

# 3. EEPROM

EEPROM is a small non-volatile memory inside the microcontroller.

Unlike SRAM:

- SRAM loses data when power is removed
- EEPROM keeps data even after reset or power loss

EEPROM is commonly used to store:

- Settings
- Calibration values
- System state

EEPROM has limited write endurance, so it should only be used sparingly

---

## EEPROM Registers

EEPROM operations are controlled using dedicated I/O registers.

---

### EEARH:EEARL — EEPROM Address Register

Stores the memory address to be accessed.

---

### EEDR — EEPROM Data Register

Stores the data to be written or read.

---

### EECR — EEPROM Control Register

Controls read and write operations.

Important control bits:

Name	Bit	Function
------	-----	----------

EERE	0	EEPROM Read Enable
EEWE	1	EEPROM Write Enable
EEMWE	2	EEPROM Host Write Enable

# EEPROM Write Operation

Writing EEPROM requires a specific sequence:

1. Wait until EEWE becomes '0'.
2. Write new EEPROM address to EEAR (optional).
3. Write new EEPROM data to EEDR (optional).
4. Write a logical '1' to the EEMWE bit while writing a '0' to EEWE in EECR.
5. Within four clock cycles after setting EEMWE, write a logical '1' to EEWE.

```
; Writes R21 to EEPROM address in 0x0052
EEPROM_write:
    SBIC  EECR, 1
    RJMP  EEPROM_write

    LDI  R22, hi8(0x0052)
    OUT  EEARH, R22
    LDI  R22, lo8(0x0052)
    OUT  EEARL, R22

    OUT  EEDR, R21

    SBI  EECR, 2
    SBI  EECR, 1
    RET
```

# EEPROM Read Operation

Reading EEPROM is simpler:

1. Load address into EEAR
2. Start read operation by setting EERE
3. Read data from EEDR

```
; Read R21 from EEPROM address in 0x0052
```

```
EEPROM_read:
```

```
SBIC  EECR, 1
```

```
RJMP  EEPROM_read
```

```
LDI  R22, hi8(0x0052)
```

```
OUT  EEARH, R22
```

```
LDI  R22, lo8(0x0052)
```

```
OUT  EEARL, R22
```

```
SBI  EECR, 0
```

```
IN   R21, EEDR
```

```
RET
```

---

Revision #1

Created 2026-04-06 01:44:48 UTC by MF

Updated 2026-04-06 01:53:02 UTC by MF