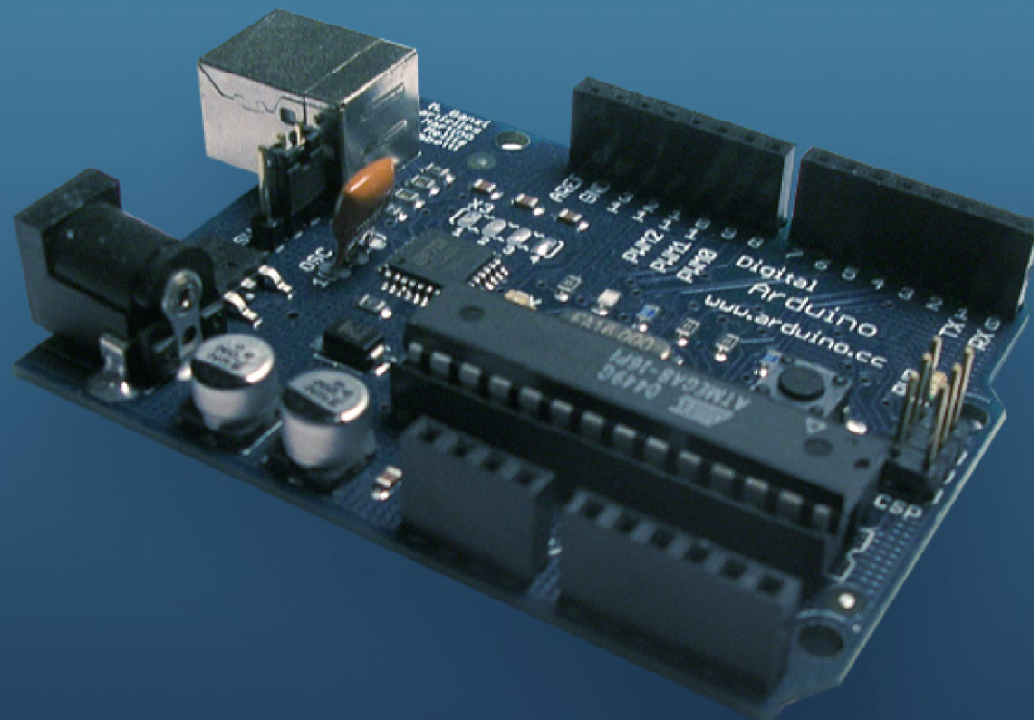


**Arduino**  
Physical Computing I/O board



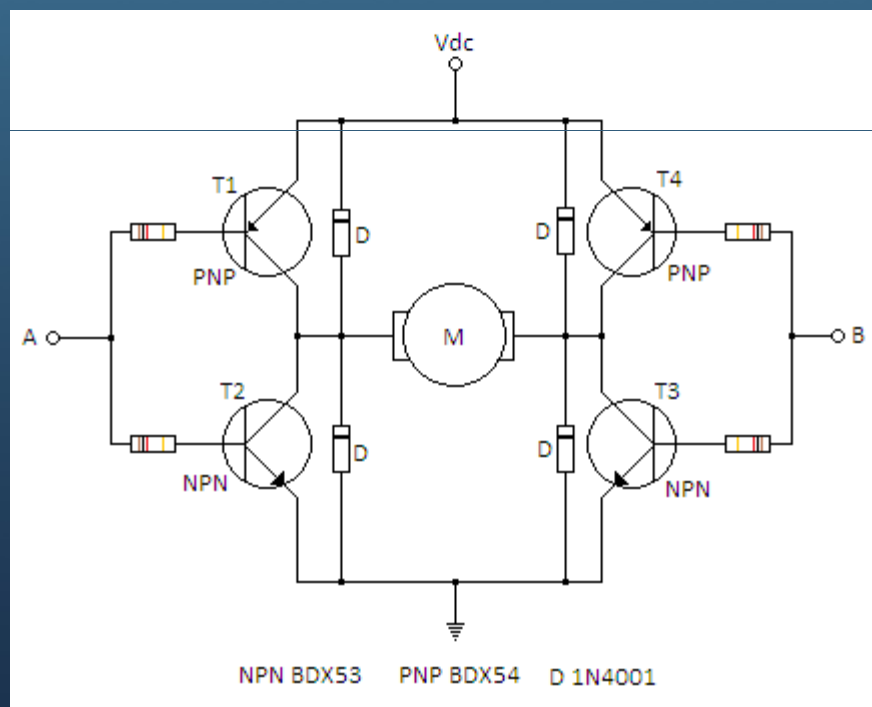
10<sup>^</sup> parte : Come gestire la rotazione di un piccolo motore DC



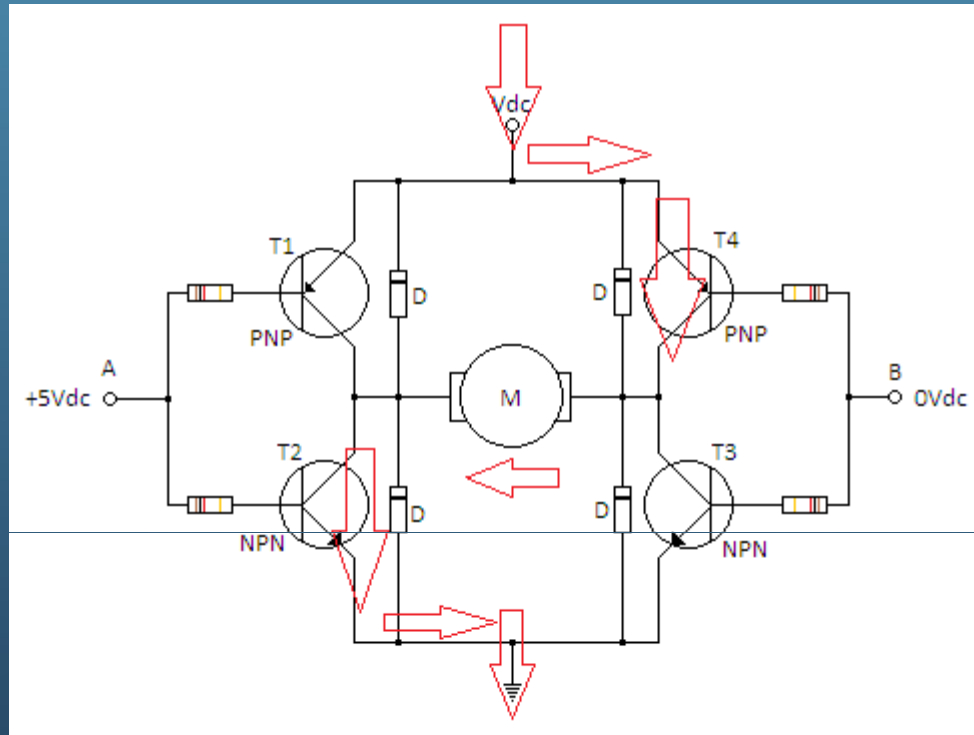
Author: Ing. Sebastiano Giannitto (ITIS "M.BARTOLO" –PACHINO)

# Come usare Arduino per gestire la rotazione di un piccolo motore DC

Come sappiamo bene le uscite di **Arduino** non possono fornire una corrente sufficiente da permettere la rotazione del motore, quindi è importante usare dei transistor (o dei circuiti integrati appositi, vedi L298) per fornire la corrente necessaria all'alimentazione del motore. Il circuito da realizzare deve anche permettere la rotazione del rotore sia in un senso che nell'altro. La soluzione più veloce è realizzare un ponte ad H che sfrutta 4 transistor, due del tipo NPN e due del tipo PNP collegati come nella figura seguente:



Il segnale di ingresso viene applicato al **pinA** e al **pinB** del ponte H. Quando A è alto e B è basso vanno in conduzione i transistor **T4** e **T2** provocando la rotazione a sinistra del motore:



I transistors che ho usato in questo schema riescono a “soportare”, se correttamente raffreddati, circa 8A ed una tensione di lavoro massima di 45 Vdc.

Testiamo il circuito usando la nostra board **Arduino**, collegando il **pinA** al pin digitale 5 e il **pinB** al pin digitale 6, in modo da utilizzare anche il segnale **PWM** disponibile su questi pin.

Il codice di esempio seguente e si avvale del serial monitor per decidere se far girare il motore a destra o a sinistra. Utilizziamo i pin5 e pin6 perché possono anche generare un segnale pwm, che utilizzerò successivamente per variare la velocità di rotazione.

Codice di esempio:

```
void setup()
```

```
{  
  //pin5 e pin6 in uscita  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  
  //pin5 e pin6 bassi  
  digitalWrite(5, LOW);  
  digitalWrite(6, LOW);  
  
  //Serial monitor init  
  Serial.begin(9600);  
}
```

```
void loop()
```

```
{  
  //attendi un comando dalla serial  
  if( Serial.available())  
  {  
    byte DataRX = Serial.read();  
    //carattere 1 avvia il motore a SX  
    if(DataRX == 49)  
    {  
      digitalWrite(5, HIGH);  
      digitalWrite(6, LOW);  
    }  
  
    //carattere 2 avvia il motore a DX  
    if(DataRX == 50)  
    {  
      digitalWrite(5, LOW);  
      digitalWrite(6, HIGH);  
    }  
  
    //Carattere 0 ferma il motore  
    if(DataRX == 48)  
    {  
      digitalWrite(5, LOW);  
      digitalWrite(6, LOW);  
    }  
  }  
}
```

Il codice precedente permette di avviare il motore e di cambiarne il verso di rotazione in base al carattere che spediamo tramite il serial monitor.

Dato che i pin 5 e 6 di Arduino possono generare un segnale PWM, modifichiamo il codice come di seguito:

```
byte Velocita = 125;
byte DataRX = -1;

void setup()
{
  //pin5 e pin6 in uscita
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

  //pin5 e pin6 bassi
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);

  //Serial monitor init
  Serial.begin(9600);
}
```

```
void loop()
{
  //attendi un comando dalla serial
  if( Serial.available())
  {
    DataRX = Serial.read();
    Serial.println(Velocita);

    //incrementa la velocita inviando il numero 5
    if(DataRX == 53)
    {
      Velocita += 10;
      if (Velocita >= 255)
        Velocita = 255;
    }

    //decrementa la velocita inviando il numero 6
    if(DataRX == 54)
    {
      Velocita -= 10;
      if (Velocita <= 0)
        Velocita = 0;
    }
  }
}
```

```

//carattere 1 avvia il motore a SX
  if(DataRX == 49)
  {
    analogWrite(5, Velocita);
    digitalWrite(6, LOW);
  }

//carattere 2 avvia il motore a DX
  if(DataRX == 50)
  {
    digitalWrite(5, LOW);
    analogWrite(6, Velocita);
  }

//Carattere 0 ferma il motore
  if(DataRX == 48)
  {
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
  }
}
}
}

```

Questo esempio permette di variare la velocità spedendo, tramite il serial monitor, il numero 5 (codice ASCII 53) per incrementare la velocità e il numero 6 (codice ASCII 54) per decrementarla. Dopo aver cambiato il valore della variabile Velocita, bisogna rispettare il carattere 1 o il carattere 2 per confermare il nuovo valore di pwm.

Naturalmente esistono dei circuiti integrati che svolgono egregiamente il pilotaggio di motori unipolari e bipolari, uno dei più utilizzati in campo hobbistico è il driver L298 . Vedremo nei prossimi tutorial come utilizzarlo con un motore bipolare.