



---

## Section 7. Data EEPROM

---

### HIGHLIGHTS

This section of the manual contains the following major topics:

7.1	Introduction .....	7-2
7.2	Control Register .....	7-3
7.3	EEADR.....	7-4
7.4	EECON1 and EECON2 Registers .....	7-4
7.5	Reading the EEPROM Data Memory .....	7-5
7.6	Writing to the EEPROM Data Memory.....	7-5
7.7	Write Verify .....	7-6
7.8	Protection Against Spurious Writes .....	7-7
7.9	Data EEPROM Operation During Code Protected Configuration .....	7-7
7.10	Initialization .....	7-7
7.11	Design Tips .....	7-8
7.12	Related Application Notes.....	7-9
7.13	Revision History .....	7-10

# PICmicro MID-RANGE MCU FAMILY

---

## 7.1 Introduction

The EEPROM data memory is readable and writable during normal operation (full VDD range). This memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers. There are four SFRs used to read and write this memory. These registers are:

- EECON1
- EECON2 (not a physically implemented register)
- EEDATA
- EEADR

EEDATA holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. The 8-bit EEADR register can access up to 256 locations of Data EEPROM. The EEADR register can be thought of as the indirect addressing register of the Data EEPROM. EECON1 contains the control bits, while EECON2 is the register used to initiate the read/write.

Some devices will implement less than the entire memory map. The address range always starts at 0h, and goes throughout the memory available. [Table 7-1](#) shows some of the possible common device memory sizes and the address range for those sizes.

**Table 7-1: Possible Data EEPROM Memory Sizes**

Data EEPROM Size <sup>(1)</sup>	Address Range
64	0h - 3Fh
128	0h - 7Fh
256	0h - FFh

Note 1: Presently, devices are only offered with 64 bytes of Data EEPROM.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write-time will vary with voltage and temperature as well as from chip to chip. Please refer to the AC specifications for exact limits.

When the device is code protected, the CPU may continue to read and write the data EEPROM memory. The device programmer can no longer access this memory.

# Section 7. Data EEPROM

## 7.2 Control Register

Register 7-1: EECON1 Register

U-0	U-0	U-0	R/W-1	R/W-1	R/W-x	R/S-0	R/S-x
—	—	—	EEIF <sup>(1)</sup>	WRERR	WREN	WR	RD
bit 7			bit 0				

- bit 7:5 **Unimplemented:** Read as '0'
- bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit  
1 = The write operation completed (must be cleared in software)  
0 = The write operation is not complete or has not been started
- bit 3 **WRERR:** EEPROM Error Flag bit  
1 = A write operation is prematurely terminated  
(any MCLR reset or any WDT reset during normal operation)  
0 = The write operation completed
- bit 2 **WREN:** EEPROM Write Enable bit  
1 = Allows write cycles  
0 = Inhibits write to the data EEPROM
- bit 1 **WR:** Write Control bit  
1 = initiates a write cycle. The bit is cleared by hardware once write is complete.  
The WR bit can only be set (not cleared) in software.  
0 = Write cycle to the data EEPROM is complete
- bit 0 **RD:** Read Control bit  
1 = Initiates an EEPROM read. Read takes one cycle. RD is cleared in hardware.  
The RD bit can only be set (not cleared) in software.  
0 = Does not initiate an EEPROM read

Legend

R = Readable bit

W = Writable bit

S = Settable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

**Note 1:** Future devices will have this bit in the PIR register.

## 7.3 EEADR

The EEADR register can address up to a maximum of 256 bytes of data EEPROM.

The unused address bits are decoded. This means that these bits must always be '0' to ensure that the address is in the Data EEPROM memory space.

## 7.4 EECON1 and EECON2 Registers

EECON1 is the control register with five low order bits physically implemented. The upper-three bits are unimplemented and read as '0's.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a  $\overline{\text{MCLR}}$  reset or a WDT time-out reset during normal operation. In these situations, following reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDATA and EEADR registers.

Interrupt flag bit EEIF is set when write is complete. It must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the Data EEPROM write sequence.

## 7.5 Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next instruction cycle, in the EEDATA register; therefore it can be read by the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

### Example 7-1: Data EEPROM Read

```
BCF    STATUS, RP0    ; Bank0
MOVLW  CONFIG_ADDR    ; Any location in Data EEPROM memory space
MOVWF  EEADR           ; Address to read
BSF    STATUS, RP0    ; Bank1
BSF    EECON1, RD      ; EE Read
BCF    STATUS, RP0    ; Bank0
MOVF   EEDATA, W       ; W = EEDATA
```

## 7.6 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte.

### Example 7-2: Data EEPROM Write

<b>Required Sequence</b> {	BSF	STATUS, RP0	;	Bank1
	BCF	INTCON, GIE	;	Disable INTs.
	BSF	EECON1, WREN	;	Enable Write
	MOVLW	55h	;	
	MOVWF	EECON2	;	55h must be written to EECON2
	MOVLW	AAh	;	to start write sequence
	MOVWF	EECON2	;	Write AAh
	BSF	EECON1, WR	;	Set WR bit begin write
	BSF	INTCON, GIE	;	Enable INTs.

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

## 7.7 Write Verify

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM be verified ([Example 7-3](#)) as the value that was intended to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit. The Total Endurance disk will help determine your comfort level.

### Example 7-3: Write Verify

```
BCF    STATUS, RP0 ; Bank0
:      ; Any code can go here
:      ;
MOVWF  EEDATA, W    ; Must be in Bank0
BSF    STATUS, RP0 ; Bank1
READ
BSF    EECON1, RD    ; YES, Read the value written
BCF    STATUS, RP0 ; Bank0
;
; Is the value written (in W reg) and read (in EEDATA) the same?
;
SUBWF  EEDATA, W     ;
BTFSS  STATUS, Z     ; Is difference 0?
GOTO   WRITE_ERR     ; NO, Write error
:      ; YES, Good write
:      ; Continue program
```

## 7.8 Protection Against Spurious Writes

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, WREN is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-outs, power glitches, and software malfunction.

## 7.9 Data EEPROM Operation During Code Protected Configuration

When the device is code protected, the CPU is able to read and write data to the Data EEPROM.

For ROM devices, there are two code protection bits. One for the ROM program memory and one for the Data EEPROM memory. See the Device Programming Specification for more information about these bits.

## 7.10 Initialization

The Data EEPROM module does not have an initialization sequence such as other modules. To do a read of the Data EEPROM refer to [Example 7-1](#). To do a write to the Data EEPROM refer to [Example 7-2](#), and to verify that the write completed successfully refer to [Example 7-3](#).

As for the General Purpose RAM, it is a good idea to initialize all Data EEPROM locations to a known state. This initialization may take place at the time of device programming or an application diagnostic mode, since on reset you may not want the Data EEPROM to be cleared.

An Application Diagnostic mode may be a condition on the I/O pins that the device tests for after the device power-ups. Then depending on this mode, the device would do some diagnostic function. The state for the I/O pins would need to be something that would not be possible without the injected levels to force this diagnostic mode.

## 7.11 Design Tips

**Question 1:** *Why do the data EEPROM locations not contain the data that I wrote?*

**Answer 1:**

There are a few possibilities, but the most likely is that you did not exactly follow the write sequence as shown in [Example 7-2](#). If you are using this code segment ensure that all interrupts are disabled during this sequence.

**Question 2:** *Why is the data in the data EEPROM is getting corrupted?*

**Answer 2:**

The data will only change when a Data EEPROM write occurs. Inadvertent writes may occur when the device is in a brown-out condition (out of operating specification) and the device is not being forced to the reset state. During a brown-out, either the internal brown-out circuitry should be enabled (when available) or external circuitry should be used to reset the PICmicro MCU to ensure that no data EEPROM writes occur when the device is out of the valid operating range.



## 7.12 Related Application Notes

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to data EEPROM are:

Title	Application Note #
EEPROM Endurance Tutorial	AN601
How to get 10 Million Cycles out of your Microchip Serial EEPROM	AN602
Basic Serial EEPROM Operation	AN536
Everything a System Engineer needs to know about Serial EEPROM Endurance	AN537
Using the Microchip Endurance Predictive Software	AN562

# PICmicro MID-RANGE MCU FAMILY

---

## 7.13 Revision History

### Revision A

This is the initial released revision of the Data EEPROM description.