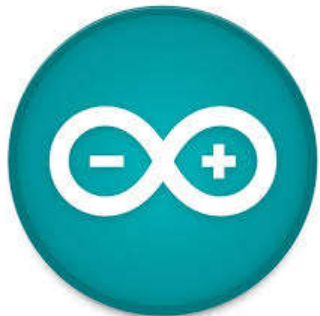


Έλεγχος διάφορων τύπων DC κινητήρων με την πλατφόρμα Arduino

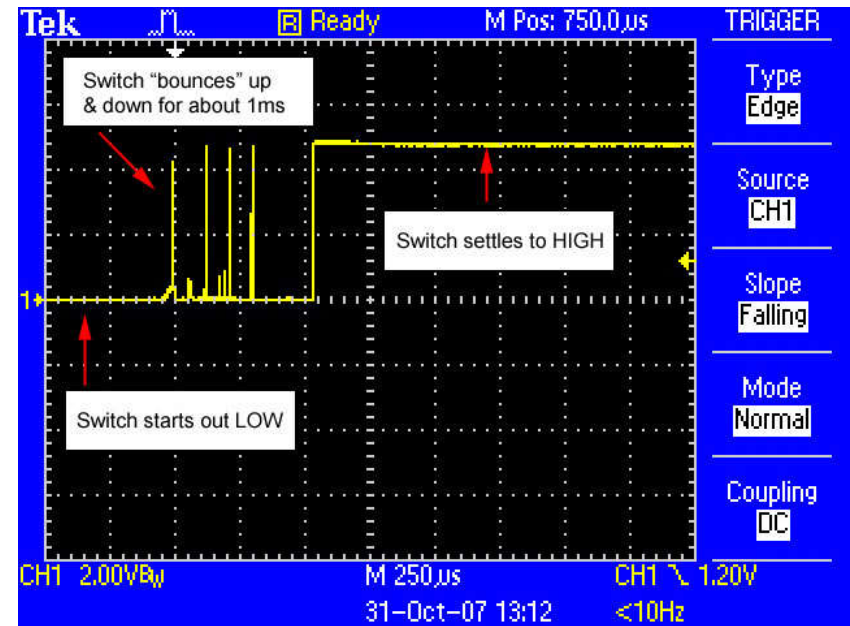
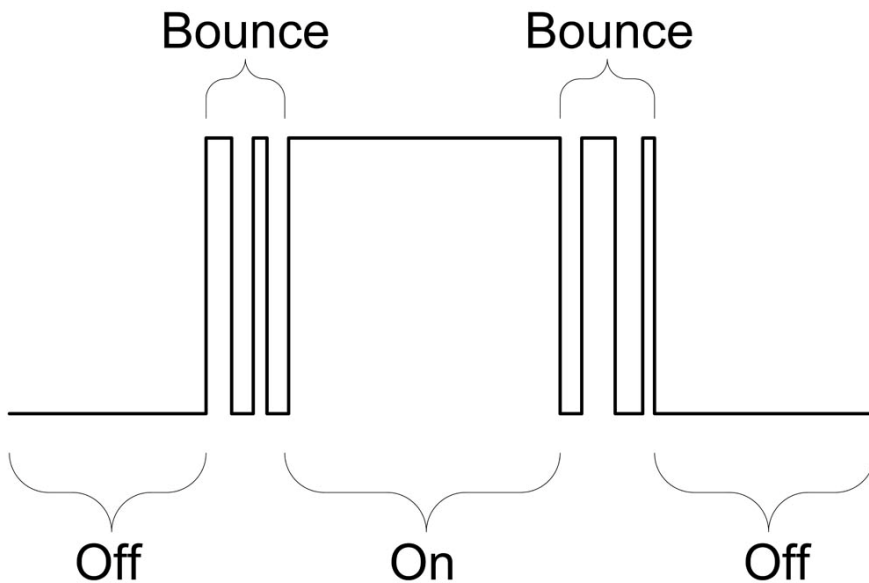
ΠΑΡΟΥΣΙΑΣΗ

Τσαλμπούρης Γεώργιος ΠΕ17.04
gtsalmpouris@gmail.com
tsal81@yahoo.com



Αναπήδηση Μηχανικών Διακοπών (Bouncing)

Ένας μηχανικός διακόπτης κανονικά ανοιχτός αν πατηθεί δεν πηγαίνει απευθείας από την κατάσταση OFF στην κατάσταση ON, όπως θα θέλαμε και θα περιμέναμε. Στην πραγματικότητα εκτελεί μερικές αναπηδήσεις μέχρι να πάει από την αρχική στην τελική του κατάσταση.



Αντιμετώπιση της Αναπήδησης(Debouncing)

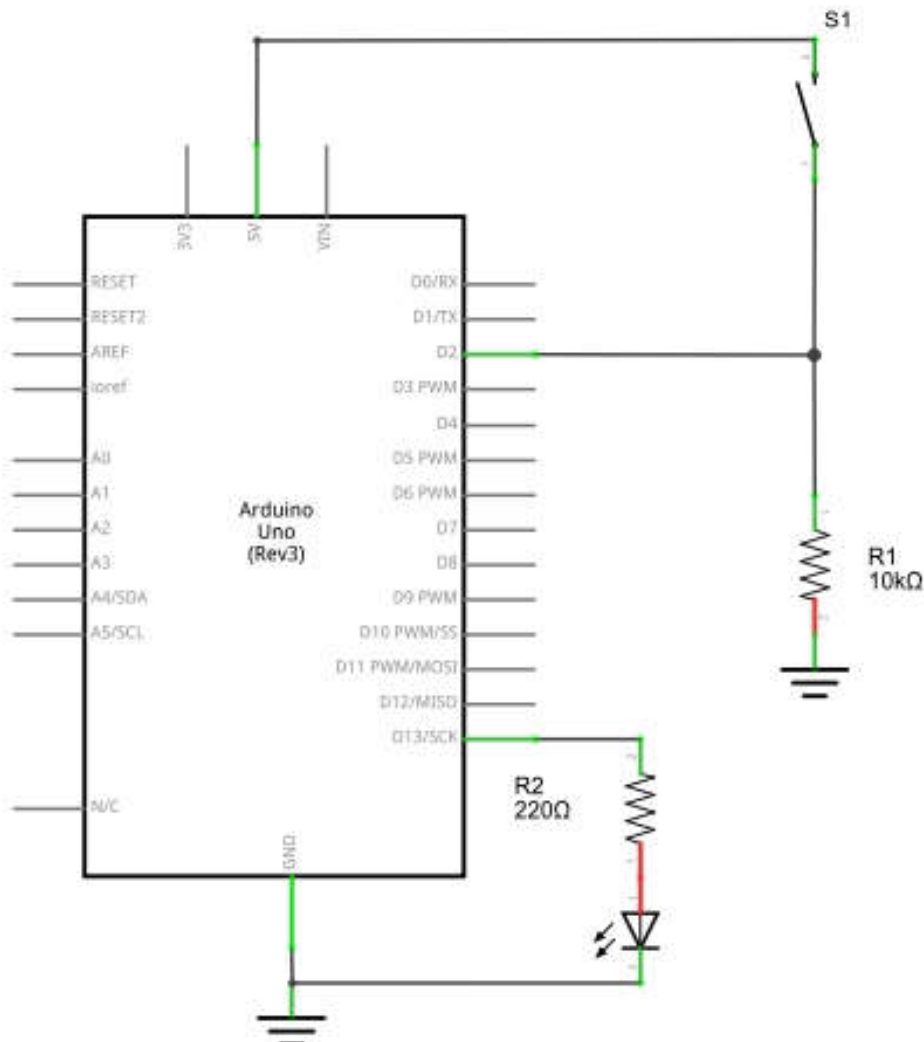
Ο μικροελεγκτής μπορεί να διαβάσει μεταβολές στις εισόδους του πάρα πολύ γρήγορα. Αυτό έχει σαν αποτέλεσμα ένα πάτημα του διακόπτη να ερμηνευτεί σαν πολλαπλά πατήματα και πρέπει να ληφθεί μέριμνα για να αντιμετωπιστεί το πρόβλημα.

Η αναπήδηση μπορεί να αντιμετωπιστεί με δύο τρόπους:

- 1) Μέσω **software** (με χρήση delay).
- 2) Μέσω **hardware** (με κύκλωμα RC κατάλληλης σταθεράς χρόνου).

Εφαρμογή 1

(Αλλαγή κατάστασης ενός Led με 1 διακόπτη)



Εφαρμογή 1

(Αλλαγή κατάστασης ενός Led με 1 διακόπτη)

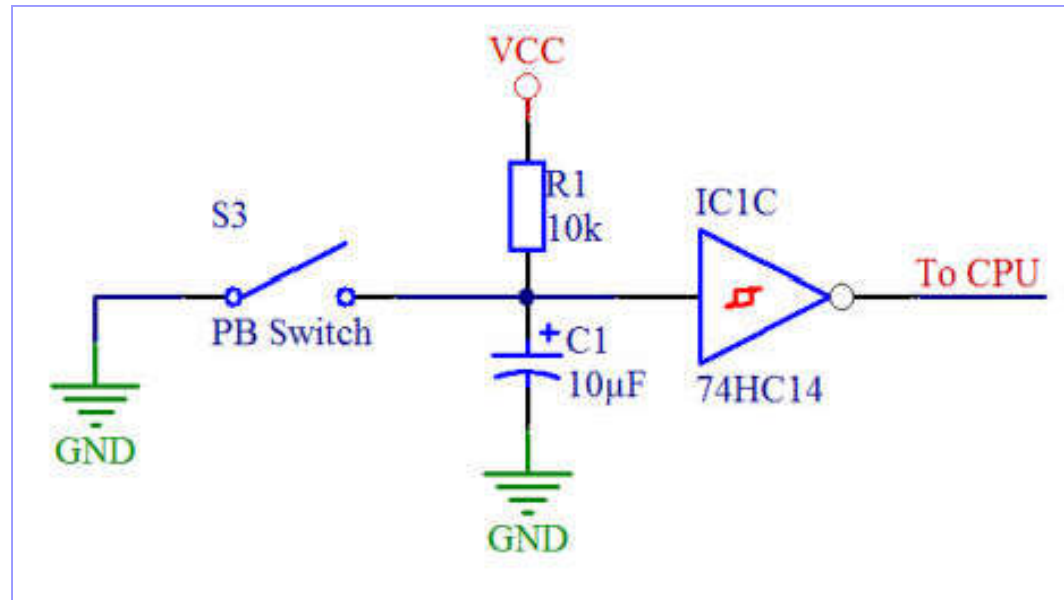
```
toggle_led
int led = 13;      // LED on pin 13
int switchPin = 2; // switch on pin 2

boolean state = false;

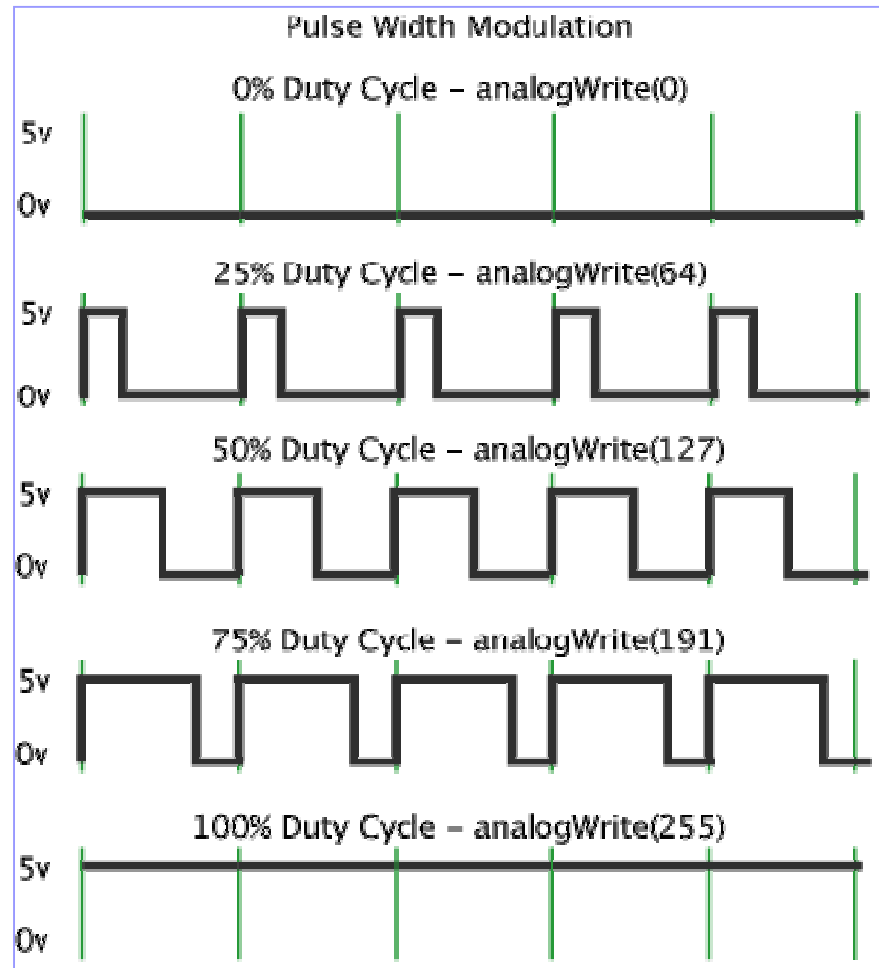
void setup()
{
  pinMode(led, OUTPUT);
  pinMode(switchPin, INPUT);
}

void loop()
{
  if (digitalRead(switchPin) == HIGH)
  {
    delay(250);           // delay to debounce switch
    state = !state;      // toggle state variable
    digitalWrite(led, state);
  }
}
```

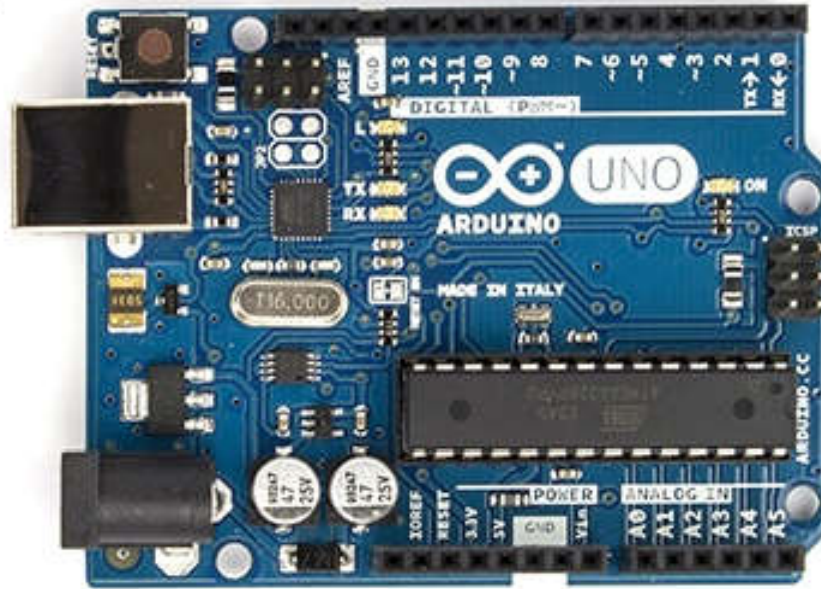
Hardware Debouncing



Διαμόρφωση PWM (Pulse Width Modulation)

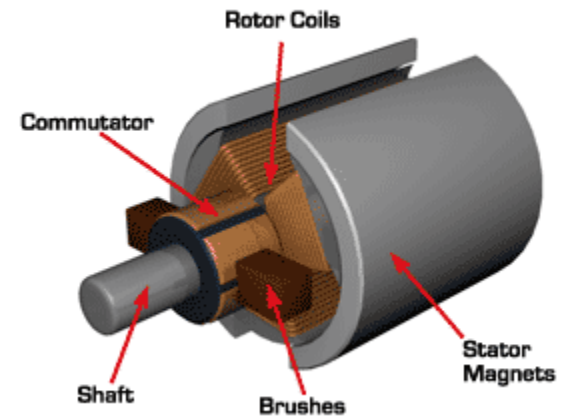
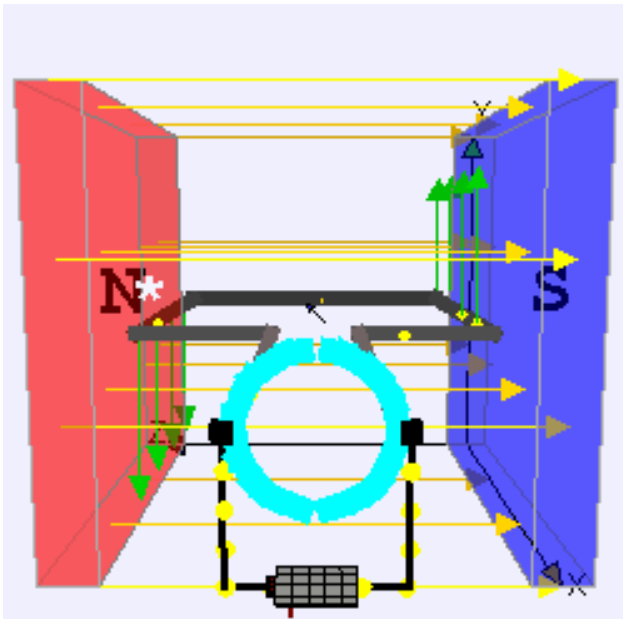
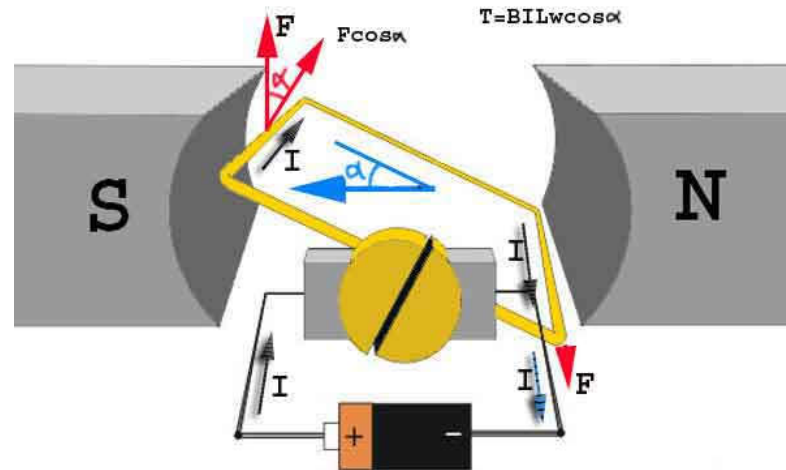


PWM στον Arduino Uno

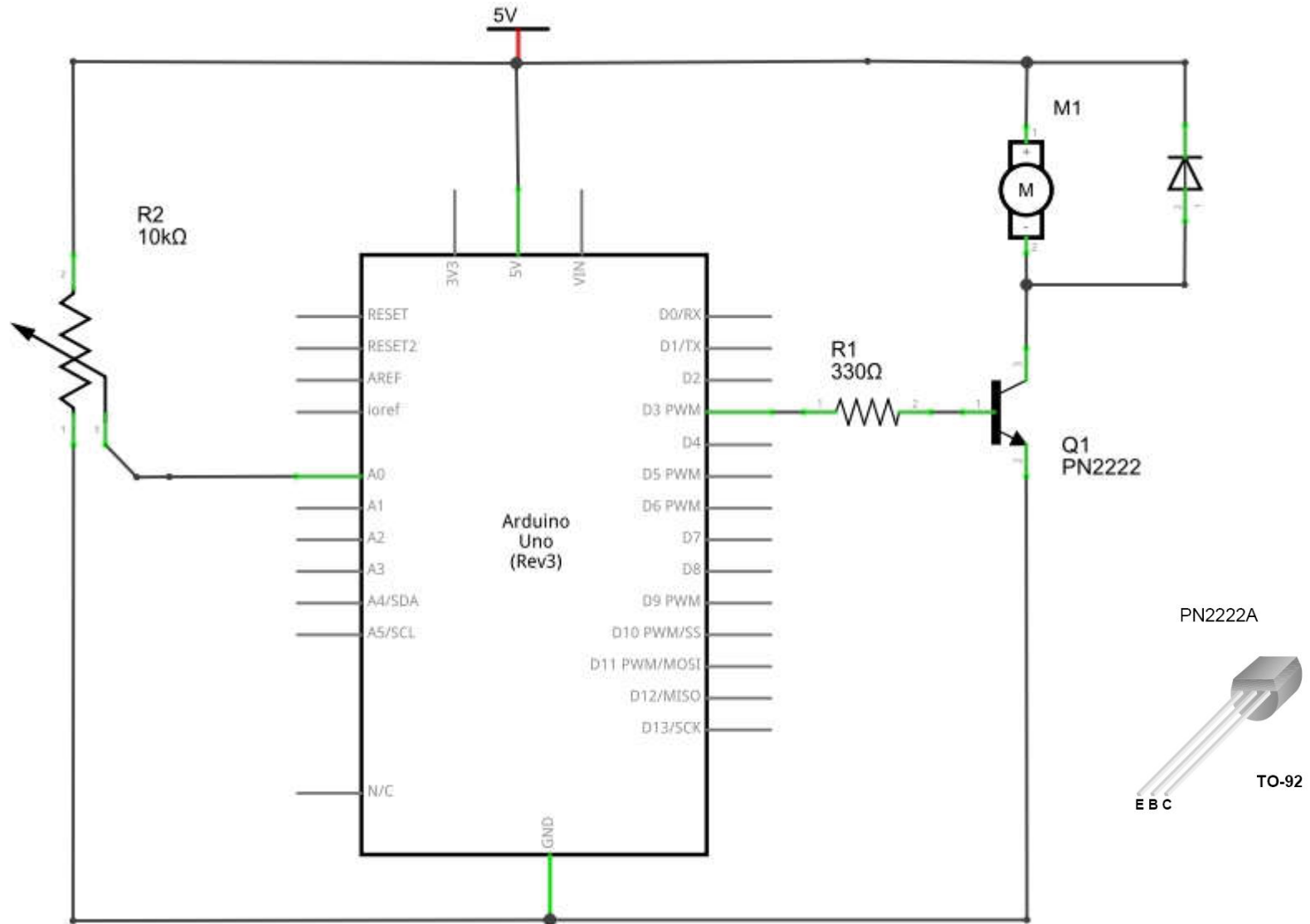


- Η χρήση της PWM στον Arduino γίνεται με την συνάρτηση `analogWrite` με τιμές από 0-255.
- Η συχνότητα PWM στα pin 5 και 6 είναι 980 Hz
- Η συχνότητα PWM στα pin 3,9,10,11 είναι 490 Hz

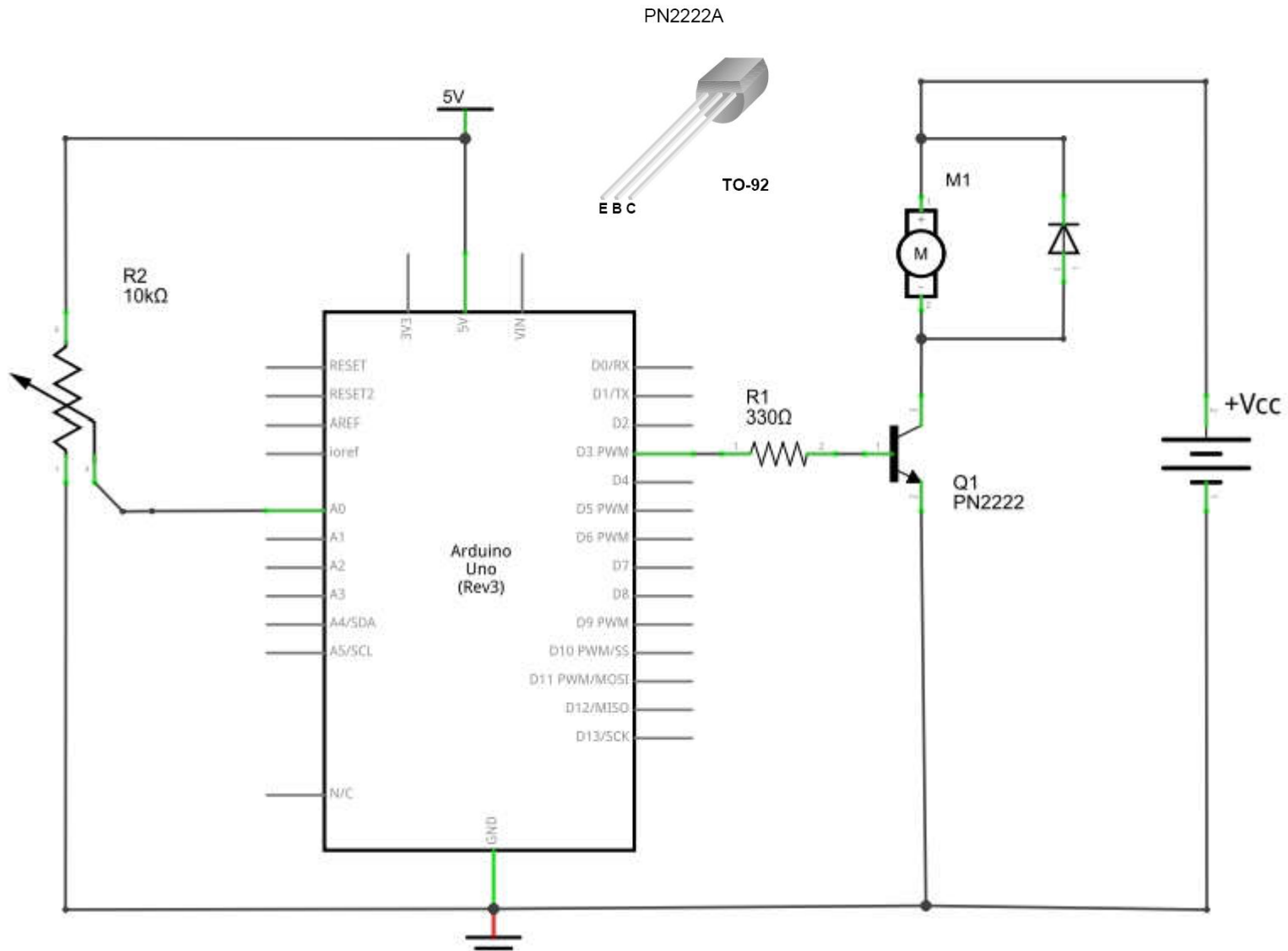
Αρχή Λειτουργίας DC Motor



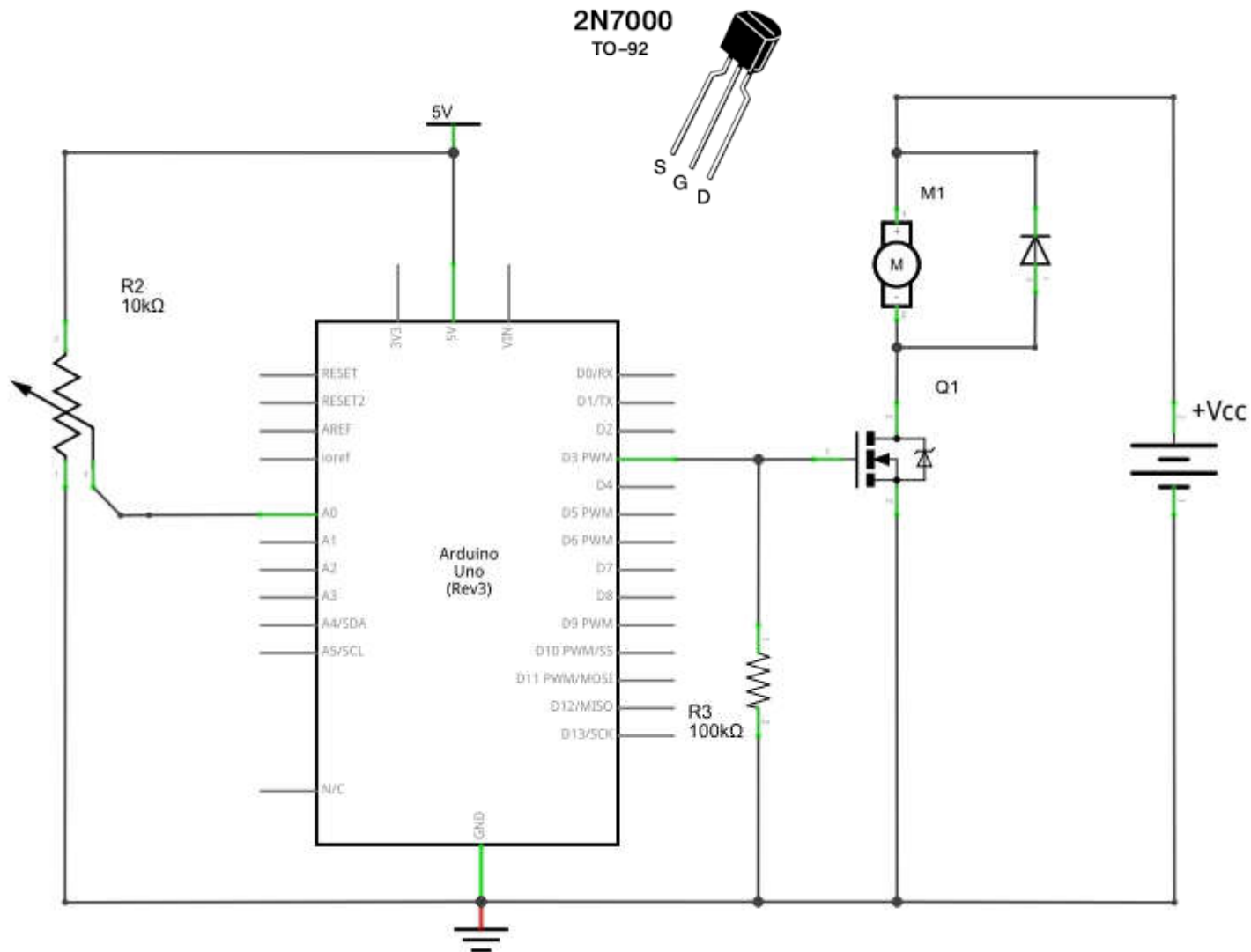
Εφαρμογή 2 (Έλεγχος Στροφών DC Motor με PWM)



Εφαρμογή 2 (Έλεγχος Στροφών DC Motor με PWM)



Εφαρμογή 2 (Έλεγχος Στροφών DC Motor με PWM)



Δίοδος Αντεπιστροφής (Flyback, FlyWheel or Snubber Diode)

Figure 1. Closed Switch, No Flyback Diode

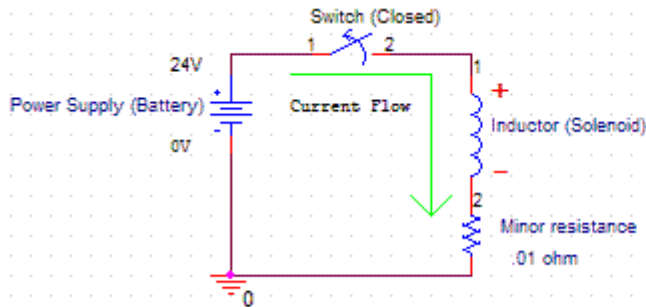


Figure 2. Open Switch, Energized Inductor, No Flyback Diode

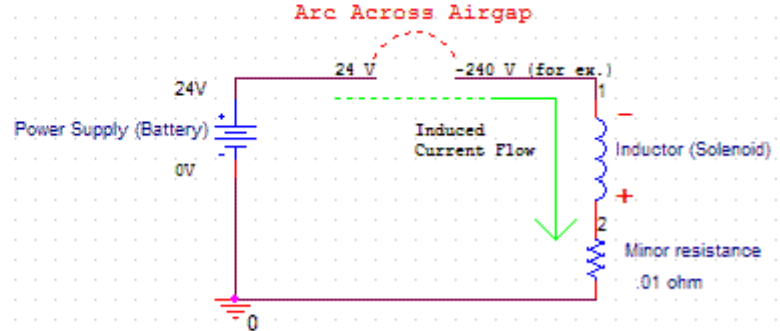
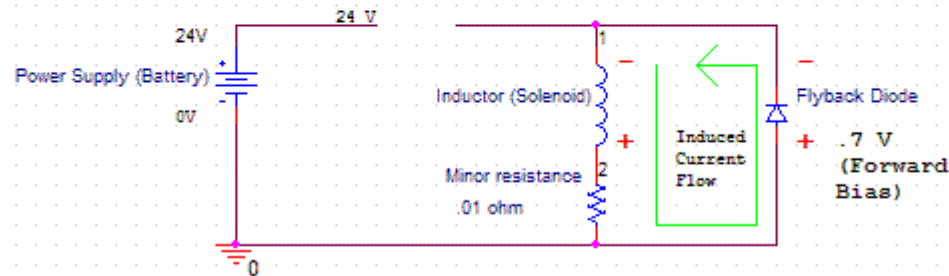
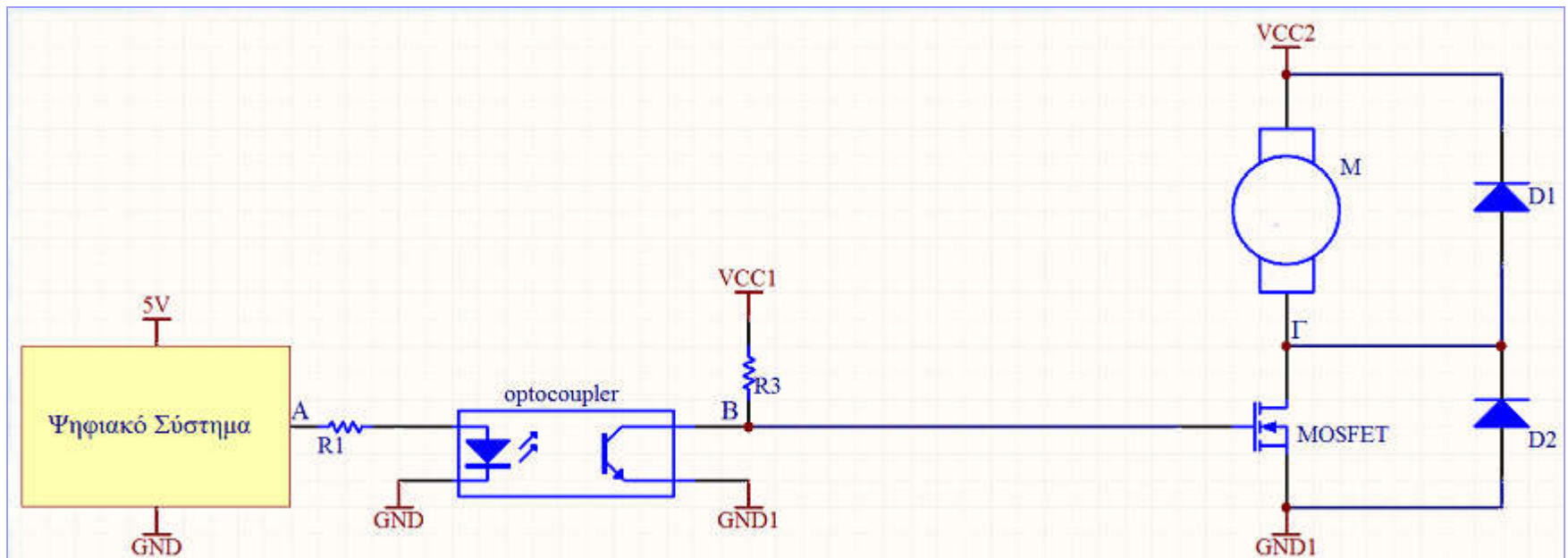


Figure 3. Open Switch, Energized Inductor, Flyback Diode Protection



Ηλεκτρική Απομόνωση Κυκλώματος Οδήγησης από το Κύκλωμα Ισχύος



Εφαρμογή 2 (Έλεγχος DC Motor με PWM)

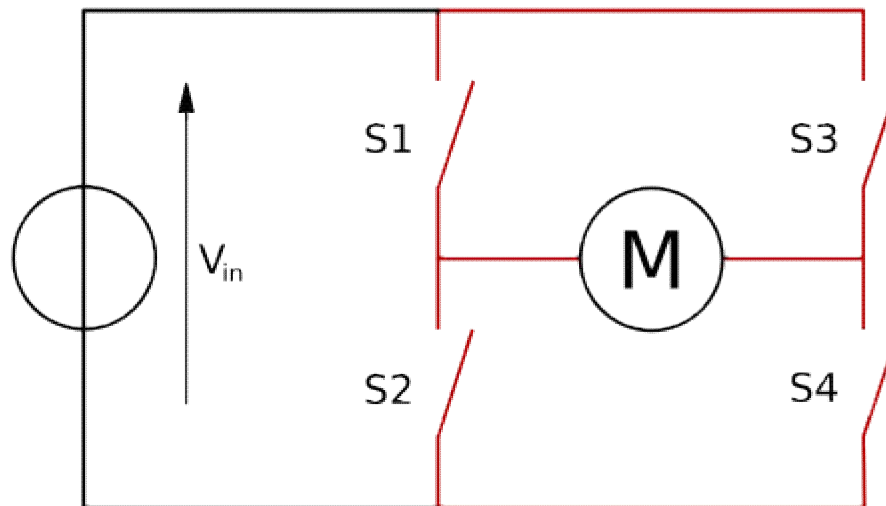
```
motor_speed_control_pot
//Elegxos taxititas mikrou moter mesw potensiometrou
//To potensiometro sindeetai sto A0 tou arduino
//To moter elegxetai mesw PWM apo to pin3 tou arduino

void setup()
{
  pinMode (A0,INPUT);
  pinMode (3,OUTPUT);
}

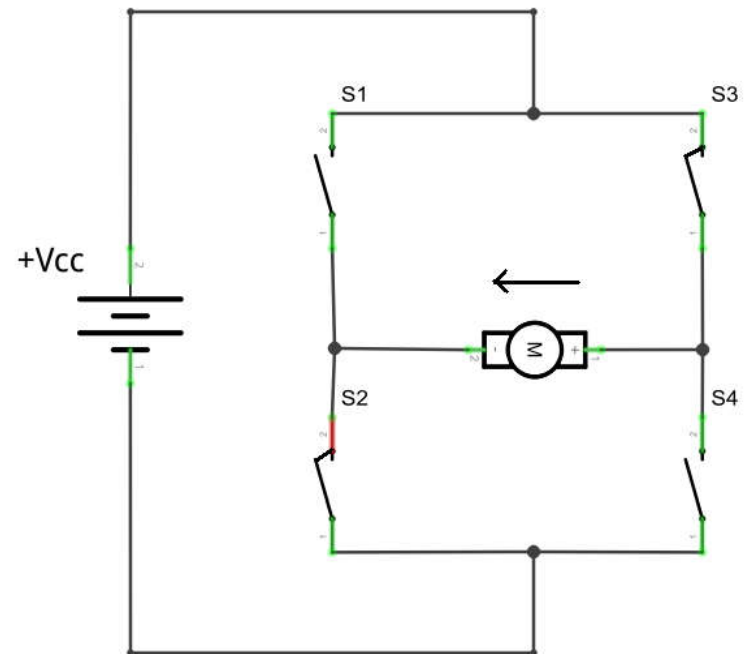
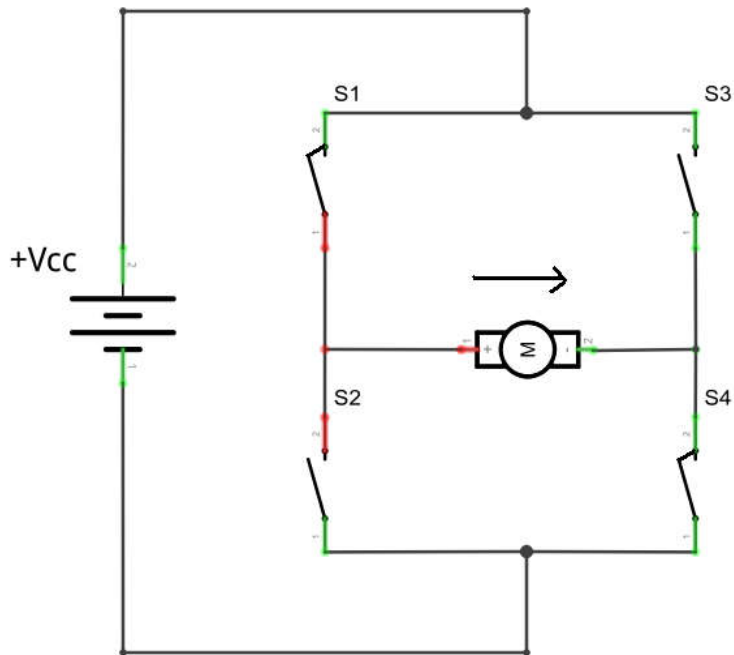
void loop()
{
  int potvalue=analogRead(A0);
  int motorspeed=potvalue/4;
  analogWrite (3,motorspeed);
  delay(5);
}
```

Γέφυρα Η

Η **γέφυρα Η** (H-Bridge) είναι ένα κύκλωμα τεσσάρων διακοπών το οποίο επιτρέπει την εφαρμογή τάσης σε ένα φορτίο (συνήθως μοτέρ) και στις δύο κατευθύνσεις.



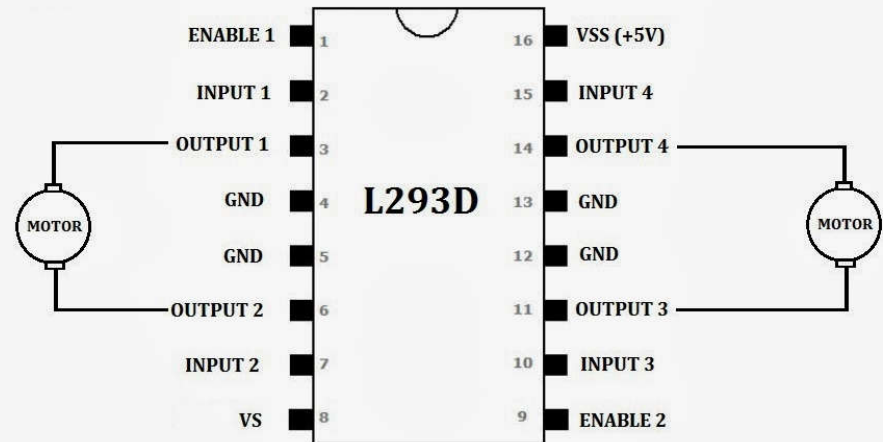
Γέφυρα Η



S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor free runs
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes
1	1	0	0	Short Circuit
0	0	1	1	Short Circuit
1	1	1	1	Short Circuit

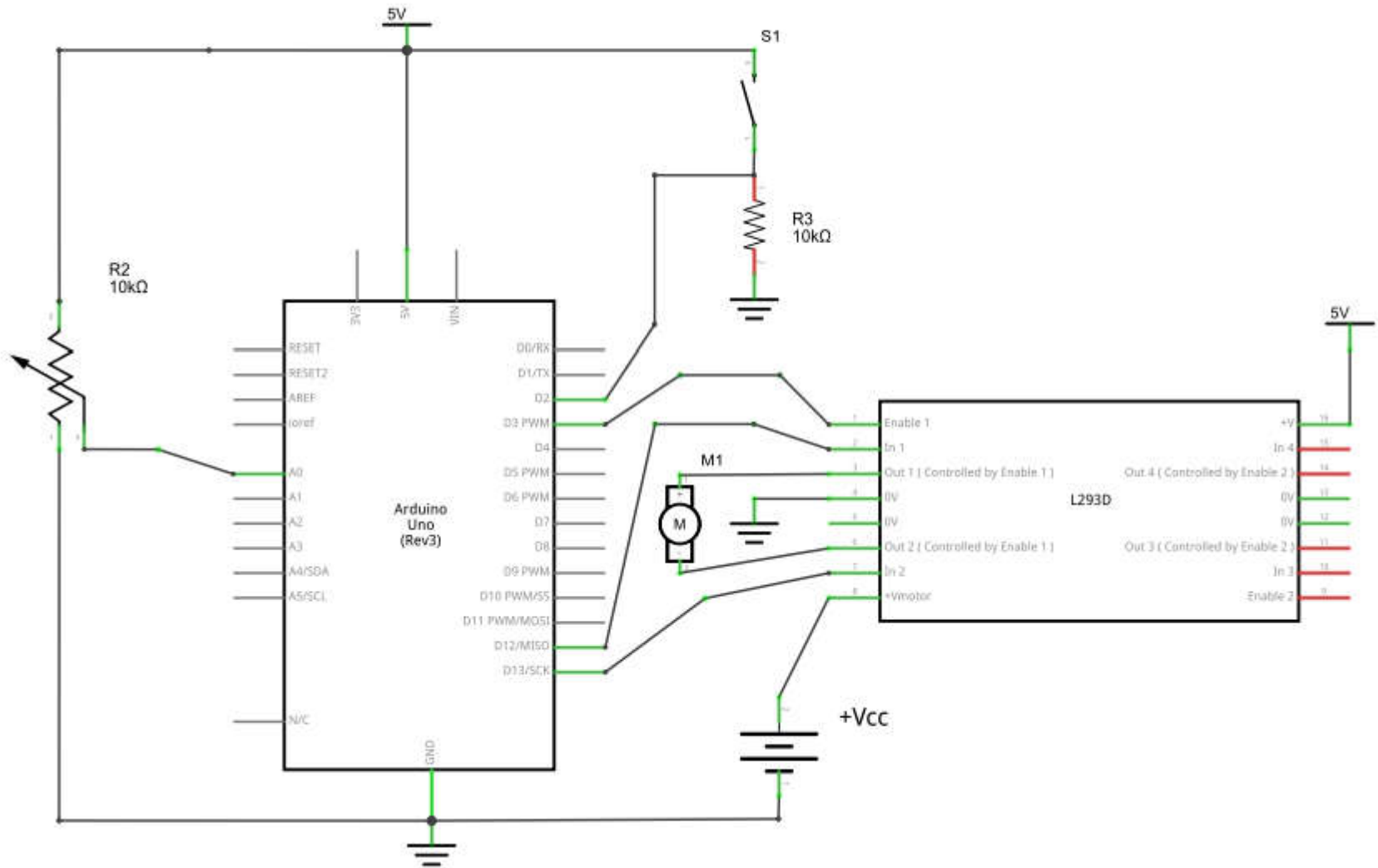
L293D (Dual H-Bridge Controller)

Pin No	Pin Name	Function
1	Enable1	Enable Pin for Motor1 (PWM)
2	Input1	Motor1 Control Pin 1
3	Output1	Motor1 Output 1
4,5,12,13	GND	Ground
6	Output2	Motor1 Output2
7	Input2	Motor1 Control Pin 2
8	Vs	Motors Power Supply (4,5V-36V)
9	Enable2	Enable Pin for Motor2 (PWM)
10	Input3	Motor2 Control Pin 3
11	Output3	Motor2 Output 3
14	Output4	Motor2 Output 4
15	Input4	Motor2 Control Pin 4
16	Vss	Power Supply for the chip (+5V)



Dual DC Motor Controller

Εφαρμογή 3 (Έλεγχος Φοράς και Περιστροφής DC Motor με Γέφυρα H και 1 διακόπτη)



Εφαρμογή 3 (Έλεγχος Φοράς και Ταχύτητας DC Motor με Γέφυρα H)

L293d

```
/* Elegchos foras peristrofis kai taxititas 1 dc motor me H-Bridge L293D
   0 elegchos tis foras ginetai me ena diakopti
   0 elegchos strofwn ginetai me ena potensiometro 5k i 10k
*/

int pwmpin=3;           //to pin 3 tou arduino sto pin 1 tou L293d
                       // elegxei tin taxitita tou moter me PWM.

//Ta pin 12 kai 13 elegxoun tin fora peristrofis tou moter
int rotation1=13;      //to pin 13 tou arduino sto pin 2 tou L293d
int rotation2=12;      //to pin 12 tou arduino sto pin 7 tou L293d
int switchpin=2;       // sto pin 2 sindeetai enas diakoptis mesw
                       // pull down resistor 10k

boolean statel=true;
boolean state2=false;
int potpin=A0;         //sto A0 sindeetai ena pot
int motorspeed;

void setup()
{
  pinMode(pwmpin,OUTPUT);
  pinMode(rotation1,OUTPUT);
  pinMode(rotation2,OUTPUT);
  pinMode(switchpin,INPUT);
  pinMode(potpin,INPUT);
  digitalWrite(rotation1,statel);
  digitalWrite(rotation2,state2);
}
```

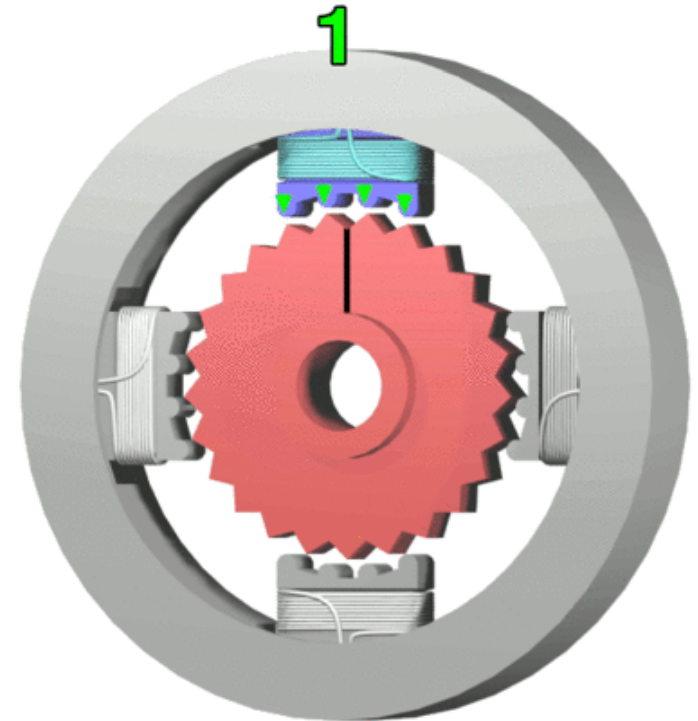
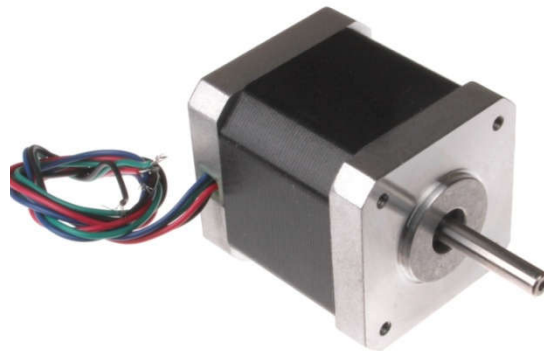
```
void loop()
{
  motorspeed=analogRead(potpin);
  analogWrite(pwmpin,motorspeed);
  if (digitalRead(switchpin)==HIGH)
  {
    delay(250); //kathisterisi gia debounce
    statel=!statel;
    state2=!state2;
    digitalWrite(rotation1,statel);
    digitalWrite(rotation2,state2);
  }
}
```

Stepper Motor (Βηματικός Κινητήρας)

Ο βηματικός κινητήρας είναι ένας DC κινητήρας ο οποίος διαιρεί μια πλήρη περιστροφή του άξονά του σε ίσο αριθμό βημάτων.

Στην πιο απλή του μορφή αποτελείται από δύο ή τέσσερις ηλεκτρομαγνήτες.

Για την περιστροφή του απαιτείται εντολοδότηση με συγκεκριμένη ακολουθία παλμών.

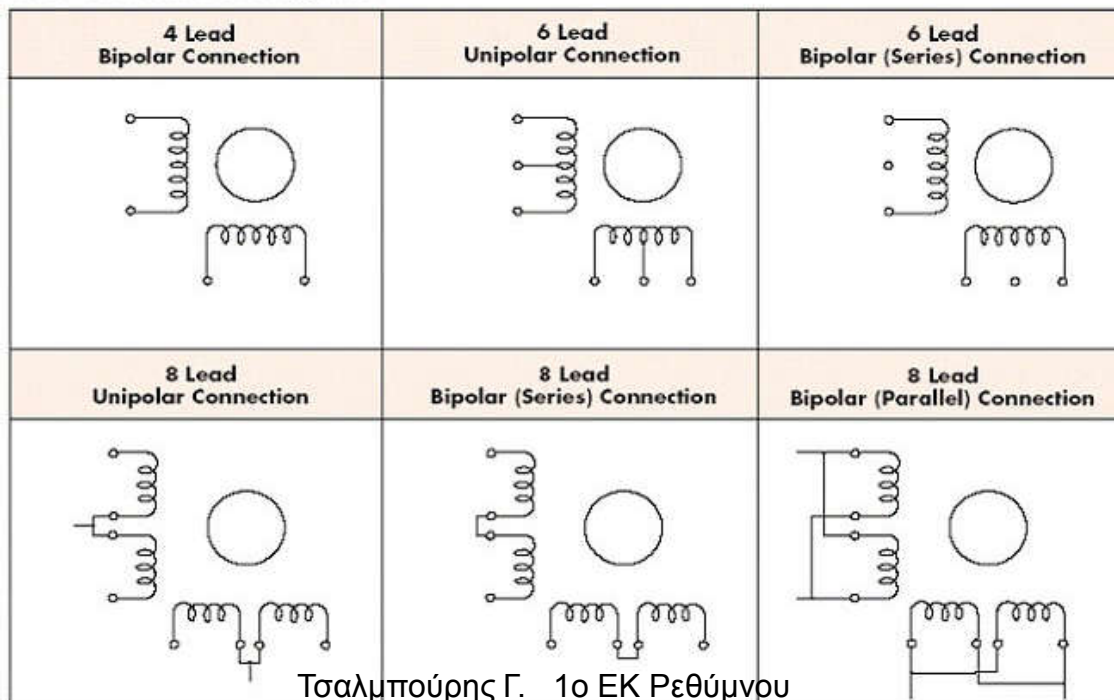


Τύποι Βηματικών Κινητήρων

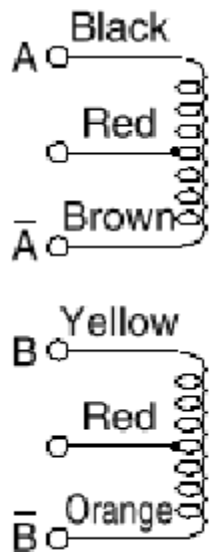
Ανάλογα με την τοπολογία των τυλιγμάτων τους χωρίζονται σε δύο βασικές κατηγορίες:

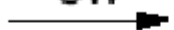
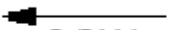
- 1) **Μονοπολικούς (Unipolar)** (Μεσαία Λήψη σε κάθε φάση)
- 2) **Διπολικούς (Bipolar)** (Χωρίς Μεσαία Λήψη)

Wire Connection Diagrams

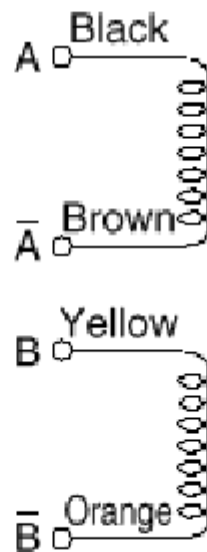


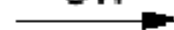

Unipolar Mode



	CW  CCW 			
Red	+	+	+	+
Red	+	+	+	+
Black	-	-		
Brown			-	-
Yellow	-			-
Orange		-	-	

Bipolar Mode

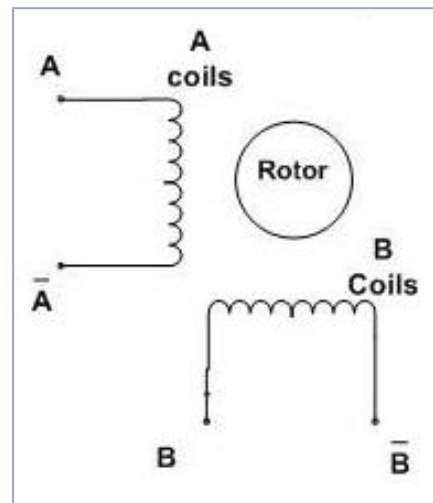


	CW  CCW 			
Black	-	-	+	+
Brown	+	+	-	-
Yellow	-	+	+	-
Orange	+	-	-	+

Ακολουθία Παλμών Bipolar Mode

Wave Drive			
A+	A-	B+	B-
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1

Full Step Drive			
A+	A-	B+	B-
1	0	0	1
1	0	1	0
0	1	1	0
0	1	0	1



Εφαρμογή 4 Οδήγηση Bipolar Stepper (Wave Drive)

stepper2

```
//Elegxos stepper se Bipolar mode Wave Drive
//me H-bridge me to L293D

int controll1=13; //pin13 του Arduino στο pin 2 του L293D
int control2=12; //pin12 του Arduino στο pin 7 του L293D
int control3=8; //pin8 του Arduino στο pin 10 του L293D
int control4=7; //pin7 του Arduino στο pin 15 του L293D

void setup()
{
  pinMode(controll1, OUTPUT);
  pinMode(control2, OUTPUT);
  pinMode(control3, OUTPUT);
  pinMode(control4, OUTPUT);
}

void loop()
{
  pulse1();
  delay(1000);
  pulse2();
  delay(1000);
  pulse3();
  delay(1000);
  pulse4();
  delay(1000);
}
```

```
void pulse1()
{
  digitalWrite(controll1,HIGH);
  digitalWrite(control2,LOW);
  digitalWrite(control3,LOW);
  digitalWrite(control4,LOW);
}

void pulse2()
{
  digitalWrite(controll1,LOW);
  digitalWrite(control2,LOW);
  digitalWrite(control3,HIGH);
  digitalWrite(control4,LOW);
}

void pulse3()
{
  digitalWrite(controll1,LOW);
  digitalWrite(control2,HIGH);
  digitalWrite(control3,LOW);
  digitalWrite(control4,LOW);
}

void pulse4()
{
  digitalWrite(controll1,LOW);
  digitalWrite(control2,LOW);
  digitalWrite(control3,LOW);
  digitalWrite(control4,HIGH);
}
```

Εφαρμογή 4 Οδήγηση Bipolar Stepper (Full Step Drive)

```
stepper
//Elegxos stepper se Bipolar mode Full Step Drive
//me H-bridge me to L293D

int controll1=13;
int control2=12;
int control3=8;
int control4=7;

void setup()
{
  pinMode(controll1, OUTPUT);
  pinMode(control2, OUTPUT);
  pinMode(control3, OUTPUT);
  pinMode(control4, OUTPUT);
}

void loop()
{
  pulse1();
  delay(1000);
  pulse2();
  delay(1000);
  pulse3();
  delay(1000);
  pulse4();
  delay(1000);
}
```

```
void pulse1()
{
  digitalWrite(controll1,HIGH);
  digitalWrite(control2,LOW);
  digitalWrite(control3,LOW);
  digitalWrite(control4,HIGH);
}

void pulse2()
{
  digitalWrite(controll1,HIGH);
  digitalWrite(control2,LOW);
  digitalWrite(control3,HIGH);
  digitalWrite(control4,LOW);
}

void pulse3()
{
  digitalWrite(controll1,LOW);
  digitalWrite(control2,HIGH);
  digitalWrite(control3,HIGH);
  digitalWrite(control4,LOW);
}

void pulse4()
{
  digitalWrite(controll1,LOW);
  digitalWrite(control2,HIGH);
  digitalWrite(control3,LOW);
  digitalWrite(control4,HIGH);
}
```

Εφαρμογή 4 Οδήγηση Bipolar Stepper (Full Step Drive)

```
stepper5
// Stepper Control se Bipolar Mode me H Bridge
//Ektelountai 20 steps deksiostrofa kai 20 aristerostrofa

int input1=13;
int input2=12;
int input3=8;
int input4=7;

void setup()
{
  pinMode(input1, OUTPUT);
  pinMode(input2, OUTPUT);
  pinMode(input3, OUTPUT);
  pinMode(input4, OUTPUT);
}

void loop()
{
  for (int i=0;i<5;i++)
  {
    pulse1();
    pulse2();
    pulse3();
    pulse4();
  }
  for (int j=0;j<5;j++)
  {
    pulse4();
    pulse3();
    pulse2();
    pulse1();
  }
}
```

```
void pulse1()
{
  digitalWrite(input1,LOW);
  digitalWrite(input2,HIGH);
  digitalWrite(input3,LOW);
  digitalWrite(input4,HIGH);
  delay(500);
}

void pulse2()
{
  digitalWrite(input1,LOW);
  digitalWrite(input2,HIGH);
  digitalWrite(input3,HIGH);
  digitalWrite(input4,LOW);
  delay(500);
}

void pulse3()
{
  digitalWrite(input1,HIGH);
  digitalWrite(input2,LOW);
  digitalWrite(input3,HIGH);
  digitalWrite(input4,LOW);
  delay(500);
}

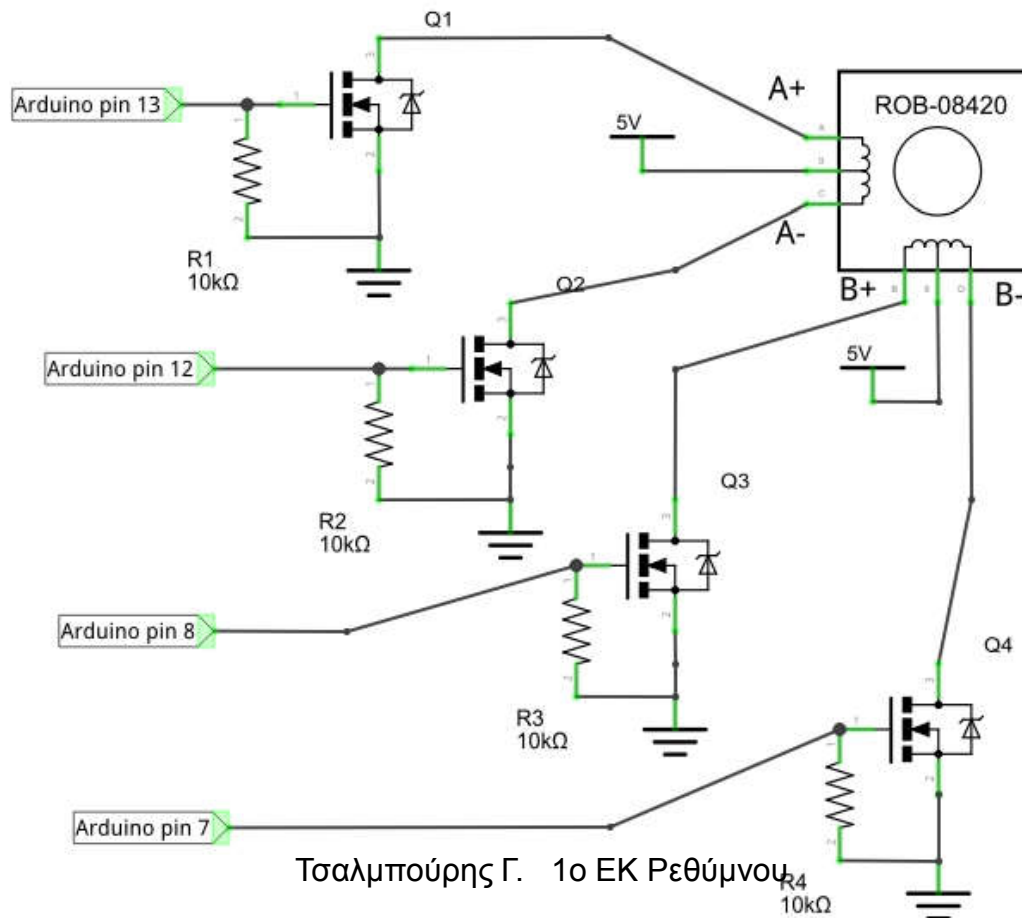
void pulse4()
{
  digitalWrite(input1,HIGH);
  digitalWrite(input2,LOW);
  digitalWrite(input3,LOW);
  digitalWrite(input4,HIGH);
  delay(500);
}
```

Οδήγηση Unipolar Stepper

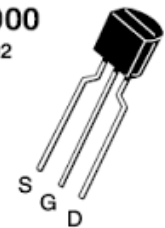
Ένας βηματικός κινητήρας σε Unipolar mode μπορεί να οδηγηθεί με 4 τέσσερα διπολικά τρανζίστορ ή mosfet.

Απαιτείται εξωτερική πηγή τροφοδοσίας.

!!! Προσοχή το GND του Arduino πρέπει να ενωθεί με τη γη του κυκλώματος.



2N7000
TO-92

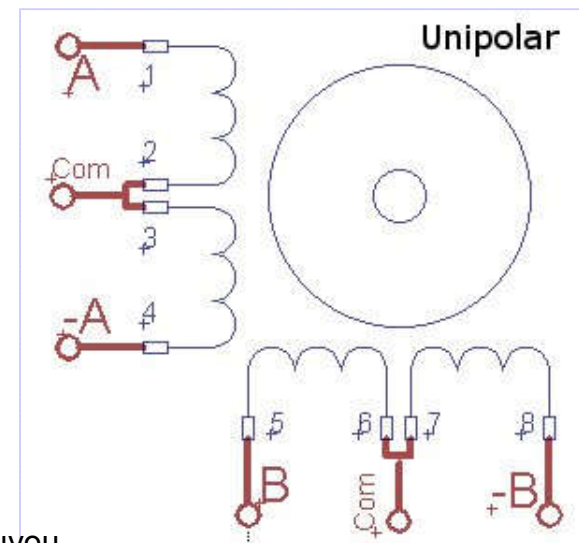


Ακολουθία Παλμών Unipolar Mode

Wave Drive			
A+	A-	B+	B-
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1

Full Step Drive			
A+	A-	B+	B-
0	1	1	0
0	1	0	1
1	0	0	1
1	0	1	0

Με αντιστροφή της σειράς των παλμών αλλάζει φορά ο κινητήρας.



Εφαρμογή 5 Οδήγηση Unipolar Stepper (Wave Drive)

```
stepper3
// Stepper Motor Unipolar Mode Wave Drive me 4 N-Channel Mosfet

int controll1=13; //to pin 13 stin pili tou Q1, elegxei to A+
int controll2=12; //to pin 12 stin pili tou Q2, elegxei to A-
int controll3=8; //to pin 8 stin pili tou Q3, elegxei to B+
int controll4=7; //to pin 7 stin pili tou Q4, elegxei to B-

void setup()
{
  pinMode(controll1, OUTPUT);
  pinMode(controll2, OUTPUT);
  pinMode(controll3, OUTPUT);
  pinMode(controll4, OUTPUT);
}

void loop()
{
  pulse1();
  delay(1000);
  pulse2();
  delay(1000);
  pulse3();
  delay(1000);
  pulse4();
  delay(1000);
}
```

```
void pulse1()
{
  digitalWrite(controll1,LOW);
  digitalWrite(controll2,LOW);
  digitalWrite(controll3,LOW);
  digitalWrite(controll4,HIGH);
}
void pulse2()
{
  digitalWrite(controll1,LOW);
  digitalWrite(controll2,HIGH);
  digitalWrite(controll3,LOW);
  digitalWrite(controll4,LOW);
}
void pulse3()
{
  digitalWrite(controll1,LOW);
  digitalWrite(controll2,LOW);
  digitalWrite(controll3,HIGH);
  digitalWrite(controll4,LOW);
}
void pulse4()
{
  digitalWrite(controll1,HIGH);
  digitalWrite(controll2,LOW);
  digitalWrite(controll3,LOW);
  digitalWrite(controll4,LOW);
}
```

Εφαρμογή 5 Οδήγηση Unipolar Stepper (Full Step Drive)

stepper4

```
// Stepper Motor Unipolar Mode Full Step Drive me 4 N-Channel Mosfet

int controll1=13; //to pin 13 stin pili tou Q1 elegxei to A+
int control2=12; //to pin 12 stin pili tou Q2 elegxei to A-
int control3=8; //to pin 8 stin pili tou Q3 elegxei to B+
int control4=7; //to pin 7 stin pili tou Q4 elegxei to B-

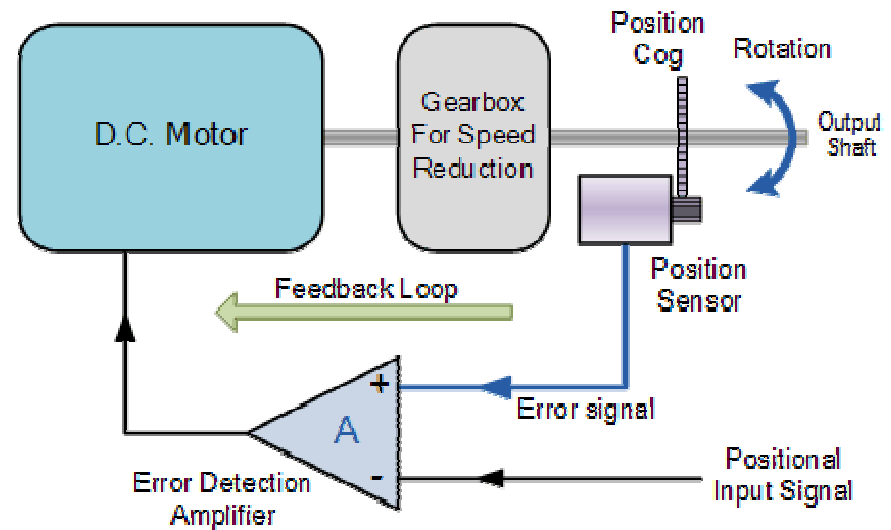
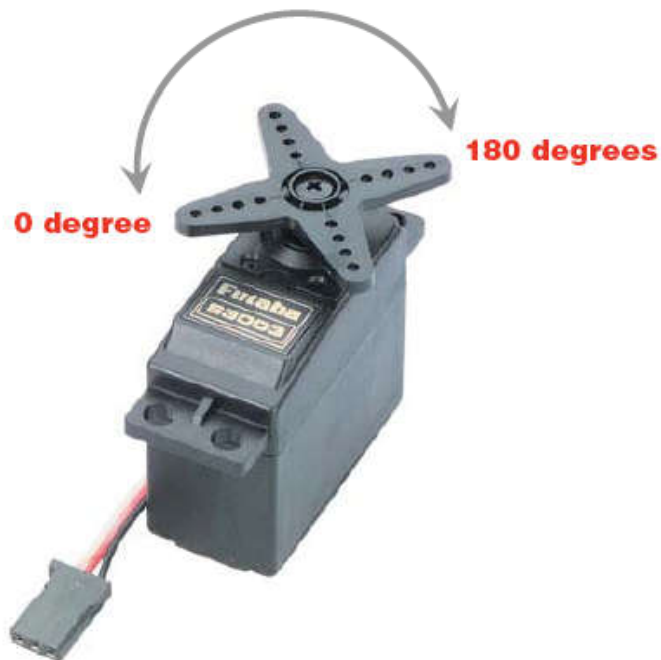
void setup()
{
  pinMode(controll1, OUTPUT);
  pinMode(control2, OUTPUT);
  pinMode(control3, OUTPUT);
  pinMode(control4, OUTPUT);
}

void loop()
{
  pulse1();
  delay(1000);
  pulse2();
  delay(1000);
  pulse3();
  delay(1000);
  pulse4();
  delay(1000);
}
```

```
void pulse1()
{
  digitalWrite(controll1,HIGH);
  digitalWrite(control2,LOW);
  digitalWrite(control3,HIGH);
  digitalWrite(control4,LOW);
}
void pulse2()
{
  digitalWrite(controll1,HIGH);
  digitalWrite(control2,LOW);
  digitalWrite(control3,LOW);
  digitalWrite(control4,HIGH);
}
void pulse3()
{
  digitalWrite(controll1,LOW);
  digitalWrite(control2,HIGH);
  digitalWrite(control3,LOW);
  digitalWrite(control4,HIGH);
}
void pulse4()
{
  digitalWrite(controll1,LOW);
  digitalWrite(control2,HIGH);
  digitalWrite(control3,HIGH);
  digitalWrite(control4,LOW);
}
```

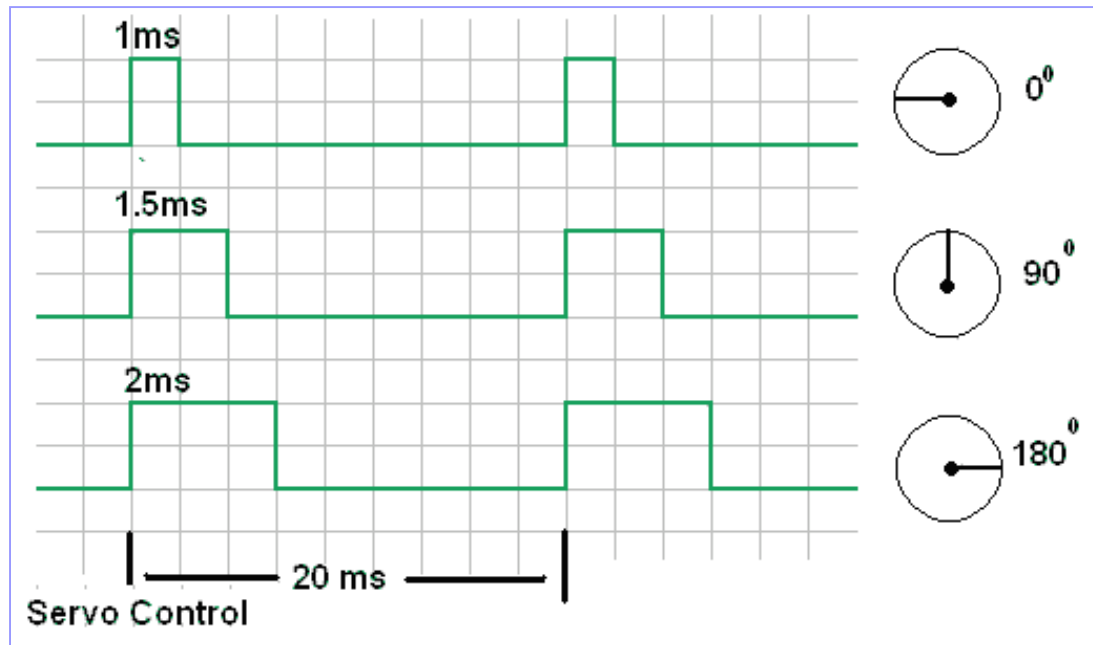

Servo Motor

Ένα **Servo Motor** είναι ένα **DC Motor** το οποίο διαθέτει επιπλέον ένα **σερβομηχανισμό**. Ο σερβομηχανισμός είναι ένα σύστημα αυτομάτου ελέγχου κλειστού βρόχου που φροντίζει για την κίνηση και την τελική θέση του άξονα του μοτέρ.

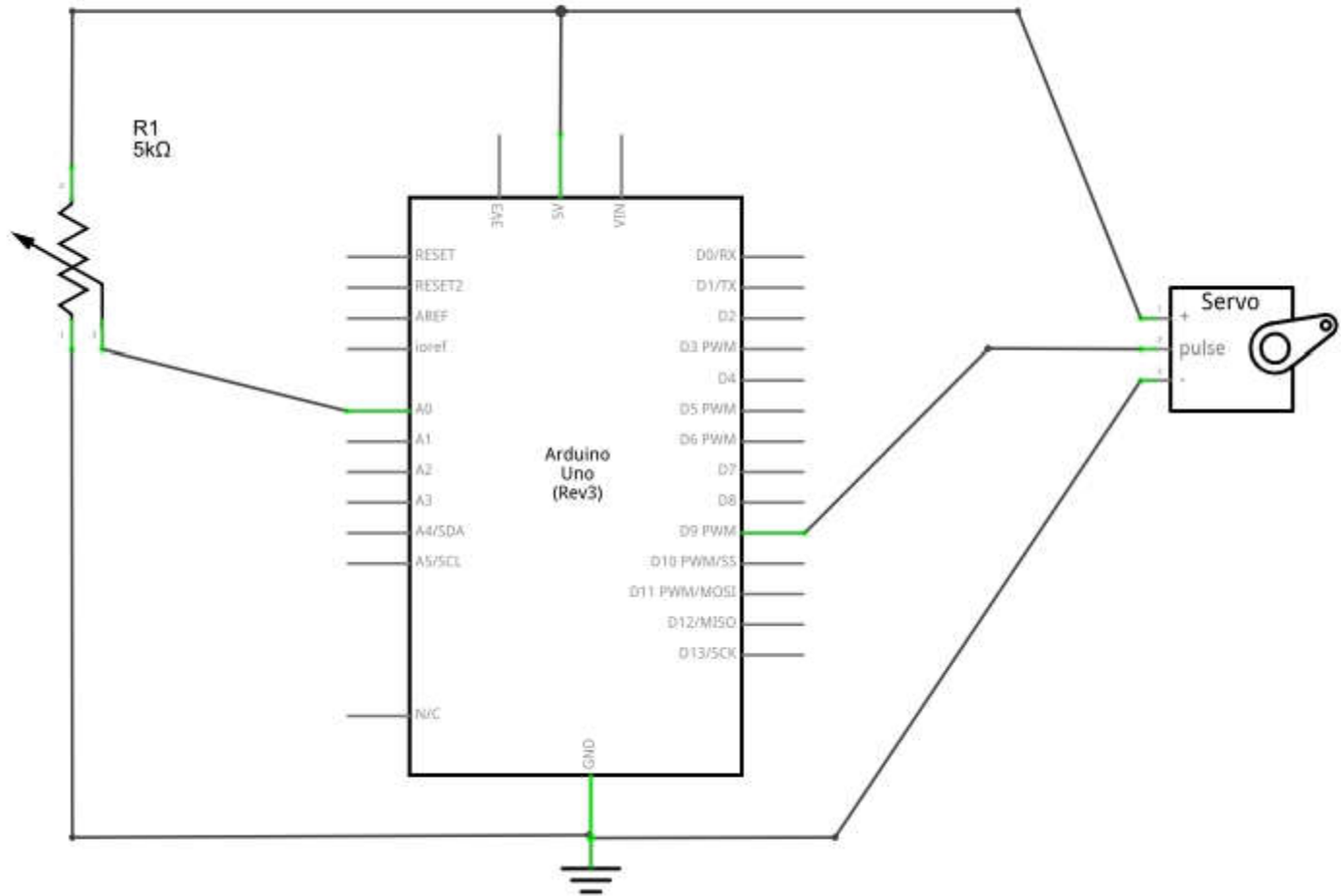


Έλεγχος Servo Motor

Τα **servo motor** διαθέτουν 3 ακροδέκτες. Οι δύο είναι για την τροφοδοσία τους ενώ το τρίτο προορίζεται για την εντολοδότησή τους. Ο έλεγχος των servo motors γίνεται με την τεχνική της PWM. Τα περισσότερα servo ελέγχονται με PWM σήμα συχνότητας 50Hz (περίοδος 20 msec).



Εφαρμογή 8 Έλεγχος θέσης Servo Motor



Εφαρμογή 6 Έλεγχος θέσης Servo Motor

```
servo_1
//Knob for Futaba S3003 Servo
// Control me pot sto A0
//Servo moves from -90 to 90 with PWM.
//T=20msec , Ton minimum=500μsec ,Ton maximum=2500μsec

int toff=0;
int ton=0;

void setup()
{
  pinMode(9,OUTPUT);
  pinMode(A0,INPUT);
}

void loop()
{
  int pot=analogRead(A0);
  ton=500+(1.96*pot);
  digitalWrite(9,HIGH);
  delayMicroseconds(ton);
  digitalWrite(9, LOW);
  toff=2500-ton;
  delayMicroseconds(toff);
  delay(18);
  delay(25);
}
```

Εφαρμογή 6 Έλεγχος θέσης Servo Motor

```
Knob
/*
Controlling a servo position using a potentiometer (variable resistor)
by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

modified on 8 Nov 2013
by Scott Fitzgerald
http://arduino.cc/en/Tutorial/Knob
*/

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); // waits for the servo to get there
}
```