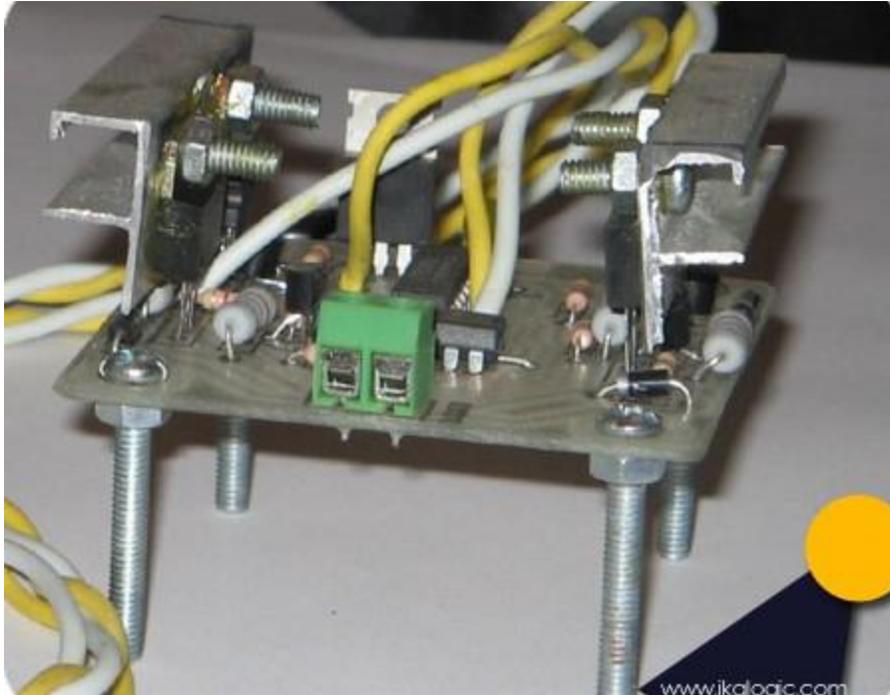


Build a 5A H-bridge Motor driver! New version

Posted on June 7, 2008, by Ibrahim KAMAL, in Motor Control, tagged



This H-bridge is easy to build, without any critical components. It is based on the famous and cheap TIP122 and TIP127 power transistors. It has been used on many of our robots and proved to be very versatile and robust. Another major advantage is that it only needs four wires for 12V power supply and direction control. Nevertheless, it allows bidirectional control, braking and freewheeling.

The hardware

The schematic **figure 1.A** can seem a little complicated at the first sight, but, practically, it only consists of two logic ICs and a bunch of transistors.

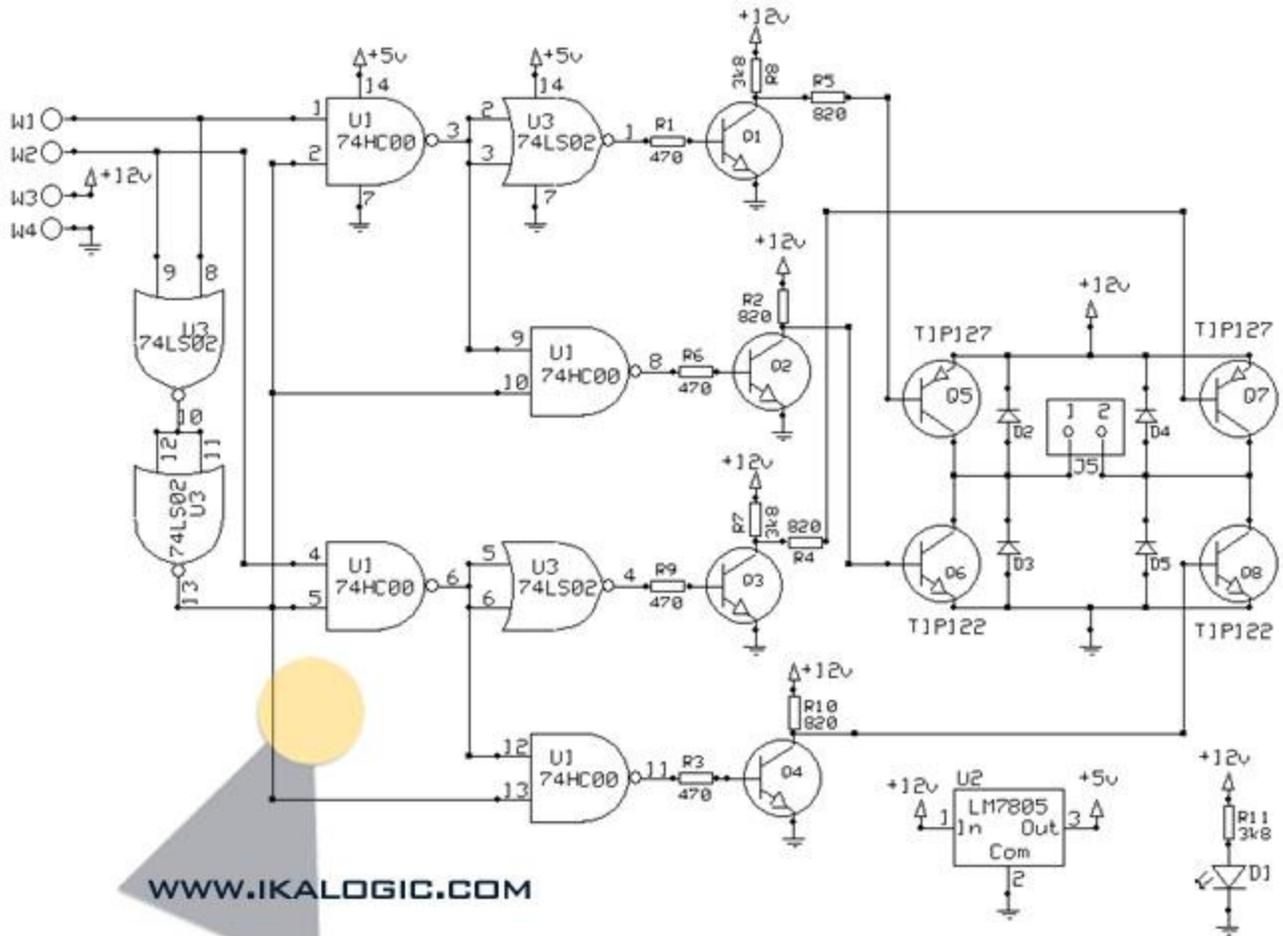


figure 1.A

Q_1 , Q_2 , Q_3 and Q_4 are **2N2222 BJT**. They can be replaced with any generic switching transistor.

Resistors R_2 , R_4 , R_5 and R_{10} have to be 1 WATT rated, to support high currents, especially if a 24V power supply is used.

The wire connection W_1 to W_4 are the wires between this H-bridge module and the controller board that provides the 12/24V power supply and the logic signals for direction control.

The wire connection W_3 can be connected to a power source from 10V to 24V, without circuit modifications.

The wire connection W_1 and W_2 are for connecting the H-bridge to a microcontroller or another logic device allowing the control of the motor. A truth table **figure 1.B**, is given below showing the effect of every combination of the logic states of W_1 and W_2 :

Inputs		Result on the connected motor
W1	W2	
0	0	Motor is freewheeling (disconnected, High impedance)
0	1	Turn the motor clockwise
1	0	Turn the motor anti-clockwise
1	1	breaking the motor

*(X mean Don't care), 1 = Logic high Voltage (5V), 0 = Logic low voltage (0V).

figure 1.B

Diodes D_2 to D_4 are simple rectifier diodes, with forward current ratings of at least 4A. LED D_1 is only used to signal the presence of power in the circuit.

The whole is assembled in a 5cm X 5cm PCB like in the picture below (**figure 1.C**). The length of the cable is not critical, it can be up to 1 meter, assuming you're PWM frequency is below 50 KHz. Heatsinks can be added to protect the TIP transistors from overheating. Be careful, the metallic back of the TIP is internally connected the collector, so be careful not to cause short circuits.

Actually, as you can see in **figure 1.C**, each pair of TIP122 and TIP127 share the same heatsink, without any risk of short circuits, since their collectors are already connected together in the circuit (If the schematic in **figure 1.a**, Q_5 and Q_6 have their collector connected together, as well as Q_7 and Q_8).

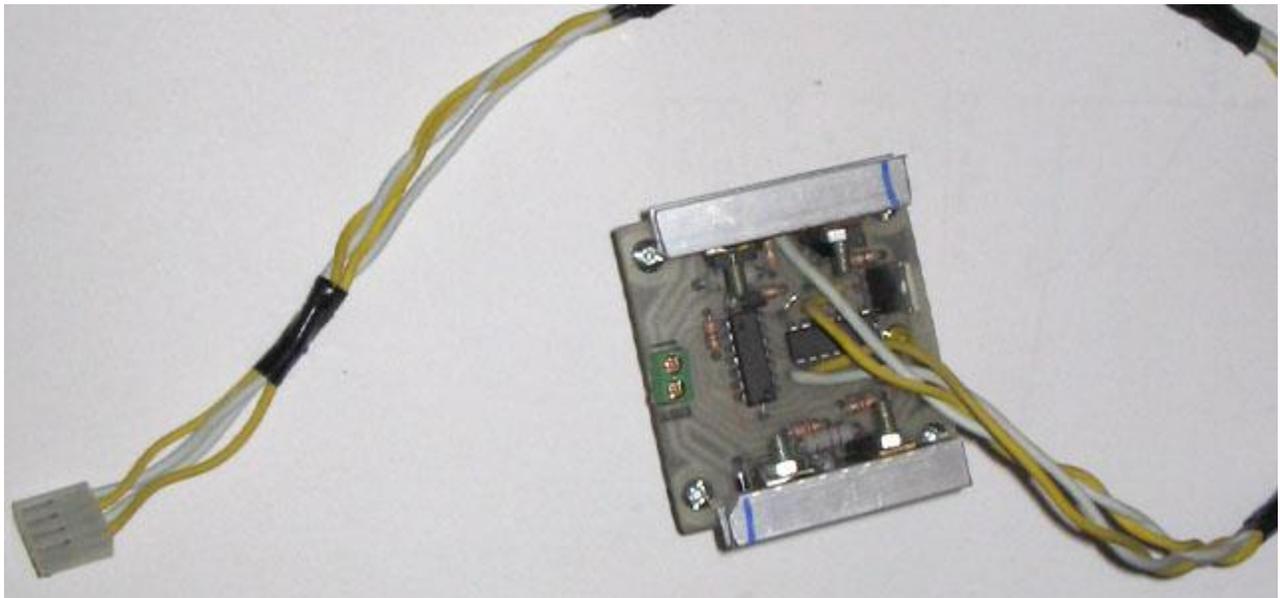


figure 1.C

Associated motor control algorithm

Here are some example C source codes showing how to control a motor using this H-bridge. The source codes are written for 8051 micro controller under the KEIL IDE, but can be easily modified for any other kind of microcontrollers or compiler.

This first example shows a function that control the direction/state of the motor, without speed variation.

```
#define w1 P1_0
#define w2 P1_1
#define motor_free 0
#define motor_break 1
#define motor_clockwise 2
#define motor_anti_clockwise 3
drive_motor(state){
    if (state == motor_free){
        w1 = 0;
        w2 = 0;
    }else if(state == motor_break){
        w1 = 1;
        w2 = 1;
    }else if(state == motor_clockwise){
        w1 = 1;
        w2 = 0;
    }else if(state == motor_anti_clockwise){
        w1 = 0;
        w2 = 1;
    }
}
```

The above function can then be called anytime in your program to change the state of the motor as in the example below:

```
drive_motor(motor_clockwise); //turn clockwise
```

The same function can be used in a more complicated code that controls the speed (or breaking torque) of a motor via PWM signals (for more information about PWM signals and speed control, check this article). The following example shows how to use this H-bridge with speed variations via PWM:

```

//global variables
unsigned char pwm_counter; //used to count from 0 to max_pwm
unsigned char max_pwm = 100; //this controls the period of one complete cycle
unsigned char pwm = 50; //this controls the duty cycle.

void main(){
    while(1){
        pwm_counter++
        if (pwm_counter > max_pwm){ pwm_counter = 0; }
        if (pwm_counter < pwm){ //ON cycle
            drive_motor(motor_clockwise); //turn clockwise
        }else{ //OFF cycle
            drive_motor(motor_free); //motor freewheeling
        }
    }
}

```

The above example drives a motor with a 50% duty cycle PWM signal. It can be integrated in any program without causing any delays or disturbance the other functions.

I hope this article was useful. Any comments and further questions are welcome in the form below.

Source: <http://www.ikalogic.com/build-a-5a-h-bridge-motor-driver-new-version/>