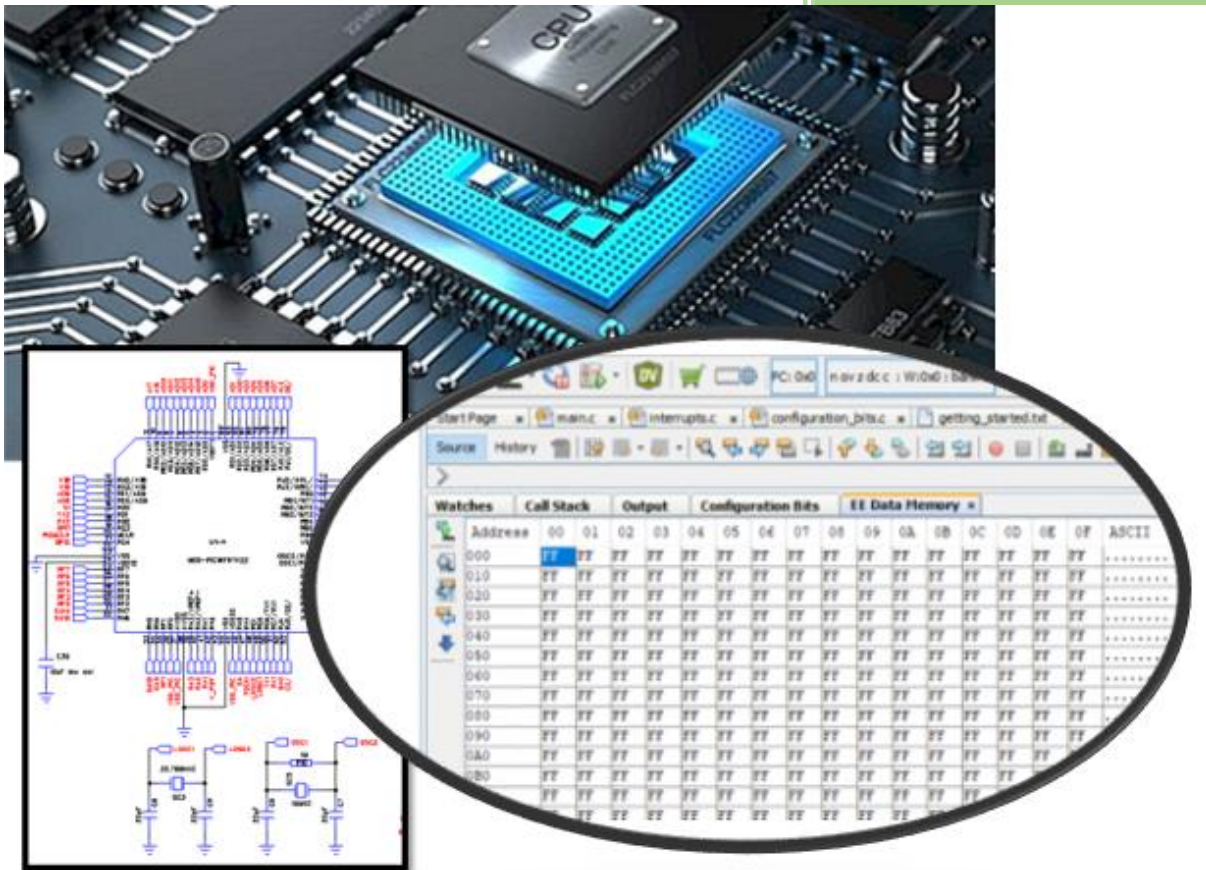


2021

Microcontrôleur EEPROM du PIC18 (Supplément)



ANTINI Nicolas

CESI

01/07/2021



EEROM pour PIC 18

Résumé

La mémoire EEPROM (Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et programmable) (aussi appelée E2PROM ou E²PROM) est un type de mémoire morte. Une mémoire morte est une mémoire utilisée pour enregistrer des informations qui ne doivent pas être perdues lorsque l'appareil qui les contient n'est plus alimenté en électricité

Le contenu de la mémoire EEPROM peut être facilement effacé et réécrit à l'aide d'un courant électrique.

Par exemple, quatre registres sont utilisés pour l'accès à la mémoire EEPROM d'un microcontrôleur PIC :

- *EEDATA contient la donnée.*
- *EEADR contient l'adresse.*
- *EECON1 est le registre de contrôle de l'accès à l'EEPROM. Cinq bits permettent cet accès :*
 - *RD et WR initialisent la lecture ou l'écriture. Ils sont mis à 1 par le programme pour initier l'accès et mis à zéro par le système à la fin de l'accès.*
 - *WREN autorise (1) ou non (0) l'accès en écriture.*
 - *WRERR est mis à 1 par le système quand une opération d'écriture est interrompue par MCLR, reset ou le chien de garde.*
 - *EEIF est un drapeau d'interruption signalant la fin de l'écriture physique dans la mémoire EEPROM. Il doit être mis à 0 par le programme.*
- *EECON2 joue un rôle spécifique lors de l'écriture.*



1 La mémoire de donnée EEPROM du PIC 18F87K22.

Cette mémoire EEPROM de données ne fait pas partie de l'espace adressable « normal » des PIC. Elle n'interfère donc en aucune manière, ni avec la mémoire de programme, ni avec les registres ou mémoire de données classique.

La taille de cette mémoire EEPROM de données peut atteindre 1 024 octets selon les versions de circuits dans lesquels elle est implantée. Son adresse la plus basse part toujours de 00 et son adresse la plus haute est déterminée par sa taille et peut donc atteindre 3FF pour 1 024 octets.

Elle n'est accessible en lecture et en écriture qu'au travers de deux registres :

- Le registre EEDATA pour les données ;
- Le registre EEADR pour les adresses.

Deux registres de contrôle, EECON1 et EECON2, permettent quant à eux 2 de définir le mode de fonctionnement de cette mémoire mais, alors que le registre EECON1 est bien réel, le registre EECON2 n'est pas un registre physique et n'est utilisé que d'une façon particulière lors de l'écriture dans la mémoire.

Cette EEPROM n'utilise aucune source de haute tension externe et fonctionne dans toute la plage de tension d'alimentation autorisée pour le PIC choisi. Sa durée de rétention des informations est annoncée par Microchip comme étant supérieure à 40 ans, soit largement plus que la durée de vie de la majorité des applications actuelles... Elle est contrôlée par un timer interne qui lui est propre, vous déchargeant ainsi de toute mesure de temps par logiciel et n'empruntant aucune ressource aux timers normaux du circuit.

Le fait de vouloir écrire à une adresse déjà occupée efface automatiquement le contenu qui s'y trouve avant d'y placer la nouvelle donnée. De ce fait, aucune commande d'effacement spécifique n'est prévue. Il n'existe pas non plus de procédure d'initialisation suite à une mise sous tension et le contenu de cette mémoire doit donc être considéré comme quelconque tant qu'il n'a pas été défini par votre programme.

1.2 Lecture dans la mémoire EEPROM de données



Nicolas ANTINI 2021:

FISA 25 Langage C bas niveau pour Microcontrôleur

Le principe d'utilisation de cette EEPROM en lecture est fort simple et respecte la chronologie suivante :

- écriture de l'adresse à lire dans le registre EEADR (EEARDH Upper et EEADR Low pour le 18f87K22) ;
- mise à 0 des bits EEPGD et CFGS dans le registre EECON1 (voir ci-après) ;
- mise à 1 du bit RD du registre de contrôle EECON1 (voir ci-après) ;
- lecture de la donnée ainsi adressée dans le registre EEDATA.

```
MOVLW DATA_EE_ADDRH      ;  
MOVWF  EEADRH              ; Upper bits of Data Memory Address to read  
MOVLW DATA_EE_ADDR      ;  
MOVWF  EEADR               ; Lower bits of Data Memory Address to read  
BCF    EECON1, EEPGD       ; Point to DATA memory  
BCF    EECON1, CFGS        ; Access EEPROM  
BSF    EECON1, RD          ; EEPROM Read  
NOP  
MOVF   EEDATA, W           ; W = EEDATA
```

Compte tenu de la vitesse de fonctionnement de cette mémoire, la donnée est disponible dans **EEDATA** dès la mise à 1 du bit RD. Il est donc possible de la lire par l'instruction qui suit immédiatement celle qui met ce bit à 1.

Remarquez que le bit RD n'a pas besoin d'être remis à 0 par logiciel. Cela se produit automatiquement lors de la lecture de la donnée contenue dans **EEDATA**.

Un programme de lecture en Assembleur PIC18 se présente donc de la façon suivante :

1.3 Écriture dans la mémoire EEPROM de données

L'écriture dans cette mémoire, qui est en fait une programmation puisque nous sommes en présence d'une EEPROM, est un peu plus complexe pour d'évidentes raisons de sécurité. Elle doit se dérouler de la façon suivante :

- écriture de l'adresse où vous souhaitez écrire dans le registre EEADR (EEARDH Upper et EEADR Low pour le 18f87K22) ;
- écriture de la donnée dans le registre EEDATA ;
- mise à 0 des bits EEPGD et CFGS dans le registre EECON1 ;
- mise à 1 du bit du bit WREN dans le registre EECON1 ;



- interdiction de toutes les interruptions par mise à 0 du bit GIE/GIEH ;
- exécution du programme type que voici, à l'exclusion de tout autre processus :
- validation des interruptions (si nécessaire).

L'instruction BSF EECON1, WR déclenche le processus d'écriture proprement dit.

Lorsqu'il est terminé :

- le bit WR contenu dans le registre EECON1 revient à 0 ;
- le bit WRERR contenu dans le registre EECON1 est mis à 0 en l'absence d'erreur d'écriture.

Dans les deux cas, et si l'interruption EEPROM a été autorisée via le bit EEIE du registre PIE_x, une interruption est générée. Le bit EEIF doit être remis à 0 par logiciel comme nous l'avons vu au chapitre précédent lors de sa description. Remarquez ici aussi que le bit WR n'a pas besoin d'être remis à 0 par logiciel. Cela se produit automatiquement lorsque l'écriture en mémoire de la donnée est terminée.

Un programme d'écriture se présente donc de la façon suivante :

	MOVLW	DATA_EE_ADDRH	;
	MOVWF	EEADRH	; Upper bits of Data Memory Address to write
	MOVLW	DATA_EE_ADDR	;
	MOVWF	EEADR	; Lower bits of Data Memory Address to write
	MOVLW	DATA_EE_DATA	;
	MOVWF	EEDATA	; Data Memory Value to write
	BCF	EECON1, EEPGD	; Point to DATA memory
	BCF	EECON1, CFGS	; Access EEPROM
	BSF	EECON1, WREN	; Enable writes
	BCF	INTCON, GIE	; Disable Interrupts
Required Sequence	MOVLW	0x55	;
	MOVWF	EECON2	; Write 55h
	MOVLW	0xAA	;
	MOVWF	EECON2	; Write 0AAh
	BSF	EECON1, WR	; Set WR bit to begin write
	BTFSC	EECON1, WR	; Wait for write to complete GOTO \$-2
	BSF	INTCON, GIE	; Enable Interrupts
			; User code execution
	BCF	EECON1, WREN	; Disable writes on write complete (EEIF set)

La procédure qui consiste à écrire 55 puis AA dans le registre EECON2 est une mesure de sécurité mise en place sur la puce afin d'éviter qu'un programme connaissant un moment « d'égarement » ne puisse modifier le contenu de la mémoire EEPROM par accident, en manipulant simplement les bits de EECON1.

Cette procédure pouvait être considérée par le passé comme un luxe presque inutile s'agissant de la mémoire de données, mais elle est aujourd'hui pleinement justifiée puisque c'est aussi celle qui est utilisée pour écrire dans la mémoire de programme et que, dans ce cas, il est indispensable de se protéger des caprices d'un programme pouvant comporter des erreurs.

1.4 Le registre de contrôle EECON1

Le registre de contrôle **EECON1** contient un certain nombre de bits de contrôle dont voici la signification :

Accès								
E : Écriture								
L : Lecture								
	L/E	L/E	L	L/E	L/E	L/E	L/E	L/E
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
État à la mise sous tension	X	X	0	0	X	0	0	0

Bit 0 ou bit RD pour Read Data

Ce bit doit être mis à 1 pour lire une donnée en mémoire EEPROM de données. Il est automatiquement remis à 0 par le circuit lorsque la lecture de la donnée est effective.

Bit 1 ou bit WR pour WRite data

Ce bit doit être mis à 1 pour écrire une donnée, que ce soit en mémoire EEPROM de données ou en mémoire Flash de programme. Il est automatiquement remis à 0 par le circuit lorsque l'écriture est effective.



Bit 2 ou bit WREN pour WRite ENable

Si ce bit est mis à 0 il interdit toute écriture dans l'EEPROM de données et dans la mémoire Flash de programme. Il doit être mis à 1 pour autoriser une écriture.

Bit 3 ou bit WRERR pour WRite ERRor

Ce bit est mis à 1 si une erreur se produit en écriture, que ce soit en mémoire EEPROM de données ou en mémoire Flash de programme, suite à un arrêt prématuré de celle-ci (reset ou action du timer chien de garde par exemple). Dans ces conditions, les contenus de EEDATA et EEADR sont inchangés pour permettre une reprise correcte de l'opération.

Bit 4 ou bit FREE pour Flash Row Erase Enable

Lorsque ce bit est mis à 1, la ligne de la mémoire Flash de programme adressée par le registre TBLPTR est effacée lors de la commande d'écriture qui suit. Si ce bit est laissé à 0, cette commande n'effectue qu'une simple écriture.

Bit 5 Bit inutilisé et lu comme étant à 0.

Bit 6 ou bit CFGS pour Configuration Select

Ce bit permet de sélectionner l'accès à la mémoire EEPROM de données ou à la mémoire Flash de programme lorsqu'il est positionné à 0 et aux registres de configuration dans le cas contraire.

Bit 7 ou bit EEPGD pour EEProm Data

Ce bit permet de sélectionner l'accès à la mémoire EEPROM de données lorsqu'il est positionné à 0 et à la mémoire Flash de programme dans le cas contraire.

Le registre EECON2 quant à lui n'a aucune existence physique et il est donc impossible de le lire. Il ne sert que pour le processus de sécurisation en écriture en utilisant le petit programme présenté ci-dessus.

Malgré la très grande sûreté offerte par ce processus d'écriture, d'autres protections ont été mises en place pour éviter toute altération involontaire du contenu de la mémoire EEPROM de données. Ainsi, lors de la mise sous tension, le bit WREN est mis à 0 interdisant de ce fait toute écriture. De plus, le timer de reset à la mise sous tension inhibe également l'écriture dans l'EEPROM pendant toute la durée de son activation.



1.5 Registres associés à l' EEROM

TABLE 9-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY								
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
EEADRH	EEPROM Address Register High Byte							
EEADR	EEPROM Address Register Low Byte							
EEDATA	EEPROM Data Register							
EECON2	EEPROM Control Register 2 (not a physical register)							
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
IPR6	—	—	—	EEIP	—	CMP3IP	CMP2IP	CMP1IP
PIR6	—	—	—	EEIF	—	CMP3IF	CMP2IF	CMP1IF
PIE6	—	—	—	EEIE	—	CMP3IE	CMP2IE	CMP1IE

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

1.6 Remarque concernant CONFIG5H

Le registre de configuration CONFIG5H est présent dans toutes les versions de PIC 18.

Accès								
E : Écriture								
L : Lecture								
	L/E	L/E	L	L	L	L	L	L
CONFIG5H	CPD	CPB	—	—	—	—	—	—

Bit 7 ou bit CPD pour Code Protect Data

Lorsqu'il est mis à 1, ce bit autorise la lecture de la mémoire EEPROM de données alors qu'il l'interdit dans le cas contraire.

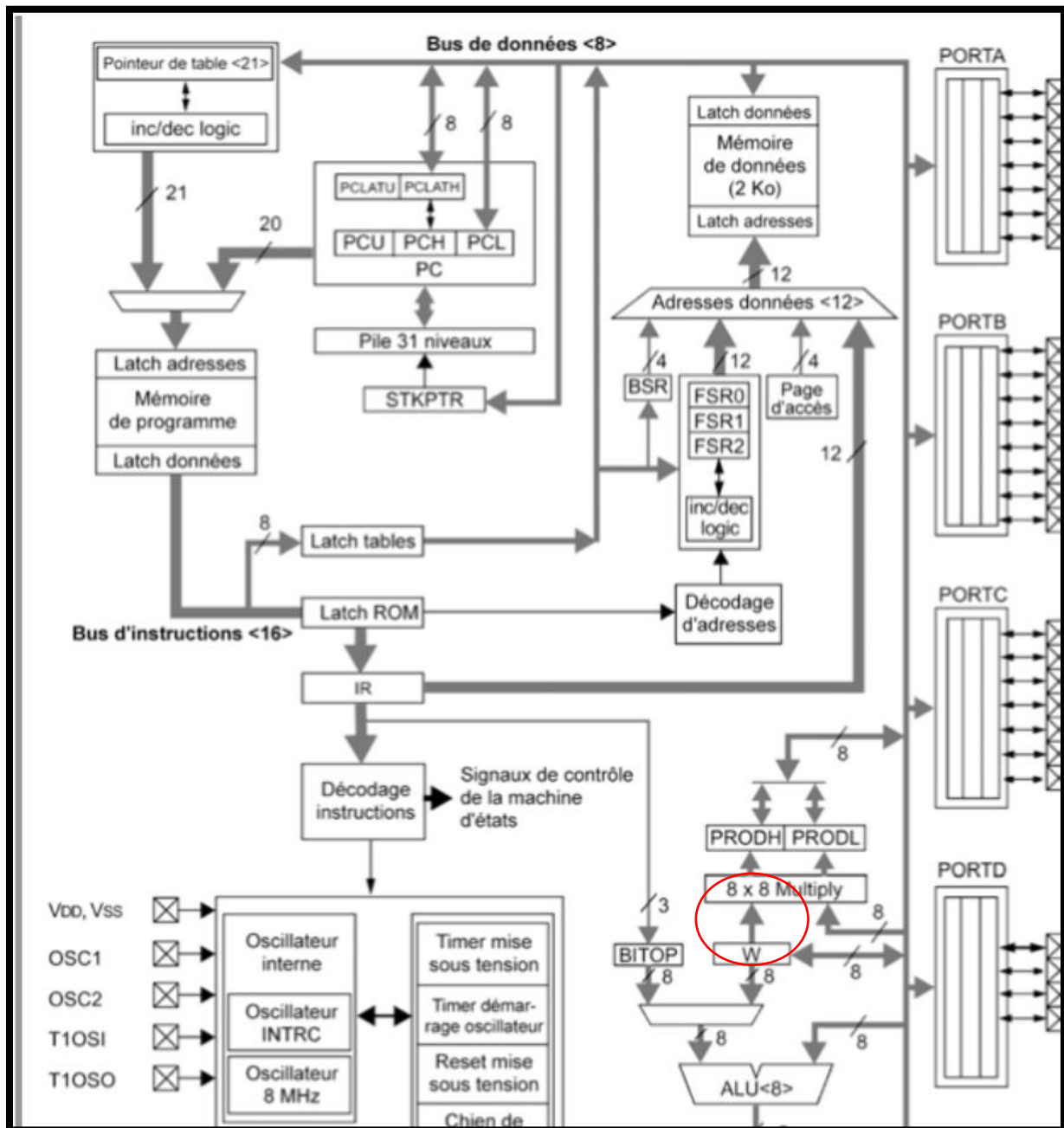
Le registre de configuration CONFIG5L est présent dans toutes les versions de PIC 18 mais le nombre de bits utiles qu'il contient varie en fonction de la taille de mémoire de programme dont dispose le circuit puisque ce registre est dédié à la protection de cette dernière en écriture.

1.7 Définition des instructions Assembleurs (programme Lecture / Ecriture EEPROM).



Rappel Le registre W

Le registre W ou WREG ou registre de travail



W appelé également WREG (8 bits) est un registre de travail.

Comme vous pouvez le constater sur le synoptique de la figure, ce registre occupe une position stratégique puisqu'il est lié directement à l'ALU (unité arithmétique et logique). Il s'appelle donc tout naturellement registre de travail, son sigle signifiant en effet Working REGister.



Ce registre va donc être partie prenante de toutes les instructions arithmétiques et logiques W que vous pourrez « récupérer » le résultat de telle ou telle opération.
Ainsi par exemple, pour remettre à zéro le contenu de ce registre, on écrit en assembleur

CLRF WREG

BCF Bit Clear F

Syntaxe : *BCF f, b*

Opérandes : $0 \leq f \leq 255$

$0 \leq b \leq 7$

Description : met à 0 le bit numéro *b* de *f*.

BSF Bit Set F

Syntaxe : *BSF f, b, a*

Opérandes : $0 \leq f \leq 255$

$0 \leq b \leq 7$

Description : met à 1 le bit numéro *b* de *f*.

MOVLW MOVE Literal to W

Syntaxe : *MOVLW k*

Opérande : $0 \leq k \leq 255$

Opération : $k \rightarrow W$

Description : charge le registre *W* avec le literal *k*.

MOVWF MOVE W to F

Syntaxe : *MOVWF f*

Opérandes : $0 \leq f \leq 255$ Opération : $(W) \rightarrow f$

Description : charge *f* avec le contenu de *W*.

MOVF MOVE F

Syntaxe : *MOVF f, d*

Opérandes : $0 \leq f \leq 255$

$d = 0$ ou 1

Opération : $(f) \rightarrow f$ si $d = 1$ ou $(f) \rightarrow W$ si $d = 0$

Bit d'état : *N, Z*

Description : déplace le contenu de *f* dans *f* si $d = 1$ ou dans *W* si $d = 0$.



Attention ! Le déplacement du contenu de f dans f qui semble a priori inutile permet en fait de tester le contenu de f pour voir s'il est négatif ou nul puisque cette instruction positionne les bits N et Z.

NOP No Operation

Syntaxe: NOP

Opérande : aucun

Cycle : 1

Description : ne fait que consommer du temps machine (un cycle dans ce cas).

BTFSK Bit Test F, Skip if Clear

Syntaxe : BTFSK f, b

Opérandes : $0 \leq f \leq 255$

$0 \leq b \leq 7$

Cycle(s) : 1 si pas de saut, 2 ou 3 si saut selon que l'instruction qui suit est codée sur 1 mot ou 2 mots.

Opération : saut si $b(f) = 0$

Bit d'état : aucun

Description : si le bit numéro b de f est à 0, l'instruction qui suit celle-ci est ignorée et traitée comme un NOP

1.8 Initialisation de l'EEROM en entête de programme

__EEPROM_DATA

Cette fonction stocke les données dans la mémoire EEPROM. Les données doivent être spécifiées en blocs de

8 octets. Un exemple est donné ci-dessous:

`__EEPROM_DATA(0x01,0x03,0x20,0x3A,0x00,0x78,0xAA,0x02);`

