

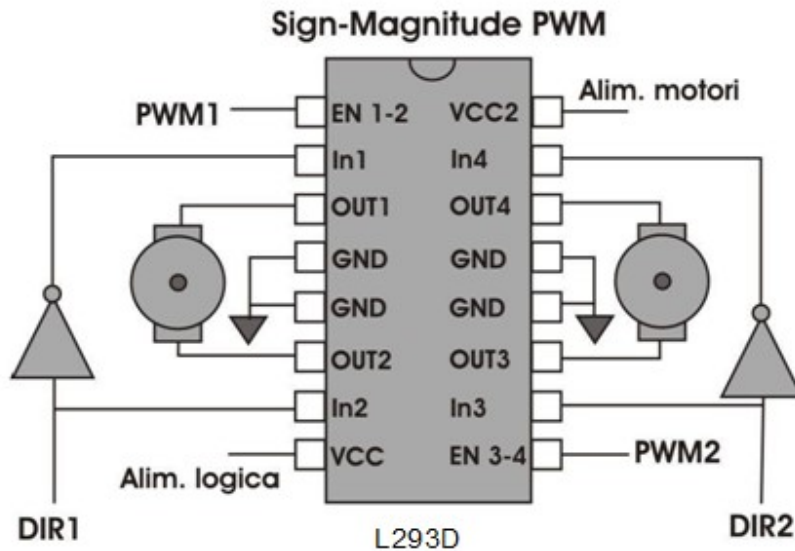
1. Pilotaggio dei motori con L293D

(http://www.tmasi.com/robotica/pwmtut/pwmtut_1.htm)

Le configurazioni per pilotare uno o più motori mediante un ponte H e un segnale PWM, sono due:

- La configurazione **Sign Magnitude PWM**:

Arduino mediante la linea digitale PWM (A o B)(IN 12) invia un segnale con periodo (e quindi anche frequenza) costanti ma con un duty cycle che varia in base alla velocità alla quale vogliamo far girare il motore e attraverso la linea digitale DIR comandiamo la direzione. Quindi inviando il segnale PWM all'ingresso enable del ponte, possiamo comandare la velocità dei motori e mediante i due ingressi di controllo la direzione . Tali due ingressi devono essere comandati da segnali invertiti, utilizzando un inverter come nello schema di figura si riduce il numero di pin del microcontrollore necessari per il controllo.



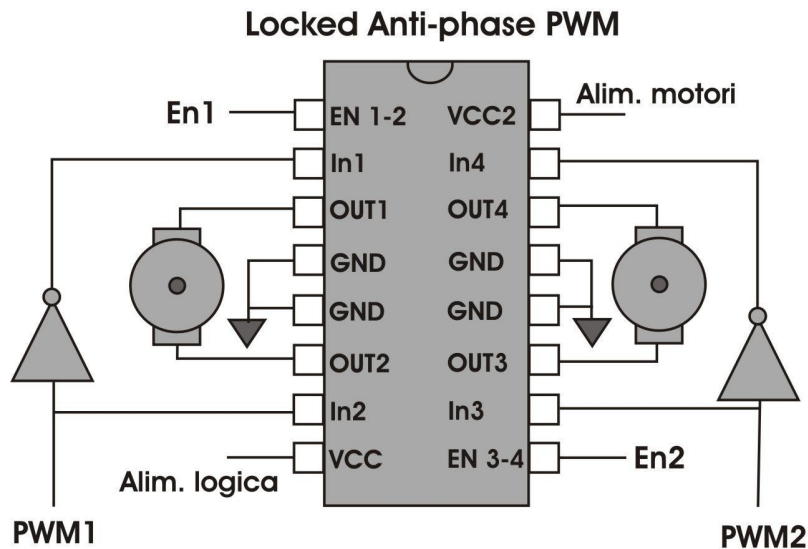
In figura DIRA corrisponde a DIR1 e DIRB corrisponde a DIR2.

- La configurazione **Locked Antiphase PWM**:

Arduino mediante la linea digitale PWM A o B (IN 12) invia un segnale con periodo (e quindi anche frequenza) costanti ma con un duty cycle che varia in base alla velocità alla quale vogliamo far girare il motore. In questo caso però la direzione del motore viene sempre decisa dal segnale digitale che viene messo in ingresso all'invertitore in modo da avere ai due lati opposti del ponte due segnali invertiti tra loro. Il verso di rotazione, viene dedotto dal duty cycle dell'onda nel seguente modo:

1. Duty cycle a 0% : rotazione alla massima velocità in un verso
2. Duty cycle al 50%: motore fermo
3. Duty cycle al 100%: rotazione alla massima velocità nell'altro verso

La linea digitale DIR (Enable) è fissa a 5V quindi agendo sull'enable è possibile spegnere il rispettivo ponte.



In figura PWMA corrisponde a PWM1 e PWMB corrisponde a PWM2.

4 Il codice per i movimenti del robot

/* Ardumoto Example Sketch

by: Jim Lindblom

date: November 8, 2013

license: **Public domain. Please use, reuse, and modify this sketch!**

Three useful functions are defined:

setupArdumoto() Setup the Ardumoto Shield pins

driveArdumoto([motor], [direction], [speed]) Drive [motor] (0

for A, 1 for B) in [direction] (0 or 1) at a [speed]

between 0 and 255. It will spin until told to stop.

stopArdumoto([motor]) Stop driving [motor] (0 or 1).

setupArdumoto() is called in the setup().

The loop() demonstrates use of the motor driving functions.

*/

// Clockwise and counterclockwise definitions.

// Depending on how you wired your motors, you may need to swap.

#define CW 0

#define CCW 1

```

// Motor definitions to make life easier:
#define MOTOR_A 0
#define MOTOR_B 1

// Pin Assignments //
// Don't change these! These pins are statically defined by shield layout
const byte PWMA = 3; // PWM control (speed) for motor A
const byte PWMB = 11; // PWM control (speed) for motor B
const byte DIRA = 12; // Direction control for motor A const
byte DIRB = 13; // Direction control for motor B

void setup()
{
  setupArdumoto(); // Set all pins as outputs
}

void loop()
{
  // Drive motor A (and only motor A) at various speeds, then stop.
  driveArdumoto(MOTOR_A, CCW, 255); // Set motor A to CCW at max
  delay(1000); // Motor A will spin as set for 1 second
  driveArdumoto(MOTOR_A, CW, 127); // Set motor A to CW at half
  delay(1000); // Motor A will keep trucking for 1 second
  stopArdumoto(MOTOR_A); // STOP motor A

  // Drive motor B (and only motor B) at various speeds, then stop.
  driveArdumoto(MOTOR_B, CCW, 255); // Set motor B to CCW at max
  delay(1000); // Motor B will spin as set for 1 second
  driveArdumoto(MOTOR_B, CW, 127); // Set motor B to CW at half
  delay(1000); // Motor B will keep trucking for 1 second
  stopArdumoto(MOTOR_B); // STOP motor B

  // Now spin both!
  driveArdumoto(MOTOR_A, CW, 255); // Motor A at max speed.
  driveArdumoto(MOTOR_B, CW, 255); // Motor B at max speed.
  delay(1000); // Drive forward for a second
  // Now go backwards at half that speed:
  driveArdumoto(MOTOR_A, CCW, 127); // Motor A at half speed.
  driveArdumoto(MOTOR_B, CCW, 127); // Motor B at half speed.
}

// driveArdumoto drives 'motor' in 'dir' direction at 'spd' speed

```

```
void driveArdumoto(byte motor, byte dir, byte spd)
{
if (motor == MOTOR_A)
{
digitalWrite(DIRA, dir);
analogWrite(PWMA, spd);
}
else if (motor == MOTOR_B)
{
digitalWrite(DIRB, dir);
analogWrite(PWMB, spd);
}
}
```

```
// stopArdumoto makes a motor stop
void stopArdumoto(byte motor)
{
driveArdumoto(motor, 0, 0);
}
```

```
// setupArdumoto initialize all pins
void setupArdumoto()
{
// All pins should be setup as outputs:
pinMode(PWMA, OUTPUT);
pinMode(PWMB, OUTPUT);
pinMode(DIRA, OUTPUT);
pinMode(DIRB, OUTPUT);

// Initialize all pins as low:
digitalWrite(PWMA, LOW);
digitalWrite(PWMB, LOW);
digitalWrite(DIRA, LOW);
digitalWrite(DIRB, LOW);
}
```