



Vores Arduino Uno Controller ATMEGA328P har indbygget 1K EEPROM.

EEPROM, står for **E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly-**M**emory (Elektrisk sletbar programmerbar læsehukommelse).

Det er en type hukommelse, der beholder sit indhold ved poweroff. Kaldes også Non Volatile Hukommelse, eller Permanent Memory.

Data kan kun skrives og slettes 1 byte ad gangen i en EEPROM, og er derfor er en del langsommere. En EEPROM write-funktion tager 3.3 ms.

Hver hylde – eller adresse kan indeholde 8 bit.

Der skal adderes et bibliotek for at få adgang til at skrive / læse i EEPROMén.

Et eksempel fra Arduinos hjemmeside:

```
/*
 * EEPROM Write
 *
 * Stores values read from analog input 0 into the EEPROM.
 * These values will stay in the EEPROM when the board is
 * turned off and may be retrieved later by another sketch.
 */

#include <EEPROM.h>

int addr = 0;          /** the current address in the EEPROM **/

void setup() {
  /** Empty setup. **/
}

void loop() {

  int val = analogRead(0) / 4; /** kun 8 bit i EEPROM !! **/

  EEPROM.write(addr, val); // Skriv værdi til adresse

  addr = addr + 1;         // Inc adressepointer

  if (addr == EEPROM.length()) { // Tjeck for sidste adresse
    addr = 0;
  }

  delay(100);
}
```



Læsning sker med ordren:

```
value = EEPROM.read(address);
```

Slet EEPROM-data – eksempel:

```
void clearEEPROM()
{
    for (int i = 0 ; i < EEPROM.length() ; i++) {
        if(EEPROM.read(i) != 0) //skip already "empty"
        addresses
        {
            EEPROM.write(i, 0); //write 0 to address i
        }
    }
    Serial.println("EEPROM erased");
    address = 0; //reset address counter
}

/*
  based on Blink without Delay
  uses EEPROM to store blink values
*/

#include <EEPROM.h>

// these values are saved in EEPROM
const byte EEPROM_ID = 0x99; // used to identify if valid data in EEPROM
byte ledPin = 13; // the number of the LED pin
int interval = 1000; // interval at which to blink (milliseconds)

// variables that do not need to be saved
int ledState = LOW; // ledState used to set the LED
long previousMillis = 0; // will store last time LED was updated

//constants used to identify EEPROM addresses
const int ID_ADDR = 0; // the EEPROM address used to store the ID
const int PIN_ADDR = 1; // the EEPROM address used to store the pin
const int INTERVAL_ADDR = 2; // the EEPROM address used to store the interval

void setup()
{
    Serial.begin(9600);
    byte id = EEPROM.read(ID_ADDR); // read the first byte from the EEPROM
    if( id == EEPROM_ID)
    {
        // here if the id value read matches the value saved when writing eeprom
        Serial.println("Using data from EEPROM");
        ledPin = EEPROM.read(PIN_ADDR);
        byte hiByte = EEPROM.read(INTERVAL_ADDR);
        byte lowByte = EEPROM.read(INTERVAL_ADDR+1);
    }
}
```



```
    interval = word(hiByte, lowByte); // see word function in Recipe 3.15
}
else
{
    // here if the ID is not found, so write the default data
    Serial.println("Writing default data to EEPROM");
    EEPROM.write(ID_ADDR, EEPROM_ID); // write the ID to indicate valid data
    EEPROM.write(PIN_ADDR, ledPin); // save the pin in eeprom
    byte hiByte = highByte(interval);
    byte loByte = lowByte(interval);
    EEPROM.write(INTERVAL_ADDR, hiByte);
    EEPROM.write(INTERVAL_ADDR+1, loByte);

}
Serial.print("Setting pin to ");
Serial.println(ledPin, DEC);
Serial.print("Setting interval to ");
Serial.println(interval);

pinMode(ledPin, OUTPUT);
}

void loop()
{
    // this is the same code as the BlinkWithoutDelay example sketch
    if (millis() - previousMillis > interval)
    {
        previousMillis = millis(); // save the last time you blinked the LED
        // if the LED is off turn it on and vice versa:
        if (ledState == LOW)
            ledState = HIGH;
        else
            ledState = LOW;
        digitalWrite(ledPin, ledState); // set LED using value of ledState
    }
    processSerial();
}

// function to get duration or pin values from Serial Monitor
// value followed by i is interval, p is pin number
int value = 0;

void processSerial()
{
    if( Serial.available())
    {
        char ch = Serial.read();
        if(ch >= '0' && ch <= '9') // is this an ascii digit between 0 and 9?
        {
            value = (value * 10) + (ch - '0'); // yes, accumulate the value
        }
        else if (ch == 'i') // is this the interval
        {
            interval = value;
            Serial.print("Setting interval to ");
            Serial.println(interval);
            byte hiByte = highByte(interval);
            byte loByte = lowByte(interval);
            EEPROM.write(INTERVAL_ADDR, hiByte);
        }
    }
}
```



```
EEPROM.write(INTERVAL_ADDR+1, loByte);  
value = 0; // reset to 0 ready for the next sequence of digits  
}  
else if (ch == 'p') // is this the pin number  
{  
    ledPin = value;  
    Serial.print("Setting pin to ");  
    Serial.println(ledPin, DEC);  
    pinMode(ledPin, OUTPUT);  
    EEPROM.write(PIN_ADDR, ledPin); // save the pin in eeprom  
    value = 0; // reset to 0 ready for the next sequence of digits  
}  
}  
}
```

Extended Lib: Se <http://thijs.elenbaas.net/2012/07/extended-EEPROM-library-for-arduino/>