

# K8048 PIC PROGRAMMER BOARD

# Velleman Kits

---

*Welcome to the exciting world of Velleman Kits.*

*Velleman Kit is known all over the world for our High Quality electronic kits. Our range goes from easy to build Mini Kits to more advanced Kits such as High-End audio circuits, domotic systems and computer interface kits.*

*Because of the huge success of our sophisticated measuring instruments like the PC- and hand-held oscilloscopes, we've founded the Velleman Instruments division.*

*The Velleman Kit team*

---

# Table of Contents

<b>généralités</b>	<b>2</b>
<b>Informations sur la Federal Communications Commission (la FCC)</b>	<b>2</b>
<b>Spécifications</b>	<b>3</b>
<b>Connexion</b>	<b>3</b>
<b>Installation de matériel</b>	<b>4</b>
<b>Procédure de programmation</b>	<b>5</b>
Ecrire/modifier un programme .....	5
Compiler .....	5
Programmation proprement dite du microcontrôleur .....	7
Programmation .....	9
Effacer un appareil PIC .....	10
<b>Expériences</b>	<b>11</b>
<b>ICSP</b>	<b>12</b>

## 1 généralités

---

Le K8048 est un programmeur multifonctions pour des microcontrôleurs FLASH PIC™ de Microchip ® \*. Ces contrôleurs se laissent reprogrammer un grand nombre de fois, permettant d'expérimenter à volonté et facilitant la mise à niveau du logiciel d'un appareil dans lequel un contrôleur est installé. Plusieurs types de CI peuvent être programmés et le CI est pourvu de LEDs et de boutons-poussoirs pour tester des circuits simples sur le circuit imprimé.

Il y a deux phases importantes dans l'utilisation de ce programmeur. D'abord, le code programme est écrit à l'aide d'un traitement de texte tel que NotePAD qui fait partie de l'ensemble standard de Windows ou vous pouvez utiliser le paquet de développement de Microchip, c.-à-d. MPLAB, que l'on sait trouver sur le site web [www.microchip.com](http://www.microchip.com).

Le K8048 permet l'installation directe de 4 types de supports de CI, 8 broches, 14 broches, 18 broches et 28 broches. Grâce au connecteur ICSP, vous pouvez également utiliser des supports d'un autre type. Cette extension permet l'usage de contrôleurs avec d'autres connexions et la programmation de contrôleurs montés dans un appareil, sans les enlever de leur support (voir ICSP).

Avec le contrôleur vous pouvez commencer à programmer immédiatement.

\* Le nom et logo de Microchip, PIC, et PICmicro sont des marques déposées de Microchip Technology Inc. aux E-U et dans d'autres pays.

## 2 Informations sur la Federal Communications Commission (la FCC)

---

Cet appareil répond à la Section 15 des Normes FCC à condition que les consignes de montage soient exécutées à la lettre. L'utilisation de l'appareil est soumise aux conditions suivantes: (1) cet appareil ne peut pas occasionner des brouillages nuisibles et (2) l'opération de l'appareil ne peut pas être influencée par des brouillages indésirables.

Visitez le site web <http://www.fcc.gov/> pour de plus amples renseignements sur la FCC

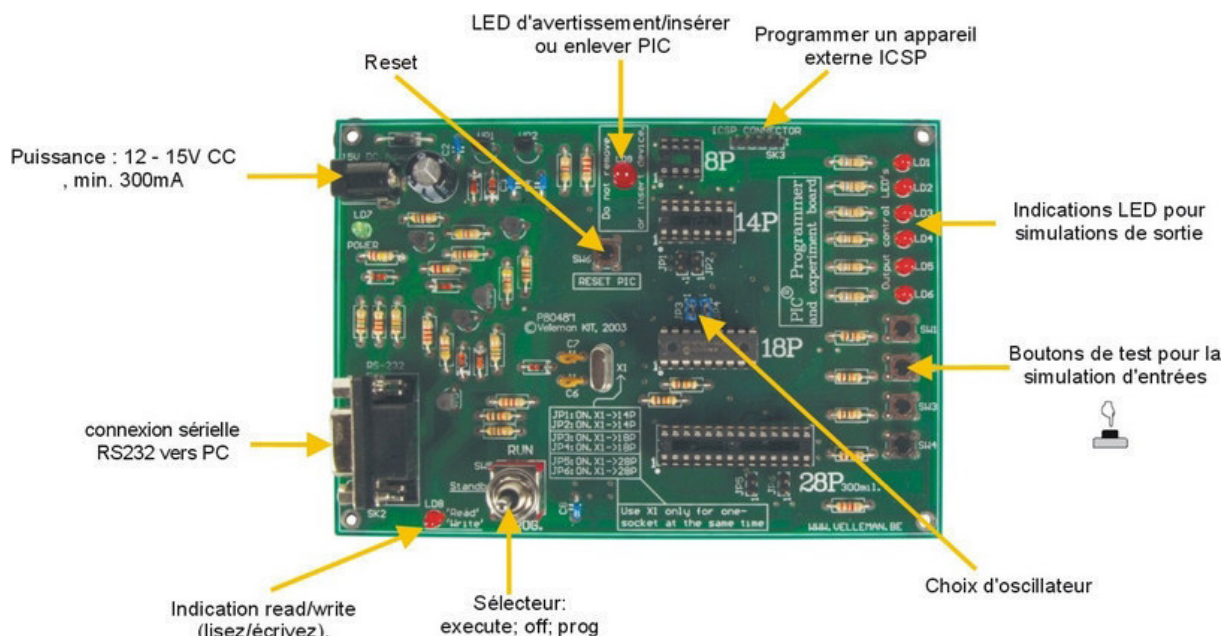
---

### 3 Spécifications

- Convient pour la programmation de microcontrôleurs FLASH PIC(tm) de Microchip®
- Supporte 4 différents PICs 300 mil.: 8, 14, 18 et 28 broches.
- Boutons de test et des indicateurs LED pour organiser des expériences p. ex. les exemples de programmation inclus.
- Connexion facile à votre PC via le port sériel.
- Un microcontrôleur Flash (PIC16F627) est inclus. Il se laisse reprogrammer jusqu'à 1000 x.
- Le logiciel pour compiler et programmer votre code source est inclus.
- Puissance: 12 of 15V CC, min. 300mA, bloc secteur non régulé:
- (PS1205 / PS1208/PS1508 (230Vca))
- (PS1208USA (115Vca))
- un PC compatible IBM, Pentium ou meilleur, avec Windows™ 95/98/ME/NT/2000/XP, CDROM et un port sériel RS232 disponible (cable non inclus p. ex. CW014).
- Supporte les microcontrôleurs FLASH suivants:
- PIC12F629, PIC12F675, PIC16F83, PIC16F84(A), PIC16F871, PIC16F872,
- PIC16F873, PIC16F874, PIC16F876, PIC16F627(A), PIC16F628(A), PIC16F630, ....\*
- Dimensions: 145 mm x 100 mm.

(\*) Visitez notre site web pour la liste à jour et le logiciel le plus récent : [www.velleman.be](http://www.velleman.be)

### 4 Connexion



## 5 Installation de matériel

---

- Connectez le K8048 à votre PC à l'aide d'un câble à 9 broches. Utilisez un port sériel libre et assurez-vous que le logiciel installé ne puisse réclamer ce port (conflit d'interruption) pendant l'emploi du K8048 p. ex. programmes directeur d'autres programmeurs, appareils de mesure, des PDA, etc ...
  - Mettez le sélecteur (SW5) dans la position centrale (standby).
  - Connectez un bloc secteur non régulé (12 à 15VCC, min. 300mA) au K8048 via SK1 (blocs secteur Velleman compatibles: [PS1205](#) / [PS1208](#)/ [PS1508](#) (230Vca) ou [PS1208USA](#) (115Vca))
  - La LED 'POWER' (LD7) doit être la seule à s'allumer pour indiquer que l'alimentation du K8048 est OK.
-

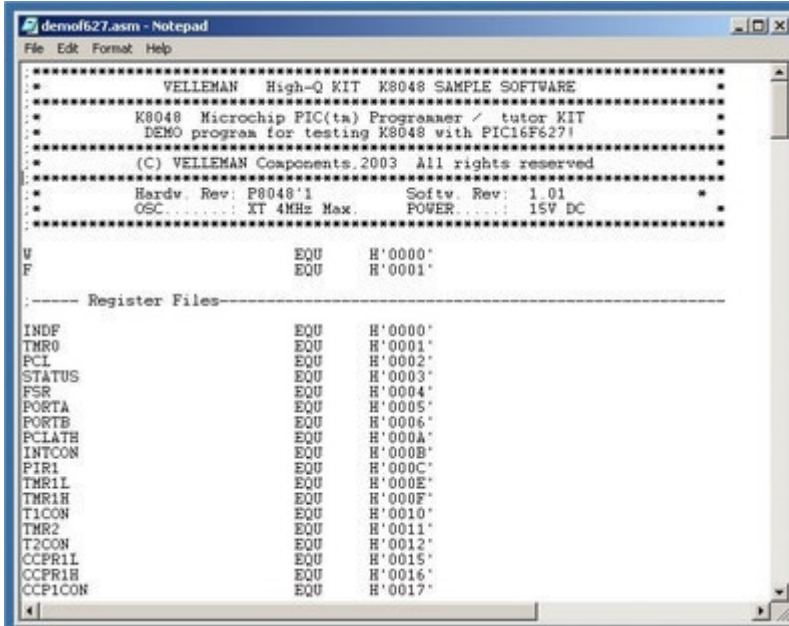
## 6 Procédure de programmation

### Programmer les microcontrôleurs: préparations

- **Ecrire ou modifier le code source:** d'habitude, le code source est écrit avec l'assembleur PIC. Jetez un coup d'oeil sur les exemples du fichier Velleman\K8048\Exemples\.
- **Compiler:** le code que vous avez écrit en langage clair est converti en langage machine. A cette fin nous utilisons le logiciel assembleur PIC de Microchip, MPASMWIN.EXE.
- **Programmer:** le code machine est programmé dans le processeur via votre PC et le K8048 et à l'aide du logiciel PROG PIC2.EXE
- **Effacer un appareil PIC:** Effacer le contenu d'un contrôleur

### 6.1 Ecrire/modifier un programme

Ecrire ou modifier un programme: \Velleman\K8048\Exemples\DEMOF627.ASM



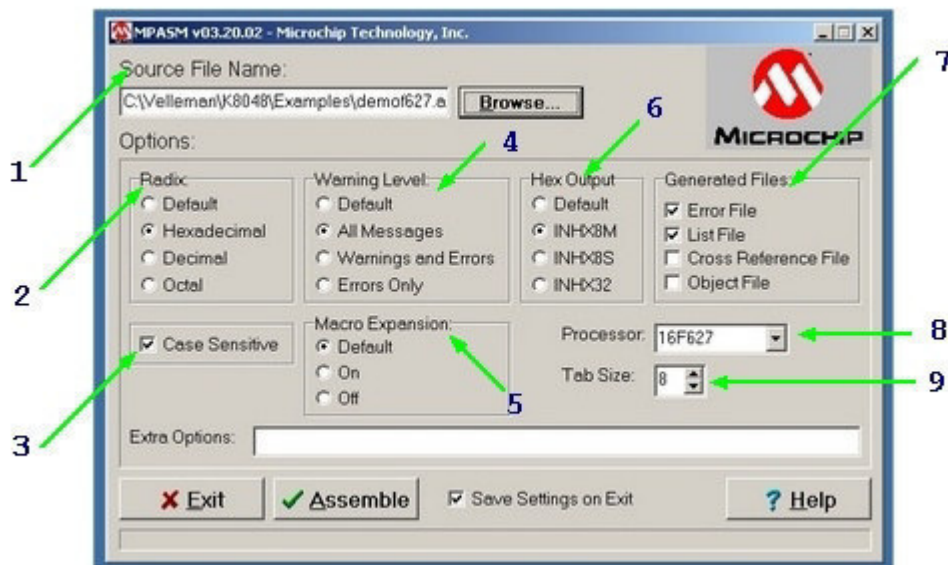
```
demof627.asm - Notepad
File Edit Format Help
*****
* VELLEMAN High-Q KIT K8048 SAMPLE SOFTWARE
* *****
* K8048 Microchip PIC(tm) Programmer / tutor KIT
* DEMO program for testing K8048 with PIC16F627
* *****
* (C) VELLEMAN Components,2003 All rights reserved
* *****
* Hardw. Rev: P8048'1          Softw. Rev: 1.01
* OSC .....: XT 4MHz Max.    POWER .....: 15V DC
* *****
W          EQU    H'0000'
F          EQU    H'0001'
----- Register Files -----
INDF      EQU    H'0000'
TMR0      EQU    H'0001'
PC1       EQU    H'0002'
STATUS    EQU    H'0003'
FSR       EQU    H'0004'
PORTA     EQU    H'0005'
PORTB     EQU    H'0006'
PCLATH    EQU    H'000A'
INTCON    EQU    H'000B'
PIR1      EQU    H'000C'
TMR1L     EQU    H'000E'
TMR1H     EQU    H'000F'
T1CON     EQU    H'0010'
TMR2      EQU    H'0011'
T2CON     EQU    H'0012'
CCPR1L   EQU    H'0015'
CCPR1H   EQU    H'0016'
CCP1CON   EQU    H'0017'
```

Exemple d'un programme traité avec 'NOTEPAD'.

### 6.2 Compiler

Démarrez MPASMWIN.EXE dans le fichier \Velleman\K8048

L'écran de réglage ci-dessous s'ouvrira (cliquez sur un numéro pour avoir plus de détails):



- 1 Le nom du fichier code source et où la trouver p. ex. C:\TEST\DEMEOF267.ASM
- 2 Sélectionnez le format du code machine à créer. Choisissez toujours 'HEXADEMICAL' pour le K8048.
- 3 Cochez cette option pour contrôler l'orthographe des variables (majuscules ou minuscules)
- 4 Détermine la mesure dans laquelle des signaux d'erreurs et des avertissements sont générés dans le fichier d'erreurs. Sélectionnez de préférence ALL MESSAGES ou ERRORS ONLY.
- 5 Gardez tout simplement le réglage d'usine pour cette option. Dans la position 'ON' les macros sont écrits complètement dans le fichier LIST (xxx.LST).
- 6 Ceci est la longueur des mots. Choisissez toujours 'INHX8M' dans le fichier en code machine pour le logiciel PROG PIC2 inclus dans l'emballage du K8048. Tout autre choix résulte en un code machine qui est illisible ou qui ne fonctionne pas.
- 7 Cochez les fichiers à générer outre le code machine HEX.

**Error file:** Les signaux d'erreurs sont saués dans ce fichier.

**List file:** Ce fichier contient votre code source, les signaux d'erreurs et de l'information additionnelle.

Il ne faut pas générer les fichiers CROSS REFERENCE & OBJECT comme ils ne sont pas compatibles avec le K8048.

8. Choisissez le processeur que vous utilisez. Remarque importante : avec certains PIC le numéro est suivi d'un code de génération. P. ex. PIC16F84A.
9. Dans les fichiers texte générés, indiquez la largeur des colonnes en nombre de caractères.

Cliquez sur **<ASSEMBLE>** quand tous les réglages sont à souhait.  
Lorsque le code source est sans erreurs, l'écran suivant sera affiché



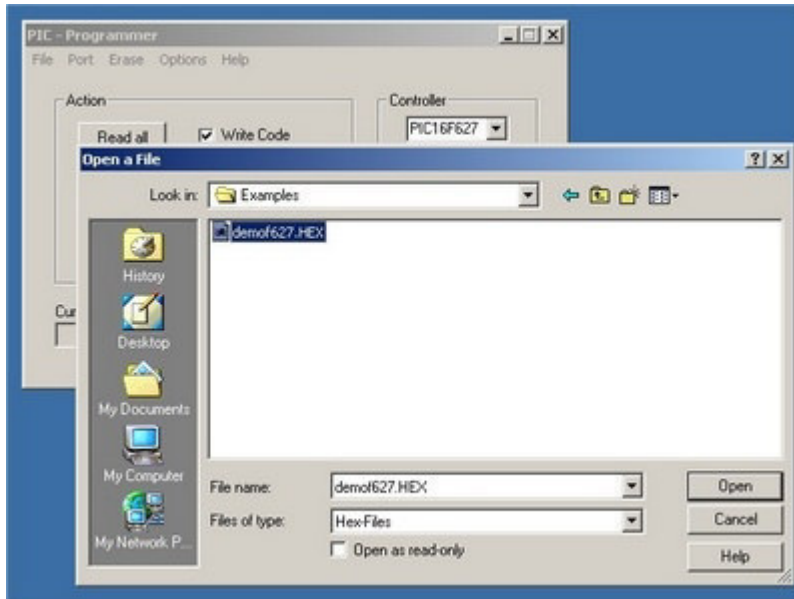


L'assembleur génère alors les fichiers suivants:

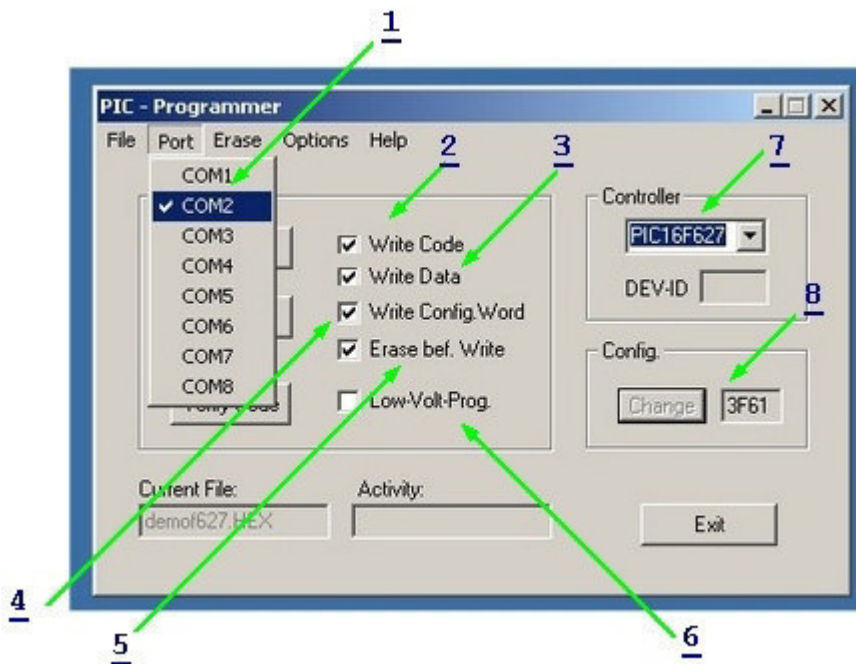
- **xxx.ERR** => fichier 'Error' ou fichier d'erreurs qui contient toutes les erreurs et/ou avertissements que l'assembleur MPASM a trouvé dans votre code source
- **xxx.Hex** => Ceci est le fichier avec le code machine que le logiciel de programmation utilisera pour la programmation du contrôleur).
- **xxx.LST** => Fichier "listing" contenant votre code source et les remarques générées par l'assembleur.
- **xxx.COD** => Fichier Code. Ce fichier est uniquement utilisé par l'environnement de programmation MPLAB et non pas par le logiciel K8048.

## 6.3 Programmation proprement dite du microcontrôleur

- Connectez le K8048 à votre PC via un port sériel disponible.
- La ligne IRQ de ce port sériel (IRQ 3 ou 4) ne peut pas être utilisé par un autre appareil p. ex. un modem interne.
- Branchez un bloc secteur compatible et vérifiez si LD7 s'allume ( LED "ON/OFF").
- Démarrez le programme 'PROGPIC2, PIC programmer' (dans le fichier C:\Velleman\K8048\).
- Cliquez sur 'FILE' -> 'OPEN' et sélectionnez le programme compilé (Hex 8M-file) que vous voulez programmer.



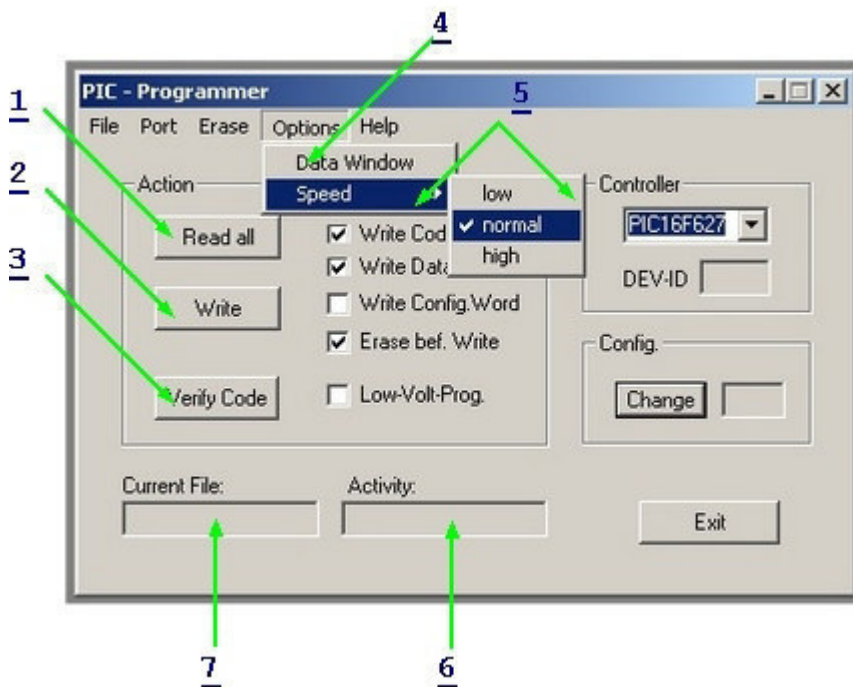
- Contrôlez les réglages du logiciel.



1. Instaurez le numéro du port sériel sur lequel le K8048 est connecté.
2. Sauvegarder CODE: (standard = ON).
3. Sauvegarder données EEPROM (standard = ON).
4. Sauvegarder les réglages de configuration (standard = ON).
5. ERASE before WRITE: (standard = ON). Indique que le contrôleur est effacé avant la programmation.
6. La programmation LOW VOLTAGE ne s'applique pas sur le K8048. Tous les contrôleurs sont programmés avec une tension de programmation (VPP) de 13V et non pas de 5V.
7. Choisissez le microcontrôleur utilisé. Le microcontrôleur inclus dans l'emballage est du type PIC16F627 ou PIC16F627A. Vérifiez si vous avez sélectionné le type correct.
8. Config: D'habitude, la configuration du contrôleur est déjà définie dans le code source avec la directive de compilation `__CONFIG`. Seuls les utilisateurs expérimentés peuvent régler ou ajuster ces options manuellement.

### 6.3.1 Programmation

- Placez SW5 dans la position centrale (standby)
- Insérez le contrôleur dans le support CI approprié:
  - o enfichez le PIC à 8 broches dans support IC1
  - o enfichez le PIC à 14 broches dans support IC2
  - o enfichez le PIC à 18 broches dans support IC3
  - o enfichez le PIC à 28 broches dans support IC4
- Placez ensuite SW5 dans la position 'PROG'. La LED d'avertissement LD9 commence à clignoter, indiquant qu'aucun contrôleur ne peut être connecté ou déconnecté d'un support quand SW5 est réglé dans cette position.
- Cliquez sur WRITE pour achever la programmation. Suivez ce processus dans la boîte 'Activity'. L'indication 'READ/WRITE' s'allume (LED LD8).
- Remettez SW5 dans la position centrale (STANDBY) après la programmation. Alors seulement pouvez-vous enlever le contrôleur ou commuter vers le mode de test (voir boutons de test et LEDs diagnostiques sous 'Expériences'). Mettez SW5 dans la position 'RUN' pour tester le programme (si celui-ci a été écrit pour l'utiliser avec des boutons de test et des LEDs).



1. Pressez ce bouton pour lire un programme et des données d'un contrôleur à condition que le dernier ne soit pas protégé par un code. Des PIC dont le code de protection bit est activé, ne peuvent plus être lus. Au plus, ils peuvent être effacés et alors tout matériel de code est également effacé. Le code machine hexadécimal est sauvé dans un fichier texte.

\* **Attention:** Mettez SW5 dans la position 'PROG' pour utiliser cette fonction !

2. Le contrôleur est programmé. Suivez cette procédure via la boîte 'Activity'. L'indication 'READ/WRITE' s'allume (LD8).
3. Cette touche permet de vérifier si le programme que vous venez de programmer a été mémorisé correctement par un PIC. Attention : il est uniquement possible de contrôler si le contrôleur N'EST PAS protégé par un code. Chaque byte est contrôlé automatiquement pendant la programmation pour voir s'il a été mémorisé correctement. Ce n'est qu'à ce

moment-là qu'est activée la protection du code.

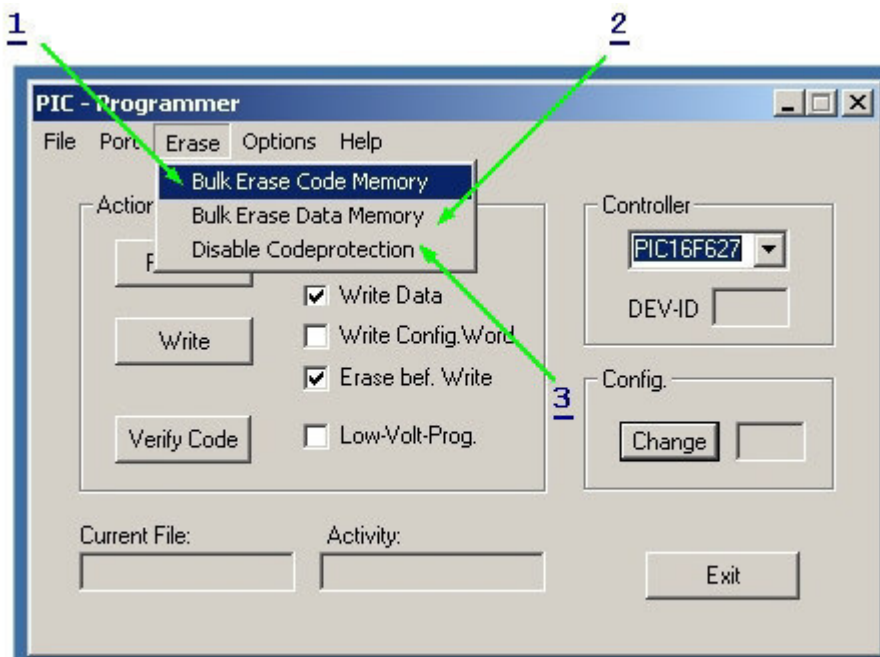
\* **Attention:** Mettez SW5 dans la position 'PROG' pour utiliser cette fonction !

4. Permet de lire et ajuster les données dans la mémoire EEPROM d'un contrôleur. \* **Attention:** Mettez SW5 dans la position 'PROG' pour utiliser cette fonction !
5. Ce réglage reste dans la position 'NORMAL'. En cas de problèmes de programmation, vous pouvez régler la vitesse sur 'HIGH' (pour un PC lent) ou sur 'LOW' (PC très rapide). Ceci permet de compenser la différence de vitesse.
6. Permet de voir quelle adresse ou zone d'adresses le logiciel utilise pour la lecture ou l'écriture.
7. Nom du fichier chargé en mémoire et qui sera utilisé pour la programmation.

## 6.4 Effacer un appareil PIC

### Effacer le contenu d'un contrôleur:

1. Pour l'effacement complet d'un programme dans la mémoire d'un contrôleur.
2. Sert uniquement à effacer les données EEPROM.
3. Peut être utilisé pour désactiver la protection par code. Retenez que le contenu devient illisible et que le programme devient inutilisable.



\***Attention:** Mettez SW5 dans la position 'PROG' pour utiliser cette fonction !

1. Pour l'effacement complet d'un programme dans la mémoire d'un contrôleur.
2. S'utilise uniquement pour l'effacement des données EEPROM.
3. Peut être utilisé pour désactiver la protection par code. Retenez que le contenu devient illisible et que le programme devient inutilisable.

Lisez le fichier C:\Velleman\K8048\Exemples pour des exemples de programmes. Vous pouvez télécharger des mises à jour du logiciel de programmation et des exemples additionnels sur notre site web [www.velleman.be](http://www.velleman.be)

## 7 Expériences

### Emploi des boutons-poussoirs pour expériences et emploi des LEDs:

Le K8048 est équipé d'un max. de 4 entrées de test que vous pouvez opérer manuellement avec SW1, SW2, SW3 et SW4.

Six LEDs diagnostiques permettent de simuler des sorties.

Cliquez sur le lien "cross-reference table" pour vérifier quelle ligne E/S est connectée à quel bouton-poussoir ou quelle LED. Vous pouvez répéter cette procédure pour chaque support CI.

	<b>Support PIC 8 broches</b>	<b>Support PIC 14 broches</b>	<b>Support PIC 18 broches</b>	<b>Support PIC 28 broches</b>
<b>LD1</b>	GP2 (pin 5)	RC0 (pin 10)	RB0 (pin 6)	RA0 (pin 2)
<b>LD2</b>	GP4 (pin 3)	RC1 (pin 9)	RB1 (pin 7)	RA1 (pin 3)
<b>LD3</b>	NC	RC2 (pin 8)	RB2 (pin 8)	RA2 (pin 4)
<b>LD4</b>	NC	RC3 (pin 7)	RB3 (pin 9)	RA3 (pin 5)
<b>LD5</b>	NC	RC4 (pin 6)	RB4 (pin 10)	RA4 (pin 6)
<b>LD6</b>	NC	NC	RB5 (pin 11)	RA5 (pin 7)
<b>SW1</b>	GP5 (pin 2)	RC5 (pin 5)	RA0 (pin 17)	RB0 (pin 21)
<b>SW2</b>	NC	RA2 (pin 11)	RA1 (pin 18)	RB1 (pin 22)
<b>SW3</b>	NC	NC	RA2 (pin 1)	RB2 (pin 23)
<b>SW4</b>	NC	NC	RA3 (pin 2)	RB4 (pin 25)

**Remarques:** Les lignes E/S venant d'un bouton-poussoir, doivent être configurés dans la source en tant qu'entrées (un 1 dans le registre TRIS pour cette entrée). Les lignes E/S connectées à une LED doivent être configurées dans la source en tant qu'entrées (un 0 dans le registre TRIS pour l'entrée en question).

## 8 ICSP

ICSP sert à programmer des contrôleurs "en circuit".

Ce processus est également connu sous le nom de "téléchargement de code", ce qui signifie que la puce en question est connectée à un bus consistant en quatre ou cinq fils, c.-à-d. le bus ICSP. Plusieurs membres de la famille PIC se laissent (re)programmer tout en restant monté sur la plaque d'alimentation. Il ne faut même pas désactiver l'alimentation.

Ces CI sont commutés vers le mode de programmation en mettant haut la broche de réinitialisation "/MCLR", d'habitude avec une tension de 12V à 14V. A partir de ce moment, RB6 sert d'entrée CLOCK tandis que RB7 a changé en connexion d'entrée/de sortie DATA.

