

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs  
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

## **Arduino – cum accesez o memorie EEPROM externa ?**

### **Ce sunt memoriile EEPROM externe ?**

Termenul EEPROM înseamnă Electrically Erasable Programmable Read-Only Memory, adică acest termen se referă la memoriile nevolatile care pot fi citite/scrise și rescrise de nenumărate ori.

Memoriile nevolatile sunt cele care își păstrează conținutul, chiar și după ce ai întrerupt alimentarea. Spre exemplu, un alt dispozitiv care își păstrează datele este hard disk-ul. Chiar dacă ai oprit calculatorul, hard disk-ul păstrează toate informațiile intacte. Ele nu se șterg și pot fi accesate din nou la următoarea pornire a calculatorului.

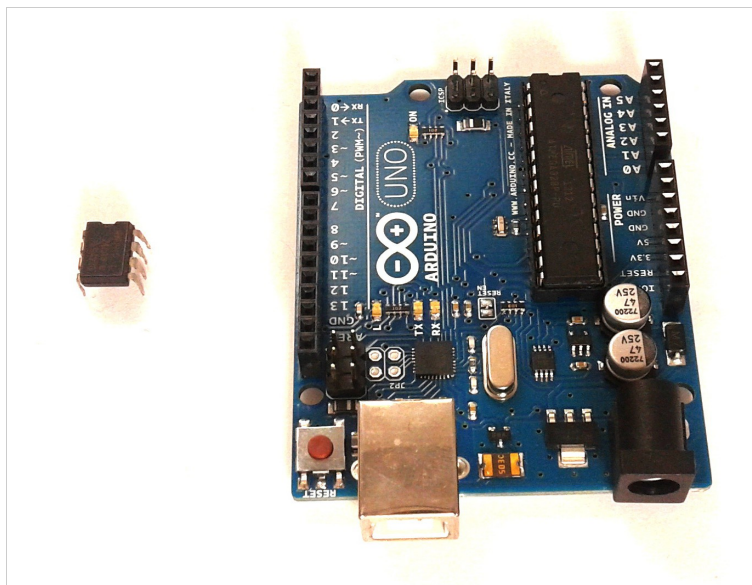
În aceeași manieră funcționează și o memorie EEPROM. Poți să salvezi date în ea și să le citești ori de câte ori vrei tu, chiar și după ce ai întrerupt alimentarea.

### **În ce condiții poți să folosești o memorie EEPROM ?**

În primul rând, memoriile EEPROM nu sunt destinate pentru a salva blocuri uriase de informații (asa cum face un hard disk), dar poți folosi o memorie EEPROM pentru a salva o cantitate minimă de informații, adică cel mult câțiva MB.

În al doilea rând, memoriile sunt simple de utilizat și sunt produse în capsule de circuite integrate, adică poți să le adaugi foarte ușor și foarte rapid în proiectul tău.

Alt avantaj al memoriilor EEPROM este că mulțumită interfeței de comunicare cu placa Arduino, poți să folosești nu doar una, ci mai multe memorii înlanțuite pe o magistrală I2C sau SPI.



Dar totusi in ce situatii poti sa folosesti aceste memorii?

Iata un data logger ce se foloseste de memoria EEPROM pentru a salva un numar mare de esantioane cu valori de temperatura:

<http://forum.arduino.cc/index.php/topic,111366.0.html>

Un alt proiect asemanator celui de mai sus este un data logger care inregistreaza viteza de deplasare a unei biciclete:

<http://www.eightlines.com/blog/2008/09/data-logging-to-the-arduino-eeeprom/>

Daca nu stiai afla ca si placa Arduino, mai exact microcontroller-ul placii, are propria memorie EEPROM pe care poti salva pana la 4096 de octeti (4KB). Poti afla mai multe detalii despre memoria interna accesand urmatoarele link-uri:

[http://arduino.cc/en/Reference/EEPROM#.UySH7Pl\\_uSo](http://arduino.cc/en/Reference/EEPROM#.UySH7Pl_uSo)

[http://arduino.cc/en/Tutorial/EEPROMWrite#.UySIYPl\\_uSo](http://arduino.cc/en/Tutorial/EEPROMWrite#.UySIYPl_uSo)

[http://arduino.cc/en/Tutorial/EEPROMRead#.UySIePl\\_uSo](http://arduino.cc/en/Tutorial/EEPROMRead#.UySIePl_uSo)

<http://www.robofun.ro/forum>

## Cum conectez o memorie EEPROM la o placa Arduino ?

In acest tutorial s-a folosit o memorie EEPROM AT24C02, a carei fisa de catalog se poate descarca de la urmatoarea adresa:

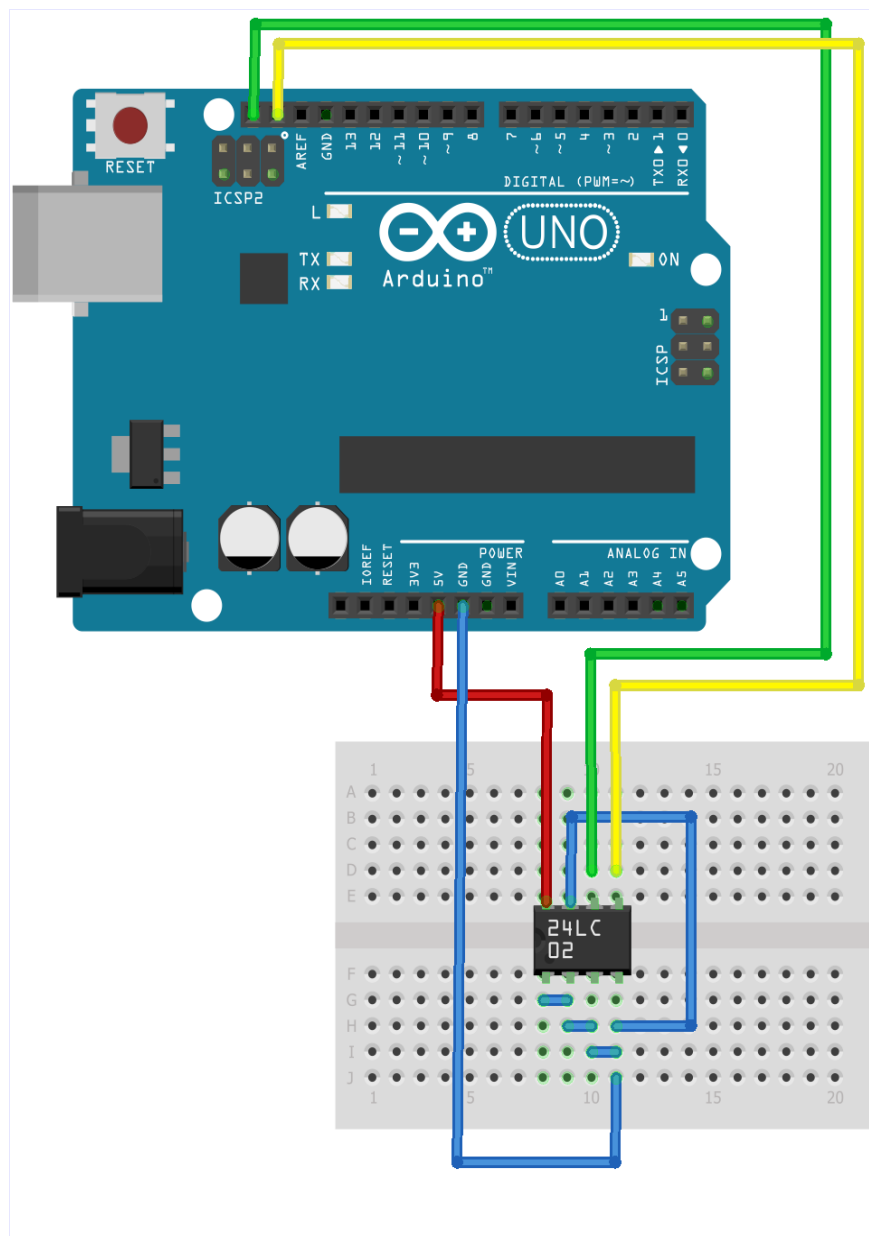
<http://www.atmel.com/Images/doc3256.pdf>

Spre exemplu, din fisa de catalog se pot afla urmatoarele specificatii ale memoriei:

- Tensiunea de alimentare: 5 V sau 2.7 V.
- Organizarea interna a memoriei care iti va fi de folos atunci cand o adresezi.
- Tipul interfetei seriale (memoria de fata necesita o interfata seriala pe 2 fire)
- Viteza de scriere a datelor in functie de tensiunea de alimentare.
- Numarul de scrieri posibile (memoria de fata suporta 1 milion de cicluri de scrieri).
- Configuratia pinilor (pinii de adresare, pinii interfetei seriale, pinii de alimentare si pinul de citire/scriere).

Dar totusi cum conectez memoria la placa Arduino ? Pentru a realiza acest lucru vei avea nevoie de o placa breadboard, fire de conexiune tata-tata si memoria in sine.

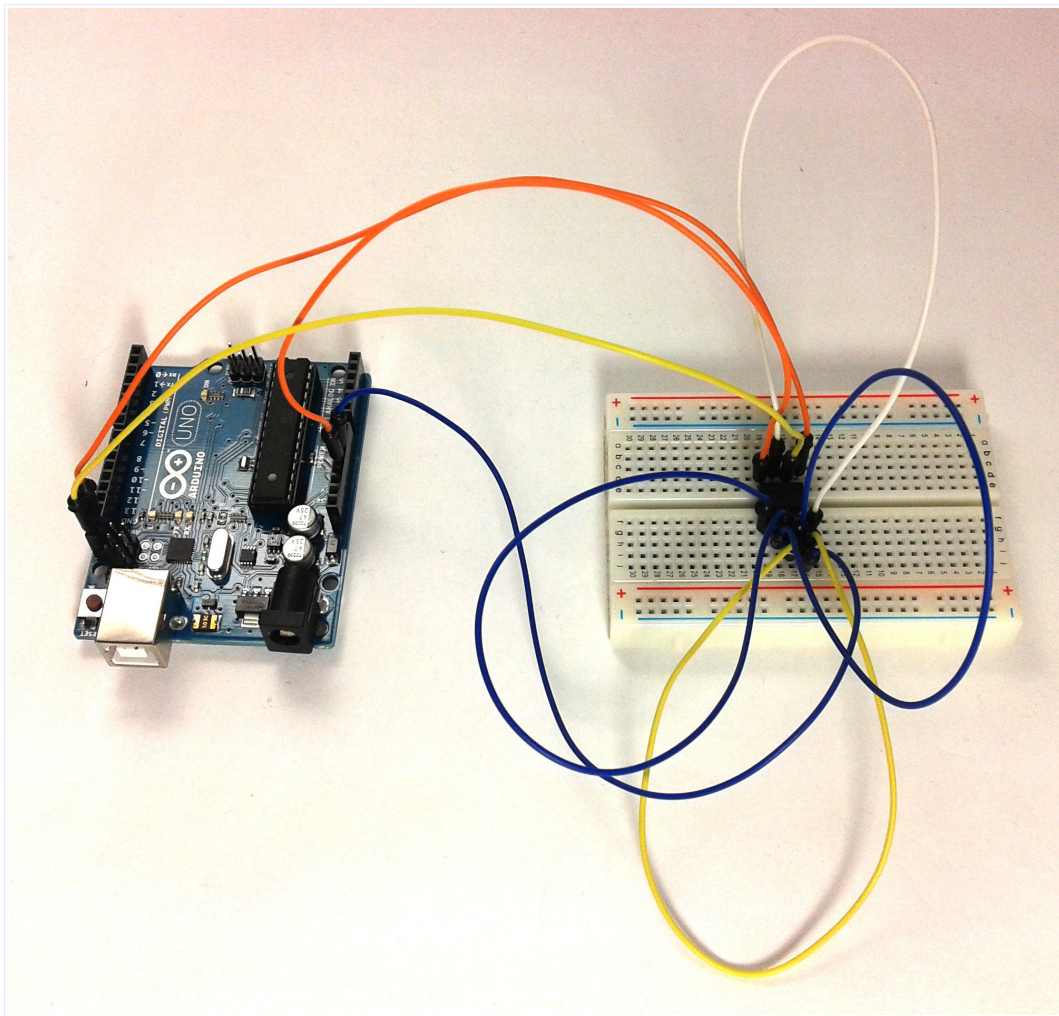
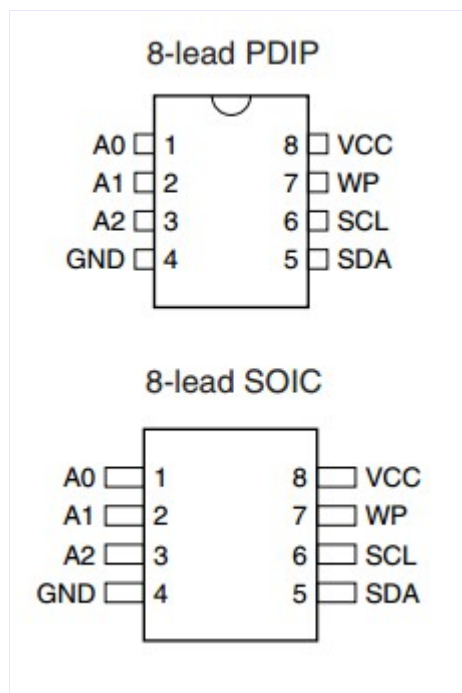
Conexiunea placii Arduino cu memoria EEPROM se va realiza dupa urmatoarea schema:



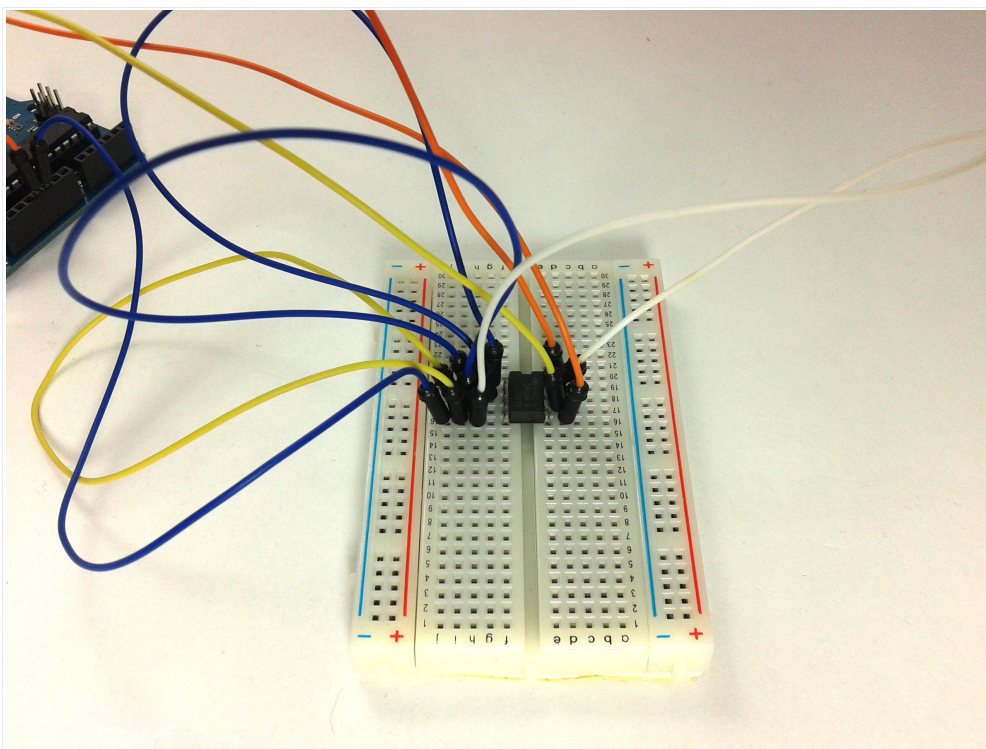
Mai exact vei conecta pinii placii Arduino cu memoria EEPROM dupa urmatorul tabel:

<b>Arduino 5V</b>	<b>Memorie EEPROM VCC</b>
<b>Arduino GND</b>	<b>Memorie EEPROM GND, A0, A1, A2, WP</b>
<b>Arduino SCL</b>	<b>Memorie EEPROM SCL</b>
<b>Arduino SDA</b>	<b>Memorie EEPROM SDA</b>

Pinii memoriei se pot afla direct din fisa de catalog (link-ul de mai sus) sau din urmatoarea diagrama:







Ce reprezinta pinii marcati cu A0, A1, A2 si WP ?

Am spus mai devreme ca o placa Arduino poate adresa mai multe memorii, iar pinii A0, A1 si A2 te pot ajuta sa inlantuiesti pe magistrala seriala pana la 8 memorii EEPROM. In cazul in care doresti sa folosesti o singura memorie, tot ce trebuie sa faci este sa conectezi pinii la GND, adica toti pinii sa capete valoarea logica „0“. Daca vrei sa adaugi o noua memorie, tot ce trebuie sa faci este sa alegi o configuratie diferita pentru noua memorie, adica sa conectezi pinii A0 si A1 la GND iar pinul ramas, A2, la VCC. Pentru o a treia memorie trebuie sa alegi o noua configuratie, diferita fata de celelalte doua. In final poti adauga pana la 8 memorii.

In cazul de fata, placa Arduino va adresa o singura memorie EEPROM, dar tie iti ramane sa descoperi cum se poate adresa o a doua memorie si chiar mai multe.

Ramane pinul WP care inseamna „Write Protect“. Acest pin iti permite sa protejezi datele de pe memoria EEPROM dupa anumite reguli pe care le poti descoperi in fisa de catalog.

In cazul de fata, pinul WP este conectat la GND si asta inseamna ca esti liber sa scrii si sa citesti memoria ori de cate ori vrei tu, adica functia de protectie la scriere este dezactivata.

## Cum citesc/scriu memoria EEPROM din Arduino ?

Iata 2 functii simple care iti permit sa scrii si sa citesti in/din memoria EEPROM:

```

void writeEEPROM(int deviceaddress, unsigned int eeaddress, byte
data )
{
    Wire.beginTransmission(deviceaddress);
    Wire.write((int)eeaddress);
    Wire.write(data);
    Wire.endTransmission();

    delay(5);
}

byte readEEPROM(int deviceaddress, unsigned int eeaddress )
{
    byte rdata = 0xFF;

    Wire.beginTransmission(deviceaddress);
    Wire.write((int)eeaddress);
    Wire.endTransmission();
    Wire.requestFrom(deviceaddress,1);

    if (Wire.available()) rdata = Wire.read();

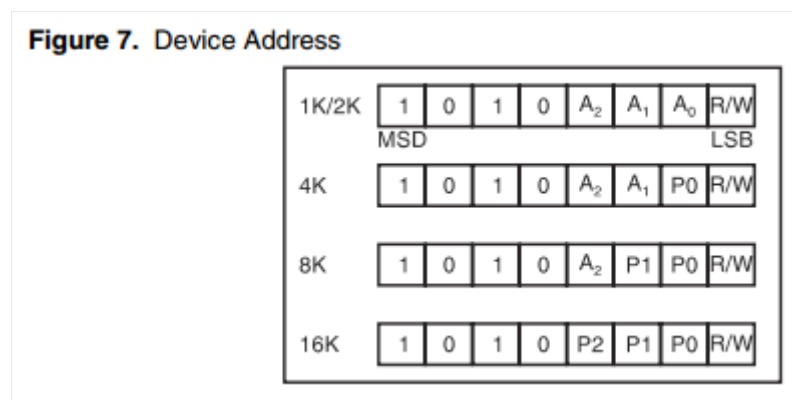
    return rdata;
}

```

Cum functioneaza functia de scriere ? La prima vedere, functia pare extrem de simpla deoarece este formata din 5 instructiuni. La apelare functia accepta 3 parametrii, dintre care primul reprezinta adresa fizica a memoriei EEPROM, al doilea parametru reprezinta adresa locatiei din memorie la care vrei sa faci scrierea, iar al treilea parametru reprezinta informatia pe care doresti sa o salvezi. Simplu ?

Cum stabilesc adresa fizica a memoriei ? In primul rand, mai devreme am vorbit despre pinii A0, A1 si A2 pe care i-ai conectat la GND, adica le-ai dat tuturor valoarea logica „0“.

Daca te uiti in fisa de catalog a memoriei vei intalni urmatoarea imagine:

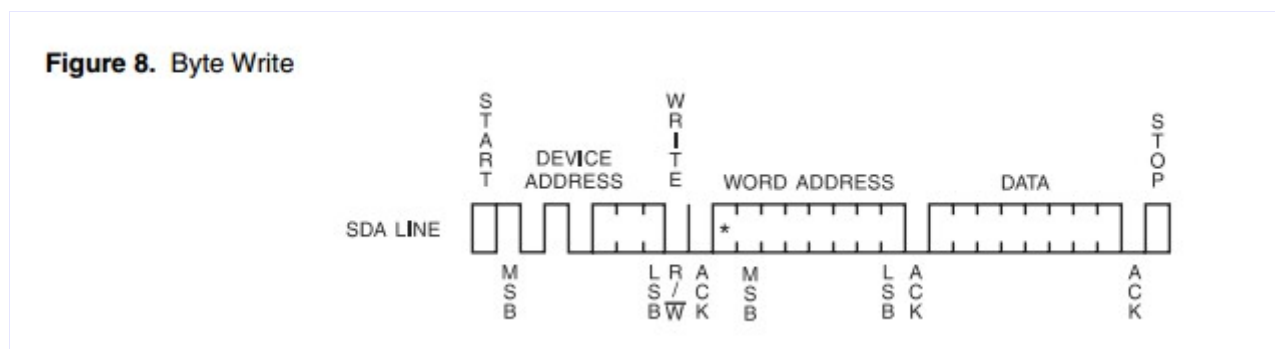




Pentru memoria de fata este valabila doar prima linie marcata cu 1K/2K in care apar bitii A2, A1 si A0. Daca acesti biti sunt conectati la GND, inseamna ca toti bitii capata valoarea „0”.

Daca realizezi conversia din binar in hexazecimal a urmatorului sir de biti „1010000”, vei descoperi adresa hardware pe care o poti folosi in functia de mai sus, respectiv 0x50.

Scrierea propriu-zisa a informatiei in memoria EEPROM se realizeaza dupa urmatoarea diagrama (pe care o poti descoperi tot din fisa de catalog). Iti ramane ca tu sa descoperi cum lucreaza instructiunile din interiorul functiei de scriere, asa cum indica si diagrama de mai jos.



Cum functioneaza functia de citire ? Diferenta intre functia de scriere si cea de citire este ca cea din urma returneaza ceva dintr-o locatie de memorie. Asta inseamna ca la apelare, functia accepta 2 parametrii, respectiv adresa hardware a memoriei (descoperita mai sus) si locatia din memorie. Instructiunile de citire sunt asemanatoare functiei de scriere, dar exista ceva in plus.

Tot ce trebuie sa faci este sa cauti diagrama de citire din fisa de catalog si sa corelezi instructiunile.

Am functiile de scriere si citire, dar cum le apelez ?

Iata un exemplu:

```
#include <Wire.h>
#define disk1 0x50    //adresa hardware a memoriei

void setup(void)
{
    Serial.begin(9600);
    Wire.begin();
    unsigned int address = 0;
    writeEEPROM(disk1, address, 123);
    Serial.println(readEEPROM(disk1, address), DEC);
}

void loop() {
}
```

Prima linie de cod, directiva preprocesor, apeleaza libraria Wire. Libraria este responsabila cu protocolul de comunicatie intre placa Arduino si memoria EEPROM.

Am mentionat mai sus ca fiecare memorie poate capata o adresa hardware (prin configurarea pinilor A0, A1 si A2). In codul de mai sus, adresa poarta numele disk1 si are valoarea hexazecimala 0x50.

In rutina setup() se initializeaza Monitorul Serial, conexiunea seriala cu memoria EEPROM, adresa locatiei la care vrei sa faci o scriere si apoi o citire.

Dupa care se apeleaza functia de scriere si dupa cum observi se scrie valoarea 123 in locatia 0, din memoria EEPROM cu adresa hardware 0x50.

Apoi se citeste si se afiseaza din memoria EEPROM, valoarea scrisa anterior prin functia: Serial.println(readEEPROM(disk1, address), DEC);

Ce altceva poti sa faci cu o memorie EEPROM ?

Pe langa a salva si a scrie date, poti spre exemplu sa salvezi un fisier wav pentru a il folosi mai tarziu la redare. Iata un exemplu:

<http://www.enide.net/webcms/index.php?page=pcm2pwm-multiplayer>