Lecture 24 objectives

- Motor interface
- PWM driver
- **Course evaluation**

Motor Specifications

Voltage
Current
Inductance
Torque
Speed
Time constant

Motor Interface

PWM to adjust delivered electrical power
Direction
Unidirectional sink
Unidirectional source
Bidirectional

Darlington TIP120 (NPN) TIP125(PNP)

h_{fe} =1000
I_{CE} up to 3A

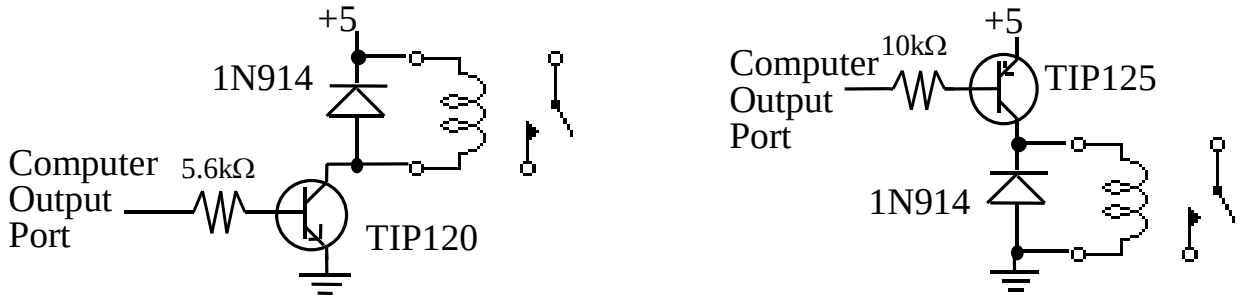


Figure 8.67. Two relay interfaces.
Also can be used to drive DC motor on/off

Any diode is fine (1N914, 1N4001, 1N4002, 1N4003)

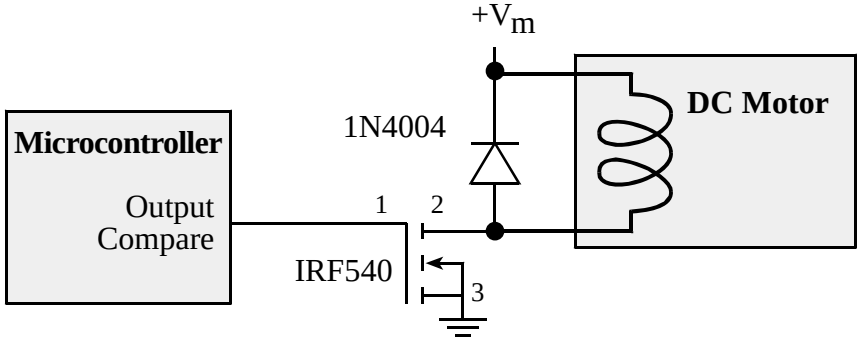


Figure 8.68. Motor interface using a high current MOSFET.

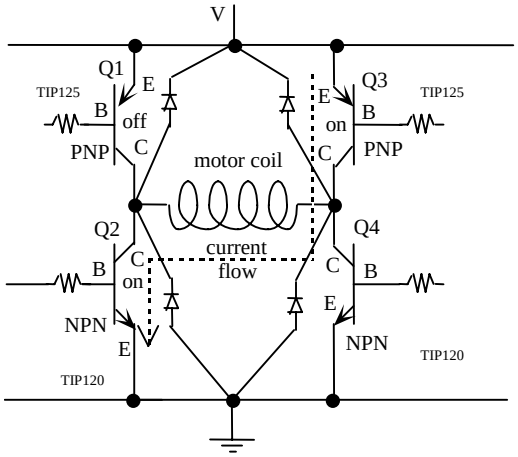
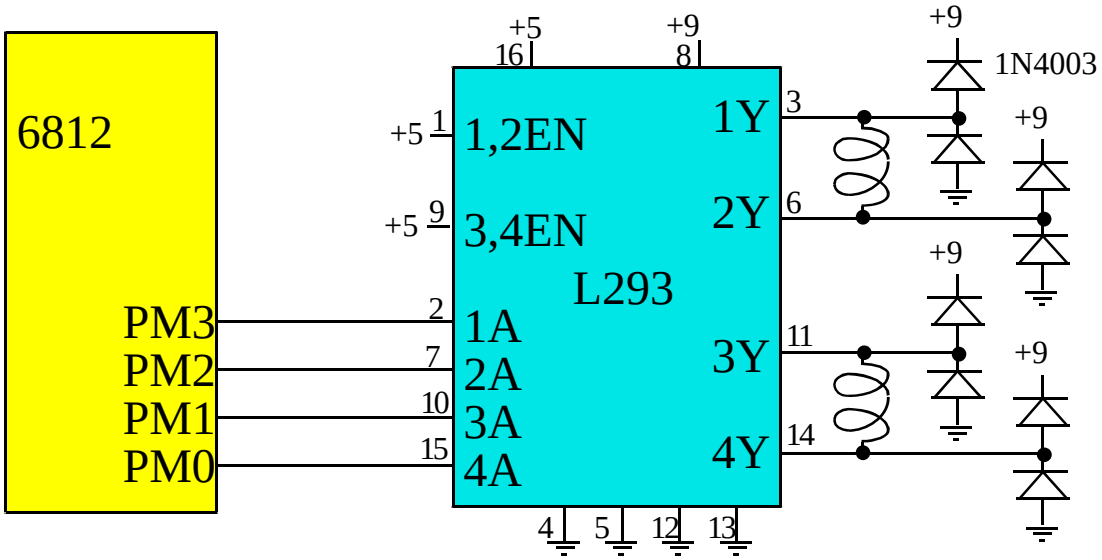


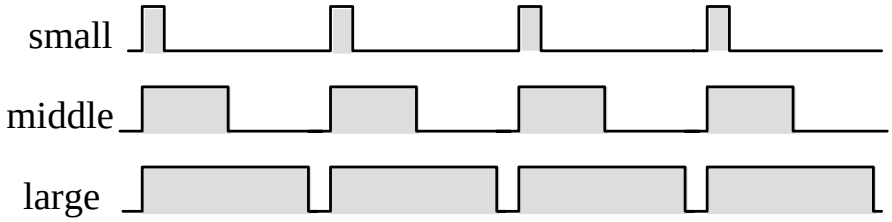
Figure 8.72. An H-bridge is used to drive current in both directions.

Outputs: two 4-wire bipolar stepper motors



Bipolar stepper motor interface using an L293 driver
DC motor interface to drive forward/backward/off

Pulse Width Modulation



`High + Low` will always equal 10000
Duty cycle is **`High/10000`**

The output compare implementation can not generate waves close to 0 or 100% duty cycle. If **T** is the number of cycles to process the output compare interrupt, then both **`High`** and **`Low`** must be greater than **T**.

****Run Metrowerks PWM project********

```
// ---------PWM_Init---------
// enable PWM channels 3,1,0
// inputs: none
// outputs: none
// concatenates 2,3
```

```
void PWM_Init(void){
  MODRR = 0x0B;    // PTT3,PTT1,PTT0 associated with PWM
  PWME = 0x0B;     // enable channels 0,1,3
  // bits 5-0, PWME5-0, Pulse Width Channel 5-0 Enable
  PWMPOL = 0x05;   // PTT2,PPT0 high/low, PTT1 low/high
  // bits 5-0, PPOL5-0, Pulse Width Channel 5-0 Polarity
  //  1 = high at beginning, then low when count reached.
  //  0 = low at beginning, then high when count reached.
  PWMCLK = 0x08;   // clock A for 0,1, clock SB for 3
  // bit 5, PCLK5, Pulse Width Channel 5 Clock Select
  //   1 = Clock SA is the clock source for PWM channel 5
  //   0 = Clock A is the clock source for PWM channel 5
  // bit 4, PCLK4, Pulse Width Channel 4 Clock Select
  //   1 = Clock SA is the clock source for PWM channel 4
  //   0 = Clock A is the clock source for PWM channel 4
  // bit 3, PCLK3, Pulse Width Channel 3 Clock Select
  //   1 = Clock SB is the clock source for PWM channel 3
  //   0 = Clock B is the clock source for PWM channel 3
  // bit 2, PCLK2, Pulse Width Channel 2 Clock Select
  //   1 = Clock SB is the clock source for PWM channel 2
  //   0 = Clock B is the clock source for PWM channel 2
  // bit 1, PCLK1, Pulse Width Channel 1 Clock Select
  //   1 = Clock SA is the clock source for PWM channel 1
  //   0 = Clock A is the clock source for PWM channel 1
  // bit 0, PCLK0, Pulse Width Channel 0 Clock Select
  //   1 = Clock SA is the clock source for PWM channel 0
  //   0 = Clock A is the clock source for PWM channel 0
  PWMPRCLK = 0x76; // B is bus/128, A is 24MHz/64 = 375kHz
  // bits 6-4, PCKB2-PCKB0, n is 0 to 7, Prescaler for B
  //           clock B is bus clock/(2**n)
  // bits 2-0, PCKA2-PCKA0, m is 0 to 7, Prescaler for A
  //           clock A is bus clock/(2**m)
  PWMCAE = 0;      // left aligned mode
  // bits 5-0, CAE5-0, Center Aligned Output on Channel 5-0
  //   1 = operates in Center Aligned Output Mode.
  //   0 = operates in Left Aligned Output Mode.
  PWMCTL = 0x20;   // Concatenate 2+3,
  // bit 6, CON45, Concatenate channels 4 and 5
  //  1 = Chan 4 and 5 concatenated to one 16-bit chan
  //  Channel 4 high byte and channel 5 low byte
```

```
  //  Channel 5 output pin used as output for 16-bit PWM
  //  Channel 5 clock select-bit determines source
  //  Channel 5 polarity bit determines polarity
  //  Channel 5 enable bit enables  output and
  //  Channel 5 center aligned enable bit determines mode.
  //  0 = Channels 4 and 5 are separate 8-bit PWMs.
  // bit 5, CON23, Concatenate channels 2 and 3
  //  1 = Chan 2 and 3 are concatenated to one 16-bit chan
  //  Channel 2 high byte and channel 3 low byte
  //  Channel 3 output pin used as output for 16-bit PWM
  //  Channel 3 clock select-bit determines clock source
  //  Channel 3 polarity bit determines polarity
  //  Channel 3 enable bit enables  output and
  //  Channel 3 center aligned enable bit determines mode
  //  0 = Channels 2 and 3 are separate 8-bit PWMs.
  // bit 4, CON01, Concatenate channels 0 and 1
  //  1 = Chan 0 and 1 concatenated to one 16-bit channel
  //  Channel 0 high byte and channel 1 low byte
  //  Channel 1 output pin used as output for 16-bit PWM
  //  Channel 1 clock select-bit determines clock source
  //  Channel 1 polarity bit determines polarity
  //  Channel 1 enable   bit enables  output and
  //  Channel 1 center aligned enable bit determines mode
  //  0 = Channels 0 and 1 are separate 8-bit PWMs.
  // bit 3, PSWAI, PWM Stops in Wait Mode
  // bit 2, PFRZ, PWM Counters Stop in Freeze Mode
  PWMSCLA = 0;     // SA prescaled A by 512
  // Clock SA = Clock A / (2 * PWMSCLA)
  PWMSCLB = 3; // SB prescaled B by 6=24MHz/128/6=31.25kHz
  // Clock SB = Clock B / (2 * PWMSCLB)
  PWMPER0 = 250;    // period0
  PWMDTY0 = 0;      // duty cycle0, off
  PWMPER1 = 250;    // period1
  PWMDTY1 = 0;      // duty cycle1, off
  PWMPER23 = 31250; // period3
  PWMDTY23 = 0;     // duty cycle23, off
}
```