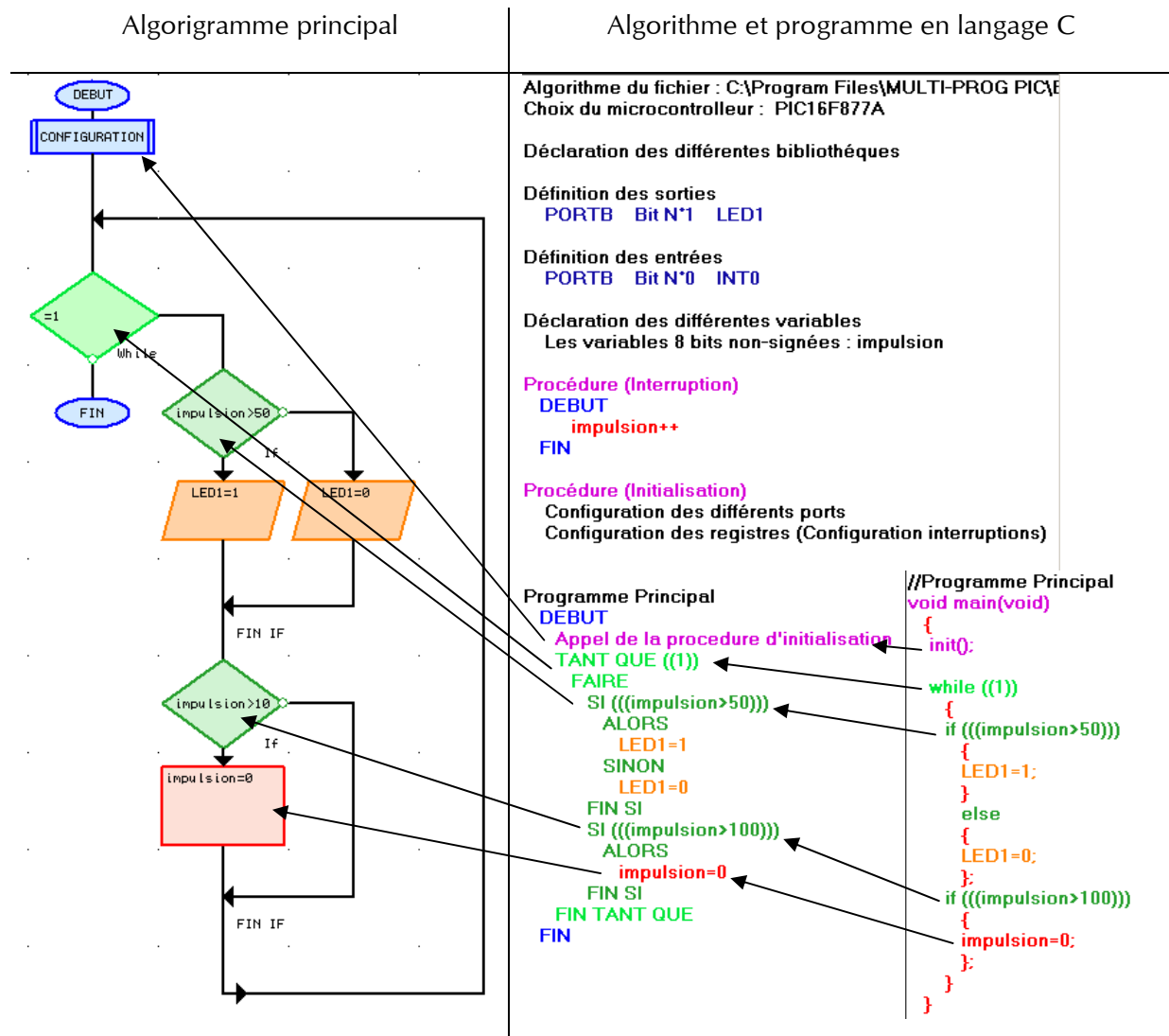
Lorsque le programme source est créé sous forme d'algorithme, la fenêtre principale est la suivante :



Fenêtre principale correspondant à la version 2

La liaison de la carte cible + une carte d'interface avec l'ordinateur (uniquement avec certaines cartes) permet de « voir » le déroulement du programme s'exécutant sur le PIC cible sur l'algorithme.

Multi-prog permet aussi l'apprentissage du C, en confrontant l'algorithme de départ et sa traduction en langage C. L'utilisation de couleurs permet d'identifier facilement les structures.

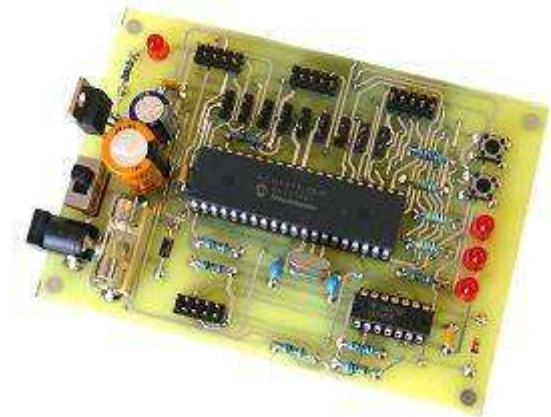


Avec certaines cartes, il est possible de

- Programmer in situ le PIC cible
- Lancer l'exécution du programme et visualiser le déroulement du programme directement sur l'algorithme. On peut aussi observer les valeurs des ports et des variables globales.

Les cartes utilisables avec Multi-prog sont :

- o La carte Castor LPT vendue par le même distributeur que Multi-prog (Micarelec)



Cette carte se connecte directement sur le port imprimante d'un PC. La partie nécessaire pour l'interfaçage est sur la carte.

- o La carte Castor USB vendue par Micrelec

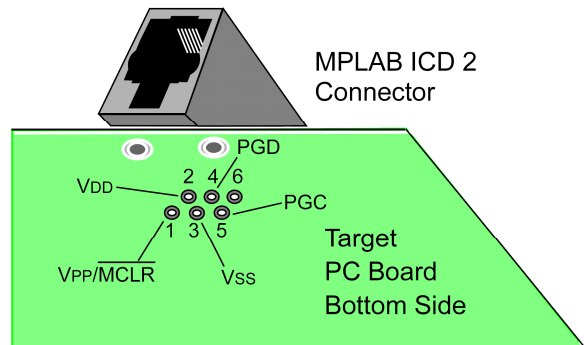


Cette carte se connecte directement sur le port USB d'un PC. La partie nécessaire pour l'interfaçage est sur la carte.

- o Une carte EasyPIC conçue par mikroElektronika →

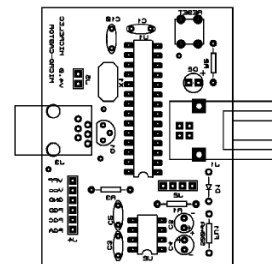
Cette carte se connecte directement sur le port USB d'un PC. La partie nécessaire pour l'interfaçage est sur la carte.

- o N'importe quelle carte avec un des PIC supportés par Multi-prog et équipée d'un connecteur pour la programmation in situ
Ce connecteur peut être du type RJ 12 au format standard Microchip:

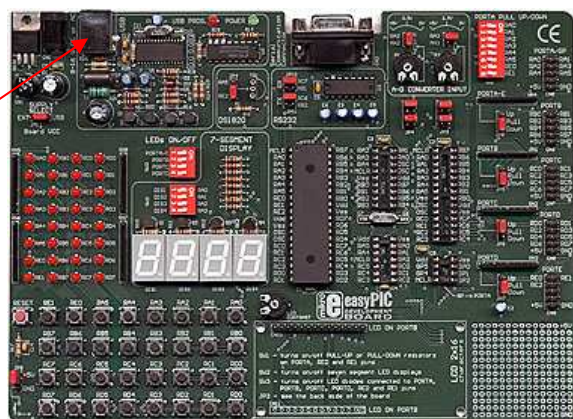


Une petite carte très simple d'interfaçage doit être utilisée entre le PC et la carte cible. L'alimentation de cette carte est prélevée sur la carte cible ou sur le port USB pour la version USB. ↓

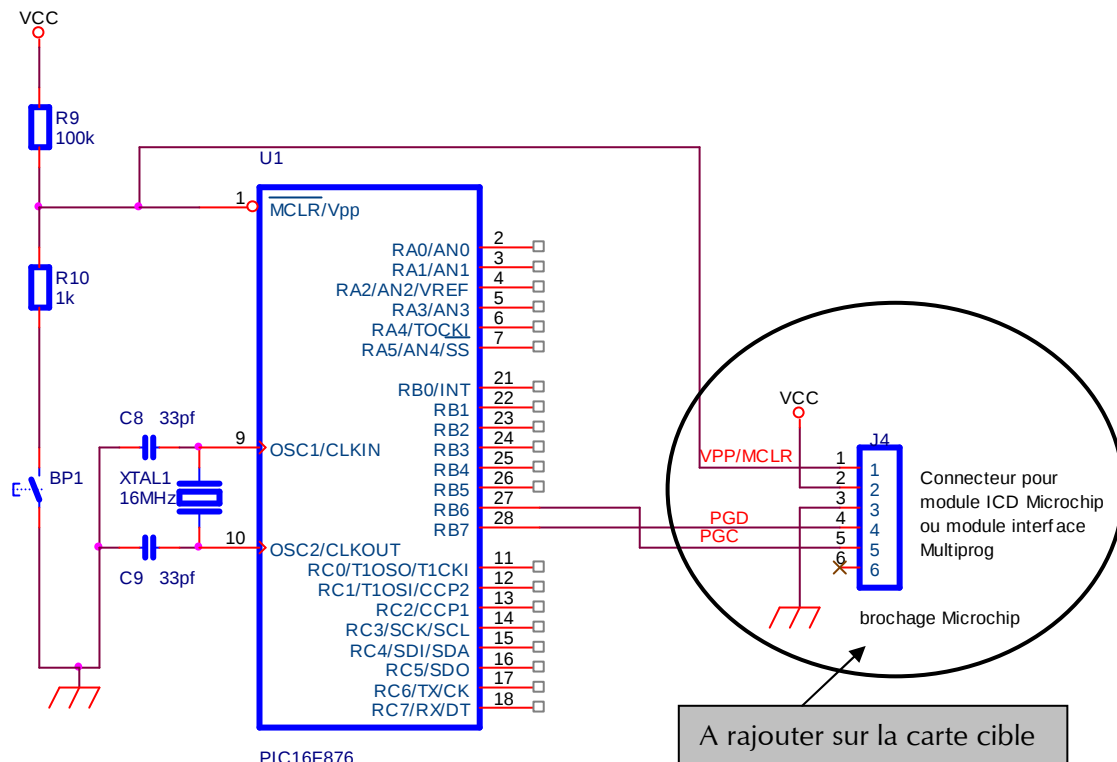
Carte µCastor pour l'interfaçage USB



Il est aussi possible de réaliser une carte d'interfaçage pour le port parallèle du PC (imprimante)



Si on désire réaliser une carte cible pouvant ensuite fonctionner de manière autonome, le schéma à utiliser pour les connexions entre les broches du PIC cible et le connecteur est le suivant :



Avec le schéma ci-dessus, le câble à utiliser entre les 2 connecteurs RJ12 (carte cible et carte interface µCastor) est un câble non croisé. Les connecteurs mâles sont placés sur les 2 côtés du câble plat. Cette disposition rend le câble compatible avec le module ICD1 ou ICD2 réalisé par Touzet. Toutes les informations pour ces modules sont disponibles sur Internet.

Multi-prog est en constante et rapide évolution. Son auteur, **Jean Yves Laronde**, est à l'écoute des utilisateurs et apporte à chaque fois qu'il le peut et qu'il le juge intéressant des modifications pour répondre à de nouveaux besoins.

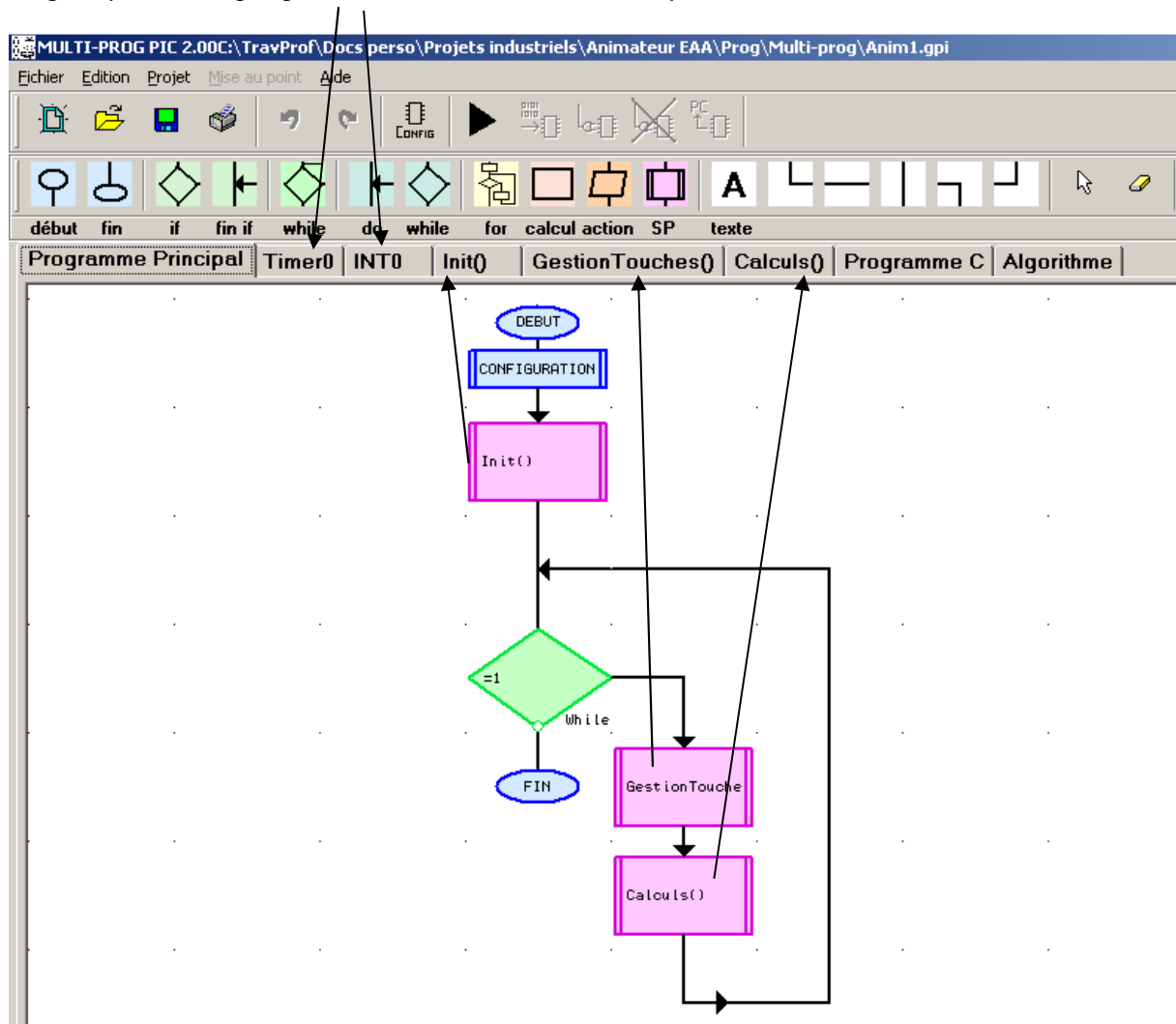
1.3 DECOMPOSITION EN ALGORIGRAMMES

Pour une meilleure lisibilité, le « programme » source peut correspondre à plusieurs algorithmes :

- Un algorithme principal
- Un algorithme pour chaque source d'interruption
- Un algorithme pour chaque bloc SP (Sous Programme = fonction ou procédure).

Tout programme peut se décomposer en algorithmes relativement simples, faciles à analyser.

Onglets pour les algorithmes de traitement d'interruptions.

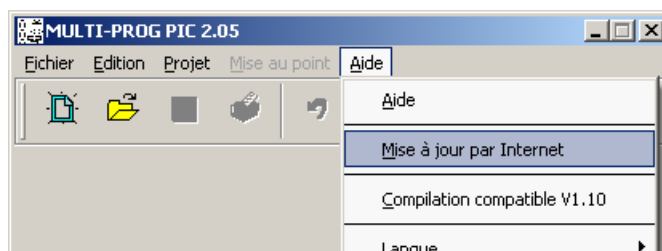


Fenêtre principale correspondant à la version 2

Dans l'exemple ci-dessus, les actions importantes (commandes de triacs) sont effectuées dans un des algorithmes de traitement d'interruption en tâche d'avant plan (foreground). L'algorithme principal correspond à l'arrière plan (background).

1.4 MISE A JOUR AUTOMATIQUE DU LOGICIEL

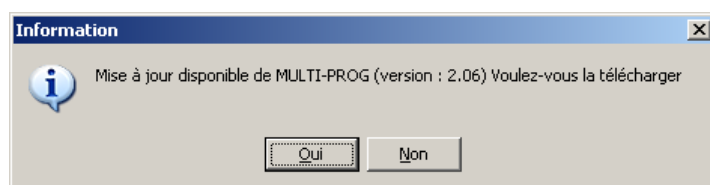
Depuis la version 1.12.6, il est possible de mettre à jour le logiciel par internet en cliquant sur Aide / Mise à jour automatique



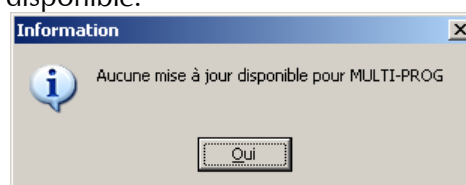
Une boîte de dialogue indique le déroulement des opérations



Si une nouvelle mise à jour est disponible, le téléchargement a lieu.



Sinon, une boîte de dialogue indique qu'aucune nouvelle mise à jour n'est disponible.



2 PRINCIPALES CARACTERISTIQUES DE MULTI-PROG PIC

Le tableau ci-dessous donne quelques unes des caractéristiques de Multi-prog.

PICs supportés	Série 16 PIC16F54*, PIC16F84(A)*, PIC16F627(A), PIC16F628(A), PIC16F648A, PIC16F818, PIC16F819, PIC16F81, PIC16F870 à 877, PIC16F876A, PIC16F877A. Série 18** PIC18F242, PIC18F252, PIC18F442, PIC18F452 <i>* : ces PICs ne permettent pas l'exécution du programme avec visualisation sur l'algorithme car ils n'ont qu'un timer et Multi-prog utilise le 2^{ème} timer.</i> <i>** A la date de rédaction de ces lignes (juin 2007), ces PICs ne sont pas supportés par le logiciel de programmation Castor-prog. Voir §5.3.</i>
Symboles pour l'algorithme	Structures alternative si (if), boucles (do while / while / for), bloc affectation de sorties, bloc traitement, bloc utilisation de fonction (au sens du C, sous programme), bloc temporisation. <i>Dans un bloc traitement ou utilisation d'une fonction, la syntaxe est celle du C (sans le ; en fin de ligne).</i>
Nombre d'alalgorithmes	1 pour le programme principal + 1 pour le gestionnaire de chacune des interruptions gérées (par exemple pour un PIC 16F87x : interruption débordement du timer0, interruption externe sur l'entrée RB0, interruption en réception RS232) + 1 algorithme pour chaque fonction ou procé-

	dure définie par l'utilisateur en mode graphique.
Compilateurs externes	Multi-prog offre le choix entre CC5X (Multi-prog est livré avec la version gratuite limitée) et SDCC (compilateur gratuit GNU sans limitation)
Types de données	Multi-prog supporte les variables de types simples et les tableaux (constantes ou variables). Pour les tableaux de constantes, seule une dimension est supportée. Les types de variables supportés dépendent du compilateur utilisé. Types simples: - entiers non signés et signés sur 8 et 16 bits (+32 bits avec SDCC) - réel en notation flottante sur 24 bits (CC5X. <i>Non pris en charge pour l'instant par Multi-prog au niveau des déclarations</i>) ou sur 32 bits (SDCC. <i>Non pris en charge pour l'instant par Multi-prog au niveau des déclarations</i>) - Tableau à 1 dimension seulement avec tous les types précédents (CC5X) ou tableaux à 2 dimensions et plus (SDCC). Structure autorisée, avec comme champ tous les types précédents, y compris un tableau. (CC5X. <i>Non pris en charge pour l'instant par Multi-prog au niveau des définitions</i>)
Visualisation	Algorithme + code en langage C automatiquement générés. Non modifiable. En mode mise au point : déroulement du programme sur algorithme + visualisation des ports et des variables. Mode continu (qqx dixièmes de secondes par bloc) ou pas à pas. <i>Le déroulement n'est pas visible sur les algorithmes des interruptions.</i>
Taille du code produit	1024 mots max avec la version gratuite du compilateur CC5X. En achetant la version complète, la seule restriction est la taille de la mémoire du PIC cible. Pas de limitation autre que la mémoire disponible dans le PIC avec le compilateur SDCC.
Fichiers produits	Toutes les indications pour l'algorithme (ou le Grafcet) sont dans le fichier NomProjet.gpi. Les fichiers suivants sont dans le dossier d'installation de Multi-prog (\Program Files\MULTI-PROG PIC) : pic.c (langage C. idem code visualisé + instructions cachées pour la mise au point) / pic.asm (listage en langage d'assemblage) / pic.hex (fichier pour la programmation). Avec certaines versions de Multi-prog, les fichiers précédents peuvent être dans le dossier du projet et se nommer NomProjet.c, NomProjet.asm, NomProjet.hex, NomProjet.occ (avec le compilateur CC5X uniquement. Fichier statistique)
Registres et bits de config du PIC directement accessibles	Oui par leur nom. Certains noms peuvent différer de leur désignation Microchip. Voir le fichier .h fourni avec le compilateur et correspondant au PIC utilisé. Avec CC5X, ce fichier est placé dans le dossier d'installation de Multi-prog. Avec SDCC, le fichier est dans Program Files\SDCC\include\PIC
Ressources réservées en mode mise au point	Timer 1 + 2 broches RB6 et RB7 déjà utilisées pour la programmation in situ
Taille du code supplémentaire en mode mise au point	≈ 200 mots + qqx mots par bloc sur l'algorithme

2.1 PARTICULARITES ET LIMITATION DU COMPILATEUR CC5X

La version fournie avec Multi-prog est la version gratuite limitée à 1kmo de code exécutable. La version gratuite a une optimisation limitée. La version complète permet de gagner de 10 à 20% sur la taille du code exécutable. Il est possible d'acheter la version complète et de placer le compilateur dans le dossier d'installation de Multi-prog. <http://www.bknd.com/cc5x/index.shtml>

La documentation du compilateur CC5X n'est pas placée automatiquement dans le dossier d'installation de Multi-prog. Il est possible de se procurer la documentation sur Internet : <http://www.bknd.com/cc5x/index.shtml>

Le manuel complet est cc5x-32.pdf. Voir aussi le document de l'auteur de ces lignes Compilateur C CC5X.doc. En voici quelques extraits :

Le compilateur a des possibilités limitées pour allouer des variables temporaires → A cause de cela les instructions complexes ont souvent à être réécrites en décomposant en plusieurs instructions plus simples. De toute façon, Multi-prog ne permet pas d'écrire des lignes longues.

En plus des bases du C ANSI, le compilateur CC5X accepte la base 2.

Notation : 0bnombre ou 0Bnombre. Ex : 0b01110011

Il est possible de séparer les bits par des points pour plus de visibilité.

Exemple : 0b1101.1001, 0b1.101.11.00

On peut donc écrire une ligne du type suivant dans une boîte Multi-prog :

Var = 0b1101.1001

Les variables globales initialisées ne sont pas supportées.

Le compilateur n'initialise pas à 0 les variables globales, comme un compilateur C ANSI. Si la valeur initiale 0 est importante, il faut placer une ligne de code dans un bloc de traitement en tout début d'algorithme.

2.2 PARTICULARITES ET LIMITATION DU COMPILATEUR SDCC

SDCC = Small Device C Compiler. C'est un compilateur gratuit, sans aucune limitation sur la taille du code produit. Il supporte tous les PICs acceptés par Multi-prog + de nombreux autres. La dernière version du compilateur peut être téléchargée à l'adresse :

<http://sdcc.sourceforge.net/>

Le PIC est le dernier µC supporté par le compilateur. La documentation actuelle donne peu d'indications sur les particularités du compilateur. Voir les prochaines versions de la documentation.

A la date de rédaction de ces lignes (avril 2006), Le code produit par le compilateur SDCC n'est pas optimisé. Le code produit est jusqu'à 75% plus important que le code produit par la version gratuite de CC5X, qui n'est pourtant pas complètement optimisée. Cette augmentation de la taille est surtout importante pour les petits programmes. Pour les programmes de tailles plus importantes, le code produit avec SDCC est de l'ordre de 50% plus important qu'avec CC5X.

3 CREATION ET CONFIGURATION D'UN PROJET

La configuration du projet, s'effectue avec :

- la boîte de Configuration du projet
- le choix de la carte cible
- le choix du compilateur

Lors du lancement de Multi-prog ou après la fermeture du projet en cours d'édition, la fenêtre de travail est grise et seules quelques rubriques des menus sont accessibles.

Pour créer un nouveau menu, il faut cliquer sur Fichier / Nouveau. Ceci a pour effet d'ouvrir la **boîte de configuration du projet**.

3.1 BOITE DE CONFIGURATION DU PROJET

La boîte de configuration du projet est automatiquement ouverte lors de la création d'un nouveau projet. Elle peut aussi être ouverte à tout moment avec la commande Projet / Configuration (Algorigramme) ou Configuration (Grafcet).

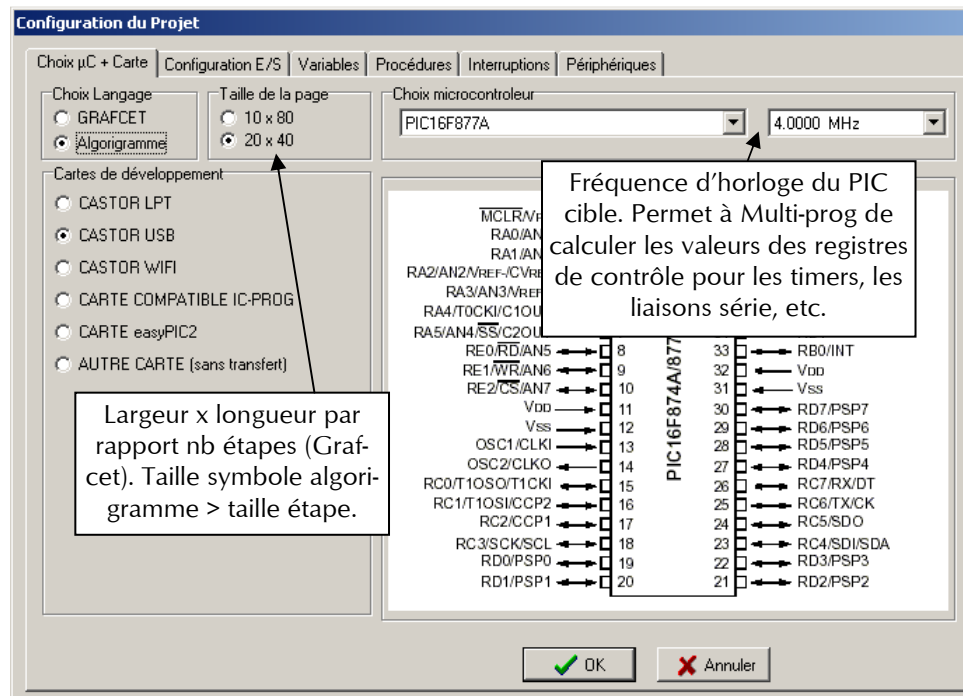
Tous les paramètres peuvent être modifiés sauf la taille de la page qui est choisie lors de la création. Elle ne peut plus être modifiée ensuite.

La boîte de configuration de projet dispose de plusieurs onglets.

3.1.1 DEFINITION DU µC CIBLE ET DE LA CARTE DE PROGRAMMATION

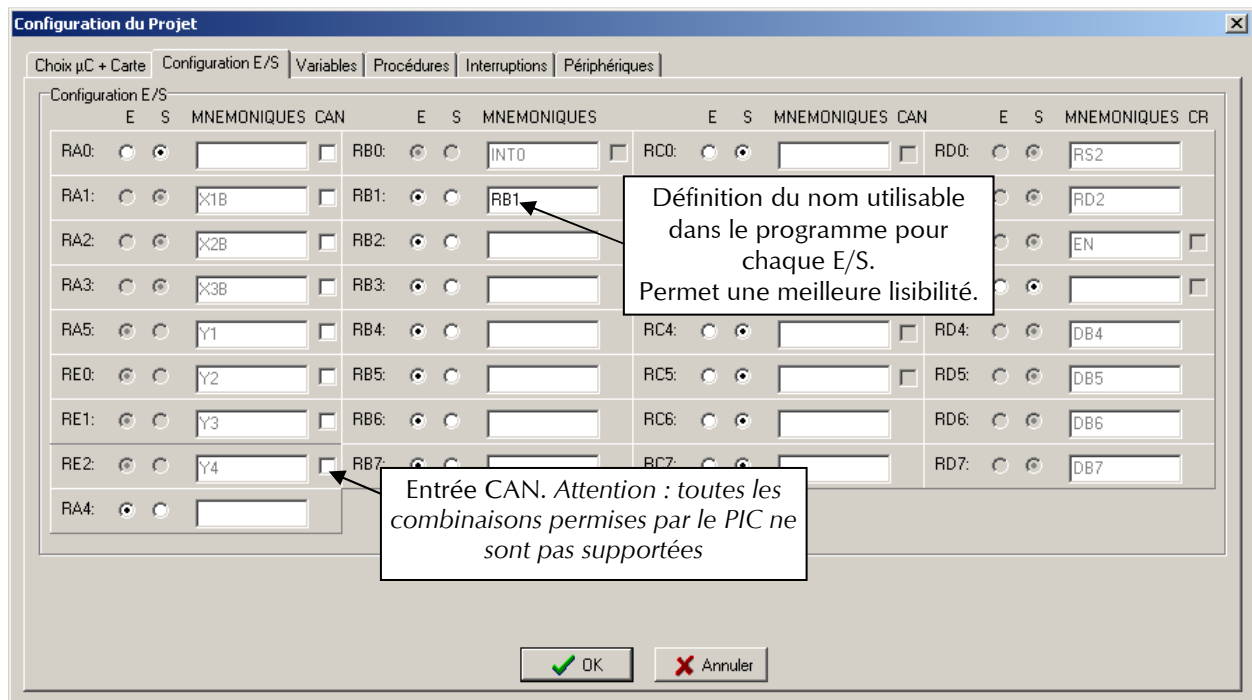
Le premier onglet permet de choisir le µC cible, la fréquence de l'oscillateur (quartz uniquement) et la carte de programmation :

Certaines cartes de développement disposent de l'interface de programmation. La carte de programmation / développement peut aussi être choisie à partir du menu.



3.1.2 DEFINITION DES ENTREES / SORTIES

Le deuxième onglet permet de définir les entrées / sorties



3.1.3 DEFINITION (DECLARATION) DES DONNEES

Le troisième onglet permet de définir les variables et les tableaux de constantes.

Définition (ou déclaration) de variables

Il suffit d'entrer un nom de variable par ligne (sans aucun signe suivant la variable – pas de ;)
On peut déclarer un tableau avec NomVar[Taille]. Taille est un entier. Les valeurs du tableau sont fixées et modifiées dans le programme.

Les types de variables dépendent du compilateur utilisé. Multi-prog permet de choisir entre CC5X (version gratuite limitée) et SDCC (compilateur gratuit non limité). Ci-dessous, l'onglet Variables avec le compilateur SDCC.

Configuration du Projet

Choix µC + Carte | Configuration E/S | **Variables** | Procédures | Interruptions | Périphériques

Variables à déclarer (Une variable par ligne-20 variables maxi par catégorie)

CHAR (-128 à 127)	UNSIGNED CHAR (0 à 255)	SHORT (-32768 à 32767)	UNSIGNED SHORT (0 à 65535)	LONG (Variable 32 bits(+))	UNSIGNED LONG (Variable 32 bits(+))
	R_Cmpt[6] Db1Ch[3]				

☐ Tableau N°1
☐ Tableau N°2

Déclaration de tableaux de constantes :

Il faut cocher la case Tableau dans la zone de déclaration des données

☒ Tableau N°1 ☐ Tableau N°2

Valeurs

Lorsqu'on clique sur Valeurs, une boîte de dialogue s'ouvre :

Tableau de constantes N°1 : Coeff[]

Nom du tableau : Coeff

Nombre de Valeurs : 5

☐ CHAR (-128 à 127)
☒ UNSIGNED CHAR (0 à 255)
☐ SHORT (-32768 à 32767)
☐ UNSIGNED SHORT (0 à 65535)

Valeur N°0	12	Valeur N°10	0	Valeur N°20	0	Valeur N°30	0
Valeur N°1	45	Valeur N°11	0	Valeur N°21	0	Valeur N°31	0
Valeur N°2	69	Valeur N°12	0	Valeur N°22	0	Valeur N°32	0
Valeur N°3	33	Valeur N°13	0	Valeur N°23	0	Valeur N°33	0
Valeur N°4	22	Valeur N°14	0	Valeur N°24	0	Valeur N°34	0
Valeur N°5	0	Valeur N°15	0	Valeur N°25	0	Valeur N°35	0
Valeur N°6	0	Valeur N°16	0	Valeur N°26	0	Valeur N°36	0
Valeur N°7	0	Valeur N°17	0	Valeur N°27	0	Valeur N°37	0
Valeur N°8	0	Valeur N°18	0	Valeur N°28	0	Valeur N°38	0
Valeur N°9	0	Valeur N°19	0	Valeur N°29	0	Valeur N°39	0

OK

Il suffit de compléter la boîte de dialogue.

L'utilisation d'un tableau supprime un type de donnée disponible. Cette particularité peut être annulée dans une prochaine version de Multi-prog.

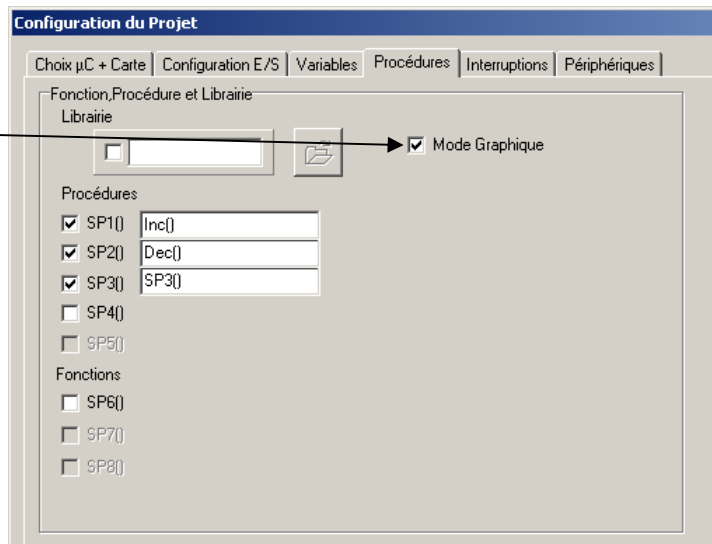
3.1.4 DECLARATION DES FONCTIONS

Le quatrième onglet permet de déclarer les fonctions

Les fonctions et procédures peuvent être définies soit toutes avec du texte soit toutes graphiquement, selon la case Mode Graphique.

Si le mode graphique est choisi, l'aspect est le suivant :

Les fonctions sont détaillées plus loin.



3.1.5 NOMS POUR LES DONNEES, LES PROCEDURES ET LES FONCTIONS

Le langage C utilise des noms réservés qui sont interdits pour nommer les données et les fonctions. Multi-prog utilise aussi des fonctions propres, de façon transparente à l'utilisateur. Les noms de ces fonctions ne doivent pas être employés par l'utilisateur pour nommer d'autres fonctions.

Les tableaux ci-dessous donnent tous les noms réservés :

Noms réservés du langage C :

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

Noms réservés pour les fonctions non documentées de Multi-prog :

init()	i2c_init()	lcd_init()	delay_ms()	delay_us()	lcd_command()		...
---------------	-------------------	-------------------	-------------------	-------------------	----------------------	--	-----

Les autres fonctions non documentées ont des noms tels qu'il est sûr que l'utilisateur ne pourra les employer pour ses propres fonctions.

Remarque : Comme le C fait la distinction entre minuscule et majuscule, il est possible d'utiliser des noms avec même orthographe, mais avec une casse différente.

Par exemple, la fonction Init() créée par l'utilisateur sera acceptée car différenciée de init() utilisée par Multi-prog.

Le compilateur **CC5X** utilise aussi les noms suivants : bit, int8, s8, int16, s16, int24, s24, int32, s32, uns8, u8, uns16, u16, uns24, u24, uns32, u32.

Le compilateur **SDCC** utilise les noms suivants : **A compléter**

3.1.6 INTERRUPTIONS ET PERIPHERIQUES

Les interruptions et les périphériques sont développés plus loin.

3.2 CHOIX DU COMPILATEUR

Le **compilateur** est choisi à l'aide du menu Projet.

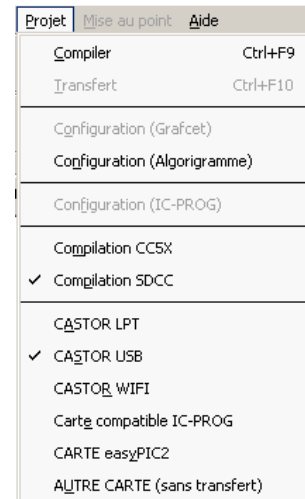
Le menu projet permet aussi de choisir la carte de programmation. La partie nécessaire à la programmation peut aussi être sur la carte de développement.

Un des items CASTOR est à cocher si on utilise une carte spécifique CASTOR ou une carte quelconque avec un connecteur compatible Microchip ICD. Dans ce cas, il faut utiliser une petite carte d'interface (µCastor par exemple).

La carte compatible IC-PROG est une carte quelconque avec un connecteur compatible Microchip ICD. Elle nécessite une carte d'interface. Voir plus haut.

La carte easyPIC2 de mikroElektronik (ou une autre carte de la même série) se connecte directement au PC par la liaison USB.

Avec AUTRE CARTE, la programmation ne s'effectue pas in situ.



4 ELABORATION D'UN ALGORIGRAMME

L'élaboration d'un algorigramme est très intuitive. Seules les grandes lignes sont données ici. Pour plus de détail, voir le fichier d'aide qu'on peut lancer depuis Multi-prog avec la commande Aide / Aide.

4.1 DEMARCHE

La démarche est la suivante :

- Placement de symboles + renseignements dans des boîtes de dialogues automatiquement ouvertes lors du placement
- Placement des liaisons
- Déplacement des blocs pour la réorganisation nécessaire en cours de saisie
- Ajout de commentaires pour une meilleure lisibilité

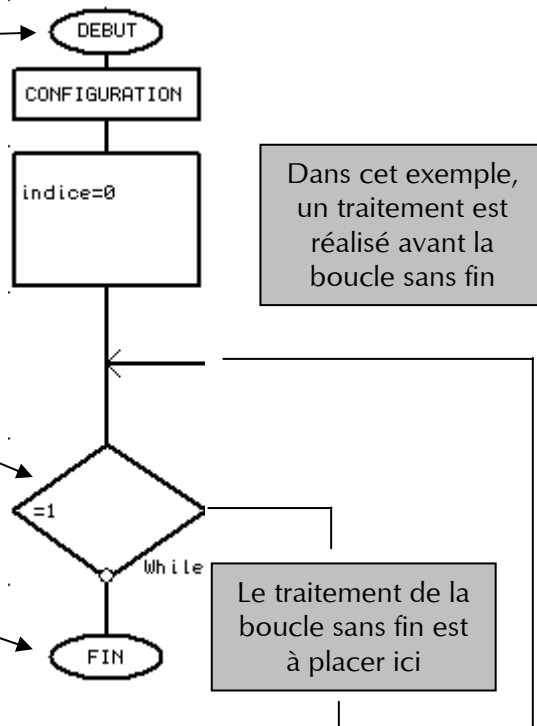
Une fois l'algorigramme terminé, on lance la traduction automatique en langage C et la compilation, puis la programmation du PIC cible et enfin l'exécution du programme.

Un algorithme commence nécessairement par un bloc début.

Le bloc configuration fait partie du bloc début. Il correspond au code qui sera automatiquement généré lors de la traduction automatique algorithme → langage C

L'algorithme principal comporte obligatoirement une boucle sans fin réalisée avec une structure while la condition est de type valeur logique = 1. On ne passe jamais par la sortie non du test.

Tout algorithme se termine par un bloc fin.



4.2 CONSTITUTION D'UN PROGRAMME

Un programme graphique est constitué d'un ou plusieurs algorithmes :

- Un algorithme obligatoire pour le programme principal (correspond au main() en langage C).
- Un algorithme pour chacun des gestionnaires d'interruption (ou programme de traitement d'interruption, ISR en anglais = Interrupt Service Routine), si une ou plusieurs interruptions sont utilisées.
- Un algorithme pour chacune des fonctions définies graphiquement

Pour l'instant les interruptions suivantes peuvent être utilisées :

- interruption provoquée par le timer 0
- interruption sur entrée externe RB0
- interruption provoquée par la réception d'un caractère sur la liaison série asynchrone (RS232)

Une nouvelle fenêtre, accessible par un onglet, est automatiquement créée lorsqu'on coche l'option d'utilisation d'une interruption.

Les interruptions sont décrites plus loin.

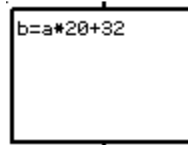
4.3 STRUCTURES DE BASE

Un algorithme est constitué d'un assemblage de structures de base. Les noms placés au-dessus des dessins des symboles dans la barre d'outils correspondent aux instructions en langage C.

Pour certains symboles placés sur la feuille de travail, le nom de l'instruction en langage C est rappelé.

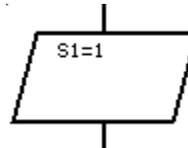
TRAITEMENT (CALCUL)

Un traitement correspond à une ou plusieurs instructions. Le symbole traitement permet d'écrire 4 lignes d'instructions. Une instruction peut utiliser un appel de fonction. Voir plus loin.



OPERATION DE SORTIE (ACTION)

Ce symbole ne permet que d'affecter une valeur à une sortie. Jusqu'à 4 affectations peuvent être réalisées par symbole



PROCEDURE

Ce symbole permet de réaliser :

- une temporisation
- un appel de sous programme sans valeur de retour (procédure au sens du Pascal)

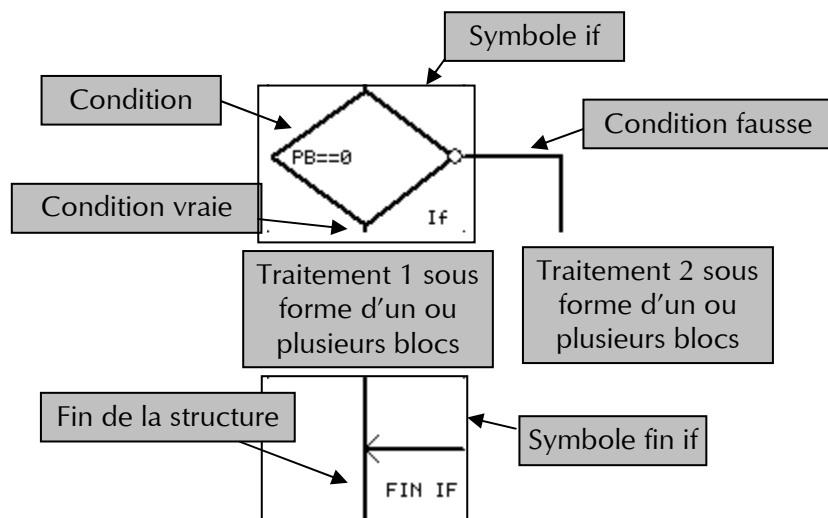


STRUCTURE ALTERNATIVE (IF)

Equivalent en C :

```
If (PB==0)
{Traitement 1}
else
{Traitement 2}
```

Le traitement 2 est facultatif.



Cette structure nécessite de placer 2 symboles : if et fin if + les traitements + les liaisons.

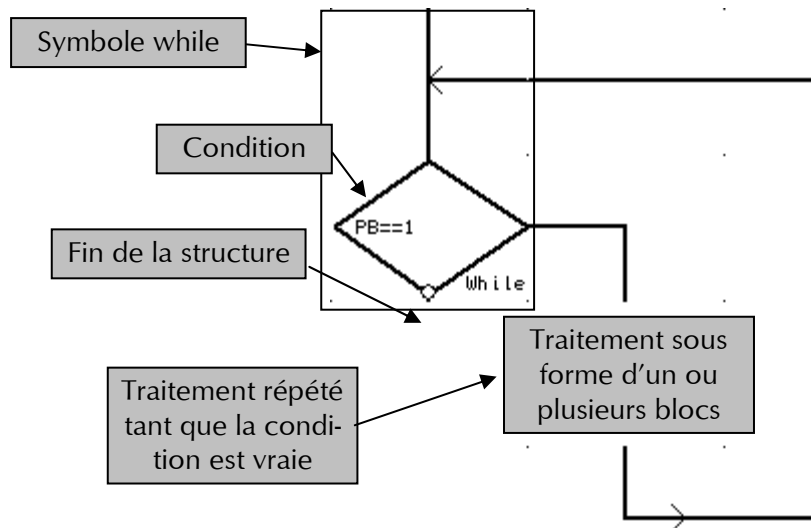
Remarque : Multiprog ne permet pas de dessiner l'équivalent d'un choix multiple (switch(...) case du langage C).

STRUCTURES ITERATIVES (BOUCLES)

Il existe plusieurs structures itératives.

Structure while

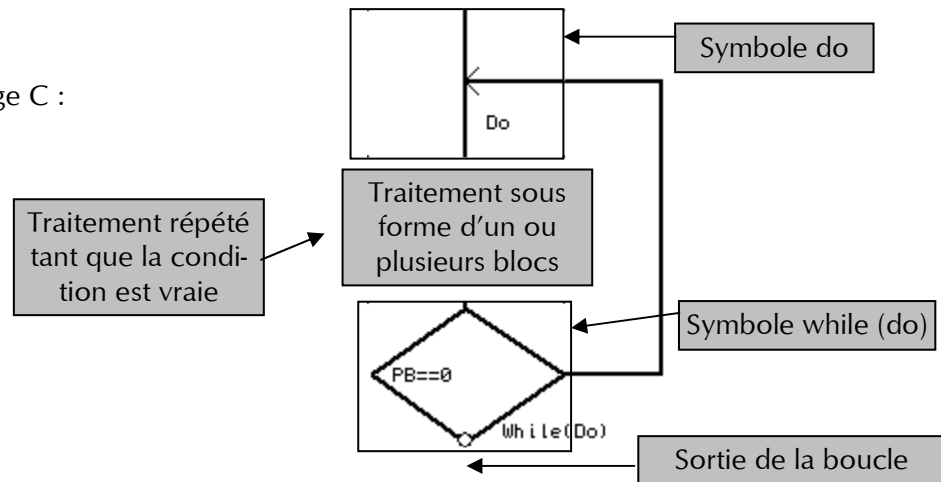
Equivalent C :
while (PB==1)
{traitement}



Cette structure nécessite de placer 1 seul symbole + le traitement + les liaisons

Structure do while

Equivalent en langage C :
do
{traitement}
while (BP==0) ;

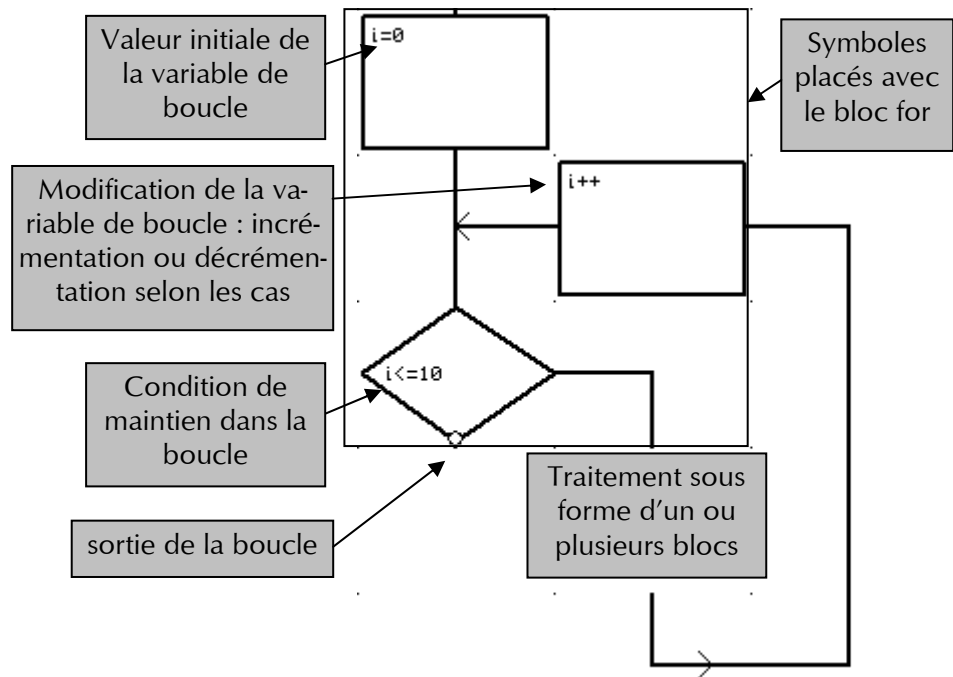


Cette structure nécessite de placer 2 symboles : do et while (do) + les traitements + les liaisons.

Structure for

Equivalent C :
`for(i=0 ; i<=10 ; i++)`
`{.....}`

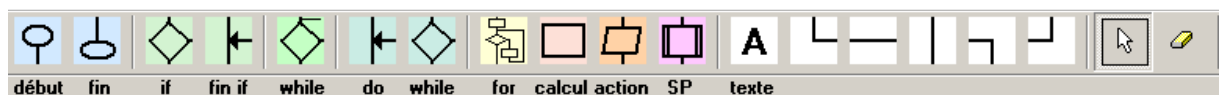
Ici, le traitement est réalisé 11 fois.



Cette structure nécessite de placer 1 seul bloc + le traitement + les liaisons

4.4 PLACEMENT ET EDITION DES SYMBOLES

Il suffit de sélectionner un symbole ou un élément de liaison dans la barre d'outils puis de la placer sur la feuille de travail en cliquant à l'endroit désiré.



Selon le type de symbole, une boîte de dialogue s'ouvre ou non lors du placement.

4.4.1 TRAITEMENT (CALCUL)

Après avoir cliqué sur la feuille de travail pour placer un symbole Traitement (Calcul), la boîte de dialogue ci-contre s'ouvre :

Il faut compléter la ligne en utilisant la syntaxe de C, sans ; en fin de ligne. Cette syntaxe est assez intuitive pour les calculs.

Pour les opérateurs, les noms des variables, des fonctions, des résultats de CAN, des tableaux de constantes, il suffit de les sélectionner dans des listes déroulantes.

Dans l'exemple ci-contre, seuls 23, 0 et 2 ont été entrés au clavier.

Une aide est disponible sur les opérateurs du langage C. En amenant le pointeur sur le bouton d'aide, la fenêtre suivante apparaît.

Opération arithmétique	symbole	syntaxe	Résultat
addition	+	a = 2 + 3	a = 5
soustraction	-	a = 5 - 2	a = 3
multiplication	*	a = 2 * 3	a = 6
division	/	a = 6 / 4	a = 1
Reste de la division	%	a = 6 % 4	a = 2

Opération logique	symbole	syntaxe	a=0011	b=0110	R=0010
et	&	R = a & b	a=0011	b=0110	R=0111
ou inclusif		R = a b	a=0011	b=0110	R=0101
ou exclusif	^	R = a ^ b	a=0011	b=0110	R=1100
complément	~	R = ~ a	a=0011		R=1100
décalage à droite	>>	R = a >> 2	a=1100		R=0011
décalage à gauche	<<	R = a << 2	a=0011		R=1100

Opération	symbole	syntaxe	Equivalent
Affectation simple	=	a = b	
opération puis affectation	+=	a+=b	a = a + b
	-=	a-=b	a = a - b
	=	a=b	a = a * b
	/=	a /= b	a = a / b
	%=	a %= b	a = a % b
	&=	a &= b	a = a & b
	=	a = b	a = a b
	^=	a ^= b	a = a ^ b
	>>=	a >>= b	a = a >> b
	<<=	a <<= b	a = a << b
incrémenter/décroître et affectation	++	a++	a = a + 1
	--	a--	a = a - 1

Une ligne peut utiliser la valeur renvoyée par une fonction.

Ex : MaVar2=MaFonc(MaVar1)*2

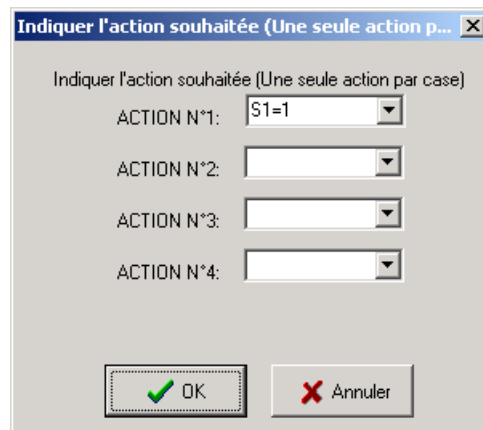
MaFonc() est une fonction qui effectue un calcul à partir de la valeur passée en argument, ici MaVar1. MaFonc() est une fonction qui est définie par l'utilisateur.

Multi-prog fournit quelques fonctions liées aux périphériques. Voir le § sur les fonctions et procédures.

4.4.2 AFFECTATION D'UNE VALEUR A UNE SORTIE (ACTION)

Après avoir cliqué sur la feuille de travail pour placer un symbole action, la boîte de dialogue ci-contre s'ouvre :

Chaque liste déroulante contient toutes les variables de sorties et toutes les affectations possibles.



Indiquer l'action souhaitée (Une seule action p...)

Indiquer l'action souhaitée (Une seule action par case)

ACTION N°1: S1=1

ACTION N°2:

ACTION N°3:

ACTION N°4:

OK Annuler

4.4.3 PROCEDURE (SP)

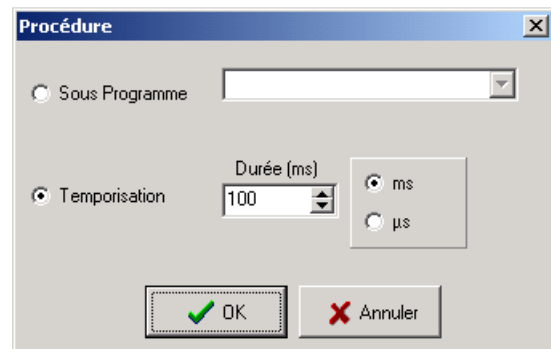
Le symbole procédure permet de réaliser une temporisation ou un appel de sous programme. L'appel de sous programme est vu plus loin.

Après avoir cliqué sur la feuille de travail pour placer un symbole procédure, la boîte de dialogue ci-contre s'ouvre :

Le choix de la durée est très simple. Elle peut être réglée jusqu'à 65535 ms (codage par un entier non signé sur 16 bits).

Cette temporisation est réalisée avec une fonction qui utilise une boucle for. La durée n'est pas d'une grande précision. Elle peut changer si on utilise la version complète du compilateur avec optimisation.

Pour un appel de sous programme, voir plus loin.



Procédure

☐ Sous Programme

☒ Temporisation

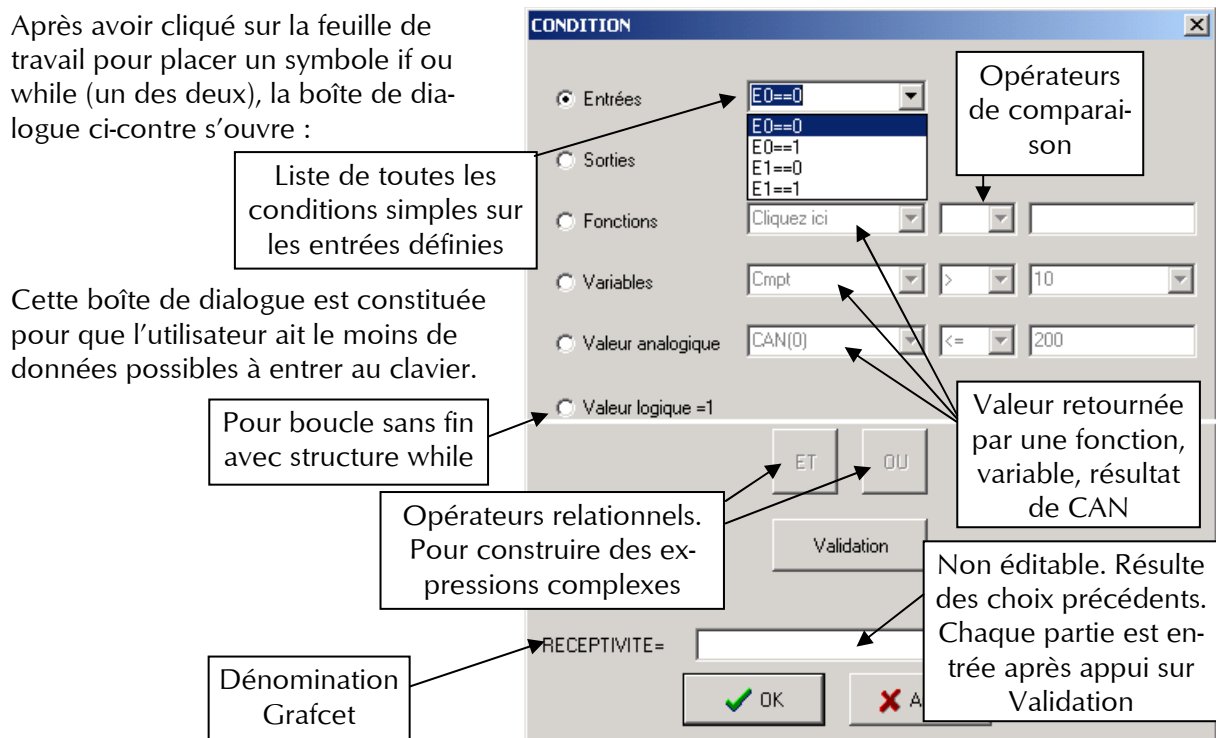
Durée (ms) 100

ms ☒
µs ☐

OK Annuler

4.4.4 STRUCTURE IF / DO ... WHILE / WHILE

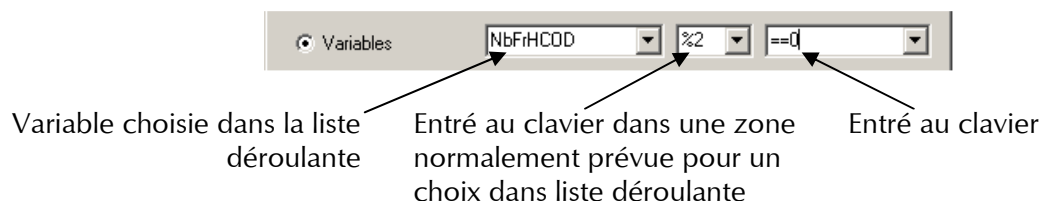
Après avoir cliqué sur la feuille de travail pour placer un symbole if ou while (un des deux), la boîte de dialogue ci-contre s'ouvre :



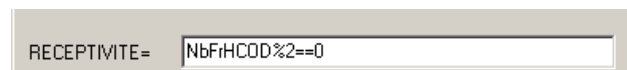
Avec cette boîte de dialogue, on peut avoir une réceptivité du type : E0==0.E1==1+S0==0
 Pour que cette expression soit facilement compréhensible, elle ne respecte pas la syntaxe du C.
 Le « et » est traduit par un point et le « ou » par un +. Comme il n'est pas possible de placer des parenthèses, les priorités des opérateurs en langage C s'appliquent.
 La syntaxe C aurait donné : E0 == 0 && E1 == 1 || S0 == 0.

Remarque : Cette boîte de dialogue est prévue pour rentrer des expressions relativement simples pour les conditions. Un utilisateur un peu expérimenté peut cependant entrer des expressions plus complexes.

Ex :



Après avoir cliqué sur Validation, on peut vérifier qu'on obtient bien l'expression souhaitée



4.4.5 BOUCLE FOR

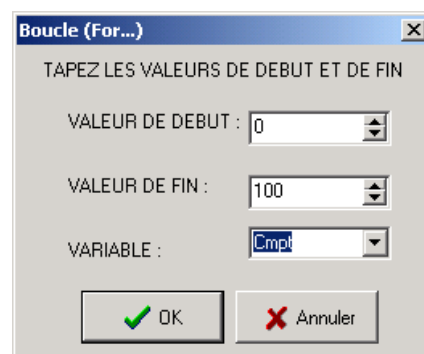
Après avoir sélectionné for dans la barre d'outils et cliqué sur la feuille de travail pour placer le symbole, la boîte de dialogue ci-contre s'ouvre :

Selon que la valeur de début est inférieure ou non à la valeur de fin, la variable est automatiquement incrémentée ou décrémentée.

Equivalent C :

```
for(Cmpt=0 ;Cmpt<=100 ;Cmpt=Cmpt+1) ...
```

Cmpt est une variable qui a été déclarée.



4.5 EDITION DES SYMBOLES APRES PLACEMENT

Pour éditer, il suffit de se placer sur un symbole, avec l'outil sélection (flèche) actif.

En cliquant sur le bouton droit, on accède à un menu contextuel. La rubrique propriétés permet d'ouvrir la même boîte de dialogue que lors du placement.

4.6 PLACEMENT DES LIAISONS

Il suffit d'utiliser les symboles « trait » de la barre d'outils.

On peut aussi utiliser un tracé automatique. Il faut choisir la fonction lien, avec n'importe quel outil actif. Il suffit d'ouvrir le menu contextuel puis de cliquer sur lien. La démarche à suivre est rappelée en haut de l'écran. Il faut cliquer avec le bouton gauche sur le point de départ puis à nouveau sur le point d'arrivée.

4.7 MANIPULATION DE SYMBOLES DEJA PLACES

Il est possible de :

- Insérer un symbole entre des symboles déjà placés (à partir de la version 2)
- Insérer une ligne ou une colonne entre les symboles
- Déplacer des symboles à la souris
- Couper ou copier / coller des symboles dans Multiprog
- Copier tout ou partie de l'algorithme dans le presse papier vers une autre application

Pour effectuer certaines de ces manipulations, il faut d'abord sélectionner un ou plusieurs blocs.

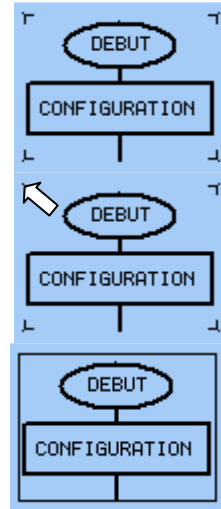
En cas d'erreur de placement ou déplacement, il est possible d'annuler les dernières opérations avec la commande Edition / Annuler.

4.7.1 SELECTION DE SYMBOLES

La sélection s'effectue avec l'outil sélection (flèche dans la barre d'outils).

Il faut :

- amener le pointeur pour que le carré de sélection encadre le symbole ou le premier symbole du groupe à déplacer
- Avec les versions 1.12 et ultérieures, il suffit de placer le pointeur sur un bloc.
- Cliquer en maintenant et tirer. Le ou les symboles sélectionnés deviennent encadrés. Il faut ensuite relâcher le bouton de la souris lorsque la sélection est terminée.



4.7.2 DEPLACEMENT DE SYMBOLES / COPIER / COLLER

Une fois le symbole sélectionné, on peut réaliser les opérations suivantes :

- Déplacement comme dans toutes les applications Windows : clic maintenu bouton gauche souris + déplacement souris
- Suppression : appui sur la touche Suppr
- Copier dans le presse papier vers une application externe : menu Edition / Copier vers... ou menu contextuel Copier vers...
- Copier pour Multi-prog : Ctrl + C ou menu contextuel, ...

4.8 INSERTION DE LIGNES OU COLONNE ENTRE LES SYMBOLES DEJA PLACES

Il suffit de placer le pointeur à l'endroit où on souhaite insérer une ligne ou une colonne, d'ouvrir le menu contextuel et de cliquer sur Insérer ligne ou insérer colonne.

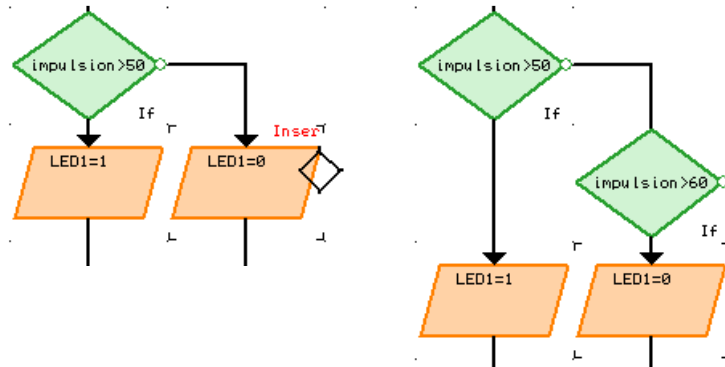
Tous les symboles sont déplacés, y compris les liaisons, pour faire apparaître une nouvelle ligne ou colonne



4.9 INSERTION DE SYMBOLES

Cette possibilité n'existe qu'à partir de la version 2.

Il faut choisir le symbole à insérer puis placer le pointeur au dessus à droite du symbole au dessus duquel le nouveau symbole doit être inséré. L'indication Inser apparaît. Il faut cliquer sur cette indication et l'insertion est automatique.



5 PRODUCTION DU FICHIER EXECUTABLE / PROGRAMMATION

5.1 VUE D'ENSEMBLE

La production du fichier exécutable se fait en 2 temps :

- Traduction automatique en langage C (et algorithmne)
- Compilation (en utilisant le compilateur CC5X ou SDCC de façon transparente à l'utilisateur)

Selon la carte cible choisie, la programmation peut suivre ou non la compilation. Selon une option choisie lors de la traduction automatique, l'exécution du programme avec animation sur l'algorithme peut être lancée ou non en fin de programmation.

Pour l'instant Multi-prog offre le choix entre 3 logiciels pour la programmation. Le choix s'effectue automatiquement selon la carte cible sélectionnée.

Il suffit de choisir une carte dans le menu Projet.

Carte	Logiciel lancé automatiquement pour la programmation	Mise au point possible (exécution avec visualisation sur algorithme)
Castor LPT	Castor_prog	Oui
Castor Wifi	Castor_prog	Oui
Castor USB	Castor_prog	Oui
Carte compatible IC-Prog	IC-Prog	Oui
Carte EasyPIC de mikroElektronika	PICFlash programmer	? (à vérifier)
Autre carte (sans transfert)	Pas de programmation depuis Multi-prog	Non

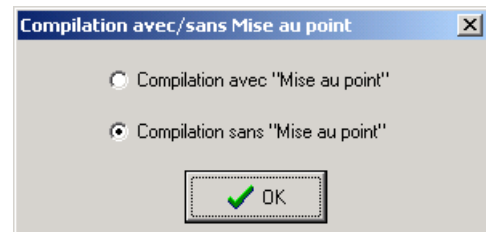
Il est préférable de choisir Castor-prog qui n'a pas besoin de configuration particulière et qui fonctionne avec une liaison USB ou par Wifi.

Avant de lancer la compilation, il est nécessaire de vérifier que les logiciels de programmation sont bien configurés. Voir § 5.4).

5.2 TRADUCTION AUTOMATIQUE EN LANGAGE C ET COMPILATION

Une fois l'algorithme terminé, il suffit de cliquer sur l'icône flèche  pour lancer la traduction automatique en langage C + la compilation.

Une boîte de dialogue demande si une mise au point in situ est prévue ou non. Lorsque aucune mise au point n'est prévue, la carte cible fonctionne seule après programmation du PIC cible.



Lorsque la mise au point est prévue, la traduction rajoute de façon cachée à l'utilisateur des instructions pour le dialogue avec le PC. La mise au point n'est possible qu'avec une carte cible qui possède la programmation in situ et qui peut se connecter au module d'interface.

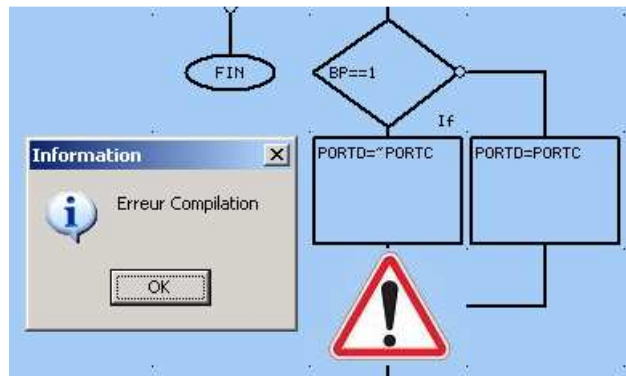
Selon la carte cible choisie au niveau du projet, l'option 'Compilation avec « Mise au point »' est accessible ou non.

Après la traduction automatique, le fichier créé (ou une partie dans le cas avec mise au point) est observable dans la fenêtre avec l'onglet Programme C.

De même, on peut aussi voir l'algorithme dans la fenêtre avec l'onglet Algorithme.

Traduction avec erreurs dans l'algorithme

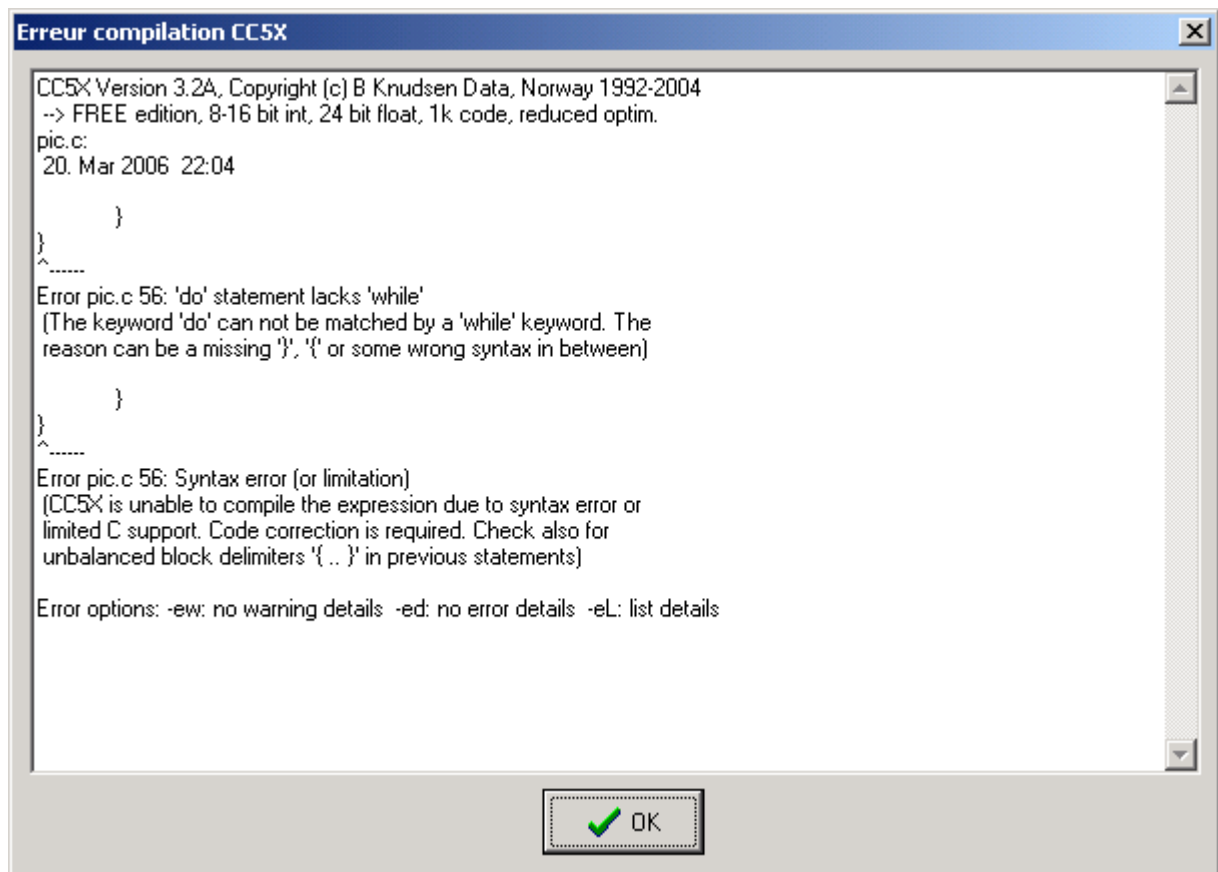
Certaines erreurs sont signalées. Il faut les corriger avant de lancer une nouvelle compilation



D'autres erreurs ne génèrent aucun message lors de la traduction. Exemple : emploi d'un symbole do à la place d'un symbole fin if. L'erreur sera générée à la compilation.

La compilation fait appel, de façon transparente à l'utilisateur, au compilateur CC5X ou SDCC. Si tout se passe bien, le fichier .hex pour la programmation est généré et la programmation commence ensuite.

En cas d'erreur, une fenêtre s'ouvre automatiquement.



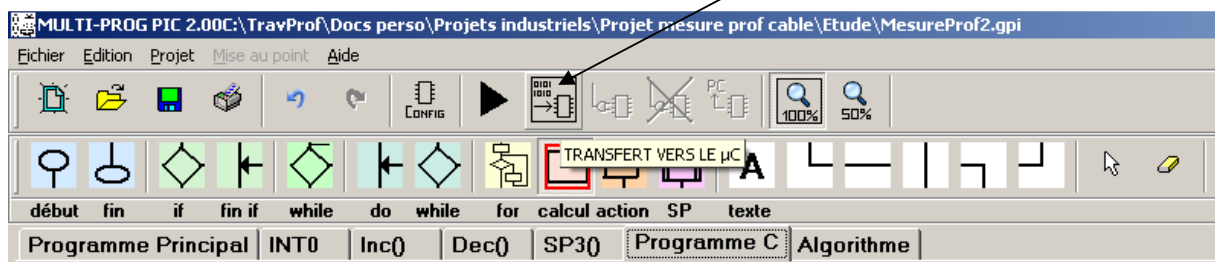
Les messages d'erreurs concernent le fichier .c non observable. On trouve l'erreur en s'aidant du fichier en langage C visible depuis Multi-prog. Il faut lancer une compilation sans mise au point et compter les lignes. Ici l'erreur est signalée sur la ligne 56.

On peut s'aider d'un éditeur de texte qui numérote les lignes. Il faut ouvrir le fichier pic.c placé dans le dossier d'installation de Multi-prog (ou le fichier NomProjet.c placé dans le dossier du projet avec certaines versions de Multi-prog).

5.3 PROGRAMMATION DU PIC CIBLE (TRANSFERT)

La programmation peut être in situ, avec les cartes qui le permettent, ou être réalisée avec un programmeur. Dans ce dernier cas, le PIC une fois programmé doit être placé sur la carte cible.

La programmation du PIC cible s'effectue avec le bouton Transfert.



Pour la programmation in situ, avec l'utilisation éventuelle d'une carte d'interfaçage, Multi-prog utilise un logiciel externe parmi 3 : PICFLASH Programmer (pour les cartes de mikroElektronika), IC-prog ou Castor-prog. Ce dernier est spécialement conçu pour travailler avec les cartes Castor mais il peut aussi s'utiliser avec une autre carte munie d'un connecteur compatible Microchip ICD + une carte d'interfaçage spécifique (µCastor pour la liaison USB).

Le choix de l'un ou l'autre de ces logiciels dépend de la carte cible retenue pour le projet (voir § 3).

IC-Prog a été choisi à l'origine du développement de Multi-prog. Castor-prog développé spécifiquement pour Multi-prog est plus adapté. Il est plus rapide, surtout sur la version USB.

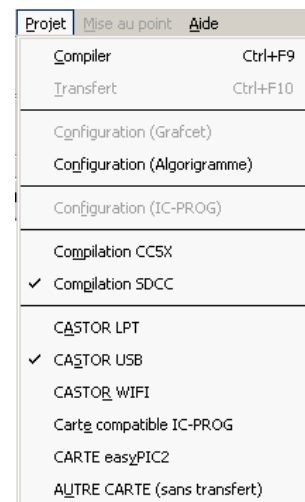
5.3.1 PROGRAMMATION AVEC UNE CARTE CASTOR OU UNE CARTE QUELCONQUE AVEC UN CONNECTEUR COMPATIBLE ICD MICROCHIP

La carte cible est choisie à l'aide du menu Projet

Un des items CASTOR est à cocher si on utilise une carte spécifique CASTOR ou une carte quelconque avec un connecteur compatible Microchip ICD. Dans ce cas, il faut utiliser une petite carte d'interface.

Avec Castor LPT, on utilise le port // (imprimante) pour la programmation. Avec la Castor USB, c'est la liaison USB qui est utilisée.

La carte easyPIC2 se programme par la liaison USB.



Les fichiers correspondant au programme Castor-prog sont automatiquement placés dans le dossier d'installation de Multi-prog.

Le logiciel Castor-prog (version 2.6 de mars 2007) permet de programmer les PIC suivants : 12F629, 12F630, 12F675, 12F676, 16F84, 16F627A, 16F628A, 16F648A, 16F818, 16F819, 16F871, 16F873, 16F873A, 16F876, 16F876A, 16F877, 16F877A, PIC18F2550 et 18F4550 (non supportés par Multi-prog).

Castor-prog utilise un fichier devices.ini (devices_lpt.ini pour l'utilisation du port // ; devices_usb.ini pour l'utilisation de la liaison USB) pour les bits de configuration du PIC cible. Pour modifier la configuration souhaitée, il faut éditer le fichier et le modifier.

Voici par exemple la partie concernant le PIC16F876 :

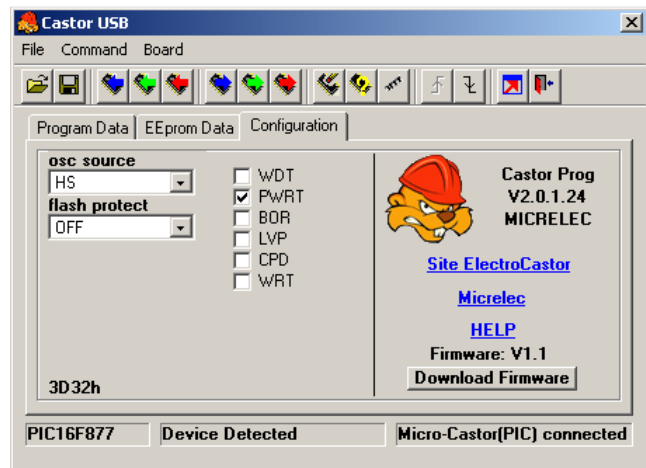
```
[09E]
device=PIC16F876
algo=2
rom=8192
eeprom=256
bits=U,U,WRT,/CPD,*LVP,BOR,*/PWRT,WDT
num bits=11,10,9,8,7,6,3,2
bits osc=RC,*HS,XT,LP
num bits osc=1,0
bits flash write=*OFF,1F00h-1FFFh,1000h-1FFFh,0000h-1FFFh
num bits flash write=13,12,5,4
```

Pour modifier une option, il faut enlever ou mettre une *.

L'utilisation de Castor-Prog est transparente à l'utilisateur. Le lancement est automatique. La programmation est très rapide.

Lors de la programmation, une barre indique la progression. En fin de programmation, une vérification est automatiquement lancée. On peut aussi utiliser Castor-prog indépendamment de Multi-prog. Le logiciel reconnaît automatiquement le PIC cible.

Castor-prog dispose des commandes habituelles : charger un fichier, programmer le composant, lire, etc.



5.3.2 PROGRAMMATION AVEC UNE CARTE COMPATIBLE IC-PROG

Pour utiliser IC-prog avec une carte quelconque, il faut une carte d'interface pour la liaison //. Cette carte d'interface est la même que celle pour Castor-prog. Depuis la mise au point de Castor-prog, cette solution est obsolète. Voir §11.

Avant la 1^{ère} utilisation d'IC-prog, il faut le configurer pour Windows. Voir plus loin.

Remarque sur la programmation par la liaison //

Pour que la programmation fonctionne correctement, il faut qu'une imprimante soit déclarée. Peu importe l'imprimante.

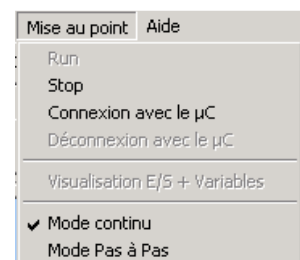
6 OBSERVATION DU DEROULEMENT DU PROGRAMME

L'observation du déroulement du programme sur l'algorithme ne peut avoir lieu que lorsque la programmation est effectuée.

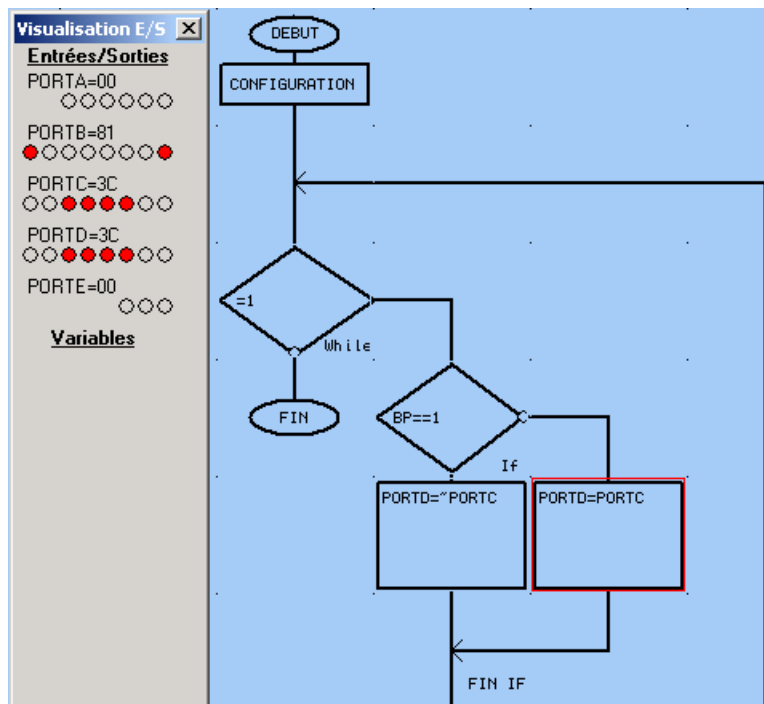
Il faut d'abord choisir le type de déroulement : continu ou pas à pas avec la commande Mise au point / Mode continu ou Mode pas à pas.

Il faut ensuite cliquer sur Connexion avec le µC.

Si le mode continu est sélectionné, le programme est lancé et on peut voir un cadre rouge se déplacer sur les différents symboles.



En cliquant sur Visualisation E/S + Variables, une nouvelle fenêtre s'ouvre :



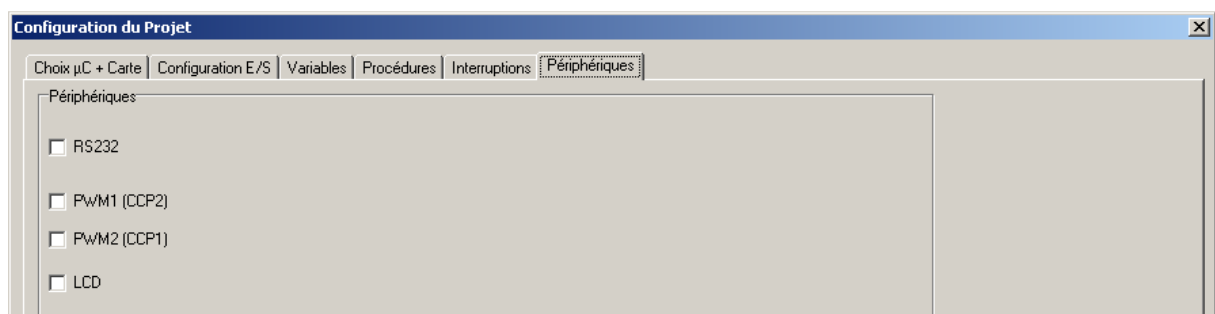
En mode pas à pas, il faut cliquer sur le bouton pas de la barre d'outil pour faire avancer le programme d'un symbole sur l'algorithme.

Pour arrêter le mode mise au point, il faut fermer la fenêtre de visualisation des E/S et des variables et cliquer sur Mise au point / Déconnexion avec le µC.

7 GESTION DE PERIPHERIQUES & INTERRUPTIONS

7.1 VUE D'ENSEMBLE

La déclaration de périphériques s'effectue en ouvrant la boîte de configuration du projet avec Projet / Configuration (Algorithme) / Onglet Périphériques.



La déclaration de l'utilisation du CAN intégré, s'effectue avec l'onglet Configuration E/S de la boîte de configuration.



Pour les interruptions, il faut utiliser l'onglet Interruptions.

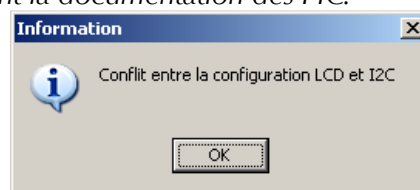
La déclaration de périphériques permet de disposer de fonctions et procédures permettant de gérer les périphériques :

- intégrés au μ C : liaison RS232, EEPROM, Timer
- ou externes : LCD, clavier

Les fonctions et procédures sont décrites dans le § 8).

Les instructions pour la configuration des périphériques intégrés sont automatiquement générées, dans la partie CONFIGURATION de l'algorithme (après le symbole DEBUT), selon les options choisies.

Remarque 1: il est impossible d'utiliser tous les périphériques dans une même application. L'afficheur nécessite 7 bits d'un port. La liaison série utilise nécessairement 2 broches spécialisées d'un port qui ne peuvent plus alors être utilisées pour l'afficheur. La programmation in situ utilise elle aussi 2 broches d'un port, etc. Il faut consulter attentivement la documentation des PIC. Si un conflit existe entre les différents périphériques, une boîte de dialogue le signale.



Remarque 2 : lors d'une interruption, le drapeau correspondant est automatiquement testé puis remis à 0 en fin de traitement de l'interruption.

7.2 CAN

Pour utiliser le CAN intégré, il suffit juste de cocher la case CAN à côté des entrées du port A et du port E, comme mentionné dans le § 7.1).

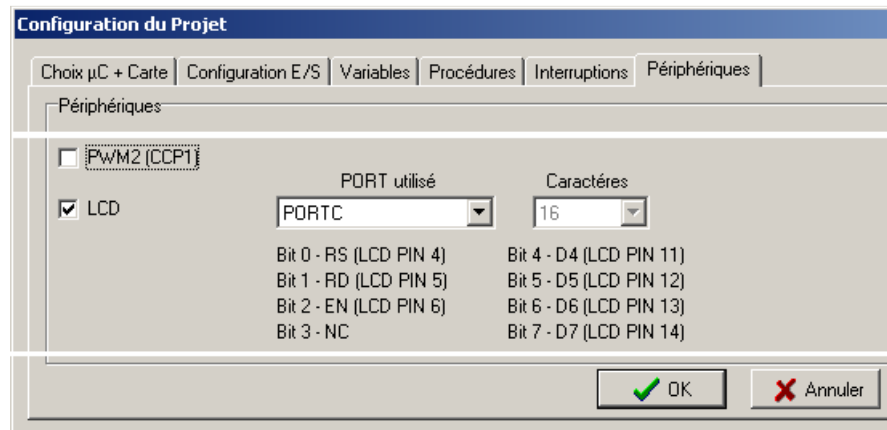
1 fonction permet de gérer la CAN. Voir §8.2), après la présentation générale des fonctions et procédures.

Attention : toutes les combinaisons permises par le PIC pour des mélanges entrées analogiques / logiques ne sont pas supportées par Multi-prog.

7.3 AFFICHEUR LCD

Multi-prog permet de commander un afficheur LCD, avec un bus de donnée de 4 bits et les commandes sur le même port du PIC.

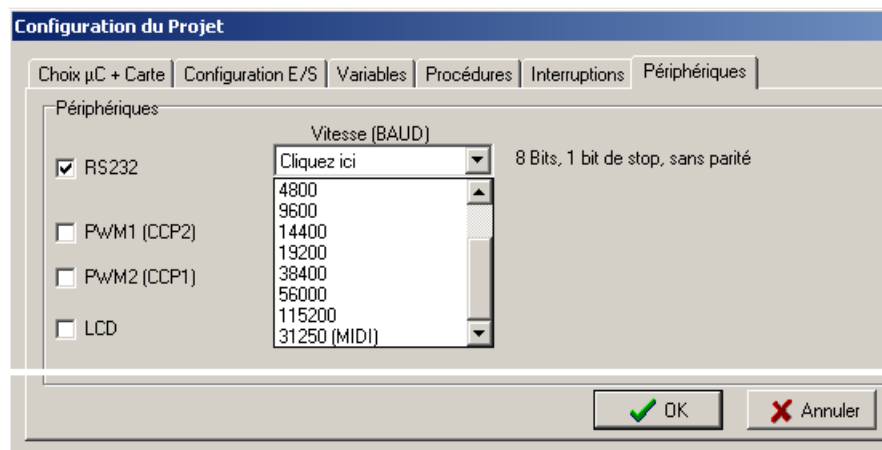
La boîte de dialogue Interruptions et périphériques permet de déclarer un LCD, de choisir le port de connexion. Les bits sont imposés. Ils sont à prendre en compte lors de la conception de la carte cible.



4 procédures permettent de gérer l'afficheur LCD. Voir §8.2), après la présentation générale des fonctions et procédures.

7.4 LIAISON RS232

La déclaration d'une liaison RS232 dans la boîte de dialogue Interruptions et périphériques ouvre une zone pour choisir la vitesse de transmission. Les broches du PIC à utiliser sont données dans la doc du PIC.



Un CI spécialisé dans la transformation des niveaux (ex : MAX232) est indispensable entre le PIC et le connecteur DB9.

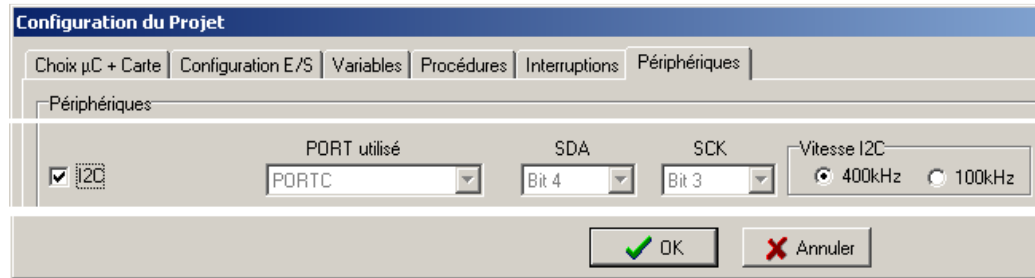
La déclaration ouvre aussi une fenêtre pour un gestionnaire d'interruption avec l'onglet RX232. Voir plus loin pour les interruptions.

Une procédure et une fonction permettent de gérer la liaison RS232. Voir §8.3), après la présentation générale des fonctions et procédures.

7.5 LIAISON I2C

La liaison I2C n'est utilisable qu'en mode maître (le plus répandu).

La vitesse de transfert peut être choisie entre 400kbits/s et 100kbits/s.



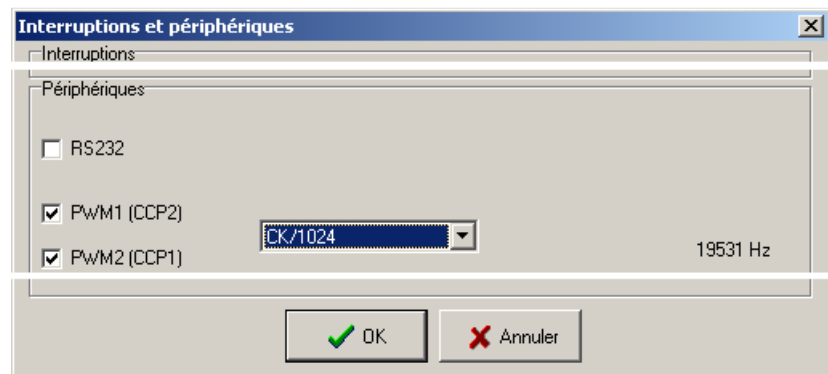
Ces deux vitesses sont les standards I2C.

Plusieurs procédures et une fonction permettent de gérer la liaison I2C. Voir plus loin, après la présentation générale des fonctions et procédures.

7.6 GENERATION D'UN SIGNAL RECTANGULAIRE A RAPPORT CYCLIQUE VARIABLE

Il est possible de générer 2 signaux.
Le Timer 2 est utilisé pour générer ces signaux.

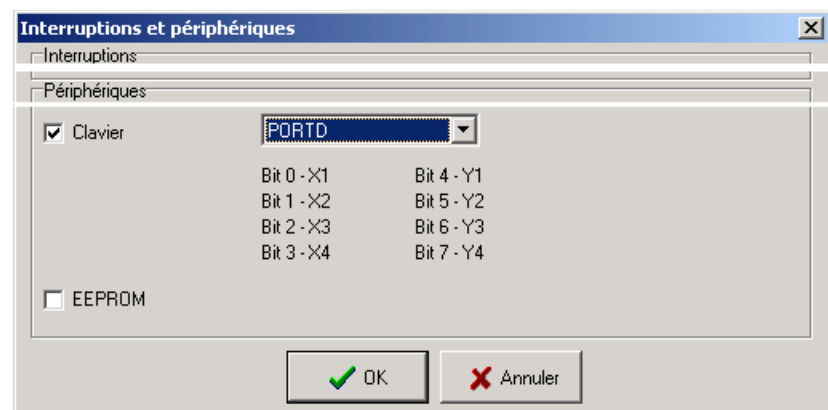
Quelques fréquences sont disponibles en choisissant une division de la fréquence d'horloge du timer 2 avec une liste déroulante.



Une fonction permet de choisir le rapport cyclique.

7.7 GESTION DU CLAVIER

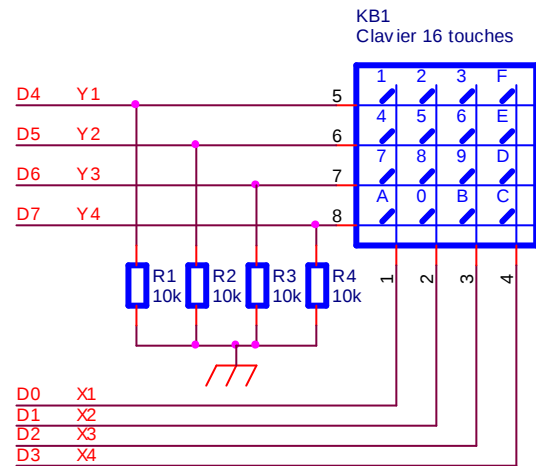
Le clavier à utiliser est un clavier matricé 4x4 (16 touches).



Multiprog est prévu pour l'utilisation d'un clavier standard.

Il faut placer des résistances (la valeur importe peu, on peut retenir 10k Ω par exemple) entre les lignes Y1, Y2, Y3, Y4 et la masse.

La déclaration d'un clavier nomme automatiquement les entrées lignes Y1, Y2, Y3, Y4 et sorties colonnes en X1, X2, X3, X4. Ces E/S apparaissent dans les listes déroulantes de certaines boîtes de dialogues présentées au §4.4).



Une fonction est disponible pour la gestion du clavier. Voir §8.4), après la présentation générale des fonctions et procédures.

Multiprog donne le code de la touche appuyée. Si plusieurs touches sont appuyées simultanément, c'est le numéro de la touche appuyée sur la colonne d'indice le plus élevé qui est renvoyé.

7.8 EEPROM

L'EEPROM est une zone mémoire qui peut être écrite et lue un très grand nombre de fois. Les informations sont conservées même après la coupure de l'alimentation.

La déclaration de l'utilisation de l'EEPROM permet d'utiliser une procédure pour l'écriture et une fonction pour la lecture.

Voir §8.5), après la présentation générale des fonctions et procédures.

7.9 INTERRUPTION PAR DEBORDEMENT DU TIMER 0

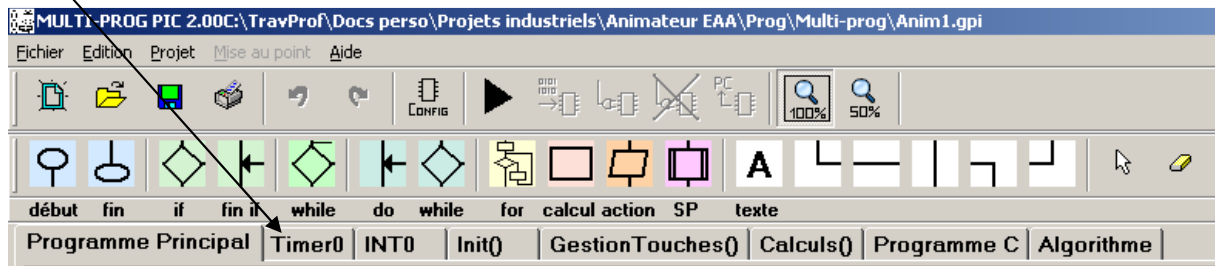
Le timer 0 est constitué d'un registre 8 bits qui est incrémenté à chaque front actif d'un signal obtenu à partir de l'horloge système dont la fréquence est divisée par un certain nombre. *Certains PICs offrent d'autres possibilités qui ne sont pas exploitées ici.*

Lorsque que le registre du timer atteint 255, il repasse à 0 au front suivant. On dit qu'il y a débordement du timer.

Il est possible de choisir 2 types d'interruption par débordement :

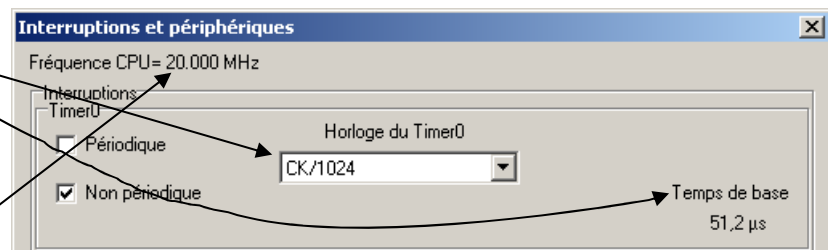
- **Interruption non périodique** : l'utilisateur doit programmer le timer dans un des algorithmes pour générer une interruption unique après un certain temps. Multi-prog fournit une procédure pour cette programmation.
- **Interruption périodique** : l'utilisateur choisit la fréquence des interruptions avec des listes déroulantes et n'a rien à programmer dans les algorithmes

Dans les 2 cas, une fenêtre pour l'algorithme de traitement de l'interruption avec l'onglet Timer0 est automatiquement créée. Voir §9 pour les interruptions.



7.9.1 INTERRUPTION NON PERIODIQUE

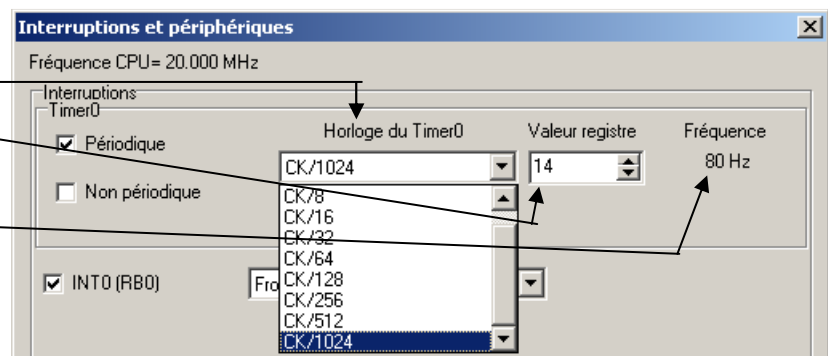
Il suffit de choisir la division en fréquence de l'horloge du timer et la période de l'horloge du timer est automatiquement calculée en fonction de la fréquence de l'oscillateur fixée dans la boîte de dialogue principale de la configuration du projet.



La procédure fournie permet de programmer la durée avant interruption en multiple de ce temps de base. La durée max est de 255 fois le temps de base. Voir § 8 sur les fonctions fournies.

7.9.2 INTERRUPTION PERIODIQUE

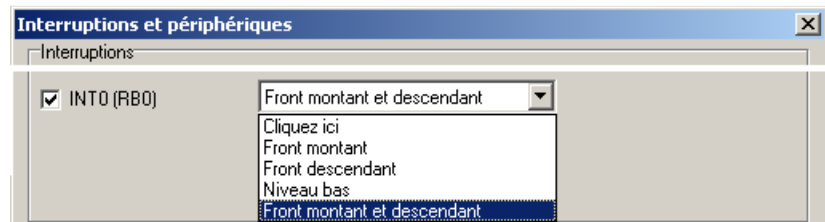
Il suffit de choisir par essais successifs la division en fréquence pour l'horloge du timer et le contenu d'un registre pour obtenir la fréquence souhaitée des interruptions périodiques



Remarque : la valeur du registre est celle qui est utilisée, de façon transparente à l'utilisateur, pour reprogrammer le registre du timer à chaque interruption pour obtenir une nouvelle interruption après la durée désirée ($1/\text{Fréquence}$).

7.10 INTERRUPTION SUR FRONT DE L'ENTREE RB0/INT

Quand l'interruption sur l'entrée RBO est choisie, une liste déroulante permet de déterminer le front ou l'état qui déclenche l'interruption.



Une fenêtre pour l'algorithme de traitement de l'interruption avec l'onglet INT0 est automatiquement créée. Voir §9 pour les interruptions.

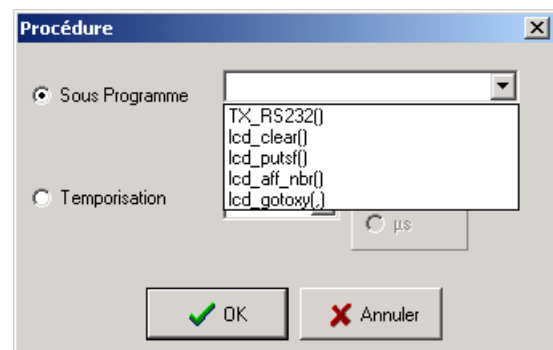
Remarque : Pour la validation des interruptions sur fronts montant ou descendant, quelques bits de registres de commande sont positionnés une fois pour toutes lors de l'initialisation en début de programme. Pour la validation des interruptions sur fronts montant et (puis) descendant, en plus de l'initialisation, un bit est modifié à chaque interruption.

8 FONCTIONS ET PROCEDURES

8.1 INTRODUCTION

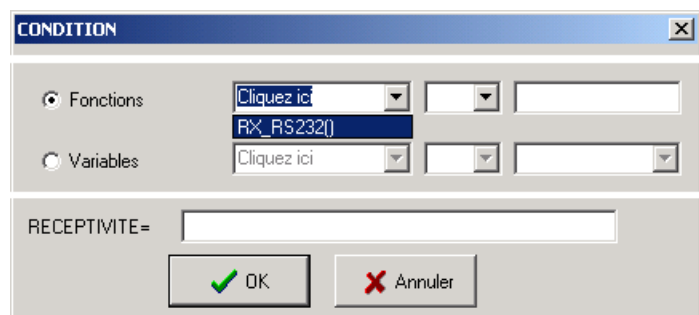
Multi-prog fournit certaines procédures et fonctions (au sens du Pascal) en relations avec les périphériques utilisés. Si un afficheur LCD est déclaré, 4 procédures sont fournies. Si une liaison RS232 est en service, une procédure + une fonction sont disponibles, etc.

Les procédures apparaissent dans la boîte de dialogue Procédure qui s'ouvre automatiquement lorsqu'on place un symbole SP sur la feuille de travail.



Les fonctions apparaissent dans deux boîtes de dialogue :

- la boîte de dialogue Condition qui s'ouvre automatiquement lorsqu'on place un symbole if ou while



- La boîte de dialogue Calcul qui s'ouvre automatiquement lorsqu'on place un symbole Calcul.

L'utilisateur peut aussi écrire ses propres procédures et fonctions. Elles apparaissent automatiquement dans les listes déroulantes des boîtes de dialogue précédentes.

Les fonctions ou procédures peuvent aussi être utilisées dans les traitements.

Les procédures ou fonctions peuvent nécessiter des paramètres. Les règles d'écritures sont celles du C.

Ex : lcd_gotoxy(0,1) 0 et 1 sont les 2 paramètres de la procédure.

Pour entrer une valeur, il suffit de sélectionner une procédure ou fonction dans une boîte de dialogue, puis de placer le point d'insertion au bon endroit et entrer le texte.

Une fonction (au sens du Pascal) renvoie une valeur qui se substitue au nom de la fonction dans l'expression.

Exemple : Valeur = CAN(0)*3. CAN est une fonction qui nécessite un paramètre (ici 0) et qui renvoie une valeur (ici le résultat d'une conversion A/N).

Les règles d'écritures des procédures et fonctions sont celles du C.

8.2 FONCTION POUR CAN

1 seule fonction CAN() est disponible. Elle renvoie un entier positif codé sur 16 bits. La valeur max renvoyée est 1023 (CAN 10 bits).

Elle nécessite un paramètre : le numéro de la voie utilisée de 0 à 7.

Syntaxe : Result_CAN = CAN(1)

Cette fonction aiguille l'entrée vers le CAN, lance une CAN, attend la fin de la conversion et renvoie le résultat. La durée totale de la conversion dépend de la fréquence de l'horloge du μ C. Elle est de l'ordre de quelques dizaines de μ s.

Remarque : Result_CAN doit être une variable codée sur 16 bits de type unsigned short ou signed short. Si on utilise une variable codée sur 8 bits, ce sont les 8 bits de poids faibles qui sont transférés. Pour obtenir les 8 bits de poids forts, il faut écrire : Result_CAN = CAN(1)>>2

8.3 PROCEDURES POUR AFFICHEUR LCD

4 procédures (au sens du Pascal = fonctions au sens du C) **sont fournies pour la gestion de l'afficheur** :

Nom procédure	Description de la procédure	Syntaxe
lcd_clear()	Effacement de tout l'afficheur	
lcd_gotoxy(,)	Positionnement du curseur	lcd_gotoxy(NumCol,NumLigne) La numérotation des colonnes et des lignes part de 0. Pour afficher sur la colonne de début de la 2 ^{ème} ligne, il faut écrire : lcd_gotoxy(0,1)
lcd_putsf()	Affichage de la chaîne transmise en argument.	lcd_putsf(« bonjour »)
lcd_puts()	Affichage du caractère ASCII dont le code est transmis en argument.	lcd_puts(0x30)
lcd_aff_nbr()	Affiche en chiffres la valeur en base 10 de la variable (entière signée codée sur 16 bits max) transmise en argument	lcd_aff_nbr(NomVariable) ex : lcd_aff_nbr(Cmpt)

8.4 LIAISON RS232

3 procédures (au sens du Pascal = fonction au sens du C) **et 3 fonctions** **sont fournies pour la gestion de la liaison série** :

Nom	Type	Description	Syntaxe
TX_RS232()	Procédure	Transmet un octet qui correspond à la valeur transmise en paramètre. S'il s'agit d'une constante, la règle d'écriture est celle du C + extension du compilateur CC5X (notation en binaire).	TX_RS232(Valeur) Ex : TX_RS232(0x33) en hexadécimal TX_RS232(Var)
TX_string_RS232()	Procédure	Transmet les codes qui correspondent à la chaîne de caractères transmise en paramètre. Le texte doit être placé entre guillemets doubles.	TX_string_RS232(«Bonjour»)
val_RS232()	Fonction	A compléter	
RX_RS232()	Fonction	Retourne la valeur (codée sur un octet) reçue par la liaison série	Var = RX_RS232()

		asynchrone RS232. La réception d'une donnée provoque une interruption (algorithme RX RS232) L'utilisation de cette fonction réalise un acquittement de l'interruption.	
RX_string_RS232()	Procédure	A compléter	
cmp_str_RS232()	Fonction	A compléter	

8.5 LIAISON I2C

3 procédures et une fonction permettent de contrôler le fonctionnement d'une liaison I2C en mode maître :

procédures : i2c_start(), i2c_stop() et i2c_write()

fonction : i2c_read()

Pour l'utilisation, voir la documentation du bus I2C.

8.6 GENERATION D'UN SIGNAL RECTANGULAIRE A RAPPORT CYCLIQUE VARIABLE

La procédure PWM1() ou PWM2() permet de fixer le rapport cyclique de la sortie.

Exemple : Pour fixer le rapport cyclique à 50%, il suffit d'écrire :

PWM1(50)

8.7 FONCTION POUR CLAVIER

La fonction Keyb() est fournie pour la gestion du clavier. Elle ne nécessite pas de paramètre. Elle renvoie une valeur codée sur 8 bits qui correspond au code de la touche appuyée.

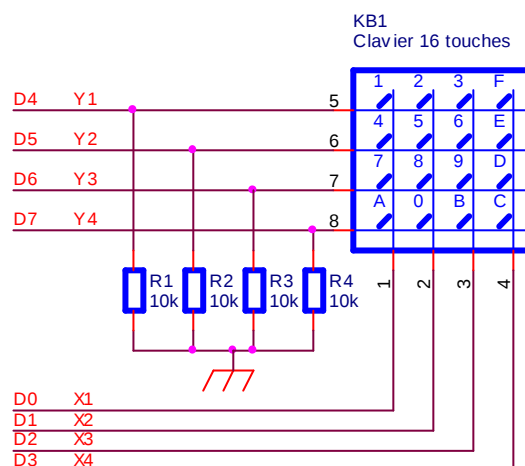
La scrutation du clavier a lieu une fois. Elle très rapide (quelques dizaines de μ S max). Les rebonds ne sont pas gérés.

Multiprog donne le code de la touche appuyée. Si aucune touche n'est appuyée, la valeur retournée est 16.

Si plusieurs touches sont appuyées simultanément, la valeur dépend des touches. Dans certains cas, c'est un numéro égal à 16 ou supérieur ; dans d'autres cas, c'est le code de la touche de la colonne de plus haut indice.

Pour un clavier standard, le code renvoyé est celui de la touche indiqué ci-contre.

Exemple d'utilisation : **CodeTouche=Keyb()**



Le tableau suivant donne le code renvoyé selon la touche enfoncée.

Code renvoyé	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Touche appuyée	Y4 X2	Y1 X1	Y1 X2	Y1 X3	Y2 X1	Y2 X2	Y2 X3	Y3 X1	Y3 X2	Y3 X3	Y4 X1	Y4 X3	Y4 X4	Y3 X4	Y2 X4	Y1 X4

Remarque : la gestion du clavier fait appel à un balayage des colonnes Y. Un état 1 est appliqué successivement sur les différentes colonnes. La lecture de l'état des lignes permet de déterminer la touche enfoncée.

8.8 EEPROM

1 procédure (au sens du Pascal = fonction au sens du C) **et une fonction** sont fournies pour la **gestion de la liaison série** :

Nom	Type	Description	Syntaxe
write_eeprom()	Procédure	Ecrit un octet transmis en paramètre à l'adresse transmise en paramètre	write_eeprom(adresse,valeur) ex : write_eeprom(10,Var)
read_eeprom(adresse)	Fonction	Retourne la valeur (codée sur un octet) lue à l'adresse donnée en paramètre	read_eeprom(adresse) ex : read_eeprom(10)

8.9 PROCEDURE POUR INTERRUPTION NON PERIODIQUE AVEC LE TIMER 0

La procédure Time_timer0() permet de fixer la durée avant une nouvelle interruption. La syntaxe est Time_timer0(Nb_Temps_de_Base). Le temps de base est indiqué dans la boîte de dialogue Interruptions et périphériques. Nb_Temps_de_Base est une valeur entière codée sur 8 bits.

Exemple : on souhaite une interruption provoquée par le timer0 dans 8ms avec un temps de base de 51,2µs.

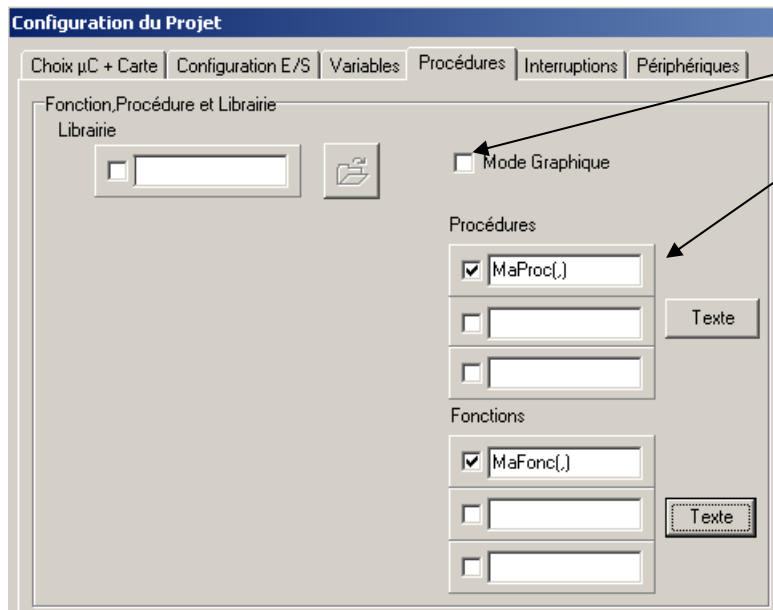
$\text{Nb_Temps_de_Base} = 8 \cdot 10^{-3} / 51,2 \cdot 10^{-6} = 156$ (valeur arrondie de 156,25).

Il faut écrire Time_timer0(156)

8.10 PROCEDURE OU FONCTION UTILISATEUR

Les procédures ou fonctions utilisateurs sont déclarées à partir de la boîte de configuration de projet. Elles peuvent être définies sous forme graphique ou sous forme de texte.

8.10.1 PROCEDURE OU FONCTION DEFINIE PAR UN TEXTE

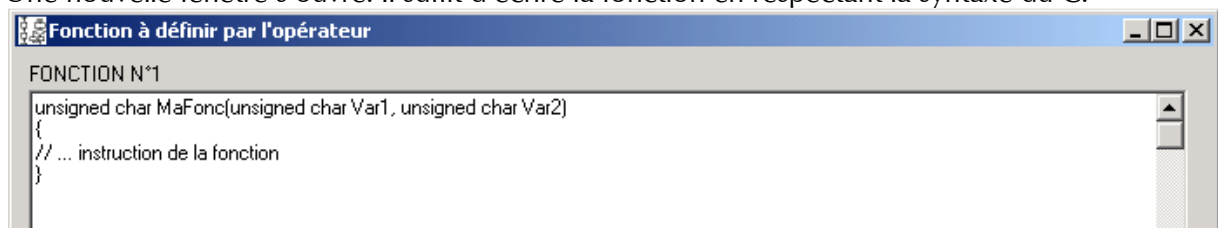


Il faut déclarer la procédure ou fonction sans cocher sur la case Mode Graphique

Dans ce champ, il faut écrire le nom de la fonction ou procédure exactement comme elle apparaîtra dans les listes déroulantes des boîtes de dialogues. Voir § 8.1).

Il est donc préférable de placer les parenthèses et une ou plusieurs virgules s'il y a plusieurs paramètres.

Pour écrire la définition de la fonction ou procédure, il faut cliquer sur le bouton Texte. Une nouvelle fenêtre s'ouvre. Il suffit d'écrire la fonction en respectant la syntaxe du C.



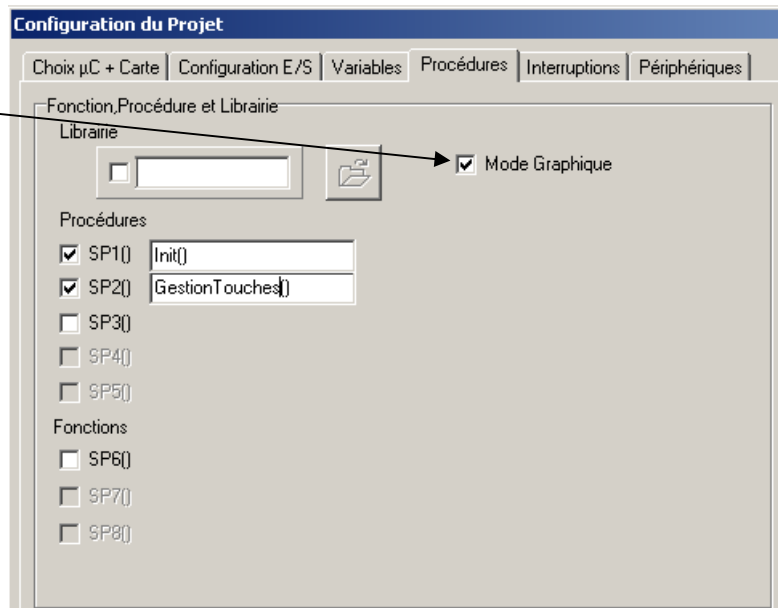
On peut utiliser des variables locales. Elles ne seront pas visibles dans la fenêtre d'observation qu'on peut ouvrir lors du déroulement du programme sur l'algorithme.

Attention : il est indispensable que l'accolade ouvrante soit sur la 2^{ème} ligne et non en fin de 1^{ère} ligne. Multi-prog se sert de la 1^{ère} ligne pour établir la déclaration de la fonction.

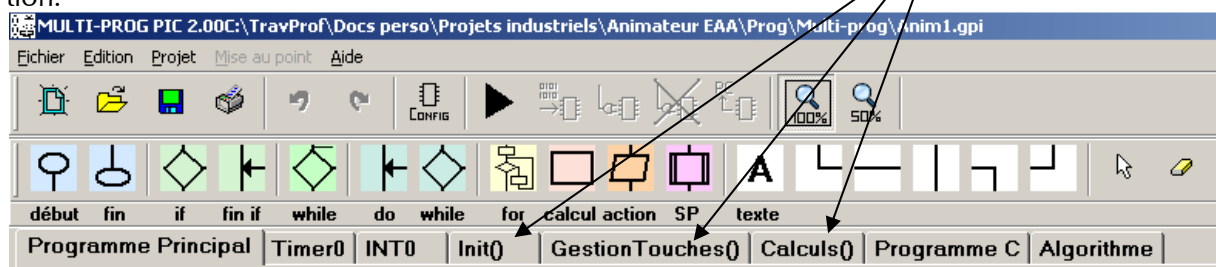
8.10.2 PROCEDURE OU FONCTION DEFINIE GRAPHIQUEMENT

Il faut déclarer la procédure ou fonction en cochant sur la case Mode Graphique.

Si on renomme une procédure ou fonction, il ne faut pas oublier les parenthèses accolées au nom.



Un onglet supplémentaire apparaît dans la fenêtre principale pour chaque procédure ou fonction.



Il suffit alors de compléter la feuille de travail pour la procédure ou la fonction.

9 ALGORIGRAMMES POUR LES INTERRUPTIONS

3 sources d'interruptions font apparaître automatiquement une fenêtre pour un algorithme de traitement d'interruption :

- débordement du timer 0
- front ou état sur l'entrée RB0/INT
- réception d'une donnée sur la liaison série asynchrone

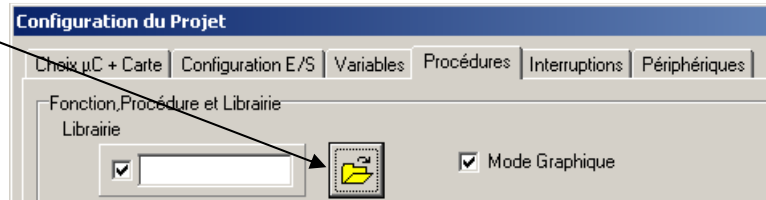
Les règles pour les algorithmes de traitement d'interruption sont identiques à celles de l'algorithme principal.

Ils doivent commencer par le symbole début et se terminer par le symbole fin. Ils ne doivent bien entendu pas contenir de boucle sans fin.

10 UTILISATION D'UNE BIBLIOTHEQUE UTILISATEUR

Multi-prog permet de définir une bibliothèque utilisateur. Il faut appeler la boîte de configuration de projet avec la commande **Projet / Configuration (Algorigramme)**.

Il faut ouvrir la boîte de dialogue de sélection et choisir le nom d'un fichier c qui contient des définitions de fonctions. Après sélection, le nom du fichier apparaît sous **Librairie**.



Avec CC5X, les définitions des fonctions doivent être placées entre les directives `#pragma library 1` et `#pragma library 0`.
Les fonctions non utilisées ne seront pas compilées.

Ex :

```
#pragma library 1
void Temporisation(unsigned int Duree) {
    unsigned int i;
    for(i=Duree;i!=0;Duree--)
        ;
}
#pragma library 0
```

Attention : Pour que la compilation soit correcte, il faut absolument placer un retour ligne après le dernier `#pragma`.

Lors de la traduction automatique de l'algorigramme en langage C, la ligne suivante est placée :
`#include «Chemin_d'accès \ MaBib.c »`

Les fonctions de la bibliothèque n'apparaissent pas automatiquement dans les boîtes de dialogues de Multi-prog.

Lors du déplacement des fichiers d'un projet, il faut penser à mettre à jour le chemin d'accès.

11 CONFIGURATION & UTILISATION D'IC-PROG

IC-Prog nécessite d'être configuré. Les fichiers pour IC-prog sont automatiquement placés dans le dossier d'installation de Multi-Prog. L'exécutable est `Icprog.exe`.

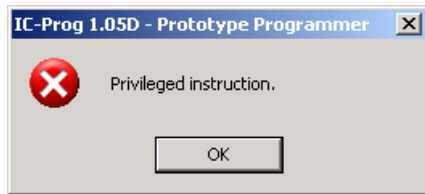
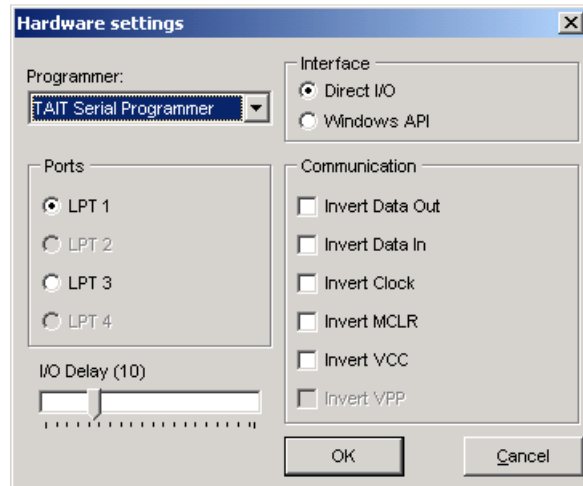
11.1 CONFIGURATION POUR WINDOWS XP

Cette configuration n'est nécessaire qu'une fois, avant la 1^{ère} utilisation.

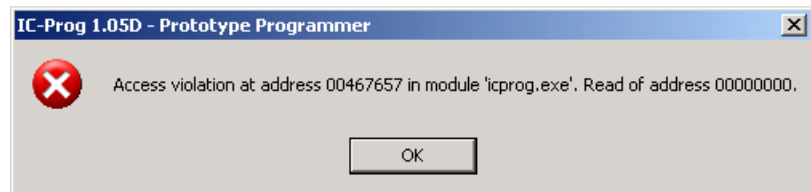
Lancer le logiciel IC-Prog. Si c'est la première utilisation, une boîte de dialogue indique qu'il faut d'abord le configurer. La boîte de dialogue suivante s'ouvre. Il faut fixer les options comme sur le dessin.

Si ce n'est pas la 1^{ère} utilisation, il faut utiliser la commande Config/ Hardware ou Configuration / Hardware.

Après avoir cliqué sur OK, le message suivant apparaît :



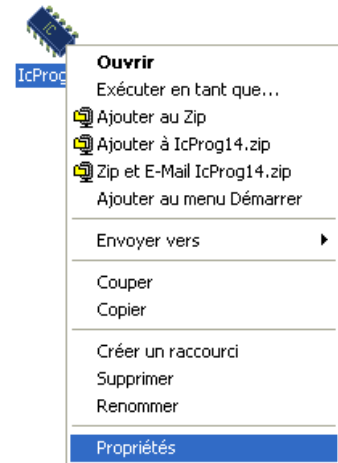
Lors de la 1^{ère} configuration, après avoir cliqué sur OK, le logiciel se lance puis le message suivant apparaît devant la fenêtre principale de IC-Prog :



Ce problème va être résolu en suivant les indications ci-dessous.

Cliquer sur OK, puis fermer la fenêtre principale de IC-Prog. Placez-vous dans le répertoire contenant l'exécutable IC-PROG.EXE, sélectionnez-le et cliquez sur le bouton droit de votre souris.

Dans le menu contextuel, cliquez sur Propriétés comme ci-contre :

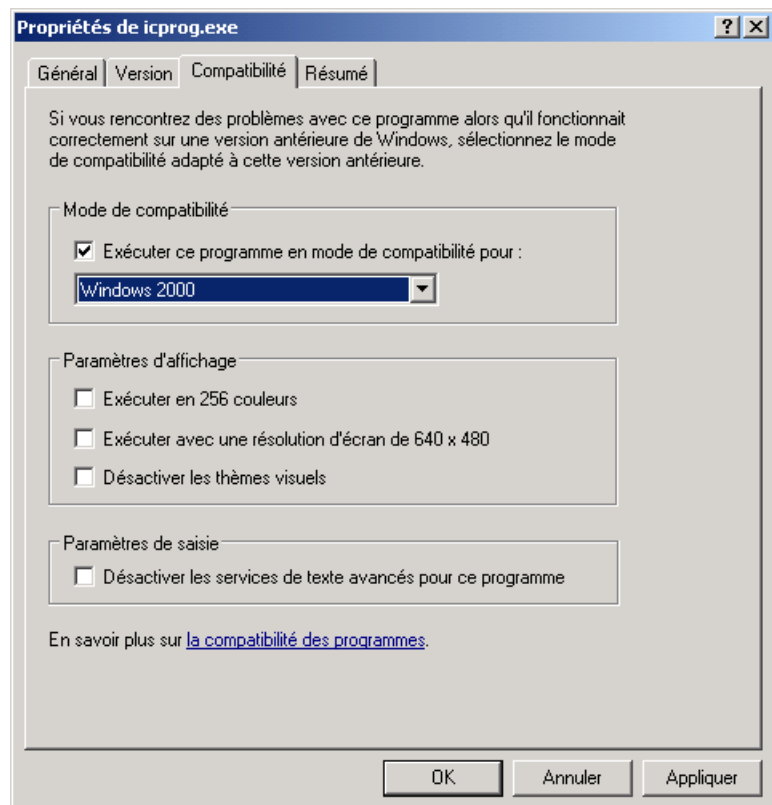
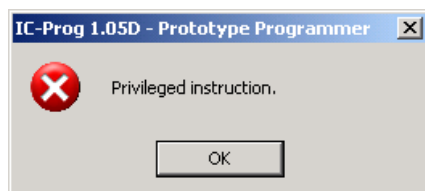


Dans la fenêtre Propriétés de IC-Prog , cliquez sur l'onglet Compatibilité, cochez "Exécuter ce programme en mode de compatibilité pour" et sélectionnez dans la liste le système d'exploitation Windows 2000.

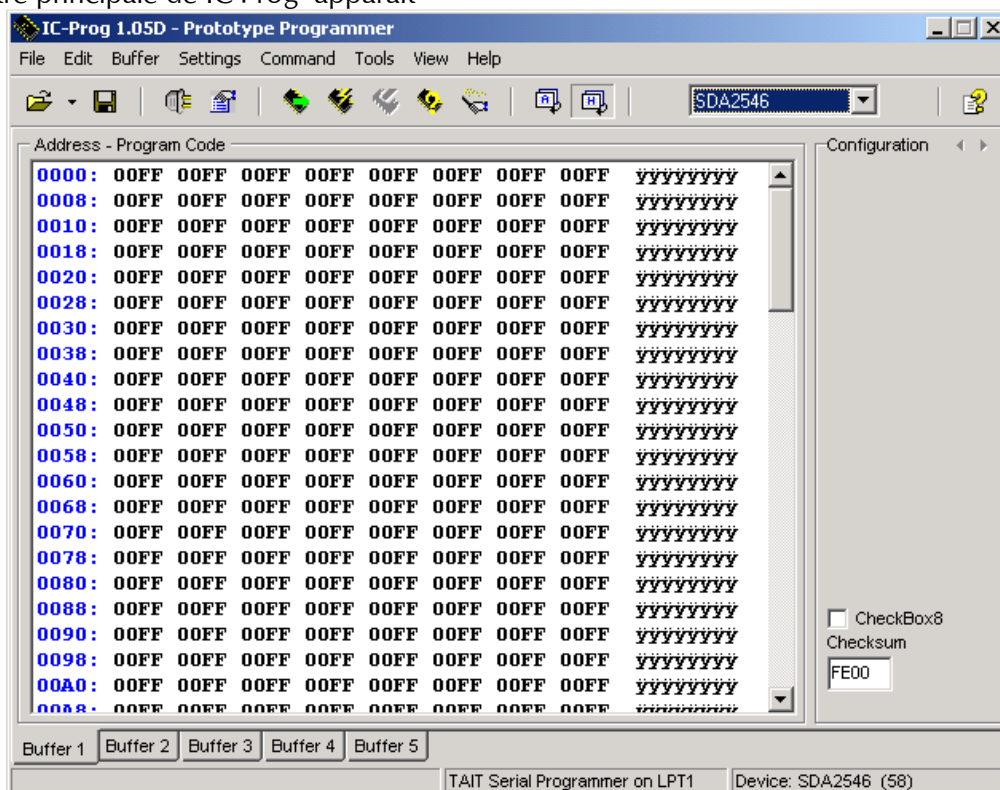
Cliquer sur OK. Deux messages apparaissent. Il faut cliquer à chaque fois sur OK.

Le logiciel peut se lancer maintenant normalement.

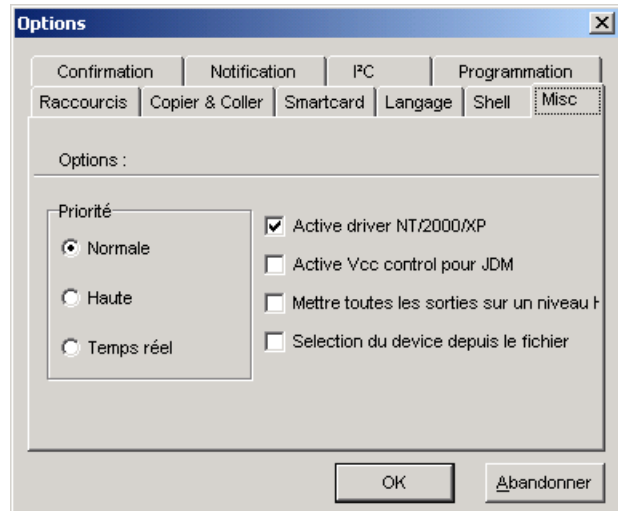
Au lancement, le message suivant apparaît :



La fenêtre principale de IC-Prog apparaît



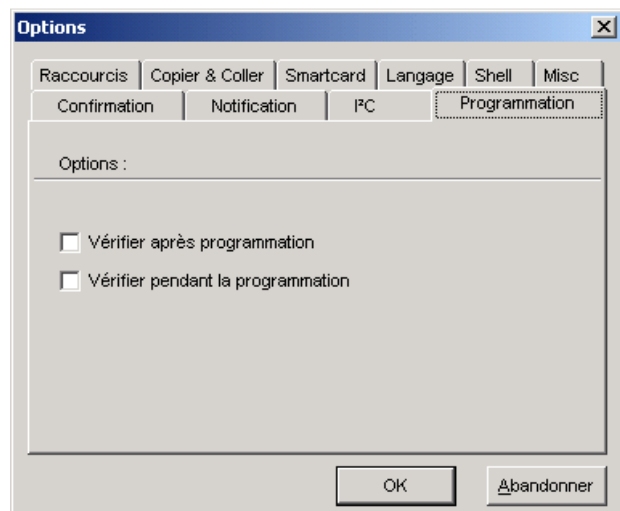
Avec la commande Settings / Option, on peut choisir la langue. Après avoir choisi French, les menus sont traduits.
Il faut configurer l'onglet Misc comme suit :



11.2 OPTIONS D'IC-PROG

Les options d'IC-prog sont définies avec la commande Settings / Option, ou Configuration / Option. La boîte de dialogue avec onglets permet de choisir la langue, la vérification de la programmation du composant, etc.

Avec Multi-prog, il est inutile de cocher la vérification après programmation dans IC-Prog car Multi-prog lance systématiquement une vérification. Si l'option est cochée, il y a 2 vérifications, ce qui ralentit inutilement le processus de programmation.



Attention : Il faut vérifier à chaque 1^{ère} utilisation d'un nouveau PIC que le PIC choisi dans IC-Prog est bien le même que celui défini dans Multi-prog.

11.3 PROGRAMMATION DU PIC CIBLE

Choix du PIC cible dans IC-prog

A chaque fois qu'on utilise un nouveau PIC par rapport au PIC du dernier projet, il faut modifier le PIC cible dans IC-prog et une valeur dans Multi-prog.

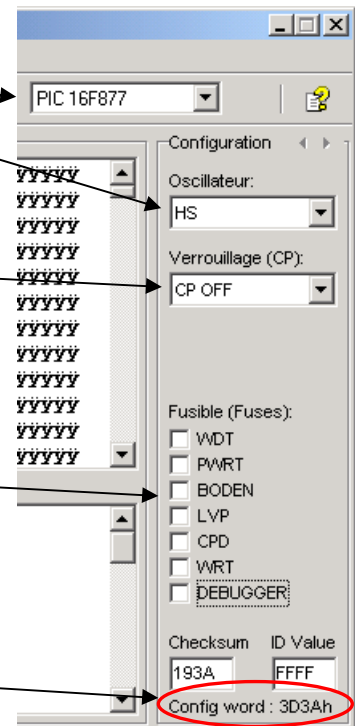
Pour lancer IC-prog, il faut se placer dans le dossier d'installation de Multi-prog et double cliquer sur icprog.exe.

Choix du PIC cible

Bits de configuration selon :

- le type d'oscillateur (HS = High Speed = quartz > 8 MHz)
- la protection des données une fois la programmation terminée (CP = Code Protection)
- différentes possibilités (WDT = Watch Dog Timer ; PWRT = PoWeR up Timer ; BODEN = Brown Out reset ENable ; LVP = Low Voltage Programming ; CPD = Code Protection Data EEPROM ; WRT = flash programm memory WRiTe enable)

la valeur correspondant aux bits de configuration est donnée dans le coin.

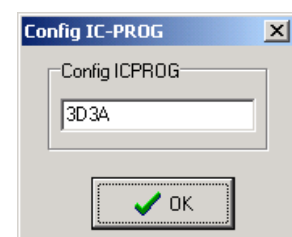


Il faut noter cette valeur qui est utilisée dans Multi-Prog.

On peut ensuite fermer IC-Prog qui n'est plus lancé manuellement par la suite. Il n'est pas nécessaire d'enregistrer une configuration ou un fichier pour que le PIC désigné soit conservé en mémoire.

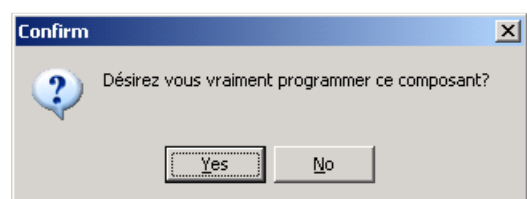
Configuration de Multi-prog pour IC-prog

Après avoir noté le mot de configuration, il faut choisir la commande Projet / Configuration (IC-prog). Il faut alors recopier le mot dans la boîte de dialogue qui s'ouvre.

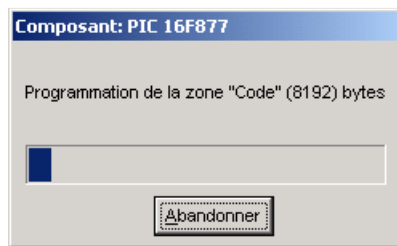


Déroulement de la programmation

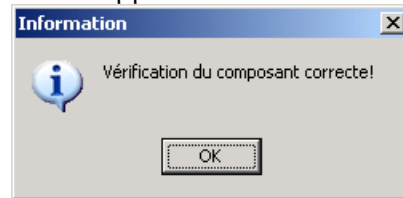
En fin de compilation, une boîte de dialogue demande si on souhaite bien programmer le composant.



Après avoir répondu oui, la programmation commence. Diverses boîtes de dialogues informent sur l'avancement de la programmation et de la vérification.



Après la programmation, une vérification est lancée. Si la programmation s'est effectuée correctement, la boîte de dialogue suivante apparaît :

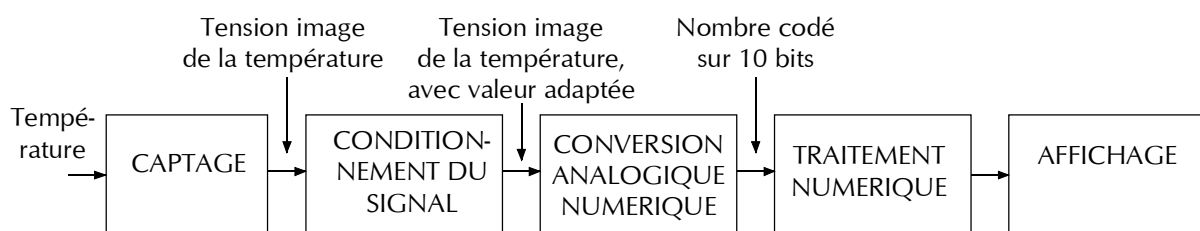


12 UN EXEMPLE DE A à Z : THERMOMETRE NUMERIQUE

12.1 INTRODUCTION

On utilise une maquette avec un microcontrôleur PIC pour réaliser un thermomètre. Les lignes ci-dessous sont extraites d'un document élève (option MPI en 2^{nde}).

Le schéma fonctionnel du thermomètre est donné ci-dessous :



La fonction CAPTAGE permet de convertir la température en une tension image de cette température. Elle est constituée avec un capteur de température intégré (de nombreux composants élémentaires sont intégrés sur une « puce » électronique dans un petit boîtier plastique).

La fonction CONVERSION ANALOGIQUE NUMERIQUE permet de convertir la tension image de la température en un nombre image de cette température. La tension à l'entrée doit être adaptée à la tension pleine échelle du convertisseur.

C'est pour permettre l'adaptation de valeur qu'est utilisée la fonction CONDITIONNEMENT DU SIGNAL. La tension en sortie est toujours image de la température, mais la valeur est adaptée au CAN. La valeur pour la température maximale mesurable correspond à la tension pleine échelle.

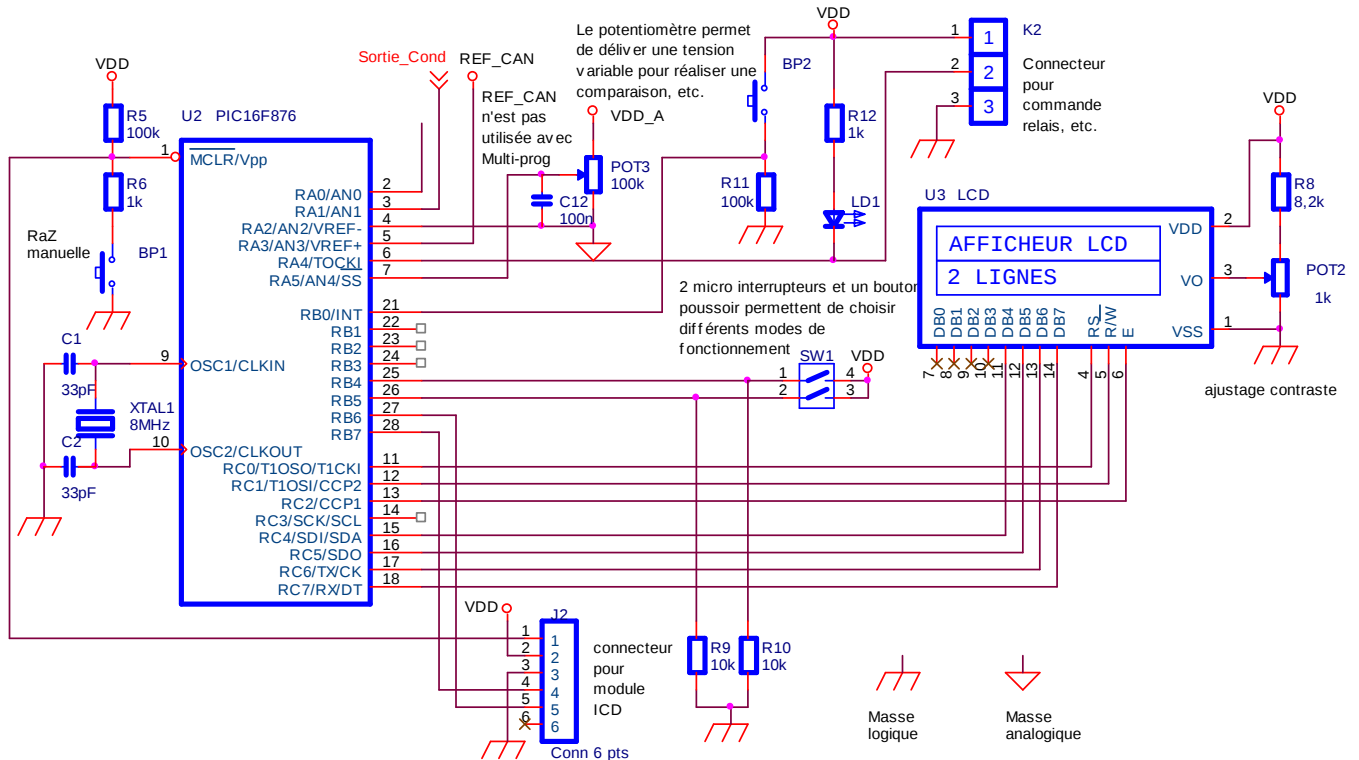
La fonction TRAITEMENT NUMERIQUE permet d'obtenir un nombre égal à la température exprimée en °C à partir du nombre en sortie de la conversion analogique numérique.

Le thermomètre est constitué de 2 cartes électroniques :

- Une carte fille qui comprend la fonction de CONDITIONNEMENT DU SIGNAL

- Une carte mère qui comprend les fonctions de CONVERSION ANALOGIQUE NUMERIQUE, TRAITEMENT NUMERIQUE et AFFICHAGE.

12.2 SCHEMA SIMPLIFIE DE LA CARTE MERE



Sur ce schéma ne sont pas représentés :

- les circuits pour les alimentations (tension de références pour le CAN du PIC, pour la carte fille, etc.)
- les condensateurs de découplage
- etc.

La tension de référence pour le CAN du PIC n'est pas utilisée par Multi-prog. Celui-ci programme le PIC pour que la tension d'alimentation soit utilisée.

Un μC PIC réalise la conversion A/N et le traitement numérique. L'affichage est réalisé avec un afficheur à cristaux liquides standard de 2 lignes avec contrôleur intégré.

12.3 VALEURS EN DIFFERENTS POINTS DE LA CHAINE DE TRAITEMENT

Le fonctionnement du capteur et du conditionneur ne sont pas décrits ici.

Le tableau ci-dessous récapitule les différentes valeurs pour une plage de mesure de température de -40°C à $+100^{\circ}\text{C}$.

Température en °C	Tension Vs2 en sortie du conditionneur du signal	Résultat CAN R	Valeur à afficher V
-50°C	0	0	-50
-40 °C	0,31V		-40
100 °C	4,6875 V		100
110°C	5V	1023	110

-50°C et 110°C sont en réalité impossibles à afficher, car il est impossible d'obtenir 0V et 5V en sortie du conditionneur, à cause des tensions de déchet de l'amplificateur utilisé.

La température peut être affichée par pas de $160/1023 = 0,16^\circ\text{C}$.

A cause des incertitudes de mesures (notamment la précision du capteur), un tel pas n'a pas de sens.

Avec un pas d'affichage de 1°C , la valeur à afficher V est égale au résultat du calcul suivant :

$$V = -50 + R/1023 * 160$$

V est un entier signé codé sur 8 bits ou plus. La conversion en caractères ASCII est simple.

Pour les calculs intermédiaires, il faut utiliser des « réels » (float), ou des entiers codés sur 32 bits (long).

On peut réaliser une approximation satisfaisante avec la formule suivante :

$$V = -50 + R/102 * 16$$

Comme R est au maximum égal à 1023, $1023*16$ est inférieur à la valeur max qu'on peut coder avec un entier non signé codé sur 16 bits (65535) ou même un entier signé (32767).

Il faut effectuer les calculs dans l'ordre :

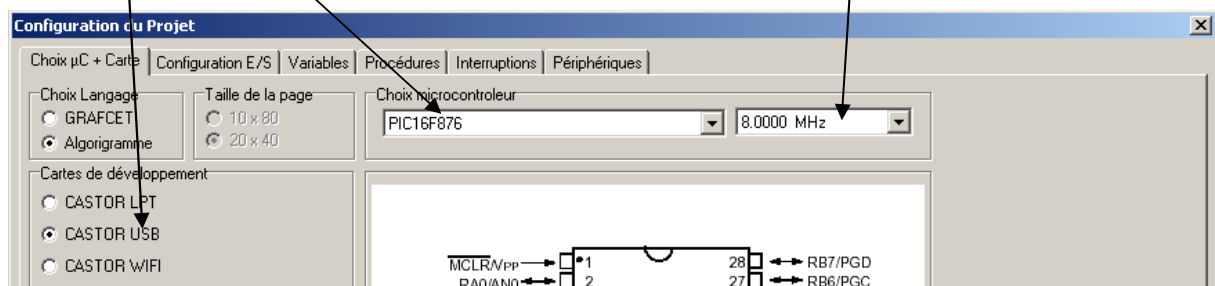
$$\begin{aligned} &R * 16 \\ &/102 \\ &-50 \end{aligned}$$

12.4 CONFIGURATION DU μC

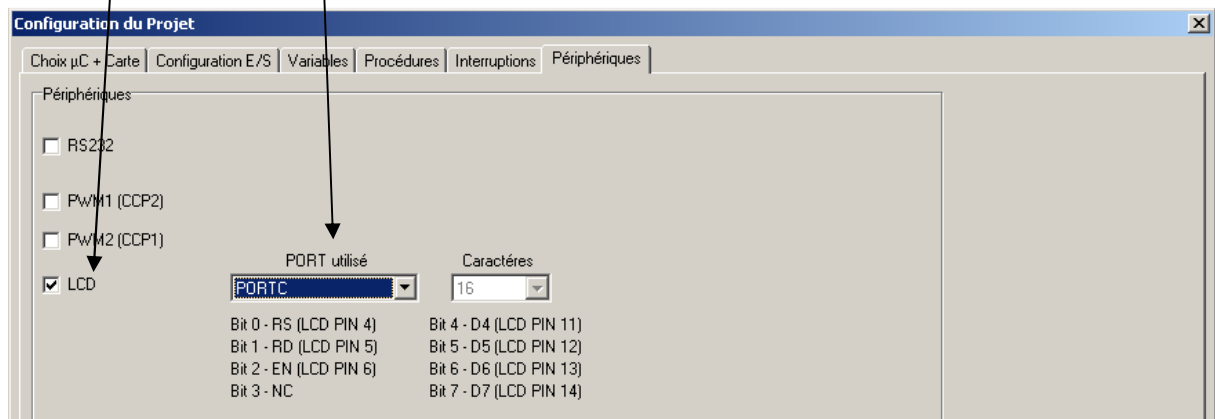
Lancer le logiciel Multi-prog puis créer un nouveau projet.

Une boîte de dialogue est automatiquement ouverte.

Il faut déclarer le μC de la carte et fixer la fréquence de l'oscillateur utilisé et indiquer la carte d'interface



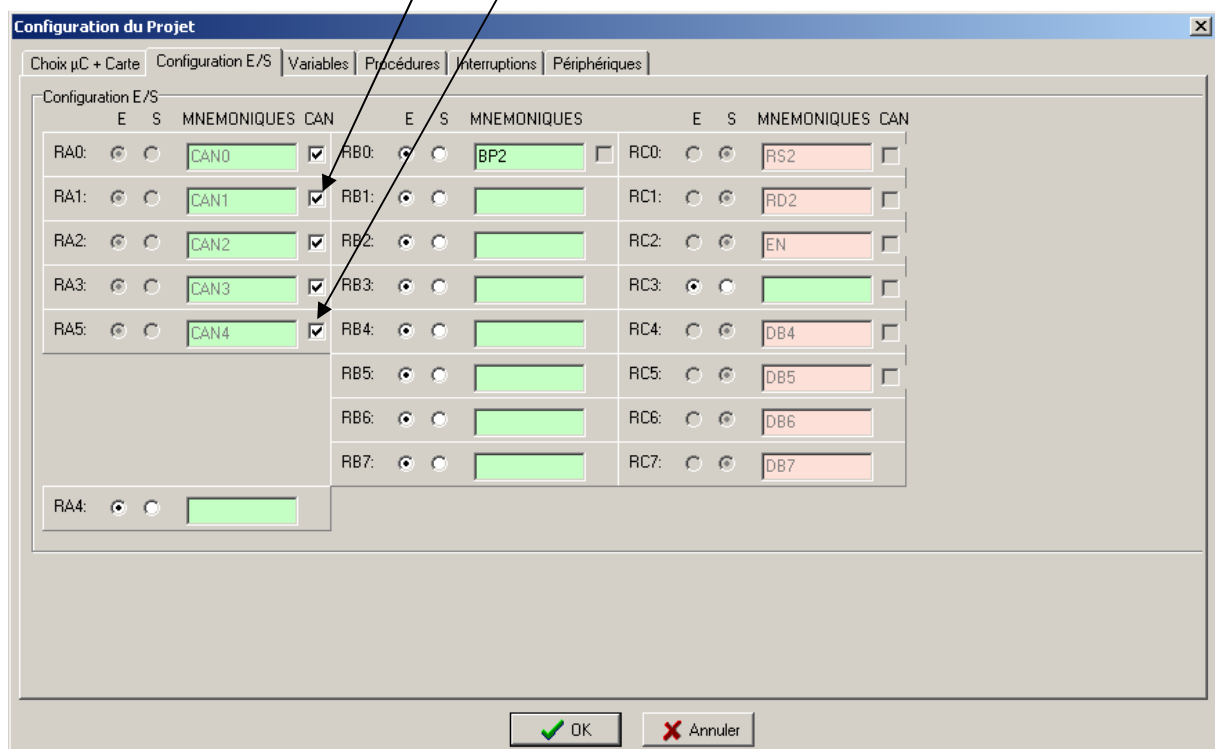
Il faut ensuite cliquer sur l'onglet périphérique pour déclarer l'utilisation d'un afficheur à cristaux liquide (LCD) sur le port C (un port = un ensemble de broches).



Il faut ensuite choisir l'onglet Configuration E/S. Certains champs ont été remplis par la déclaration de l'afficheur.

Pour les manipulations, on utilise 2 entrées qui peuvent être aiguillées en entrée du CAN intégré au µC. Il s'agit de RA1 et RA5.

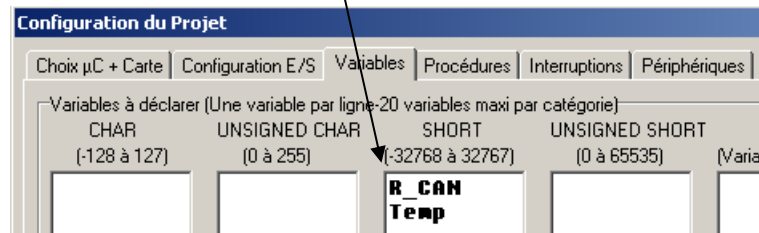
Il faut déclarer que ces entrées RA1 et RA5 sont utilisées en entrées du CAN. Les autres E / S doivent être fixées comme suit :



Remarque : la sélection des 2 entrées désirées du CAN entraîne automatiquement la sélection des autres entrées.

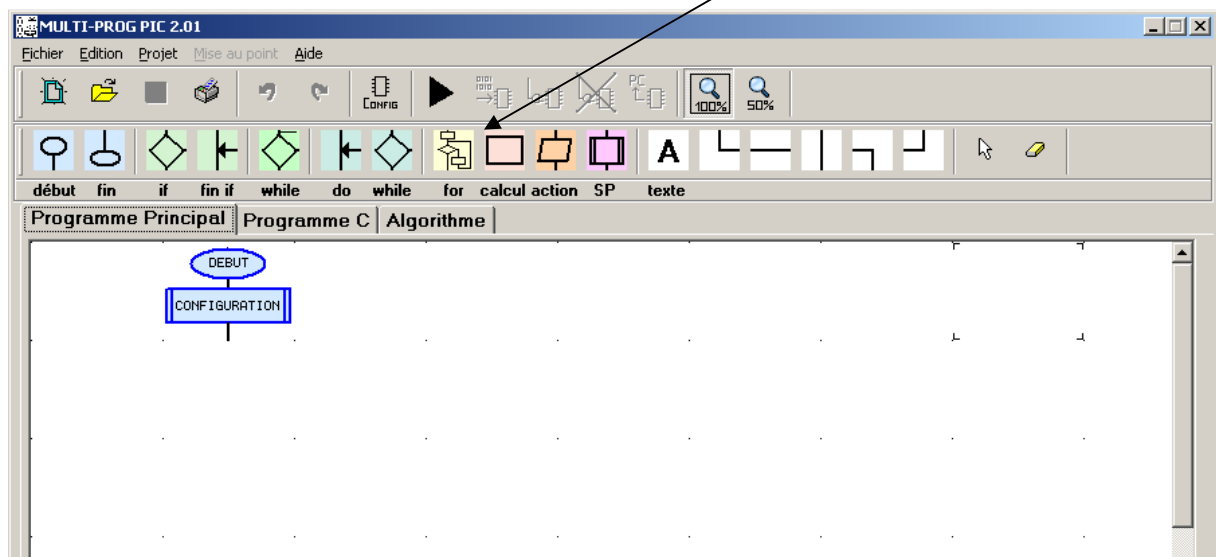
BP2 va permettre de tester le bouton poussoir.

Pour les calculs, il faut utiliser des variables signées codées sur 16 bits qu'il faut déclarer dans le projet, avec l'onglet Variables.



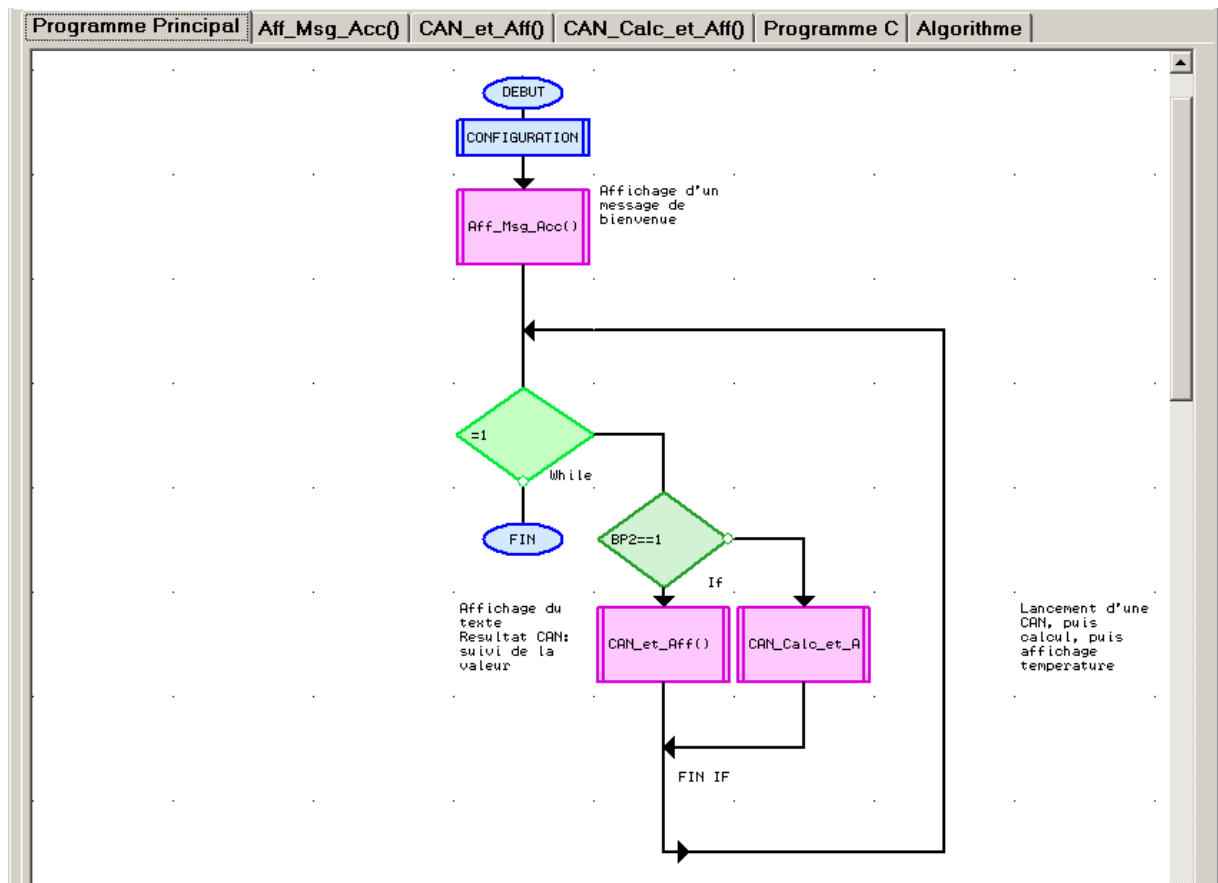
Après avoir complété la boîte de dialogue, il faut cliquer sur OK.

Un page vierge apparaît avec une barre d'outils contenant des symboles qu'on place sur la feuille de travail pour réaliser l'algorithme.

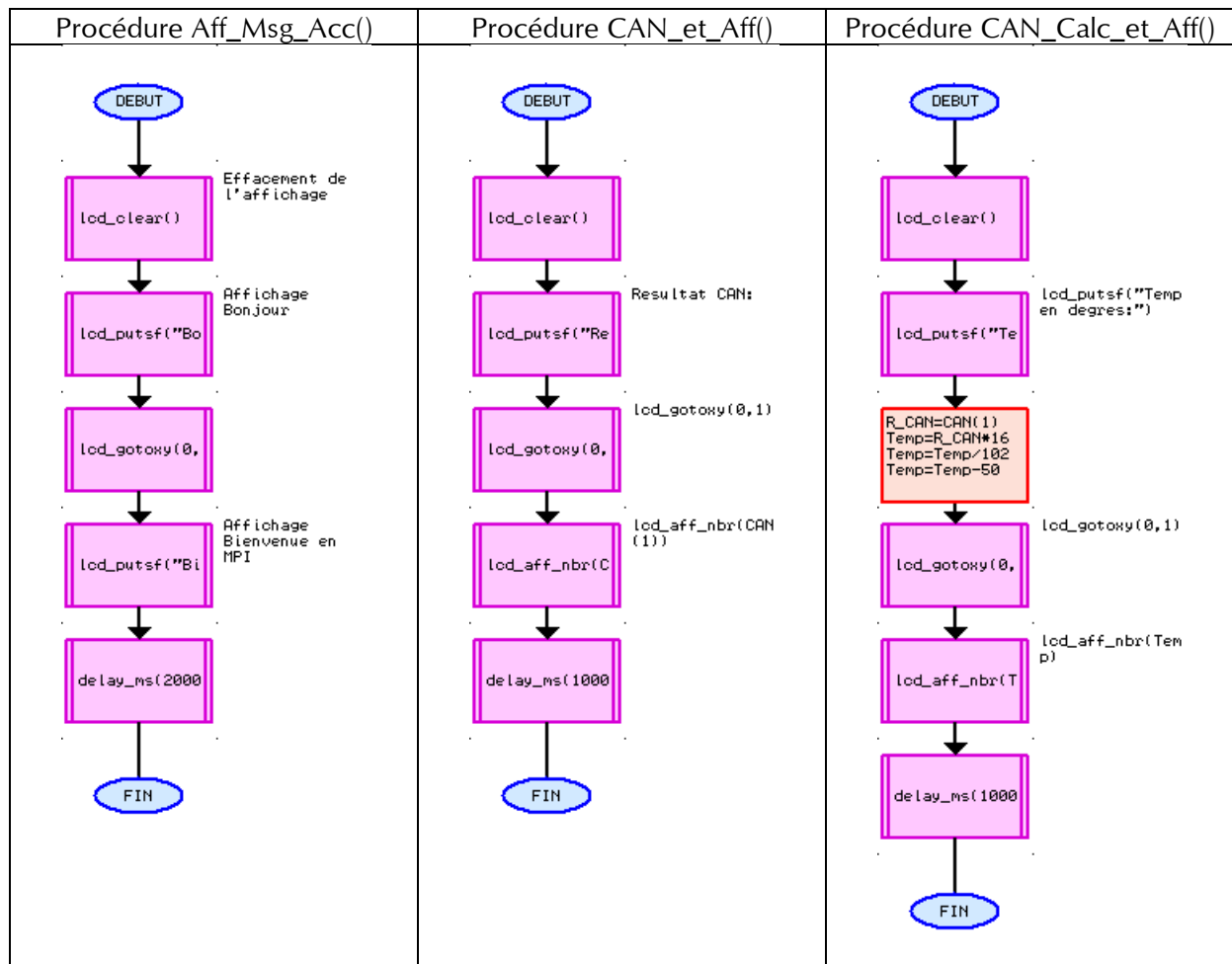


12.5 LE PROGRAMME COMPLET

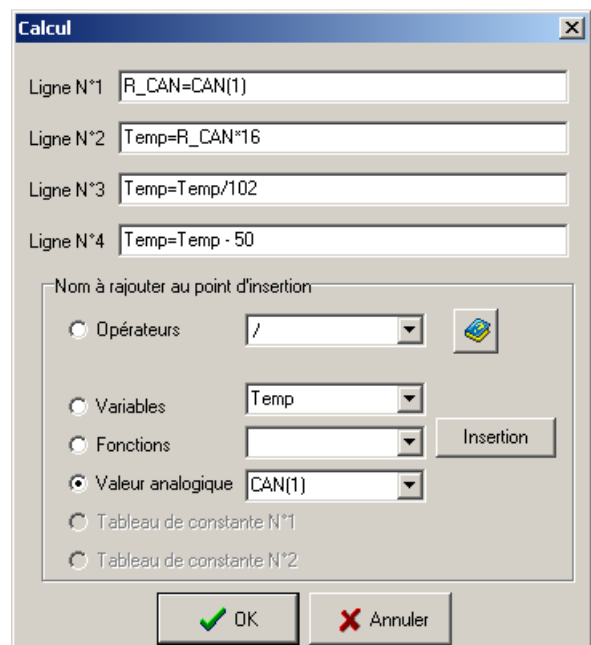
3 procédures utilisateurs ont été définies.



Les procédures sont décrites ci-dessous :



Toutes les informations ont été entrées avec les boîtes de dialogues. Exemple :



13 FONCTIONNEMENT EN MODE MISE AU POINT

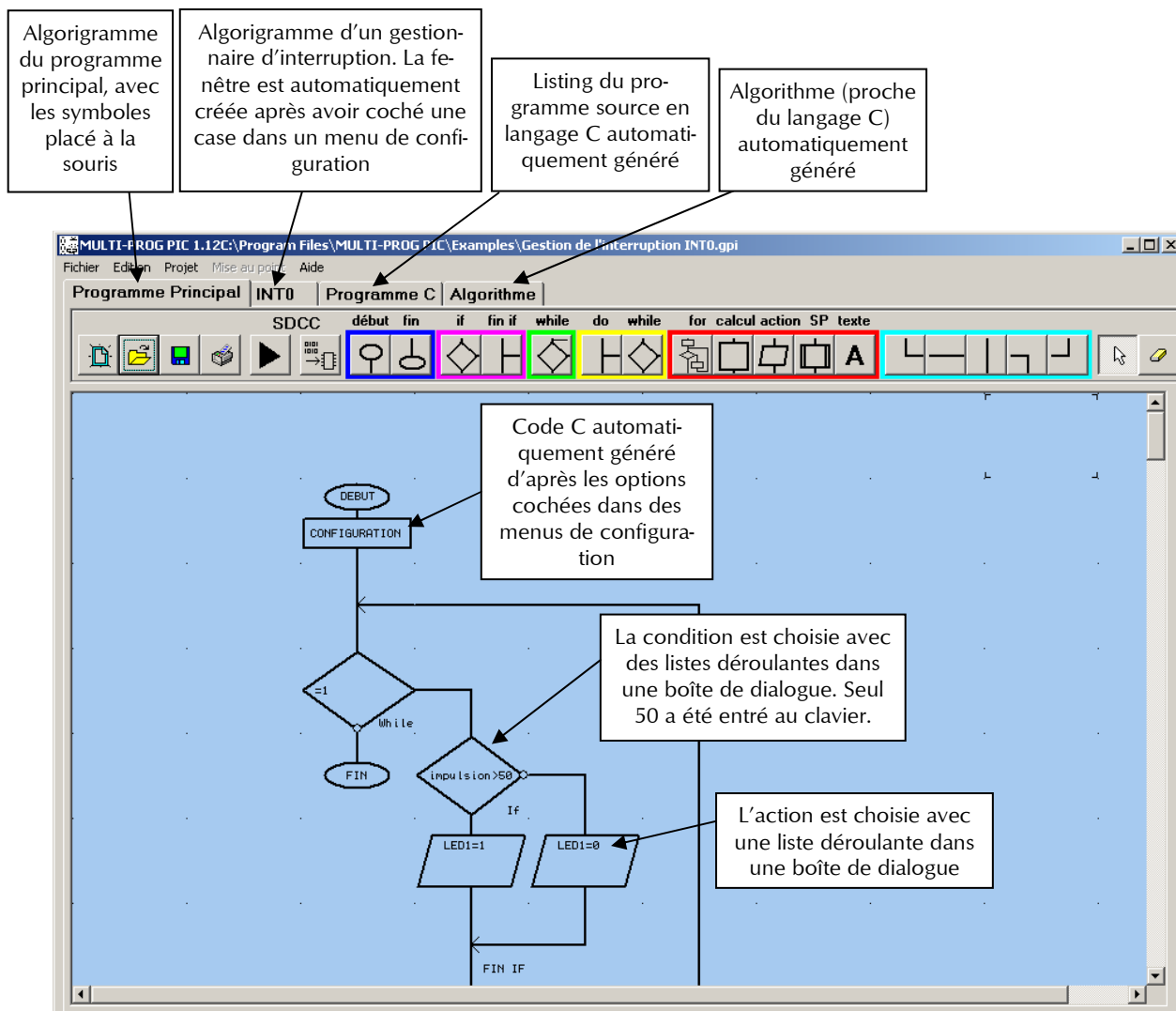
A compléter

ANNEXES

Comparaison compilateurs

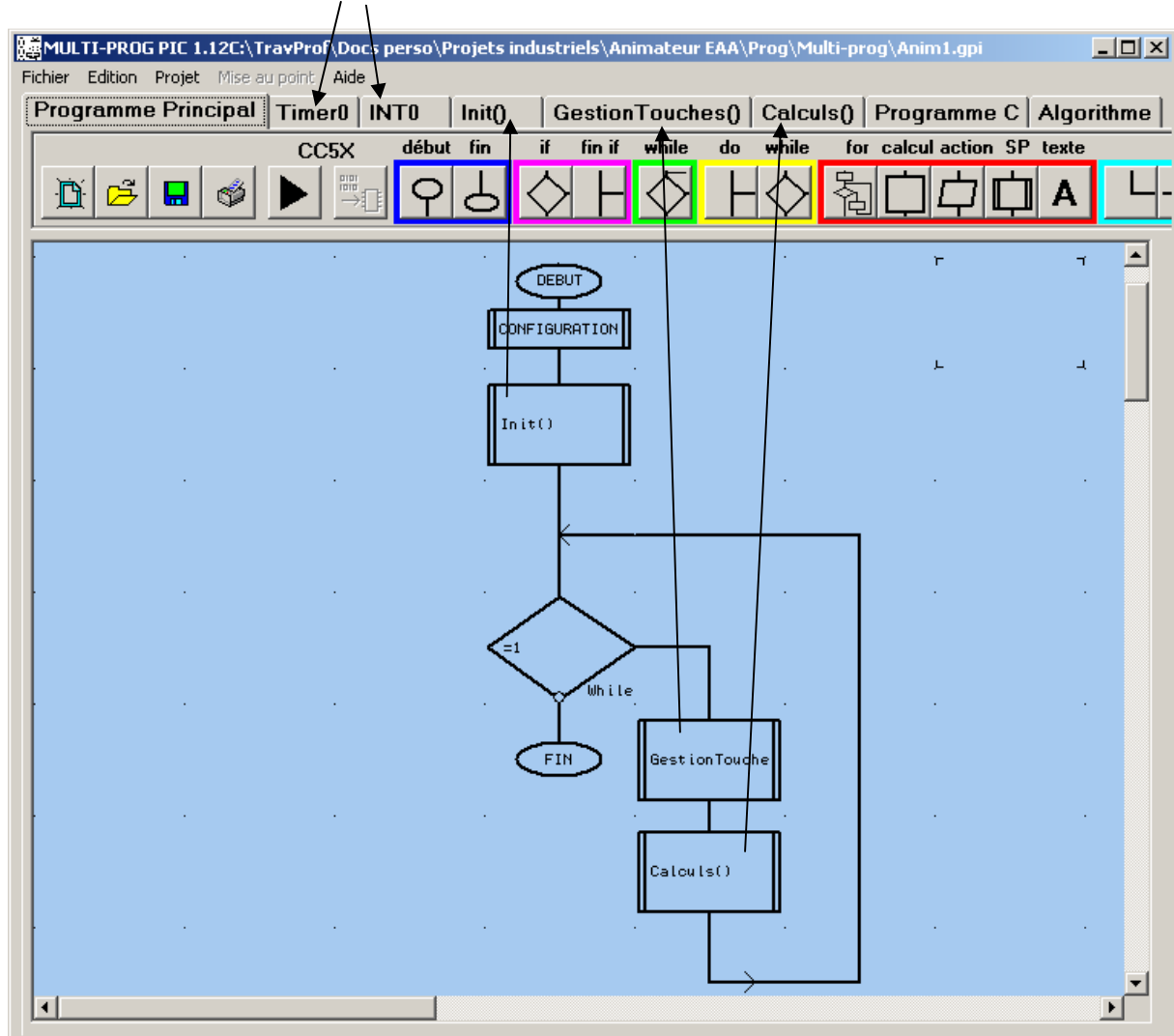
Programme	Compilateur CC5X (<i>estimation avec version complète optimisée</i>)	Compilateur SDCC 2.7 (<i>estimation, il n'y a pas de fichier produit avec la taille exacte</i>)
Gradateur avec BP	FLASH : 162 mots (<i>138 mots</i>) RAM : 13	FLASH : 231 mots (218 mots avec la version 2.6) RAM :
	FLASH : RAM :	FLASH : RAM :

Fenêtre principale



Fenêtre principale correspondant à la version 1.12

Onglets pour les algorithmes de traitement d'interruptions.

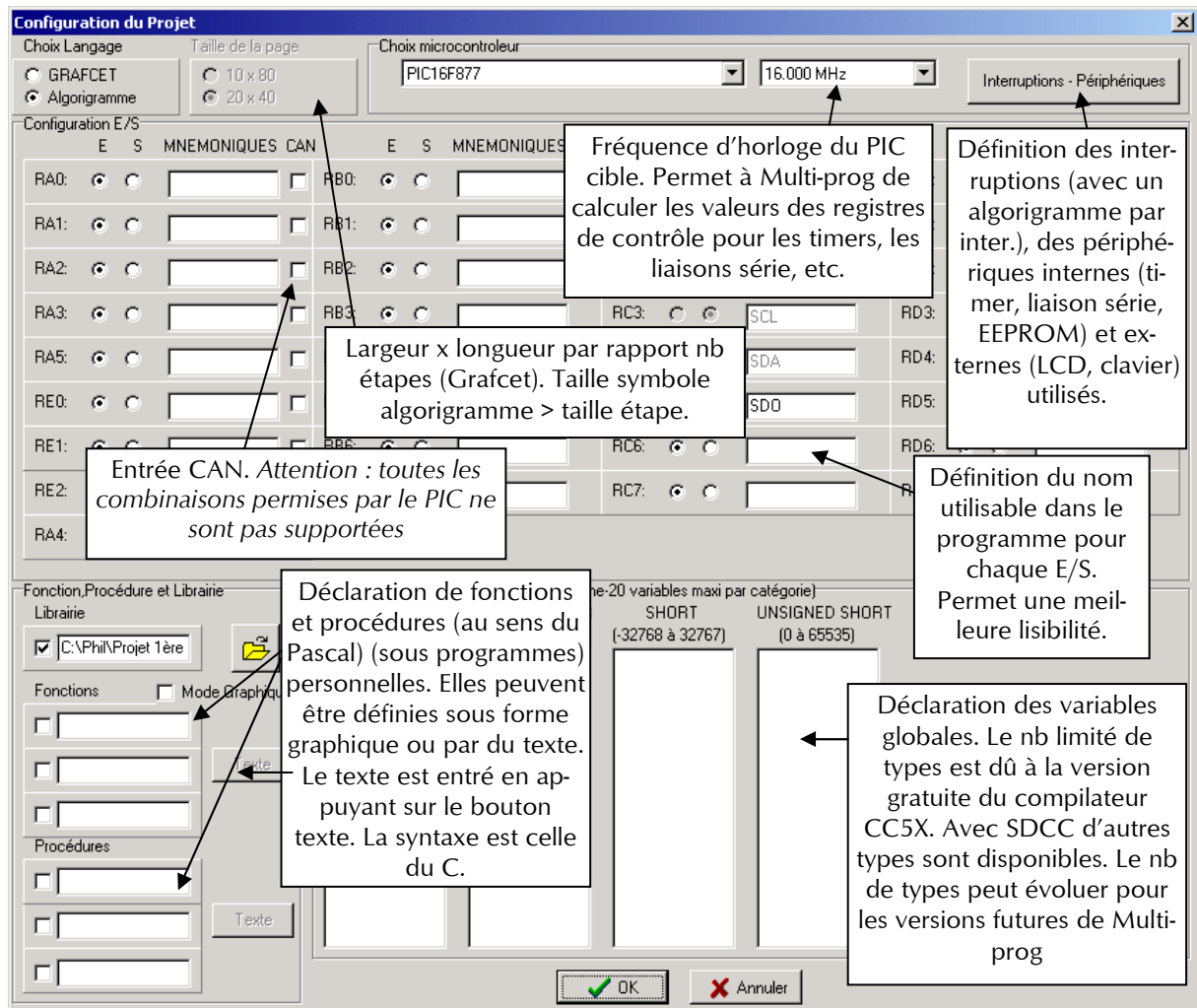


Fenêtre principale correspondant à la version 1.12

Boîte de configuration du projet

La boîte de configuration du projet est automatiquement ouverte lors de la création d'un nouveau projet. Elle peut aussi être ouverte à tout moment avec la commande Projet / E-S Algorithme ou E-S Grafset.

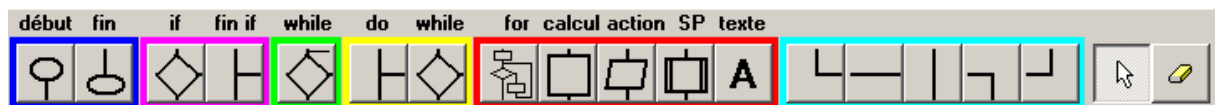
Tous les paramètres peuvent être modifiés sauf la taille de la page qui est choisie lors de la création. Elle ne peut plus être modifiée ensuite.



Fenêtre pour les versions 1.x

Placement et édition des Symboles

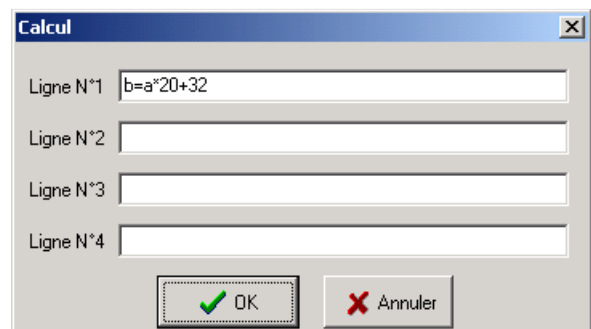
Il suffit de sélectionner un symbole ou un élément de liaison dans la barre d'outils puis de la placer sur la feuille de travail en cliquant à l'endroit désiré.



Selon le type de symbole, une boîte de dialogue s'ouvre ou non lors du placement.

La boîte de dialogue pour le symbole calcul est la suivante :

Il faut tout entrer au clavier.

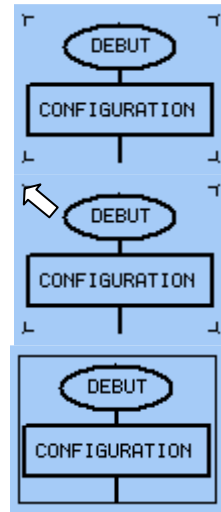


Sélection des symboles

La sélection s'effectue avec l'outil sélection (flèche dans la barre d'outils).

Il faut :

- amener le pointeur pour que le carré de sélection encadre le symbole ou le premier symbole du groupe à déplacer
- avec les versions 1.11 et antérieures, placer le pointeur dans le coin supérieur droit ou inférieur gauche, très près d'un repère
- Cliquer en maintenant et tirer. Le ou les symboles sélectionnés deviennent encadrés. Il faut ensuite relâcher le bouton de la souris lorsque la sélection est terminée.

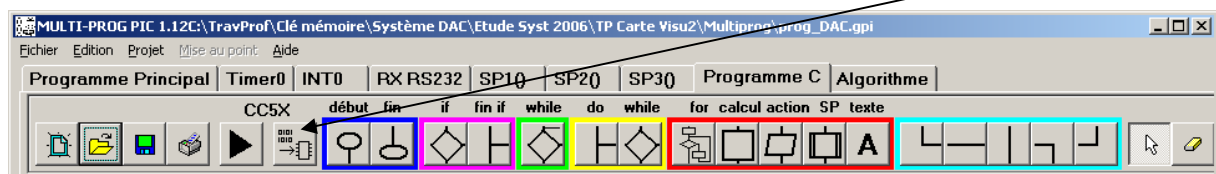


Fichiers produits

Toutes les indications pour l'algorithme (ou le Grafcet) sont dans le fichier NomProjet.gro (version 1.11 et précédentes) ou NomProjet.gpi (depuis la version 1.12).

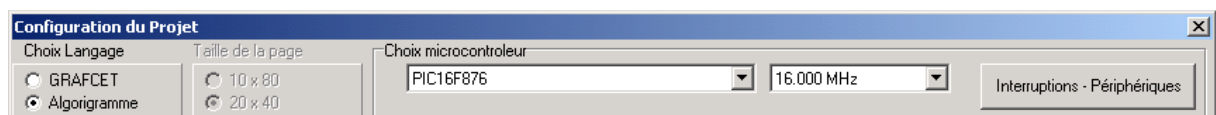
Programmation du PIC cible

Avec la version 1.12, la programmation du PIC cible s'effectue avec le bouton Transfert.

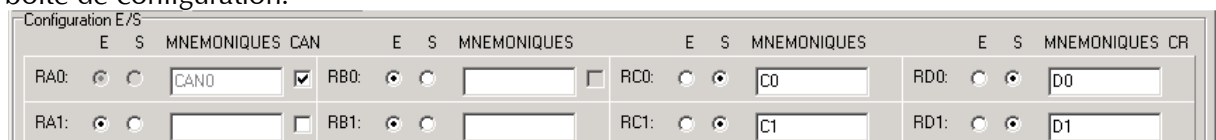


Déclaration de périphériques intégrés

La déclaration de périphériques s'effectue en ouvrant la boîte de configuration du projet avec Projet / E-S Algorithme.



La déclaration de l'utilisation du CAN intégré, s'effectue dans la partie Configuration E/S de la boîte de configuration.



Pour les autres périphériques et interruptions, il faut cliquer sur le bouton Interruptions – Périphériques.

Utilisation d'une interruption

Une fenêtre pour l'algorithme de traitement de l'interruption (ex : onglet Timer0) est automatiquement créée.

