

Lego motor library

DC Motor

Library LEGOmotor for driving DC motors using with the board. This library need to be included in the beginning of the Arduino sketch, eg.

```
#include "LEGOmotor.h"  
void setup() {}  
void loop() {}
```

This is the description of the functions used for driving DC motors used in the robot. To run the motor two wires of DC motor required to connect with the connector for motors. These motor connectors are in pairs and are labelled "A" and "B". There are four (4) pairs of connectors to accommodate four (4) DC motors and output of these connectors are controlled by Pulse Width Modulation (PWM) through microcontrollers pins.

**void motor_1_Forward(int speed) or void motor_2_Forward(int speed) or
void motor_3_Forward(int speed) or void motor_4_Forward(int speed)**

Function for running the motor in a direction by setting the DC motor connector "A" to "HIGH" and connector "B" to "LOW". By swapping the wires connected to "A" and "B" the expected forward direction can be achieved. The rotational speed can be changed by setting the speed variable with an integer value from 0 to 255.

Example code: *Run motor forward for 1 second with 50% of the highest speed*
`motor_1_Forward(125); // here PWM speed is 125
delay(1000);`

**void motor_1_Backward(int speed) or void motor_2_Backward(int speed) or
void motor_3_Backward(int speed) or void motor_4_Backward(int speed)**

Function for running the motor in a direction by setting the DC motor connector "A" to "LOW" and connector "B" to "HIGH". By swapping the wires connected to "A" and "B" the expected backward direction can be achieved. The rotational speed can be changed by setting the speed variable with an integer value from 0 to 255.

Example code: *Run motor backward for 1 second with 100% of the highest speed*
`motor_1_Backward(255); // here PWM speed is 255
delay(1000);`

**void motor_1_Brake(void) or void motor_2_Brake(void) or
void motor_3_Brake(void) or void motor_4_Brake(void)**

This motor_X_Brake(void) function will try to stop the motor like a brake. This software based brake will pull the output for DC motor to ground. While breaking the motor, PWM on connector "A" will be fully on and pulled to LOW. Also Connector "B" will be set LOW.

```
motor_1_Brake();
```

**void motor_1_Release(void) or void motor_2_Release(void) or
void motor_3_Release(void) or void motor_4_Release(void)**

motor_X_Release(void) function will stop rotating and leave the rotation of the motor on a floating state. Connector “A” will set LOW with PWM set to 0 and connector “B” will bet set LOW without any PWM value set.

```
motor_1_Release();
```

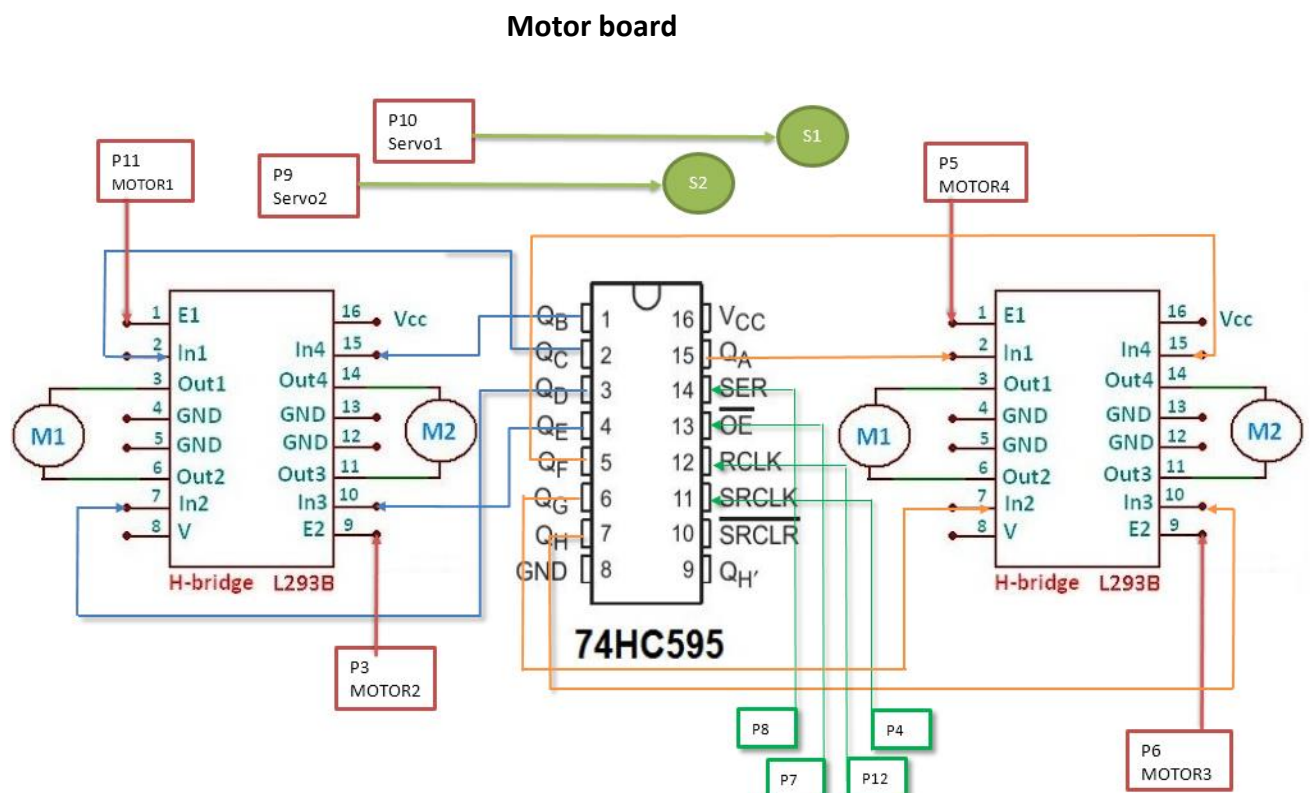


Figure 1 Diagram for motor shield used in the system; P12, P11, P10, P9, P8, P7, P6, P5, P4 and P3 are microcontroller pins

Servo motor

Library LEGOServoLib for driving servo motors using with the board. This library need to be included in the beginning of the Arduino sketch, eg.

```
#include "LEGOServoLib.h"
void setup() {
    servo_init();
    servo_1_attach();
    servo_2_attach();
}
void loop() {}
```

This is the description of the functions used for driving servo motors used in the robot. There are two (2) connectors to accommodate two (2) servo motors and output of these connectors are controlled by Pulse Width Modulation (PWM) through microcontrollers pins 9 and 10.

Functions for servo motors:

void servo_init()

This function is mandatory to call in the setup (), it assigns a servo index to the servo and initializes required functionalities.

uint8_t servo_1_attach() or uint8_t servo_2_attach()

This function attaches the servo with the microcontroller pin. This function is required to call before using the servo motor. This function attaches the given pin to the next free channel, sets pin mode as output, returns channel number or 0 if failure

void servo_1_detach() or void servo_2_detach()

This function detaches the servo with the microcontroller pin. If the certain servo motor is not being used it can be detached to save power, **servo_x_attach()** is required to call before using the servo motor.

void servo_1_write(int value) or void servo_2_write(int value)

Assign any value from 0 to 180 as the argument of this function to perform any angular movement.

bool servo_1_attached() or bool servo_2_attached()

Function for checking the servo motor is attached or not, this function returns true if this servo is attached, otherwise false.