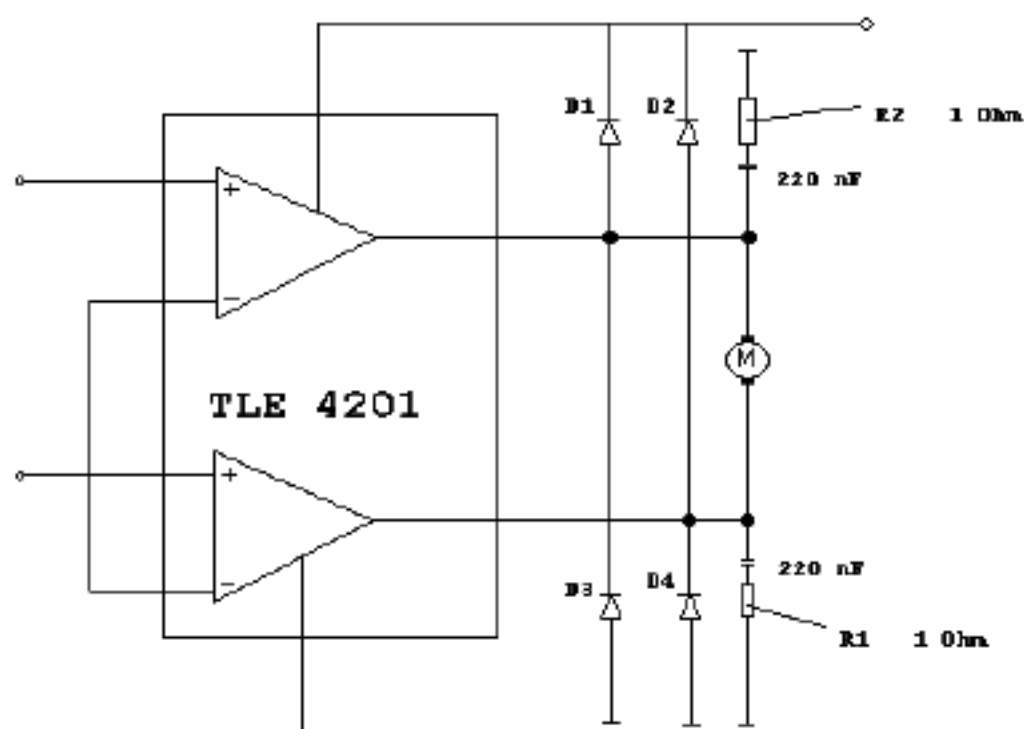


MANUAL

für TLE 4201 SI



Einleitung

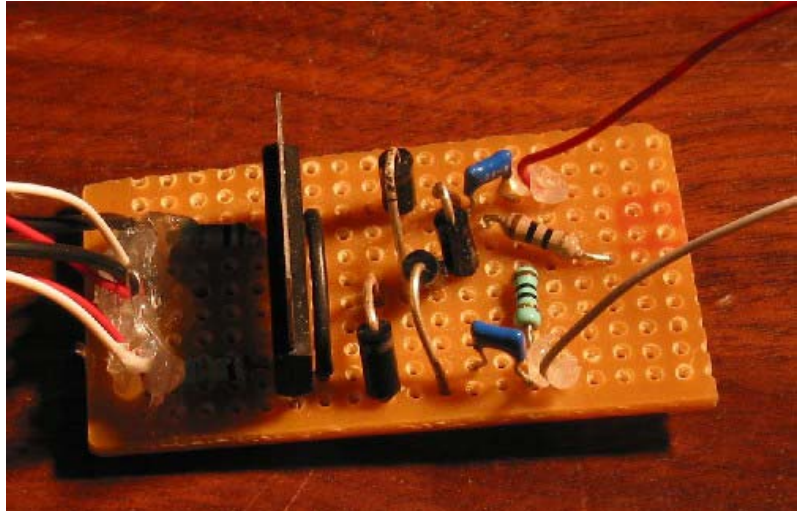
Diese elektronische Schaltung invertiert die Polarität der angelegten Spannung über 2 Datenleitungen. Genauere Erklärung der Software und Hardware folgt in den anfolgenden Kapiteln. Das von mir geschriebene Manual obliegt keinerlei Garantie auf Richtigkeit des geschriebenen Textes und der eingefügten Zeichnungen. Bei Fragen bitte unter potatoface@chello.at melden.

Bauteile

Lötset (Lötkolben, Lötzinn, usw.)

2x 220 nF Kondensatoren
4x 1N 4002 Dioden
2x 10 – 100 KOhm Widerstände
2x 1 – 100 Ohm Widerstände
(Hauptsache gleich groß)
Draht in verschiedenen Farben
Lochmaske o. Experimentierbrett
TLE 4201 SI Chip

optional Heißkleberpistole



Arbeitsweise Software

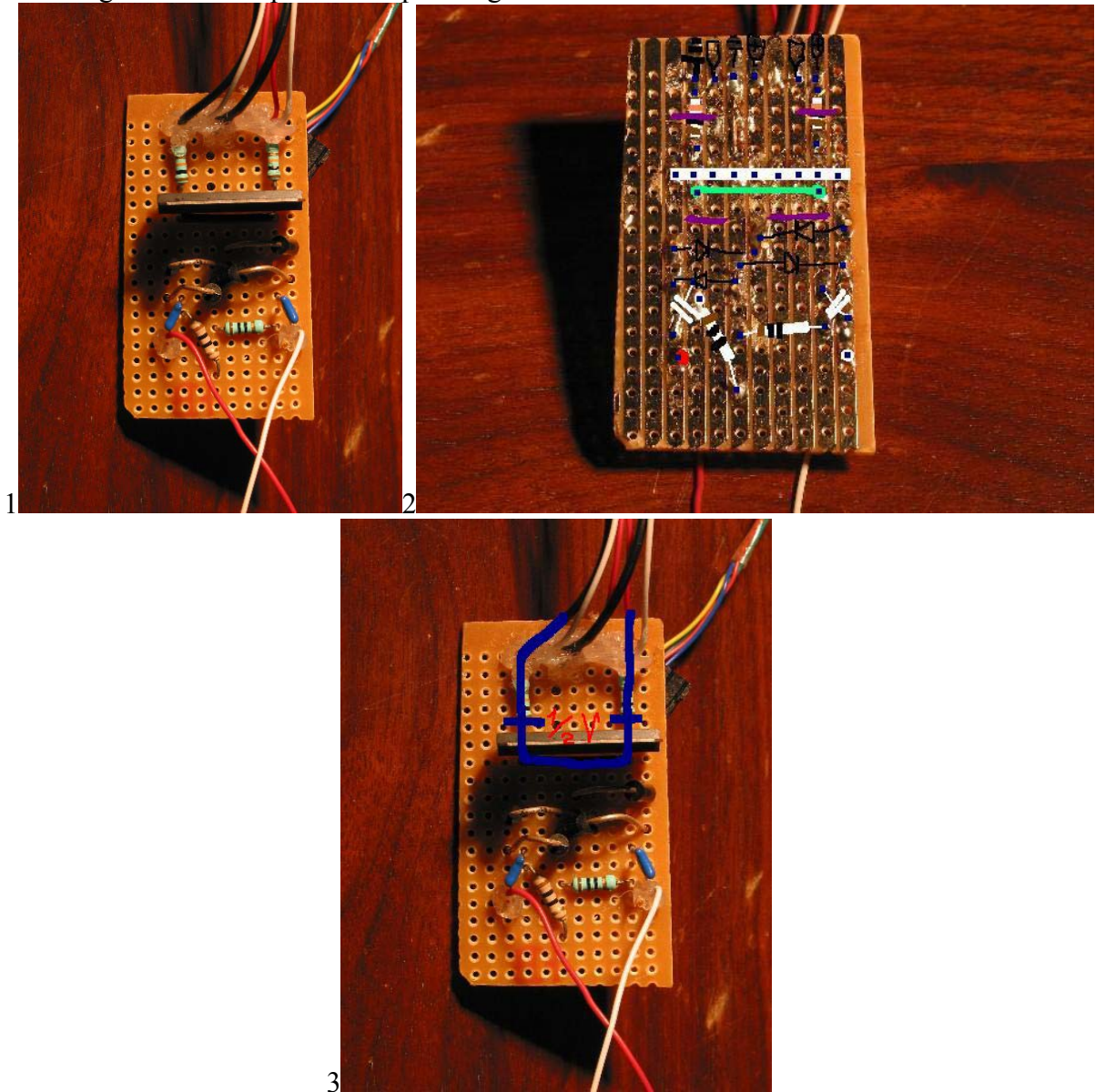
Mit der Pic-Betriebsspannung wird über den Datenstecker und den zwei weißen Kabeln der Motorbrücke mitgeteilt, in welcher Weise sie die ausgehende Spannung zu polen hat. Durch die 2 Datenkabeln können insgesamt 4 Zustände gesetzt werden. Das schwarze Kabel ist der “Ground“ und das rote Kabel die “+5V“ vom Pic. Die “+5V“ werden, erreicht indem man den passenden PIN auf high setzt und dann am TLE 4201 SI mit Hilfe von einen Spannungsteiler die PIN's 2 und 8 mit dieser halben Pic-Spannung versorgt. Dadurch erkennt der TLE 4201 SI über und unter welcher Spannung ein logische 0 und 1 ist. Deshalb ist es auch wichtig, dass die Widerstände gleich groß sind.



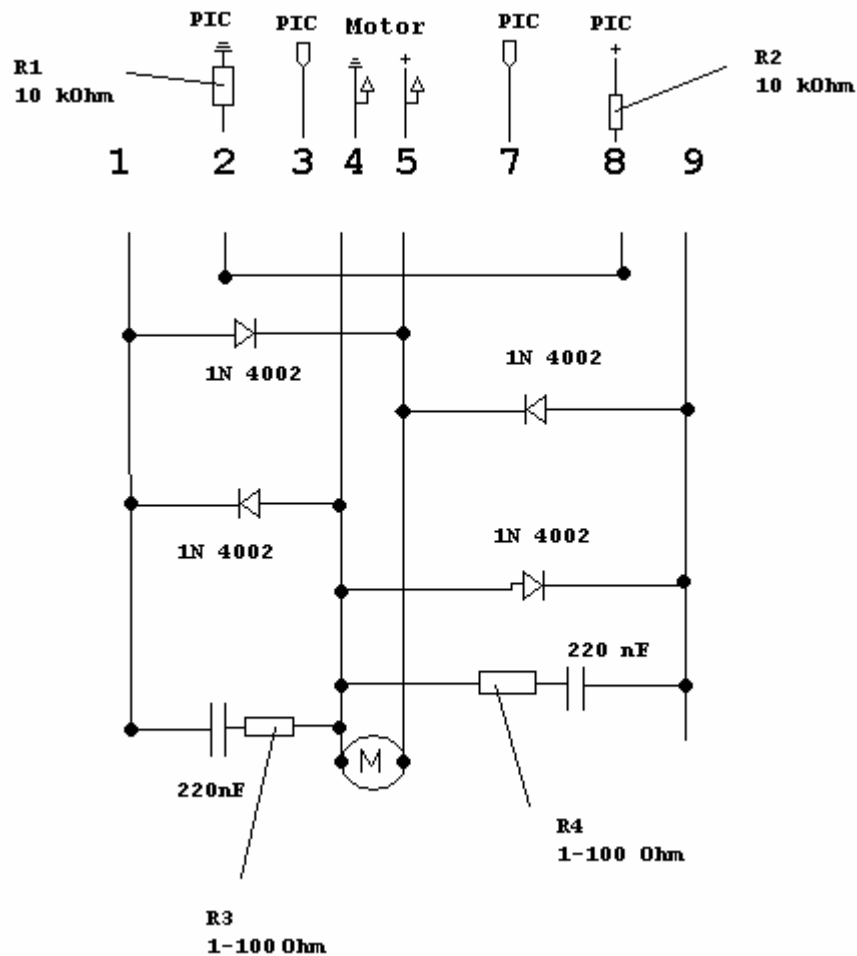
Datenleitung 1	Datenleitung 2	Ergebnis
output_low(DATENPIN_1);	output_low(DATENPIN_2);	Motor stop
output_low(DATENPIN_1);	output_high(DATENPIN_2);	Drehrichtung 1
output_high(DATENPIN_1);	output_low(DATENPIN_2);	Drehrichtung 2
output_high(DATENPIN_1);	output_high(DATENPIN_2);	Motor stop

Hardware Erklärung

Die Dioden auf der Schaltung verhindern, dass durch Induktion Spannung in die falsche Richtung fließt. Der Spannungsteiler sind die 2 10KOhm Widerstände und die schwarze Brücke (siehe unten: 3). Die Kondensatoren wirken beim Einschalten wie ein Kurzschluss und “fangen” beim Umpolen die Spannung ab.

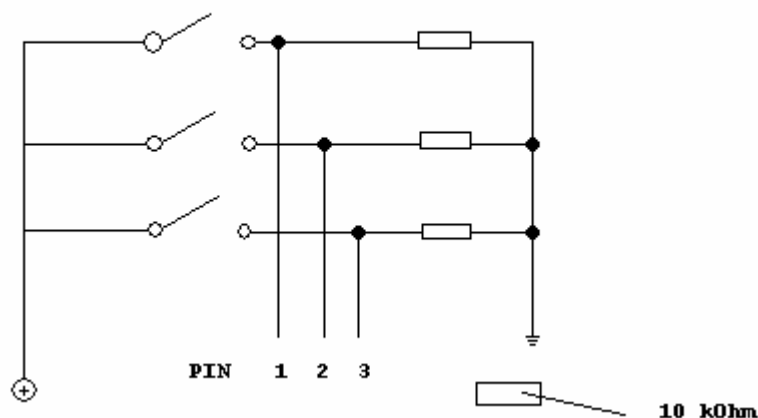


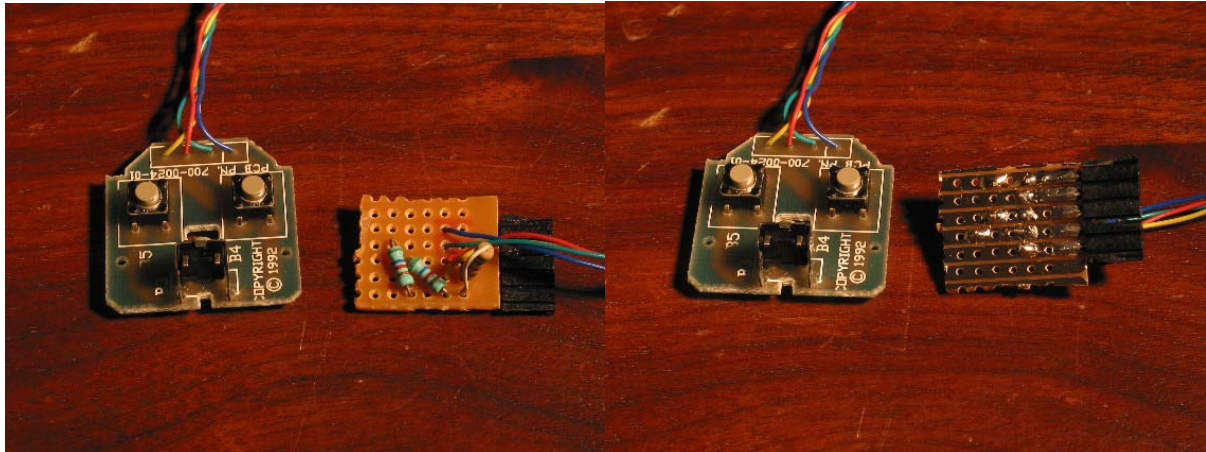
Die rechte Zeichnung (2) ist Spiegelverkehrt !!!



Steuerung per Buttons

Ich habe zur Steuerung der Motorbrücke einen kleinen Button konstruiert und den mit dem "Ground" verbunden ist. Dadurch konnte ich die Interferenzen, die durch Berührung der Leitungen entstehen würden, verhindern. In der Software wird das, dann so gelöst, dass ich zwischen den 2 Richtungen wählen kann oder stoppen.





Source

```
// Button.c
// Aigner Sebastian
// Dienstag, 18. Februar 2003

#include "D:\Picc\Examples\16F877.H"
#fuses HS,NOPROTECT,NOWDT

#use delay(clock=20000000)
//#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)

main()
{
    set_tris_c(0b01110000);
    output_high(PIN_D7);
    while(1)
    {

        if( input(PIN_D6) )
        {
            output_high(PIN_A0);
        }
        else
        {
            output_low(PIN_A0);
        }

        if( input(PIN_D5) )
        {
            output_high(PIN_A1);
        }
        else
        {
            output_low(PIN_A1);
        }

        if( input(PIN_D4) )
        {
            output_high(PIN_A2);
        }
        else
        {
            output_low(PIN_A2);
        }

    }
}
```

```

// MotorbrueckemitButton.c
// Aigner Sebastian
// Mittwoch, 19. Februar 2003

#include "D:\Picc\Examples\16F877.H"
#fuses    HS,NOPROTECT,NOWDT

#use delay(clock=20000000)
//#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)

main()
{
    int pressed=0;                                // 0 for stopped both low
                                                    // 1 for forward  PIN_D6-high PIN_D7-low
                                                    // 2 for backward PIN_D6-low  PIN_D7-high

    output_low(PIN_D6);        // set for
    output_low(PIN_D7);        // motor stopped
    output_high(PIN_D5);       // for h bridge command power

    output_high(PIN_A3);       // set buttonpower high
    set_tris_a(0b00000111);   // set PIN_A2 PIN_A1 PIN_A0 to input
    while (true)
    {
        if(input(PIN_A2) && pressed != 1)
        {
            if(pressed == 2)
            {
                output_low(PIN_D6);
                output_low(PIN_D7);
                delay_ms(500);        // 0,5 sec pause for motorstopping
                                      // if running vice versa
            }
            output_high(PIN_D6);
            output_low(PIN_D7);
            pressed=1;
        }

        if(input(PIN_A1) && pressed != 0)
        {
            output_high(PIN_D6);
            output_high(PIN_D7);
            pressed=0;
        }
        if(input(PIN_A0) && pressed != 2)
        {
            if(pressed == 1)
            {
                output_low(PIN_D6);
                output_low(PIN_D7);
                delay_ms(500)        // 0,5 sec pause for motorstopping
                                      // if running vice versa
            }
            output_low(PIN_D6);
            output_high(PIN_D7);
            pressed=2;
        }
    }
}

```

Fazit

Ich danke meinen Kollegen **Andreas Gansterer** für die Zeichnungen der Schaltungen und auch den Herrn **Adrian Dabrowski** der mir sehr lange das Prinzip der Schaltung und den Aufbau erklärt hat. Ich hoffe, ihr könnt auch diese Schaltung nun einfacher nachbauen und falls ihr noch Fragen habt, könnt ihr euch einfach unter potatoface@chello.at melden.