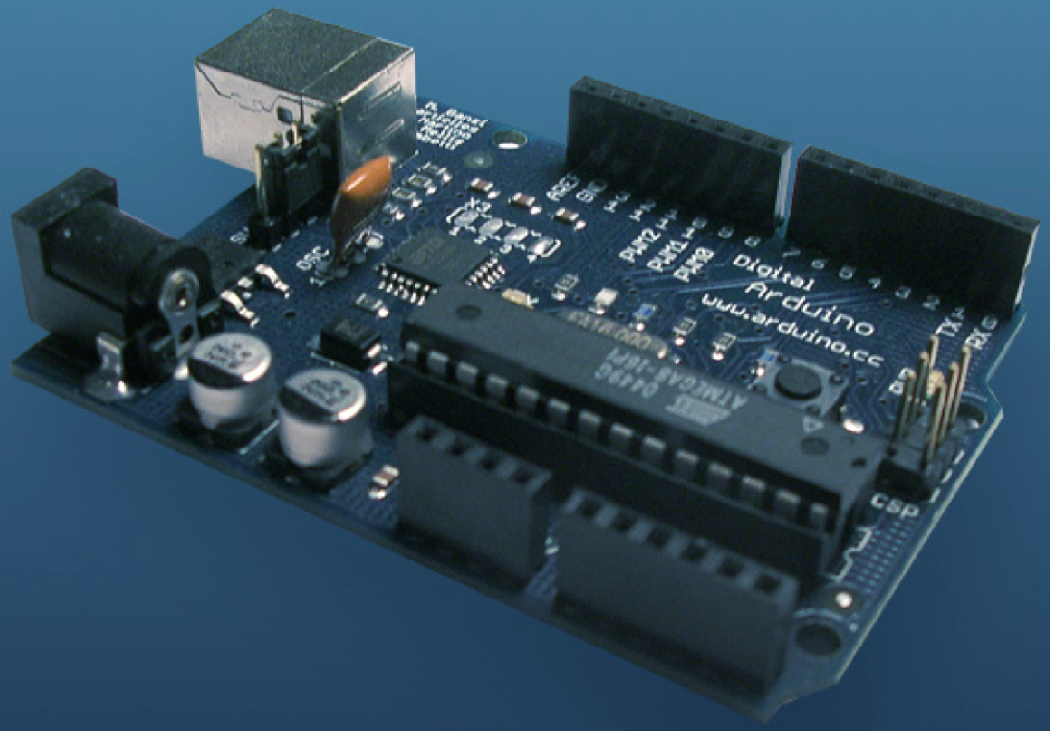
Arduino

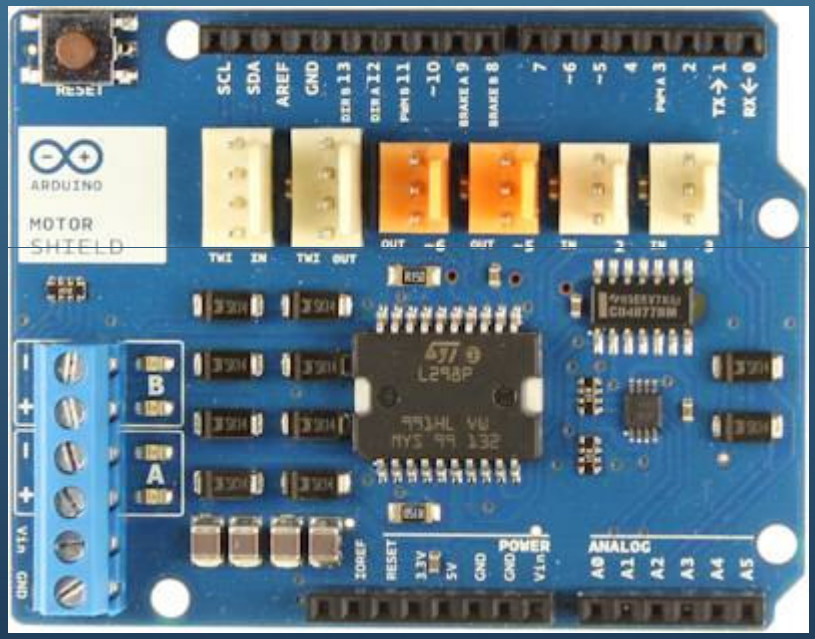
Physical Computing I/O board



9^ parte : Come pilotare motori dc utilizzando il motor shield

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

I motorini dc sono molto comuni nei vecchi giocatoli, possiamo recuperarli per costruire nuovi giochi, dei robot o dei rover utilizzando una scheda Arduino o Netduino. Queste schede, come è noto, non permettono di pilotare direttamente un motore dc perché la corrente in uscita dai pin è insufficiente ed il rischio di danneggiare irreparabilmente la scheda è inevitabile.
Esistono in commercio numerosi shield che dispongono di driver capaci di alimentare questi motorini dc, genericamente chiamati motor shield.



Quello che utilizziamo per questo tutorial è il motor shield ufficiale versione R3. Lo shield si basa sul driver L298P prodotto da ST microelectronics, configurabile anche per pilotare motori passo passo

Una volta collegato il motor shield all'Arduino UNO dobbiamo semplicemente procurarci due motorini dc con assorbimenti inferiori a 2A (solitamente quelli nei giocatoli assorbono circa 150-200 mA) e collegarli utilizzando un semplice cacciavite alla morsettiera.
Il primo motorino lo colleghiamo alla morsettiera A il secondo alla morsettiera B. Nella morsettiera contrassegnata da Vin dobbiamo collegare un alimentatore che fornisca la corrente sufficiente per alimentare Arduino e i motorini.

Come usare il motor shield

La tabella seguente elenca i pin che dobbiamo configurare per poter pilotare i motorini:

Funzione	Canale **A**	Canale **B**
Direzione	**pin 12**	**pin 13**
Segnale PWM	**pin 3**	**pin 11**
Freno	**pin 9**	**pin 8**
Assorbimento Corrente	**pin A0**	**pin A1**

Il motorino collegato nella morsettiera A verrà comandato dai pin 12, 3 e 9. Il pin 12 imposta la direzione di rotazione, settando a livello alto il motorino girerà in un senso settando a livello basso girerà in senso contrario. Il pin 3 (PWM) serve a far girare il motore, ponendo l'uscita alta il motorino girerà alla massima velocità, per fermarlo dovremmo settarlo a livello logico basso. Avvalendoci invece del segnale PWM potremo variare la velocità del motorino cambiando la percentuale del duty cicle. Per il canale B valgono le stesse regole solo che i pin interessati sono il 13, 11 e 8.

Sugli ingressi analogici A0 e A1 sarà presente un valore di tensione proporzionale alla corrente assorbita dai motori pari a 1.65V per Amper assorbito.

Codice di esempio:

```
void setup()
{
  //imposta il pin 12 come uscita
  pinMode(12, OUTPUT);
  //imposto la direzione del motorino
  //pin 12 a livello logico basso
  digitalWrite(12, LOW);
  //imposta il pin 13 come uscita
  pinMode(13, OUTPUT);
  //imposto la direzione del motorino
  //pin 13 a livello logico basso
  digitalWrite(13, LOW);

  //avvio i motori collegati alla morsettiera A e B
  //imposto la velocità cambiano il valore di duty cicle
  analogWrite(3,254);
  analogWrite(11,254);
}

void loop() { }
```

il codice precedente fa girare i motorini in un senso, per cambiare verso è sufficiente impostare a livello logico alto il pin12 e il pin13:

```
void setup()
{
  //imposta il pin 12 come uscita
  pinMode(12, OUTPUT);
  //imposto la direzione del motorino
  //pin 12 a livello logico basso
  digitalWrite(12, HIGH);
  //imposta il pin 13 come uscita
  pinMode(13, OUTPUT);
  //imposto la direzione del motorino
  //pin 13 a livello logico basso
  digitalWrite(13, HIGH);

  //avvio i motori collegati alla morsettiera A e B
  //imposto la velocità cambiano il valore di duty cicle
  analogWrite(3,254);
  analogWrite(11,254);
}

void loop() { }
```

a velocità invece viene variata cambiando il valore di duty cicle del segnale pwm dei pin3 e pin11:

```
void setup()
{
  //imposta il pin 12 come uscita
  pinMode(12, OUTPUT);
  //imposto la direzione del motorino
  //pin 12 a livello logico basso
  digitalWrite(12, LOW);
  //imposta il pin 13 come uscita
  pinMode(13, OUTPUT);
  //imposto la direzione del motorino
  //pin 13 a livello logico basso
  digitalWrite(13, LOW);
   //avvio i motori collegati alla morsettiera A e B
  //imposto la velocità cambiano il valore di duty cicle
  //velocità ridotta a metà
  analogWrite(3,127);
  analogWrite(11,127);
}

void loop() { }
```