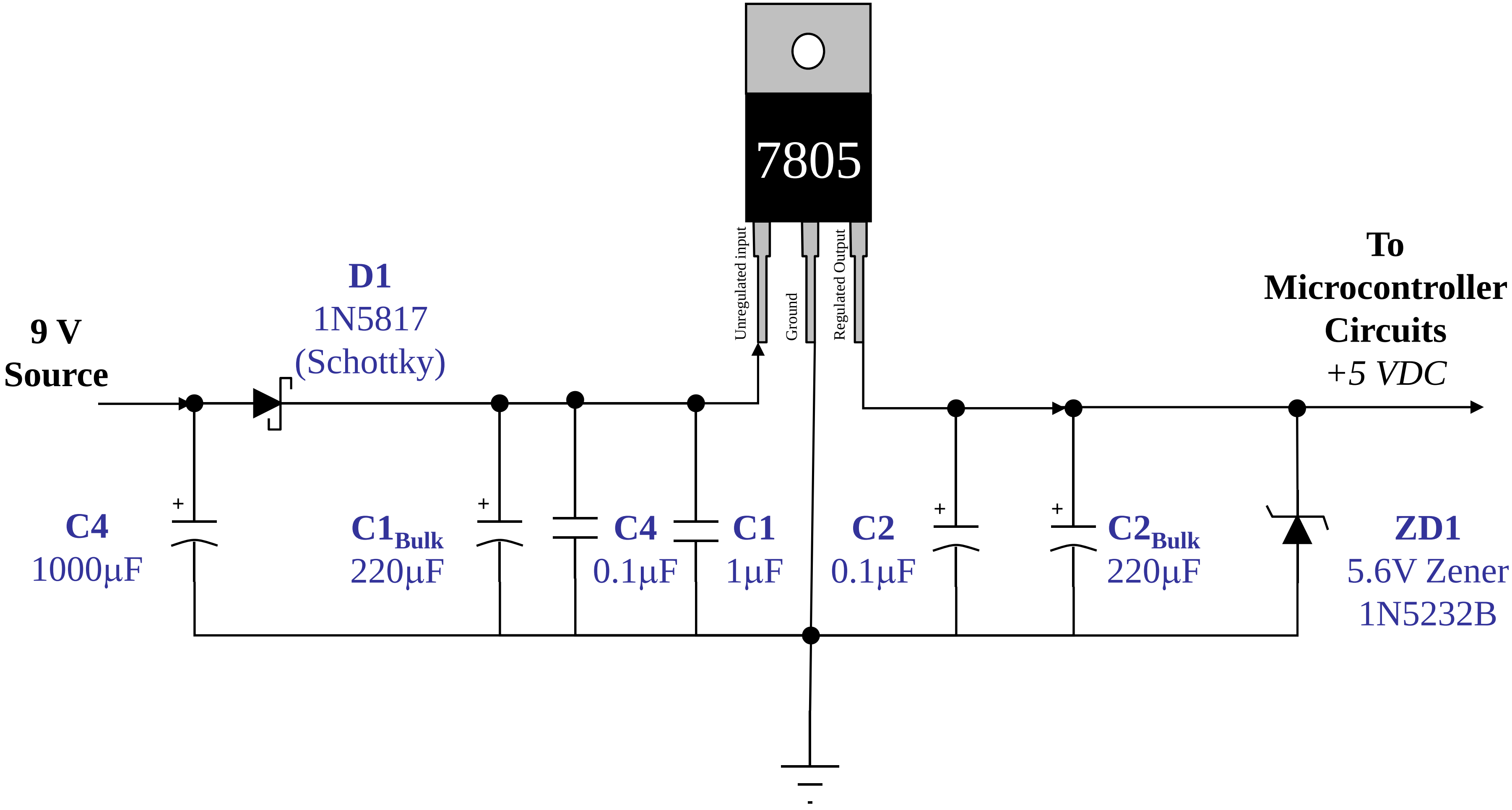
Lab 1 - Microcontrollers

- Complete the program template below being sure to select the correct parameters to meet the requirements (see the Microcontroller slides). ***Show either to TA or myself your completed code when entering the lab.***
- Test the code using the simulation capability of AVR Studio. Demonstrate this simulation to either the TA or myself and be ready to explain what is happening when the blocks change color. (Note: be sure to turn off the optimization by going to Project > Configuration Options > and selecting "-00" for optimization.
- Construct the circuit shown below being careful to separate the 9V and 5V lines. Build your circuit with the idea of limiting noise. (i.e., be careful where you place your components, use short leads, and avoid ground loops). Your layout should be easy to compare to the schematic.
- When you push and release the button switch, the motor will toggle between full speed forward and full speed reverse in half steps. Demonstrate your robot to either the TA or myself.
- Lab Report
  - Show using the schematic on the page titled "ATmega168 & SN764410" how you would connect a second motor being sure to show the connections between the motor, H-bridge and microcontroller.
  - This lab uses a single chip to operate the motors. This is also the most expensive chip in your packet! Look on-line and determine the highest current available to a motor that can be drawn from a single-chip H-Bridge. Provide its manufacturer, part number, cost, Voltage range, max current, and URL of the datasheet.
- Explain what a ground loop is and how you can limit its effect.

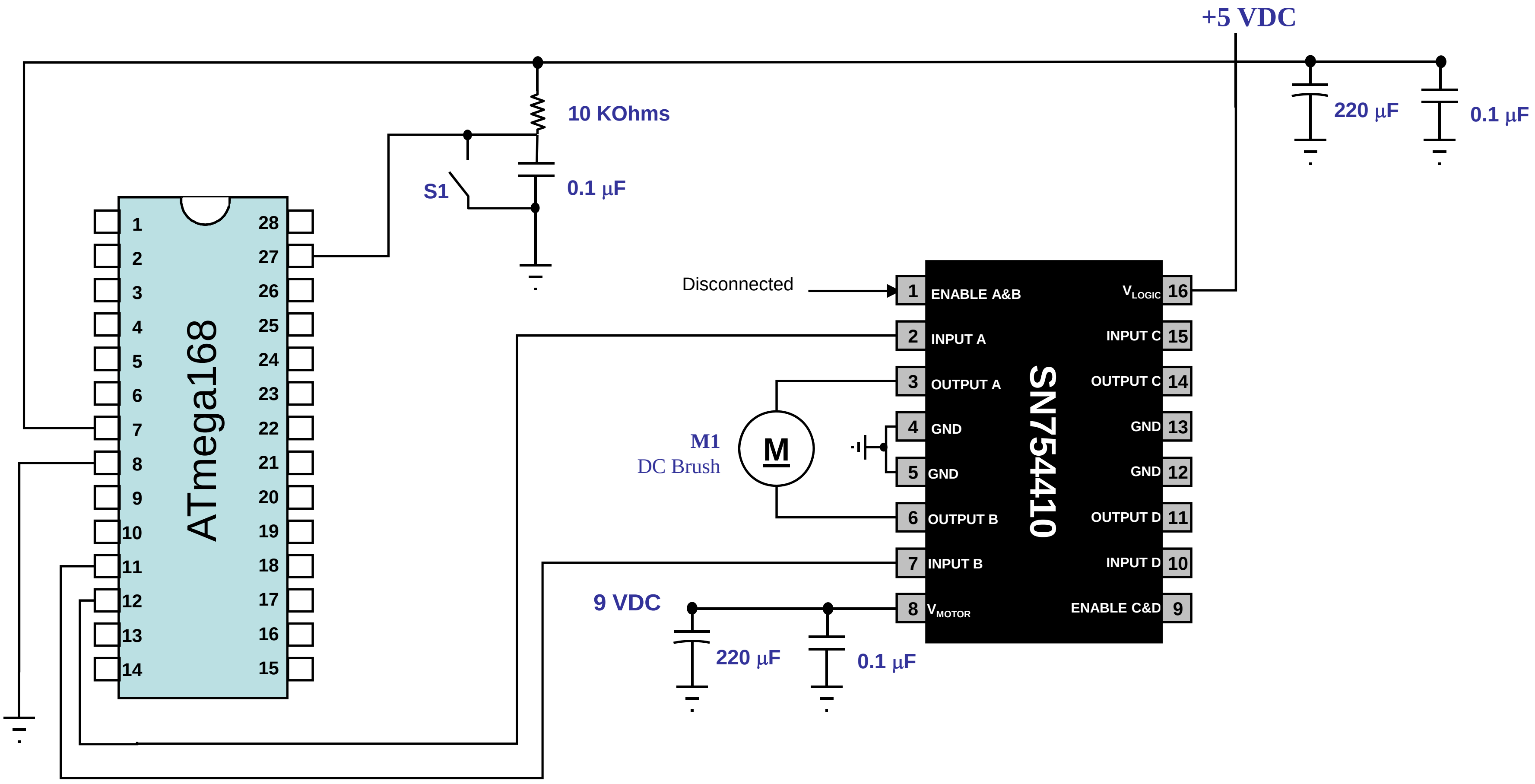
Voltage Regulator

5V for microcontroller circuit

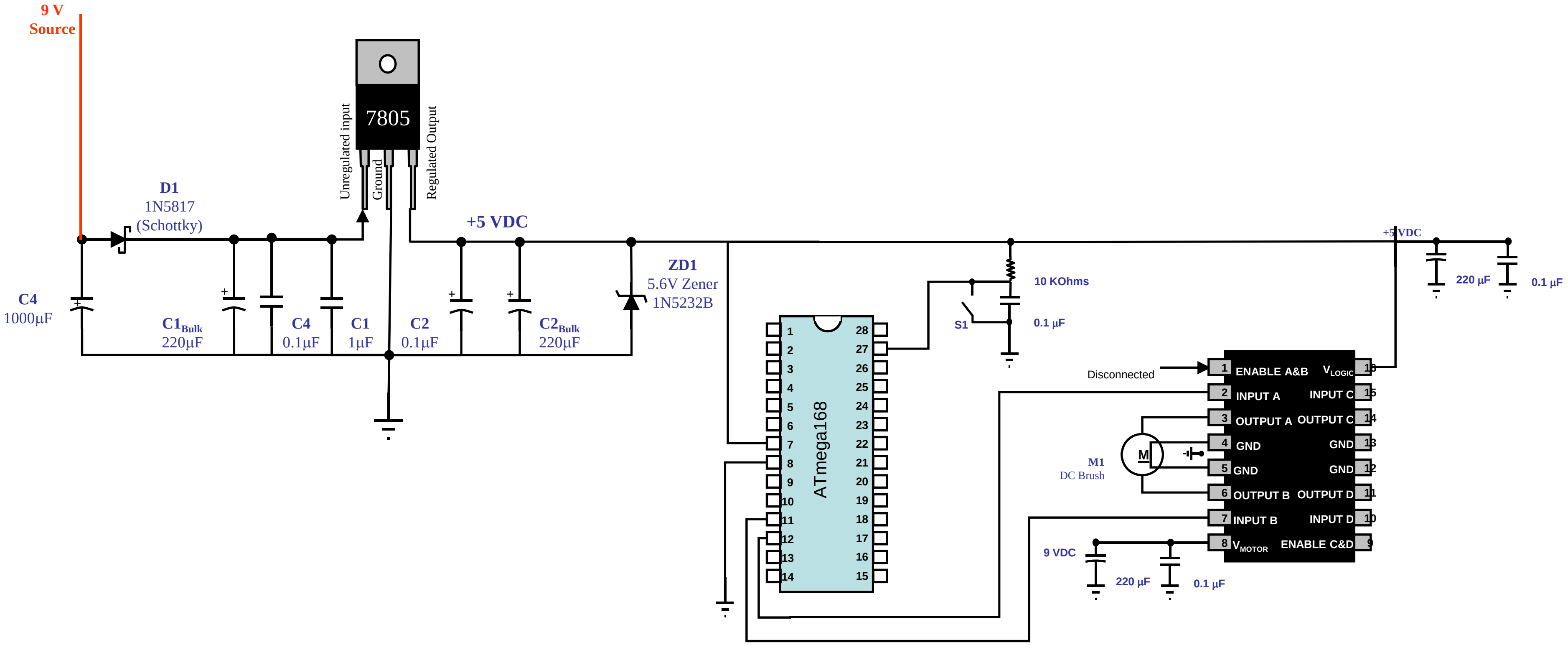


ATmega168 & SN764410

Microcontroller Connections for H-Bridge Driver



Complete Circuit for Microcontroller Lab



Code for Microcontroller Lab

motor routine

```
#include <avr/io.h>
#include <avr/interrupt.h>

#define LEFT_MOTOR 1

void motor( int mtr, int speed)
{
        static int up;
        static int back;

        if (speed > 0)
        {       up = speed;
                back = 0;
        }
        else if (speed<0)
        {       up = 0;
                back = -speed;
        }
        else
        {       up = 0;
                back = 0;
        }

        if (mtr == LEFT_MOTOR)
        {       OCR0A = up;
                OCR0B = back;
        }
}
```

Code for Microcontroller Lab

interrupt routine

```
ISR(  External Interrupt Vector ?)
{   static int dutycycle = 0;

    dutycycle += 1;
    if (dutycycle > 7)
        dutycycle = 0;

    switch (dutycycle)
    {
        case 0:
            motor(LEFT_MOTOR, ? );  //Off
            break;
        case 1:
            motor(LEFT_MOTOR, ? );  //50% Forward
            break;
        case 2:
            motor(LEFT_MOTOR, ? );  //100% Forward
            break;
        case 3:
            motor(LEFT_MOTOR, ? );  //50% Forward
            break;
        case 4:
            motor(LEFT_MOTOR, ? );  //Off
            break;
        case 5:
            motor(LEFT_MOTOR, ? );  //50% Reverse
            break;
        case 6:
            motor(LEFT_MOTOR, ? );  //100% Reverse
            break;
        case 7:
            motor(LEFT_MOTOR, ? );  //50% Reverse
            break;
    }
}
```

Code for Microcontroller Lab

main routine

```
int main (void)
{
        /* Left Motor */
                //Setup the PWM on pin 12 (PD6)
                        TCCR0A =  ? ;   // Setup for 0 Volts when counter is low; Phase Correct PWM
                        TCCR0B =  ? ;   // Use an 8-bit prescaler
                        OCR0A = ?;                // Initialize motor so that it is off
                        DDRD = ? ;                // Don't forget to set this pin as an output

                //Setup the PWM on pin 11 (PD5)
                        TCCR0A = ? ;              // Setup for 0 Volts when counter is low; Phase Correct PWM
                        OCR0B = ? ;               // Initialize motor so that it is off
                        DDRD =? ;  // Don't forget to set this pin as an output

        // Setup for External Interrupt PCINT12 (uses PC4 which is also pin 27)
                SREG =     (  ?  );
                PCICR =    (  ?  );
                PCMSK1 = (  ?  );
                PORTC =   (  ?  ); //Use PC4's pull-up resistor


        while (1)  { }
}
```