الباب الثامن CHAPTER (8)

الذاكرات الغير متطايرة
Non-Volatile Memories

عرفنا فى الباب الأول أن الـ PIC18F452 Microcontroller يوجد بداخلها نواعان من الذاكرة وهم: الذاكرة المتطايرة Volatile memory والتى تمحى محتوياتها بفصل الكهرباء عن الـميكروكنترولر مثل الـ RAM، حيث تستخدم هذه الذاكرة لتخزين البيانات Data أو المتغيرات Variables. والذاكرة الدائمة Non-volatile ويوجد منها نوعين وهم الـ Flash memory وتستخدم لتخزين البرنامج والـ EEPROM يستخدم لتخزين فى الـ Data المهمة التى لا تريد مسحها عن فصل الكهرباء من الـ Microcontroller مثل الـ Password. ولكن يجب أن تعرف أن الـ Flash memory يمكن أن يستخدم لتخزين البيانات المهمة عن طريق البرنامج ولكن هذه الجزئية خارج نطاق هذا الباب. لذلك فى هذا الباب سوف نتعلم كيفية تخزين البيانات التى نريد الحفاظ عليها فى الـ EEPROM.

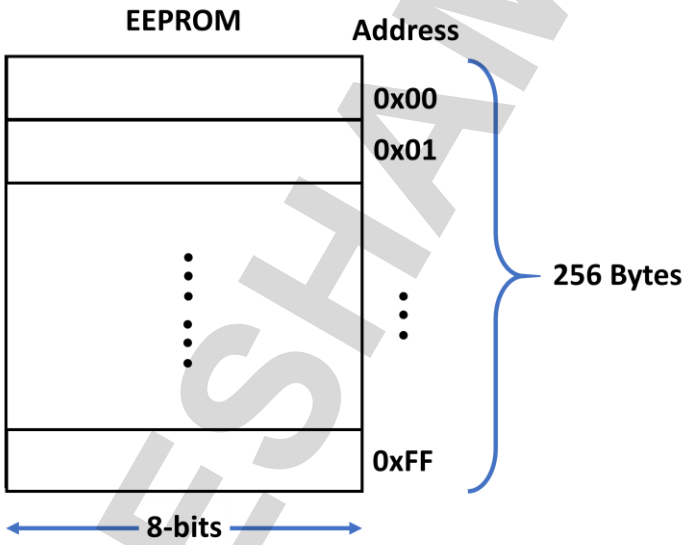
8.1-Reading and Writing Data to and from EEPROM Memory

الغالبية العظمى من عائلة الـ PIC18 تحتوى على EEPROM. وحجم هذه الذاكرة يتراوح من 256 bytes إلى أكثر من 1KB. وحيث أننا سوف نتعامل مع الـ PIC18F452، فإننا يجب أن تعرف أن حجم الـ EEPROM الموجودة 256 bytes كما هو موضح فى شكل (1-7). وبسبب أن الـ EEPROM **<u>أبطأ بكثير من الـ RAM</u>**، فإن الـ CPU لا يستطيع التعامل معه بشكل مباشر إلا عن طريق مجموعة من الـ Registers وهم:

1. **EEADR** register: وهو الـ Register المسئول عن تحديد المكان الذى نريد أن نقرأ أو نسجل عليه فى الـ EERPOM.

2. **EEDATA** register: وهو الـ Register الذى سوف يحتوى على الـ Data التى تريد تخزينها أو التى تم قرائتها من الـ EEPROM.
3. **EECON1** register: يستخدم هذا الـ Register للتحكم بالـ EEPROM والـ Flash memory أيضاً.
4. **EECON2** register: هذا الـ Register وهمى (أى أنه لا يوجد فى الأساس) ولكن يتم إستخدامه عند التعامل مع كلا الذاكرتين (Flash and EEPROM).

قبل أن يتم البدء فى التعامل مع الـ EEPROM يجب أن نتعرف على أولاً الـ EECON1 register وهو المسئول عن التحكم فى القراءة والكتابة على الـ EEPROM والـ Flash memory. وفيما يلى شرح الـ EECON1 register.



شكل (1-8): تركيب الـ EEPROM.

EECON1 Register

EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

شرح الـ EECON1 Register

EEPGD (Flash program or Data EERPOM memory select) bit:

يستخدم هذا الـ Bit لإختيار هل نريد التعامل مع الـ Flash memory أو EEPROM، فإذا كانت EEPGD=1 فهذا يعنى أننا نريد التعامل مع الـ Flash memory وإذا كانت قيمة الـ EEPGD=0 فهذا يعنى أننا نريد التعامل مع الـ EEPROM. وحيث أننا نريد التعامل فى هذه الجزئية مع الـ EEPROM، فإننا **سوف نجعل EEPGD=0**.

CFGS (Flash Program/Data EE or Configuration Select) bit:

يستخدم هذا الـ Bit لإختيار هل نريد الدخول على الـ Configuration Registers المسئولة عن التحكم فى الـ Microcontroller بشكل عام أو الدخول على كلا الذاكرتين (Flash and EEPROM). فإذا كانت قيمة CFGS=1 فهذا يعنى أننا سوف نتعامل مع الـ Configuration registers، وإذا كانت الـ CFGS=0 فهذا يعنى أننا سوف نتعامل مع الـ Flash أو EEPROM. لذلك يجب **جعل CFGS=0** وذلك لفتح إمكانية التعامل مع كلا الذاكرتين.

FREE (Flash Row Erase Enable bit) bit:

يستخدم هذا الـ Bit لمسح أجزاء من الـ Flash memory (لذلك هذا خارج نطاق الشرح هنا).

WRERR (Write Error) bit:

يستخدم هذا الـ Bit للتعرف على حدوث خطأ أثناء عملية الكتابة على كلا الذاكرتين. فإذا كانت قيمة هذا WRERR=1 فهذا يعنى أن عملية الكتابة Writing على أحدى الذاكرتين لم تتم بشكل جيد (قد يحدث نتيجة عمل Reset أو فصل الكهرباء عن الـ Microcontroller أثناء عملية الـ Writing). وإذا كانت قيمة الـ WRERR=0 فهذا يعنى أن عملية الكتابة تمت بشكل جيد.

WREN (Write Enable) bit:

يستخدم هذا الـ Bit للسماح بإمكانية الكتابة على كلا الذاكرتين، فإذا كانت قيمة WREN=1 فهذا يعنى السماح بإمكانية الكتابة على كلا الذاكرتين. وإذا كانت قيمة الـ WREN=0 فهذا يعنى عدم السماح بالكتابة على كلا الذاكرتين.

WR (Write control) bit:

يستخدم هذا الـ Bit لبدء عملية الـ Writing على كلا الذاكرتين. لذلك عندما يتم جعل WR=1 فهذا يعنى أننا نريد بدء عملية الكتابة Writing على إحدى الذاكرتين. وعند إنتهاء عملية الـ Writing، فإن هذا الـ Bit يتحول بشكل ألى إلى صفر.

RD (Read control) bit:

يستخدم هذا الـ Bit لبدء عملية الـ Reading من كلا الذاكرتين. لذلك عندما يتم جعل RD=1 فهذا يعنى أننا نريد بدء عملية الـ Reading من إحدى الذاكرتين. وعند إنتهاء عملية الـ Reading، فإن هذا الـ Bit يتحول بشكل ألى إلى صفر.

8.1.1 Writing Data to EEPROM

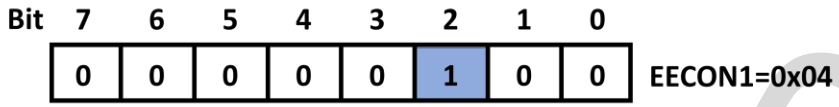
Example (1)

Write a microcontroller program to store the value (44) in location (0) in the EERPROM memory.

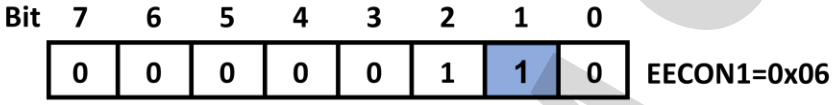
Solution:

فى هذا المثال يريد تسجيل قيمة 44 فى المكان 0 بداخل الـ EEPROM لذلك يجب أن يتم عمل الخطوات الآتية:

1. يتم تخزين العنوان المطلوب فى الـ EEADR.
2. يتم تخزين الـقيمة المطلوبة فى الـ EEDATA.
3. يتم إختيار التعامل مع الـ EEPROM عن طريق (EEPGD=0 و CFGS=0) مع تفعيل خاصية الكتابة على الـ EEPROM عن طريق (WREN=1) وهذه الـ Bits موجودة بداخل الـ EECON1 register وبذلك سوف تكون قيمة الـ EECON1=0x04.



4. يتم تحميل قيمة 0x55 على EECON2.
5. ثم يتم تحميل قيمة 0xAA على EECON2.
6. يتم بدأ الـ Writing على الـ EEPROM عن طريق جعل WR bit=1 والموجود بداخل الـ EECON1.



7. يتم الإنتظار حتى تصبح قيمة WR=0 وهذا يعنى إنتهاء عملية الكتابة على الـ EEPROM.

Program:

```
void main()
{
    EEADR=0;
    EEDATA=44;

    EECON1=0x04;
    EECON2=0x55;
    EECON2=0xAA;
    EECON1=0x06;

    while (EECON1.b1==1);
     while(1)
    {

    }
}
```

1. وضع العنوان المطلوب
2. وضع القيمة المطلوبة.

1. ضبط الـ EECON1
2. وضع 0X55 فى الـ EECON2
3. وضع 0XAA فى الـ EECON2
4. بدء عملية الـ writing.

الإنتظار حتى تنتهى عملية الـ Writing.

Example (2)

Write a microcontroller program to store the values (56, 77, 49, 66, 105) in locations (0 to 4) in the EEPROM memory.

Solution:

Program:

```
void main()
{
    char i;
    char x[]={56, 77, 49, 66, 105 };

    for(i=0;i<=4;i++)
    {
     EEADR=i;
     EEDATA=x[i];

    EECON1=0x04;
    EECON2=0X55;
    EECON2=0XAA;
    EECON1=0X06;

    while (EECON1.b1==1);
     }

    while(1);
     }
}
```

1. إنشاء مخزن للعد.
2. تخزين القيم المطلوبة فى عداد.

1. وضع العنوان المطلوب
2. وضع القيمة المطلوبة من Array.

6. ضبط الـ EECON1
7. وضع 0X55 فى الـ EECON2
8. وضع 0XAA فى الـ EECON2
9. بدء عملية الـ writing.

الإنتظار حتى تنتهى عنلية الـ Writing.

8.1.2-Reading Data From EEPROM

Example (6.3)

Write a microcontroller program to read from the location (0x04) in the EEPROM memory and put the result on the LEDs connected to PORTB pins.

Solution:

فى هذا المثال يريد قراءة القيمة المخزنة فى المخزن الذى عنوانة 0x04 فى الـ EEPROM وإخراجها على الـ LEDs لذلك يجب أن يتم أن يتم عمل الخطوات الآتية:

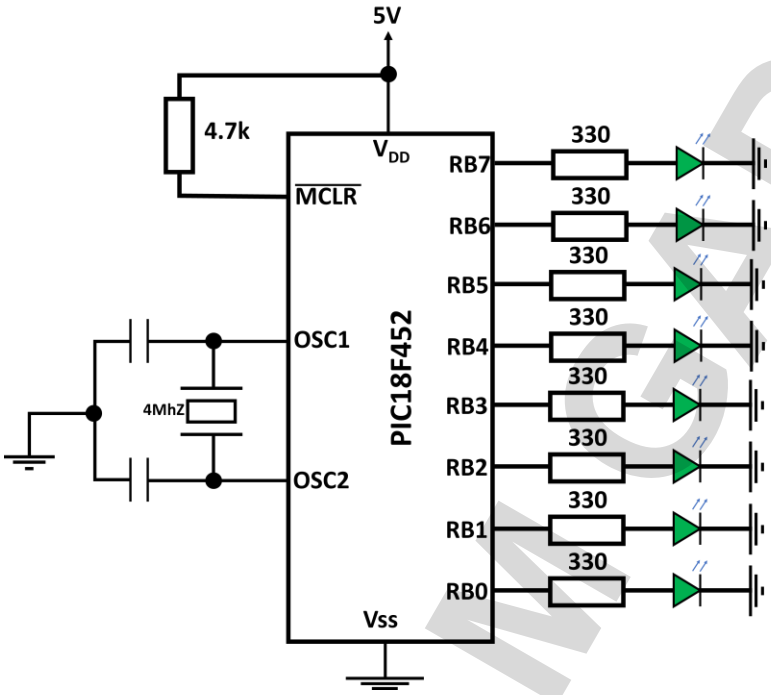
1. يتم تخزين العنوان المطلوب فى الـ EEADR.
2. يتم إختيار التعامل مع الـ EEPROM عن طريق (EEPGD=0 و CFGS=0)

Bit	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	0	EECON1=0x00

3. يتم بدء عملية القراءة عن طريق جعل (RD=1) والموجودة بداخل الـ EECON1 register وبذلك سوف تكون قيمة الـ EECON1=0x01.

Bit	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	1	EECON1=0x01

4. يتم الإنتظار حتى تصبح قيمة الـ RD=0 وهذا يعنى أنه تم الإنتهاء من عملية القراءة.
5. يتم قراءة القيمة من الـ EEDATA وإخراجها على LEDs.



Program:

```
void main()
{
    TRISB=0;
    PORTB=0;

    EEADR=0x04;

    while(1)
    {
        EECON1=0x01;
        while(EECON1.RD==1);
        PORTB=EEDATA;
    }
}
```

وضع العنوان المطلوب

1. ضبط الـ EECON1 مع بدء عملية القراءة.
2. يتم الإنتظار حتى تنتهى عملية القراءة.
3. إخراج القيمة على الـ LEDs