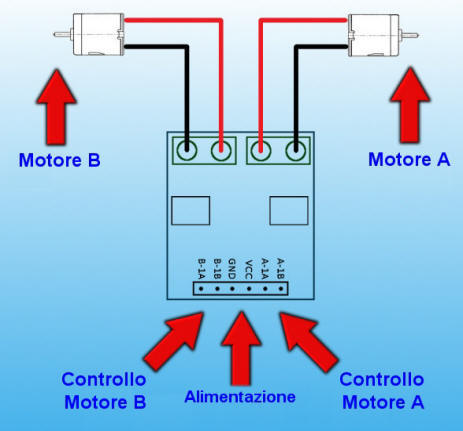
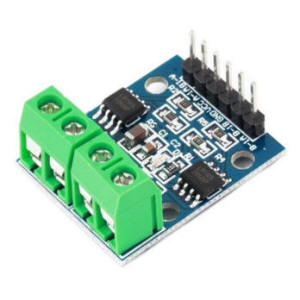
**ARDUINO H BRIGE (Ponte ad H)** fonte [www.adrirobot.it](http://www.adrirobot.it)

Il modulo L9110 consente di comandare in modo distinto due motori DC con consumo massimo di corrente di circa 750 mA ciascuno.   
E’ possibile regolare anche il numero di giri di rotazione dei motori utilizzando le uscite PWM di Arduino.

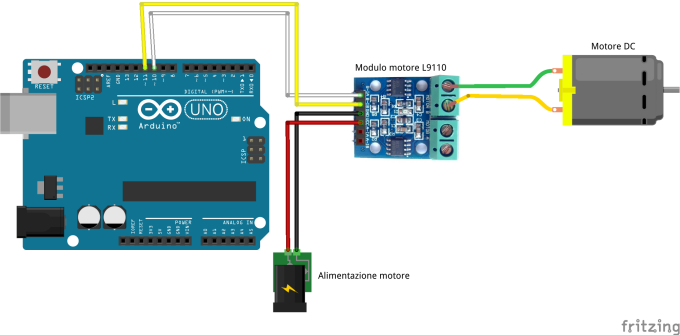


|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Pin** | **Descrizione** | | **B-IA** | Motore B Input A (IA) | | **B-IB** | Motore B Input B (IB) | | **GND** | Negativo alimentazione | | **VCC** | Positivo alimentazione 2.5-12V | | **A-IA** | Motore A Input A (IA) | | **A-IB** | Motore A Input B (IB) | |

**Foto del modulo**



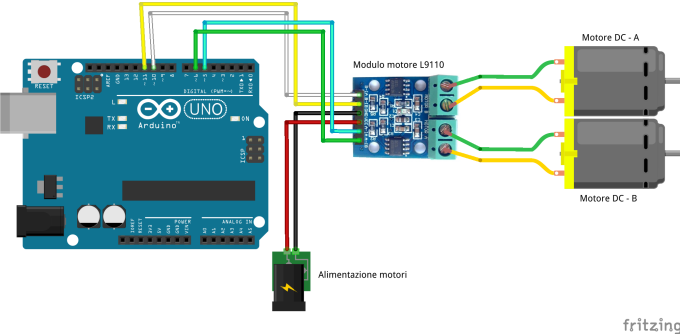
**Gestione di un motore**



[Listato programma](https://win.adrirobot.it/motor_driver/L9110_HG7881/Test_L9110_singolo.zip)

|  |  |
| --- | --- |
| |  | | --- | | // wired connections  #define HG7881\_B\_IA 10 // D10 --> Motor B Input A --> MOTOR B +  #define HG7881\_B\_IB 11 // D11 --> Motor B Input B --> MOTOR B -  // functional connections  #define MOTOR\_B\_PWM HG7881\_B\_IA // Motor B PWM Speed  #define MOTOR\_B\_DIR HG7881\_B\_IB // Motor B Direction  // the actual values for "fast" and "slow" depend on the motor  #define PWM\_SLOW 50  // arbitrary slow speed PWM duty cycle  #define PWM\_FAST 200 // arbitrary fast speed PWM duty cycle  #define DIR\_DELAY 1000 // brief delay for abrupt motor changes  void setup()  {   Serial.begin( 9600 );   pinMode( MOTOR\_B\_DIR, OUTPUT );   pinMode( MOTOR\_B\_PWM, OUTPUT );   digitalWrite( MOTOR\_B\_DIR, LOW );   digitalWrite( MOTOR\_B\_PWM, LOW );  }  void loop()  {   boolean isValidInput;   // draw a menu on the serial port   Serial.println( "-----------------------------" );   Serial.println( "MENU:" );   Serial.println( "1) Avanti veloce" );   Serial.println( "2) Avanti" );   Serial.println( "3) Soft Stop (Ruote libere)" );   Serial.println( "4) Indietro" );   Serial.println( "5) Indietro veloce" );   Serial.println( "6) Hard stop (Ruote bloccate)" );   Serial.println( "-----------------------------" );   do   {     byte c;     // get the next character from the serial port     Serial.print( "?" );     while ( !Serial.available() )       ; // LOOP...     c = Serial.read();     // execute the menu option based on the character receved     switch ( c )     {       case '1': // 1) Fast forward         Serial.println( "Avanti veloce..." );         // always stop motors briefly before abrupt changes         digitalWrite( MOTOR\_B\_DIR, LOW );         digitalWrite( MOTOR\_B\_PWM, LOW );         delay( DIR\_DELAY );         // set the motor speed and direction         digitalWrite( MOTOR\_B\_DIR, HIGH ); // direction = forward         analogWrite( MOTOR\_B\_PWM, 255 - PWM\_FAST ); // PWM speed = fast         isValidInput = true;         break;       case '2': // 2) Forward         Serial.println( "Avanti..." );         // always stop motors briefly before abrupt changes         digitalWrite( MOTOR\_B\_DIR, LOW );         digitalWrite( MOTOR\_B\_PWM, LOW );         delay( DIR\_DELAY );         // set the motor speed and direction         digitalWrite( MOTOR\_B\_DIR, HIGH ); // direction = forward         analogWrite( MOTOR\_B\_PWM, 255 - PWM\_SLOW ); // PWM speed = slow         isValidInput = true;         break;       case '3': // 3) Soft stop (preferred)         Serial.println( "Soft Stop (Ruote libere)..." );         digitalWrite( MOTOR\_B\_DIR, LOW );         digitalWrite( MOTOR\_B\_PWM, LOW );         isValidInput = true;         break;       case '4': // 4) Reverse         Serial.println( "Indietro..." );         // always stop motors briefly before abrupt changes         digitalWrite( MOTOR\_B\_DIR, LOW );         digitalWrite( MOTOR\_B\_PWM, LOW );         delay( DIR\_DELAY );         // set the motor speed and direction         digitalWrite( MOTOR\_B\_DIR, LOW ); // direction = reverse         analogWrite( MOTOR\_B\_PWM, PWM\_SLOW ); // PWM speed = slow         isValidInput = true;         break;       case '5': // 5) Fast reverse         Serial.println( "Indietro veloce..." );         // always stop motors briefly before abrupt changes         digitalWrite( MOTOR\_B\_DIR, LOW );         digitalWrite( MOTOR\_B\_PWM, LOW );         delay( DIR\_DELAY );         // set the motor speed and direction         digitalWrite( MOTOR\_B\_DIR, LOW ); // direction = reverse         analogWrite( MOTOR\_B\_PWM, PWM\_FAST ); // PWM speed = fast         isValidInput = true;         break;       case '6': // 6) Hard stop (use with caution)         Serial.println( "Hard stop (Ruote bloccate)..." );         digitalWrite( MOTOR\_B\_DIR, HIGH );         digitalWrite( MOTOR\_B\_PWM, HIGH );         isValidInput = true;         break;       default:         // wrong character! display the menu again!         isValidInput = false;         break;     }   } while ( isValidInput == true );   // repeat the main loop and redraw the menu...  } | |

**Gestione di due motori**



[Listato programma](https://win.adrirobot.it/motor_driver/L9110_HG7881/Test_L9110_doppio.zip)

|  |  |
| --- | --- |
| |  | | --- | | /\* http://www.bajdi.com L9110 motor driver controlling 2 small DC motors  Modificato da www.adrirobot.it \*/  const int AIA = 5;  // (pwm) pin 5 connected to pin A-IA  const int AIB = 6;  // (pwm) pin 6 connected to pin A-IB  const int BIA = 10; // (pwm) pin 10 connected to pin B-IA  const int BIB = 11;  // (pwm) pin 11 connected to pin B-IB  byte speed = 150;  // change this (0-255) to control the speed of the motors  void setup() {   pinMode(AIA, OUTPUT); // set pins to output   pinMode(AIB, OUTPUT);   pinMode(BIA, OUTPUT);   pinMode(BIB, OUTPUT);   Serial.begin( 9600 );  }  void loop() {   Serial.println( "Avanti..." );   forward();   delay(2000);   STOP();   Serial.println( "Indietro..." );   backward();   delay(2000);   STOP();   Serial.println( "Sinistra..." );   left();   delay(2000);   STOP();   Serial.println( "Destra..." );   right();   delay(2000);   STOP();  }  void backward()  {   analogWrite(AIA, 0);   analogWrite(AIB, speed);   analogWrite(BIA, 0);   analogWrite(BIB, speed);  }  void forward()  {   analogWrite(AIA, speed);   analogWrite(AIB, 0);   analogWrite(BIA, speed);   analogWrite(BIB, 0);  }  void left()  {   analogWrite(AIA, speed);   analogWrite(AIB, 0);   analogWrite(BIA, 0);   analogWrite(BIB, speed);  }  void right()  {   analogWrite(AIA, 0);   analogWrite(AIB, speed);   analogWrite(BIA, speed);   analogWrite(BIB, 0);  }  void STOP()  {   analogWrite(AIA, 0);   analogWrite(AIB, 0);   analogWrite(BIA, 0);   analogWrite(BIB, 0);   delay (2000);  } | |