

1 Porturi de intrare iesire al microcontrolelor AVR

1.1 Scopul lucrării

Lucrarea prezintă porturile de intrare-iesire ale microcontrolelor AVR. Sunt prezentate modurile de transfer specificate de acestea. Se vor testa porturile de intrare și iesire și se vor realiza aplicații.

1.2 Considerații teoretice

1.2.1 Prezentare generală

O importanță deosebită este acordată modalităților de conectare a unui sistem de achiziție de date la un calculator PC, sau microcontrolelor. Interfațarea calculatoarului sau a microcontrolelor cu aplicațiile externe se poate realiza utilizând mai multe căi: portul paralel, portul serial, interfața USB, precum și sloturile ce permit accesul direct pe magistralele ISA sau PCI. Fiecare din aceste soluții au avantajele și dezavantajele lor, de exemplu accesul pe magistrală necesită o realizare tehnologică complexă a plăcii de achiziție (pentru accesul pe placa de bază) și o cunoaștere temeinică a structurii hardware a PC, iar pentru microcontrolele AVR o temeinică înțelegere a structurilor porturilor de intrare și ieșire. Portul paralel este mai folosit pentru că oferă mai multe linii de comandă, de stare și control precum și o viteză mult mai mare decât a portului serial, iar în cazul microcontrolelor utilizarea porturilor de intrare-iesire de asemenea permit o utilitate mult mai largă în interconectivitatea cu diferite aplicații și circuite. În continuare ne vom axa pe porturile de intrare/ieșire a microcontrolelor AVR.

1.2.2 Structura și caracteristici

Toate porturile AVR-urilor au o funcționalitate de tipul *citire-modificare-scriere* când acestea sunt utilizate ca și porturi generale digitale de intrare sau ieșire (I/O). Acest aspect ne sugerează faptul că oricărui pin al unui port îi se poate schimba întrebuințarea (ieșire sau intrare) fără să fie afectată întrebuințarea altui pin din același port, prin instrucțiuni de tipul SBI (*set bit in register = setați bitul în registru*) sau CBI (*clear bit in register = ștergeți bitul din registru*). Același principiu se aplică și când se schimbă valoarea de ieșire sau când este activată respectiv dezactivată funcționalitatea de tipul “*pull-up resistors*” dacă este configurat ca și intrare. Fiecare pin poate să fie programat ca și intrare sau ieșire. Au destulă pentru a alimenta un LED direct, iar din punct de vedere a intrărilor fiecare pin poate fi configurat sau nu ca intrare de tipul “*pull-up resistor*”. Toți pini de intrare/ieșire au diode de protecție atât pe alimentarea (V_{cc}) cât și către masă (GND) după cum este indicat în figura 1.1.

În dicuțiile generale despre port și biți, se va folosi notațiile generale sunt la forma următoare: litera “x” mic reprezintă litera portului, iar “n” reprezintă numărul bitului, sub forma PORTxn. Dar când se folosește un port sau un bit definit într-un program se va folosi notația exactă (de exemplu pentru bitul 5 din portul D, se folosește PORTB3).

Adresele de memorie pentru fiecare port, se regăsesc astfel: registru de date (Data Register) – PORTx, tipul de date (Data Direction Register) – DDRx, și stare pinilor de intrare (Port Input Pins) – PINx. Registru pentru starea pinilor este de tipul *citire (read only)*, iar restul sunt de tipul *citire/scriere (read/write)*.

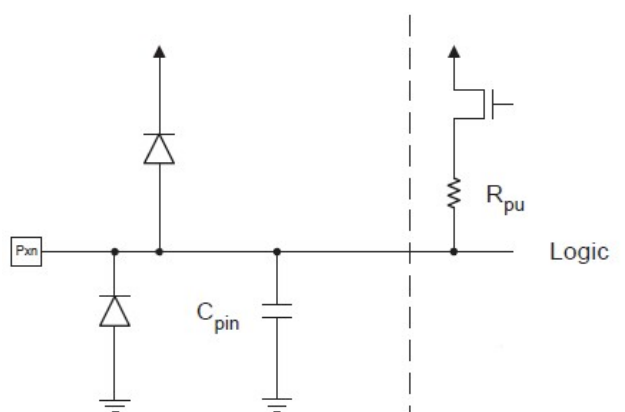


Fig. 1.1 Schema echivalentă a unui pin de intrare/ieșire

În funcție de tipul microcontroler-lui putem avea caracteristici electrice diferite. Principalele caracteristici sunt prezentate în tabelul 1.1.

Simbol	Parameteru	Conditionare	Min.	Tipic	Max.	Unitate de măsură
V_{IL}	Tensiune intrare 0 logic ₂ (excepție pini XTAL1 și RESET)	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}$ $0.3V_{CC}$	V
V_{IH}	Tensiune intrare 1 logic ₂ (excepție pini XTAL1 și RESET)	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}$ $0.6V_{CC}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{OL}	Tensiune de ieșire 0 logic (excepție pinul de RESET)	$I_{OL} = 20mA,$ $V_{CC} = 5V$	$T_A = 85^{\circ}C$		0.9	V
			$T_A = 105^{\circ}C$		1.0	
		$I_{OL} = 10mA,$ $V_{CC} = 3V$	$T_A = 85^{\circ}C$		0.6	
			$T_A = 105^{\circ}C$		0.7	
V_{OH}	Tensiune de ieșire 1 logic (excepție pinul de RESET)	$I_{OH} = -20mA,$ $V_{CC} = 5V$	$T_A = 85^{\circ}C$	4.2		V
			$T_A = 105^{\circ}C$	4.1		
		$I_{OH} = -10mA,$ $V_{CC} = 3V$	$T_A = 85^{\circ}C$	2.3		
			$T_A = 105^{\circ}C$	2.1		
I_{IL}	Curentul de pierdere pe pinul I/O	$V_{CC} = 5.5V, \text{ logic } 0$			1	μA
I_{IH}	Curentul de pierdere pe pinul I/O	$V_{CC} = 5.5V, \text{ logic } 1$			1	μA
R_{PU}	Rezistența de tragere a pinului I/O		20		50	$k\Omega$

Arhitectura generală a porturile de intrare-ieșire este prezentată în figura 1.2.

Pentru lucrarea de față se va folosi microcontroler-ul Arduino Nano. Schema electrică a acestuia este prezentată în figura 1.3. Acesta prezintă 3 porturi, pe 8 biți. PortD, PortB și PortC. Biți portului D sunt legați la ieșirile digitale D0-D7. În cazul biților portului B, ultimi 6 (PB0-PB5) sunt legați la ieșirile digitale D8-D13, iar primi doi biți ai portului PB6 și PB7 sunt legați la ieșirile de 3,3V, respectiv Aref și astfel nu avem acces la ei. În aceeași situație sunt și biți portului

C, ultimi 6 sunt legați la pini Analogici A0-A5, al doilea pin al portului (PC6) la pinul de RESET, iar la PC7 nu avem acces acesta nefiind legat la vreo ieșire de pe placuța de dezvoltare.

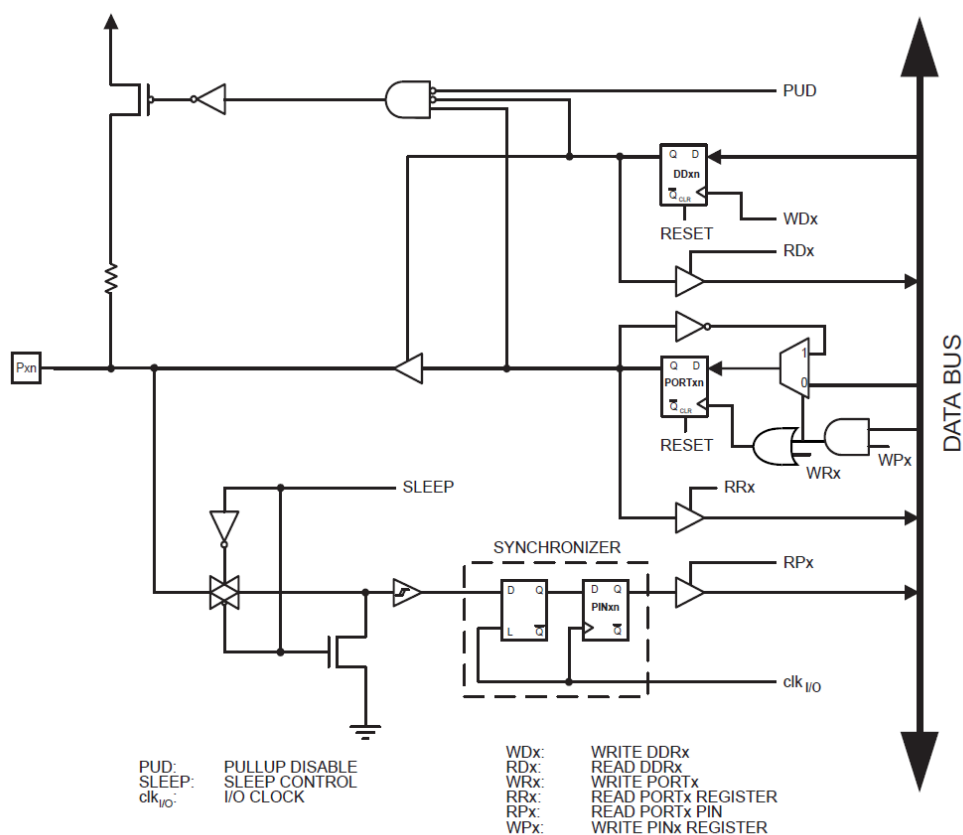


Fig. 1.2 Arhitectura generală a porturilor de intrare-ieșire

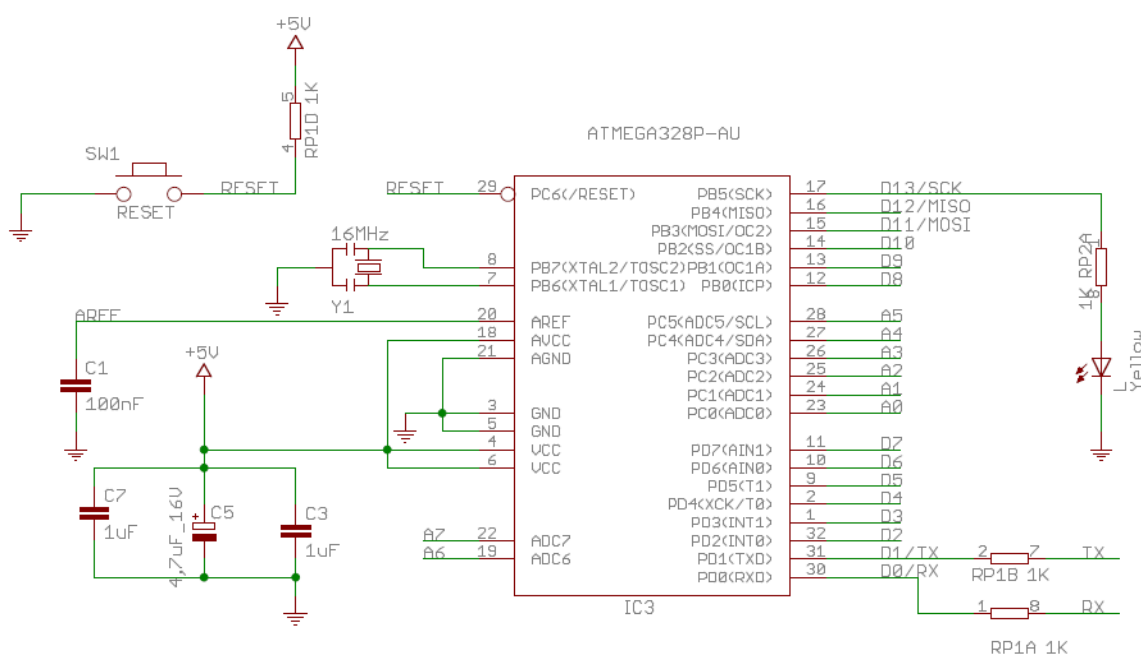


Fig. 1.3 Schema electrică a microcontroler-ului Arduino Nano

În concuzie exista 3 porturi bidirectionale cu un număr de 20 de biți care pot fi accesați. Pentru iesire se va folosi comanda de PORTx , iar pentru intrare se va folosi comanda de PINx .

1.3 Desfășurarea lucrării

1.3.1 Ieșiri digitale

Se vor testa ieșirile digitale ale microcontroler-ui Arduino Nano pentru porturile D (biți 7 și 6) și B (biți 0-5). Se vor scrie vor scrie programe în mediul de programare Arduino CC folosind librariile io.h . Pe ieșirile digitale testate se conectează diode luminiscente ca și în figura 1.4, respectiv conform schemei electrice din figura 1.5.

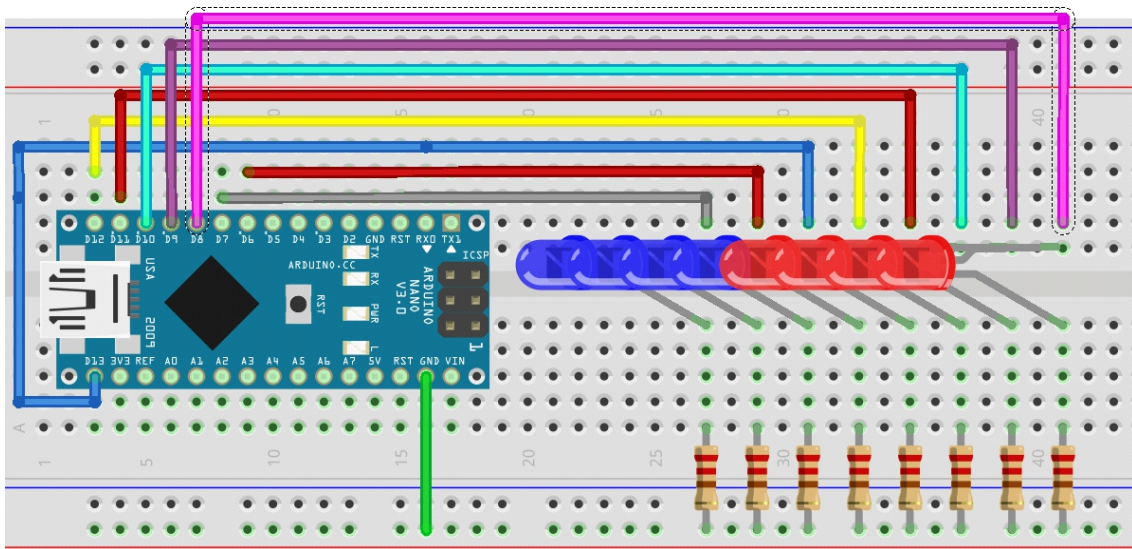


Fig. 1.4 Montajul experimental de testare

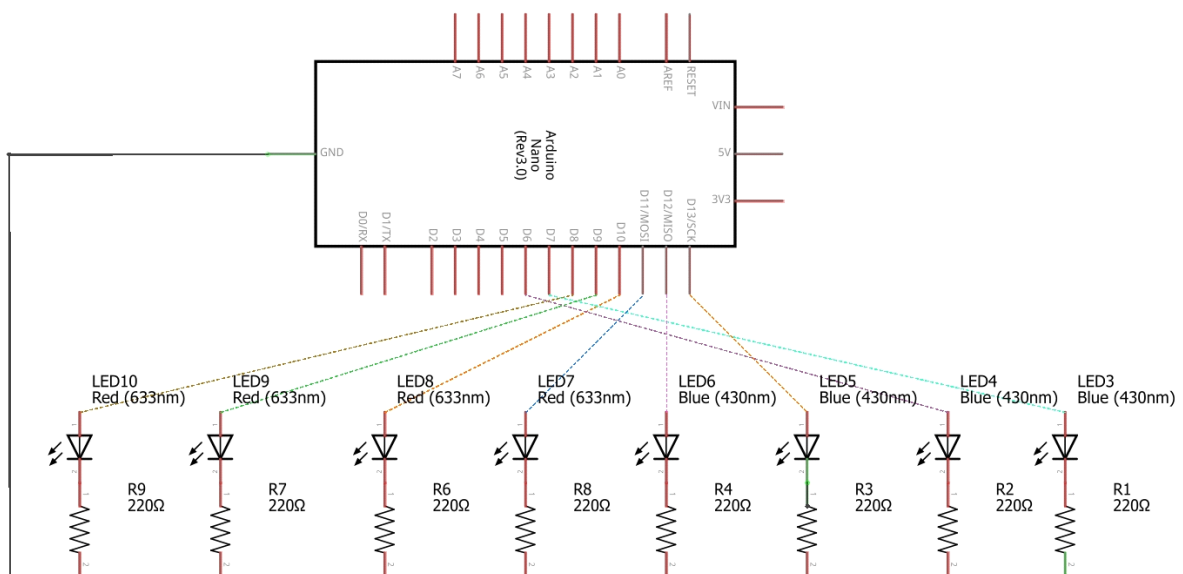


Fig. 1.5 Schema electrică de conectare

Utilizând consola serială (*Serial Monitor*) a aplicației Arduino CC se va testa ieșirile prin comenzi cu valoare în binar, hexa-zecimal, și zecimal pentru cei 8 biți legați la cele 8 leduri. Corespondekța cifrelor în cele trei baze numerice este reprezentată în tabelul 1.2:

BINAR	HEXA	ZECIMAL
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Se vor scrie programe prin care diodele luminescente să fie realizate o lumină dinamică. Un exemplu de astfel de program este prezentat mai jos:

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>

byte val = 0x55;
byte val1 = 0;
byte te;
char t;
int i = 1;
int main(void)
{
    Serial.begin(9600);
    DDRB |= 0x3f;
    DDRD |= 0xC0;
    while(1)
    {
        i=1;
        while(i<128)
        {
            val = i;
```

```

    val1 = val;
    val1 = val;
    val &= 0xC0;
    val1 &= 0x3F;
    val |= 0x3F;
    val1 |= 0xC0;
    PORTD &= 0x3F;
    PORTB &= 0xC0;
    PORTD |= val;
    PORTB |= val1;
    i*=2;
    _delay_ms(100);
  }
  i=128;
  while(i>=1)
  {
    val = i;
    val1 = val;
    val1 = val;
    val &= 0xC0;
    val |= 0x3F;
    val1 &= 0x3F;
    val1 |= 0xC0;
    PORTD &= 0x3F;
    PORTB &= 0xC0;
    PORTD |= val;
    PORTB |= val1;
    i/=2;
    _delay_ms(100);
  }
}
}

```

1.3.2 Intrari digitale

Se va realiza montajul din figura 1.6, respectând schema electrică din figura 1.7

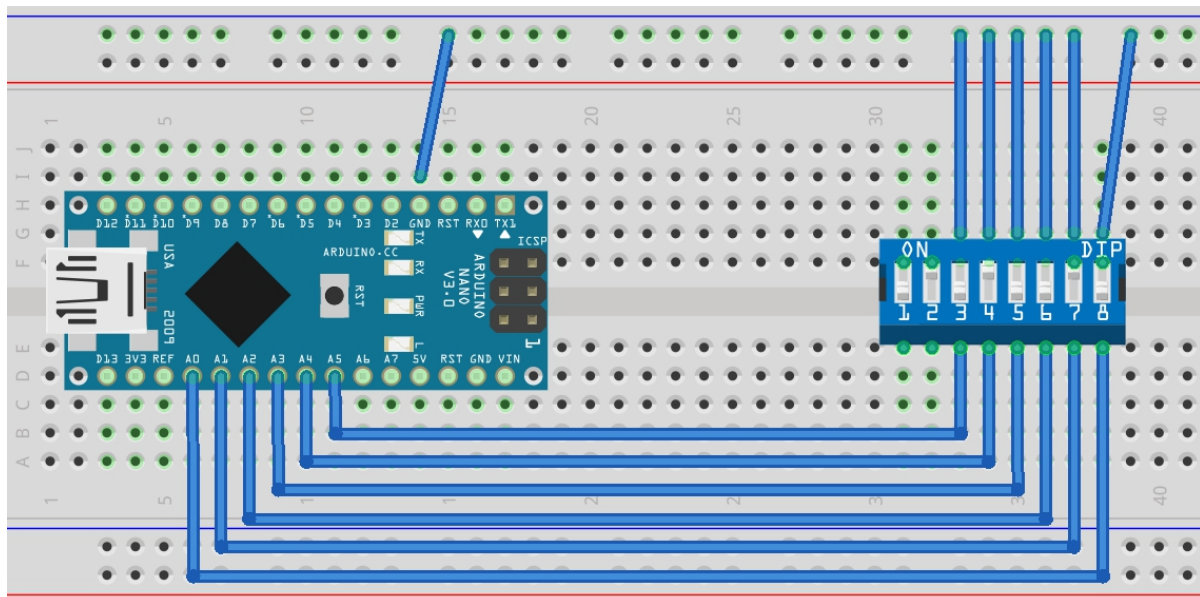


Fig. 1.6 Montaj pentru testarea intrarilor

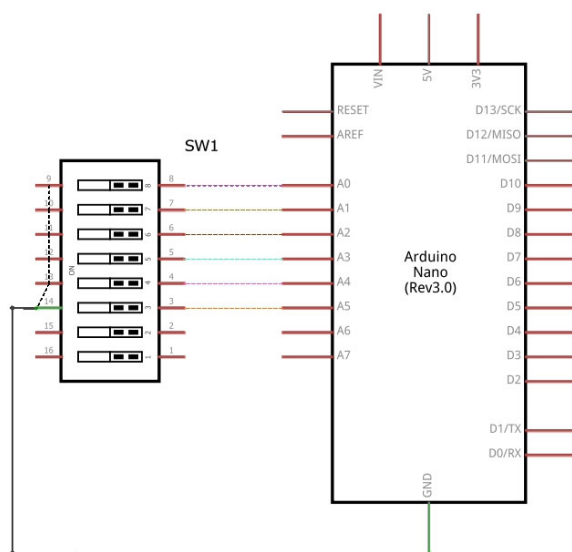


Fig. 1.7 Schema electrică pentru testare intrarilor digitale

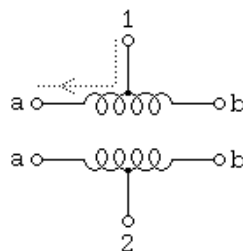
În realizarea programelor se vor declara intrari de tipul *pull-up resistor*. Cu ajutorul comandai PINC, se va citi poziția comutatoarelor 1-6 de pe placuța de test, iar rezultatul se va afișa în consola serială. Se vor realiza programe astfel încât se se afișeze poziția comutatoarelor în sistem binar, zecimal și hexazecimal

1.3.3 Comandarea unui motor pas cu pas

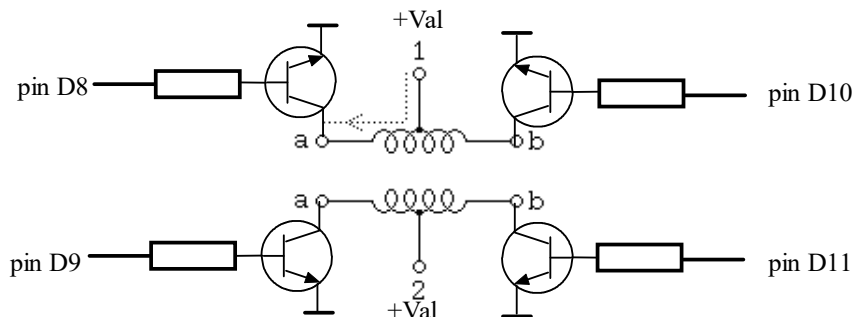
Se va comanda un motor pas cu pas unipolar cu montajul ilustrat in figura de mai jos.



a)



b)



c)

Fig. 1.8 Motorul pas cu pas unipolar: a) montaj experimental, b) constructie, c) schema electrică

Motoarele unipolare sunt atât cele cu magnet permanent cât și cele hibride cu cinci sau șase fire cablate ca în figura de mai sus, cu legătura centrală pe fiecare din cele două înfășurări. În general legătura centrală este conectată la borna pozitivă și celelalte două capete ale înfășurării sunt puse la masă alternativ pentru a reversa direcția.

Secțiunea prezentată arată un motor cu magnet permanent sau hibrid cu unghi de 30 de grade/pas. Diferența dintre aceste două tipuri de motoare este nerelevantă pentru acest nivel de abstractizare.

Înfășurarea 1 este distribuită între polii de sus și de jos ai statorului în timp ce înfășurarea 2 este distribuită între polii din stânga și din dreapta ai statorului.

Rotorul este un magnet permanent cu șase poli, trei sud și trei nord, dispuși în jurul circumferinței, pentru rezoluții mai mari rotorul va avea mai mulți poli; 30 de grade/pas este o rezoluție des întâlnită la aceste motoare dar se mai pot întâlni și motoare cu rezoluții de 15 și 7,5 grade/pas. Motoarele cu magnet permanent pot fi și cu rezoluție de 1,8 grade/pas, motoarele hibride sunt de obicei construite cu rezoluții între 3,6 și 1,8 grade/pas.

Așa cum se poate vedea în figură, curentul care trece prin legătura centrală a înfășurării 1 spre terminal, face ca polul de sus al statorului să fie un pol nord iar polul de jos să fie un pol sud. Acest lucru atrage rotorul în poziția arătată. Dacă alimentarea înfășurării 1 este decuplată iar înfășurarea 2 este alimentată rotorul se va roti cu 30 de grade. Pentru a roti rotorul în mod continuu vom alimenta înfășurările în secvențe. Presupunând 1 logic atunci când prin înfășurare trece curent, următoarele două secvențe vor roti rotorul în sens orar, 24 de pași, sau patru rotații complete:

Înfășurarea 1a. 1000100010001000100010001

Înfășurarea 1b. 0010001000100010001000100

Înfășurarea 2a. 0100010001000100010001000

Înfășurarea 2b. 0001000100010001000100010

Ambele jumătăți ale înfășurării nu vor fi alimentate în același timp. Ambele secvențe vor roti un magnet permanent, un pas, o dată. Secvența de sus alimentează o singură înfășurare în același timp; va utiliza o putere mai mică.

- Se va scrie un program în mediul de programare arduino CC care să implementeze secvența de mai sus
- Se va scrie un program în care să inverseze sensul de rotație al motorului.

A doua secvență alimentează ambele înfășurări în același timp și va produce un cuplu de 1,4 ori mai mare decât prima, utilizând putere dublă.

Înfășurarea 1a. 1100110011001100110011001

Înfășurarea 1b. 0011001100110011001100110

Înfășurarea 2a. 0110011001100110011001100

Înfășurarea 2b. 1001100110011001100110011

Poziția pașilor produsă de cele două secvențe nu este identică; ca rezultat, prin combinarea celor două secvențe se obține o comandă cu jumătate de pas, cu motorul pășind

alternativ la poziții indicate de una sau cealaltă dintre secvențe. Secvența combinată este prezentată mai jos:

Înfășurarea 1a. 11000001110000011100000111

Înfășurarea 1b. 00011100000111000001110000

Înfășurarea 2a. 01110000011100000111000001

Înfășurarea 2b. 00000111000001110000011100

- Se va scrie un program pentru a implementa secvența de pași menționată anterior.