

Lucrarea 10.

Studiarea facilitatilor senzoriale, de calcul si de comunicatie ale unei placi de tip Arduino

1. Obiectivul lucrarii

Lucrarea isi propune sa prezinte facilitatile de calcul, senzoriale si de comunicatie ale unei placi de dezvoltare de tip Arduino.

2. Consideratii teoretice

Platforma Arduino este un microsystem de calcul dedicat, cu arhitectura deschisa (nu este proprietatea unei firme) care s-a dezvoltat cu scopul de a permite unor persoane mai putin avizate in domeniul calculatoarelor (in speta al arhitecturilor hardware) sa programeze aplicatii simple de monitorizare si control.

O placa de tip Arduino contine de obicei un microcontrolor (din familia ARM) si o serie de interfete pentru achizitia si generarea de semnale analogice si digitale. In plus contine 1-2 canale seriale de comunicatie si eventual o interfata de retea.

Firma Intel (c) a dezvoltat o placa de tip Arduino denumita Intel Galileo care are ca si element central un circuit de tip SoC (system-on-chip) Intel Quark x1000. Acest circuit este de fapt un intreg calculator de tip PC integrat pe un singur circuit; foloseste un procesor compatibil cu familia ISAx86 care lucreaza la o frecventa de 400MHz. In figura 1 se observa componentele fizice ale placii Arduino Intel Galileo.

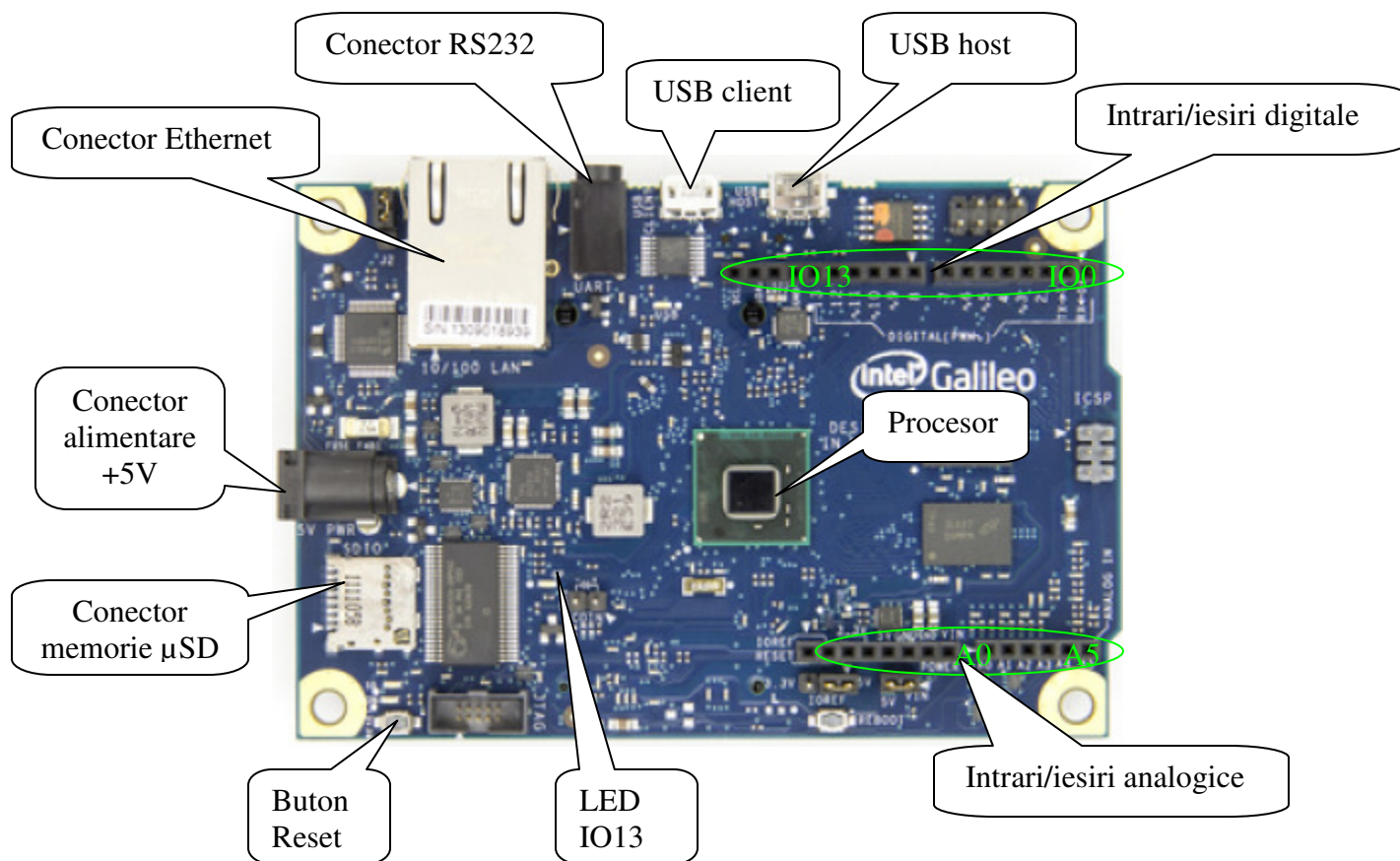


Figura 1. Placa Arduino Intel Galileo

Figura 2 prezinta schema de conectare a componentelor placii Arduino Galileo. Se observa ca intrarile si iesirile digitale si analogice nu sunt conectate direct la microprocesor (cum ar fi cazul la o placa Arduino obisnuita, bazata pe microcontrolor) ci sunt controlate de niste circuite specializate de tip GPIO. Dialogul dintre microprocesor si circuitele GPIO se realizeaza pe un canal serial. Din aceasta cauza frecventa de comutare a semnalelor de iesire este mult mai mica decat in cazul unor placii Arduino obisnuite. Frecventa maxima de comutare a unui singur semnal digital este de 230 Hz. In cazul in care programul incerca sa comute mai multe semnale, frecventa de comutare scade invers proportional cu numarul de iesiri comutate.

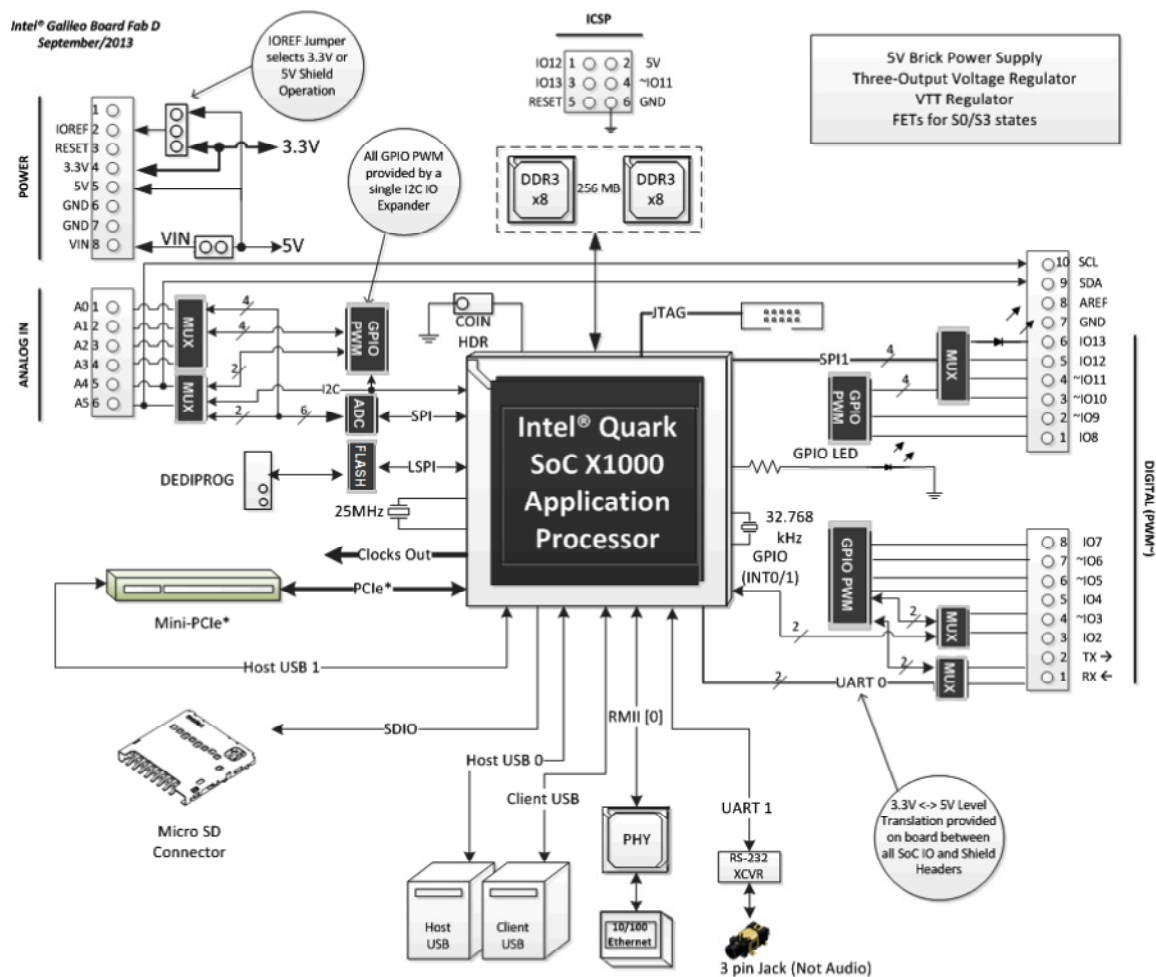


Figura 2 Schema de principiu a placii Arduino Galileo

Pe o placa Arduino Galileo s-a instalat o varianta minimala de sistem de operare Linux. O varianta extinsa de Linux se poate instala folosind o memorie externa de tip SD. Sistemul va raspunde la comenzi uzuale Linux (ls, dir, cd) pe un canal serial de tip RS232. Pentru operare de pe un terminal de tip PC se va conecta interfata seriala a placii (conectorul de tip microfon de pe placa) cu interfata seriala RS232 a calculatorului PC printr-un cablu special realizat in acest scop. Pe PC se

va lansa o aplicatie de tip hiperterminal (sau monitor de canal serial) si se va apasa tasta Enter. In acest mod sistemul va identifica terminalul de pe care se face operarea.

O alta modalitate, mult mai frecventa de operare si de executie a aplicatiilor pe placa Arduino, este prin intermediul mediului de programare (IDE) Arduino. Acest mediu care ruleaza pe un PC permite editarea, compilarea si incarcarea in vederea executiei a aplicatiilor scrise pentru placa Arduino. Descarcarea programelor se face prin intermediul unui cablu USB conectat intre conectorul USB al PC-ului si intrarea “USB client” a placii. Pe conexiunea USB se deschide un canal serial virtual care asigura transferul programului ce urmeaza sa se execute.

Important!!! Inainte de cuplarea cablului USB placa trebuie alimentata, folosind adaptorul din dotare. Nu este permisa cuplarea altor tensiuni la placa (inclusiv prin USB) pe durata cat placa nu este alimentata.

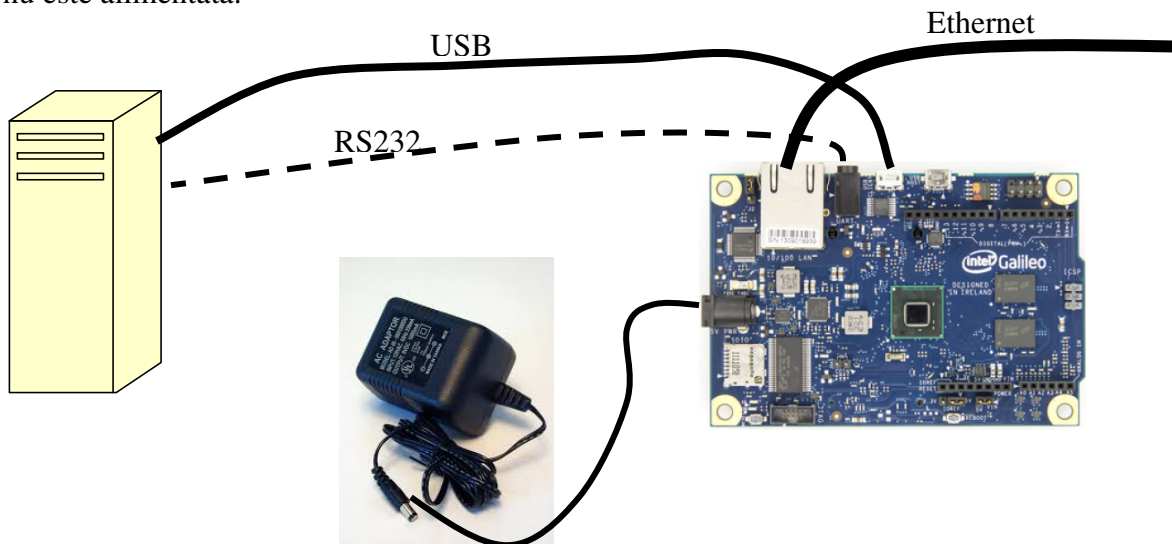


Figura 3 Conectarea placii Arduino

Limbajul de programare este unul asemanator cu limbajul “C”. In structura unui program exista doua parti:

- o parte care se executa o singura data – functia “setup()”
- o parte care se executa in mod repetitiv – functia “loop()”

In functia setup() se vor include initializarile necesare pentru executia programului, iar in functia loop(), corpul aplicatiei, care insemna o secventa de operatii executate intr-o bucla infinita. Acest mod de executie se bazeaza pe observatia ca un program de monitorizare sau de control are o parte de initializare si configurare si o alta parte care efectueaza repetitiv operatii de citire semnale, procesare si generare de comenzi.

Mediul de programare Arduino contine o biblioteca bogata de functii (proceduri) prin intermediul carora programatorul poate sa acceseze resursele fizice ale placii: semnale digitale de intrare/iesire, semnale analogice de intrare si de iesire, interfete seriale, sau alte interfete care pot fi atasate placii Arduino.

Mediul de programare pune la dispozitia utilizatorilor un numar mare de exemple de programare (din meniu: File->Examples). Aceste exemple au menirea de a exemplifica modul de accesare a diferitelor resurse ale placii si modul de utilizare a functiilor de biblioteca. Exemplele variaza de la unele simple care demonstreaza cum se controleaza un semnal digital de iesire (exemplul Blink care aprinde si stinge un LED) sau cum se citeste o intrare digitala (Button) si pana la exemple mai complexe in care se utilizeaza o interfata de retea Ethernet pentru a implementa un clients sau server de web (exemplele WEB server si WEB client).

O aplicatie scrisa in acest mediu poarta numele de schita (eng. sketch). Prin intermediul mediului de programare utilizatorul poate sa compileze programul scris (Sketch->Compile) si apoi poate sa-l lanseze in executie (butonul ->).

Mai jos este un exemplu simplu de program care aprinde si stinge un LED.

```
// semnalul de intrare/iesire digitala IO13 pe cele mai multe placi Arduino are conectat un LED.  
int led = 13;
```

```
void setup() {  
  // initializarea semnalului digital IO13 ca si iesire  
  pinMode(led, OUTPUT);  
}
```

```
// bucla care se executa la infinit:
```

```
void loop() {  
  digitalWrite(led, HIGH); // aprinde LED-ul (HIGH – tensiune ridicata)  
  delay(1000);           // asteapta o secunda  
  digitalWrite(led, LOW); // stinge LED-ul  
  delay(1000);           // asteapta o secunda  
}
```

3. Mersul lucrarii

3.1. Se vor testa exemplele din mediul de programare pentru controlul semnalelor digitale si analogice; vor fi efectuate operatii pentru:

- aprinderea si stingerea unui LED,
- citirea pozitiei unui buton,
- citirea si scrierea unor semnale analogice
- transmiterea unei informatii pe canalul serial (virtual)
- controlul unui motor pas-cu-pas

Pentru fiecare exemplu se va realiza o schema electrica in conformitate cu necesitatile exemplului.

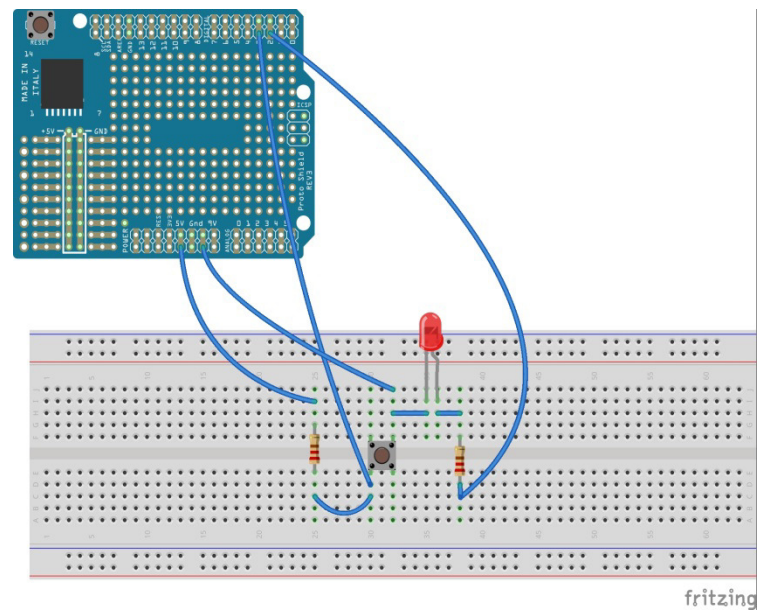
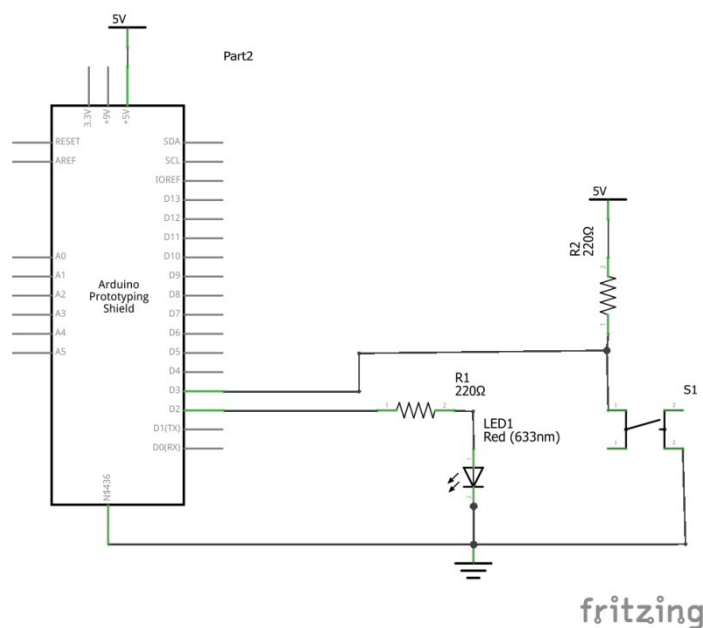
3.2 Se vor scrie si testa programe pentru placa de extensie (shield) dotata cu:

- doua afisoare 7 segmente
- 4 butoane
- circuit de adaptare (driver) pentru comanda in punte a unor motoare
- conectori de extensie pentru semnale analogice

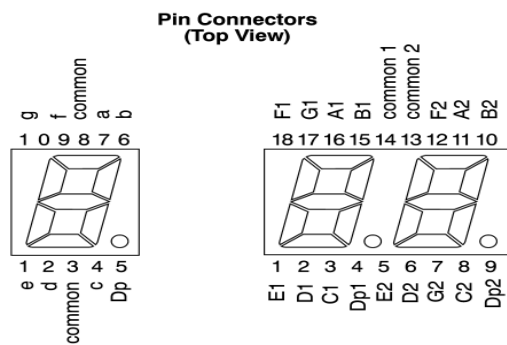
Schema electrica a placii de extensie se gaseste in anexa 1.

Anexa 1 Circuite si scheme utilizate in lucrarea de laborator

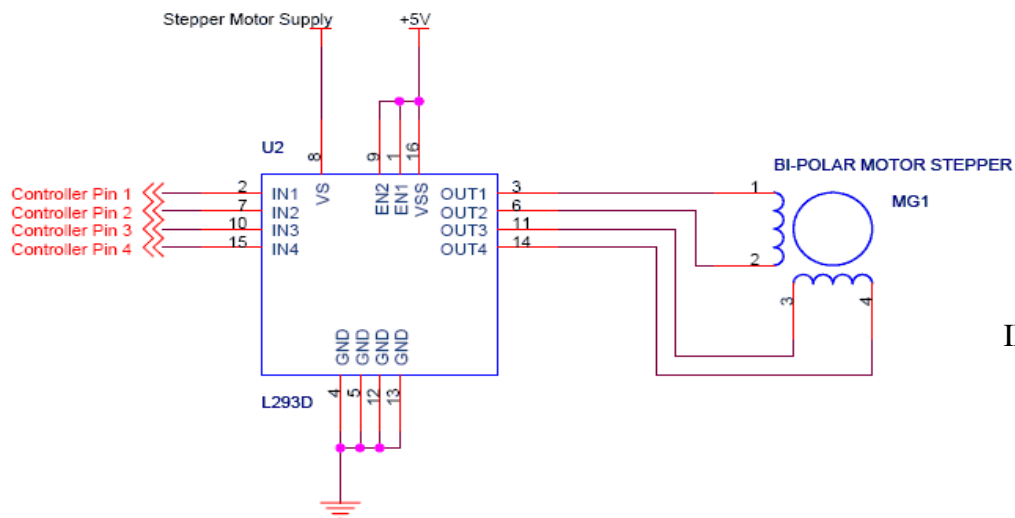
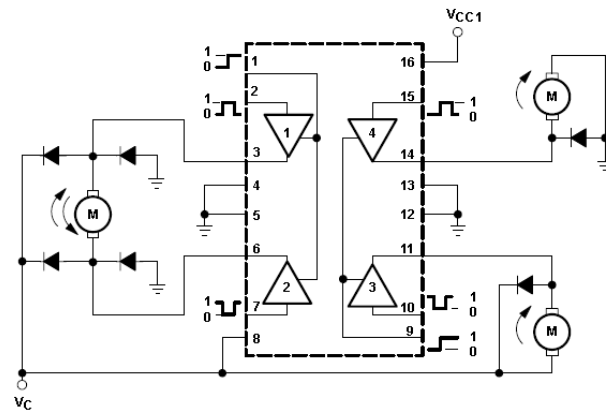
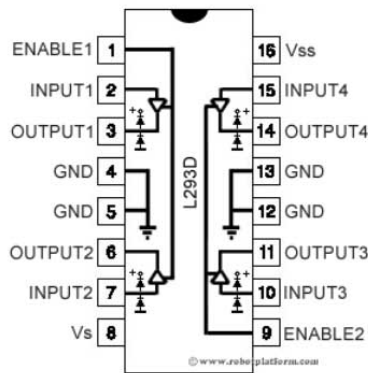
1. Schema pentru LED si Buton



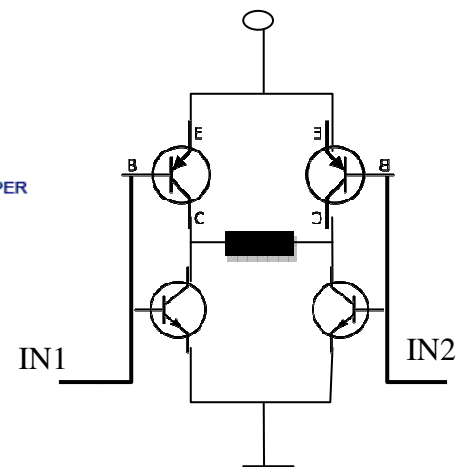
2. Circuit de afisare cu 7+1 segmente



3. L293D Driver pentru motoare bipolare (de C.C. sau pas cu pas)

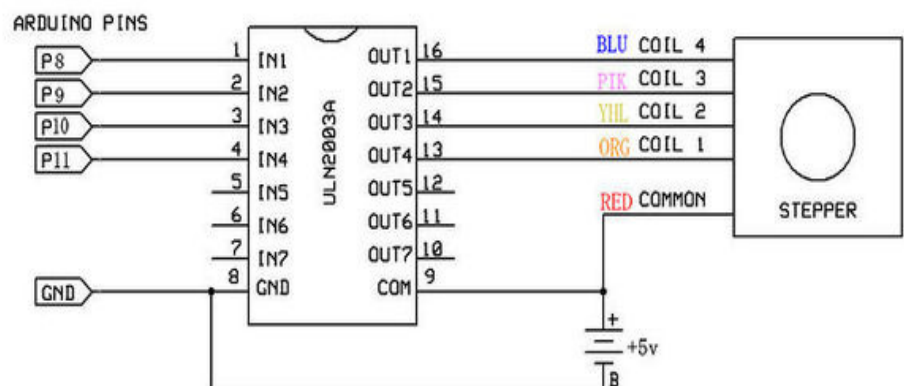
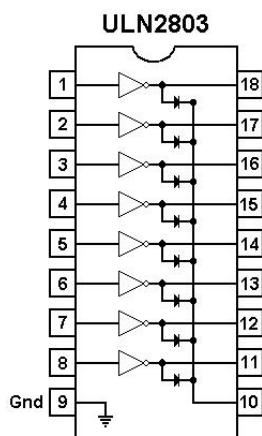


Controlul unui motor pas-cu-pas bipolar

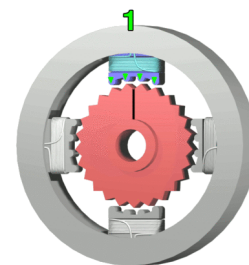
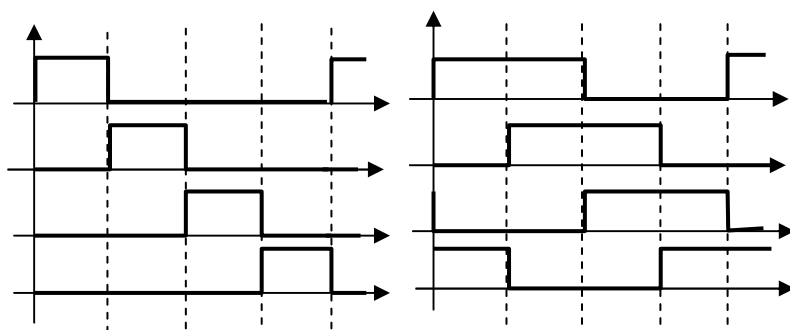


Comanda in punte H

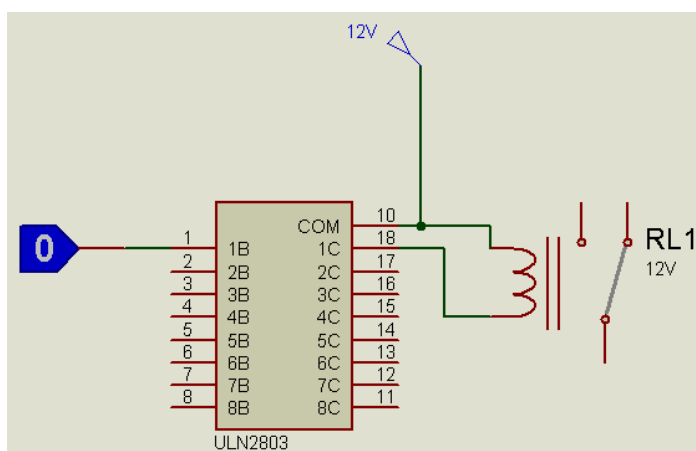
4. ULN2803 Driver pentru motoare unipolare, relee si alte sarcini inductive



Principiul de control al unui motor pas-cu-pas unipolar



Comanda unui releu



Anexa 2 Schema placii de extensie

Contine:

- Placa de extensie (shield) Arduino
- Circuit de afisare 2*7 segmente – utilizat in regim de multiplexare in timp
- Driver pentru motor bipolar
- 4 mini-butoane (intrerupatoare)
- Sir de cleme pentru intrari analogice
- Sir de cleme pentru conectare motor
- Sir de cleme pentru alimentare externa cu 12V

