

MECH-554 Microprocessors for Mechanical Systems

Winter 2003

Lab 3 PWM, Optical Sensor and DC Motor

Demonstration Date: 11th April, 2003
Report Due: 11th April, 2003

1. Objectives

This experiment takes us into the world of creating systems to enhance interaction and functionality with the analog world using digital control systems. An open-loop DC motor control system is implemented using the HC11 MCU and the ST L298 motor driver kit. The following steps are involved in this experiment:

- Configure and operate HC11 timer circuits.
- Calibrate and generate pulse wave modulation (PWM) waveforms using HC11.
- Gain experience with DC motors, optical encoders, PWM motor controllers.
- Implement an open loop speed controller with a DC servo motor.

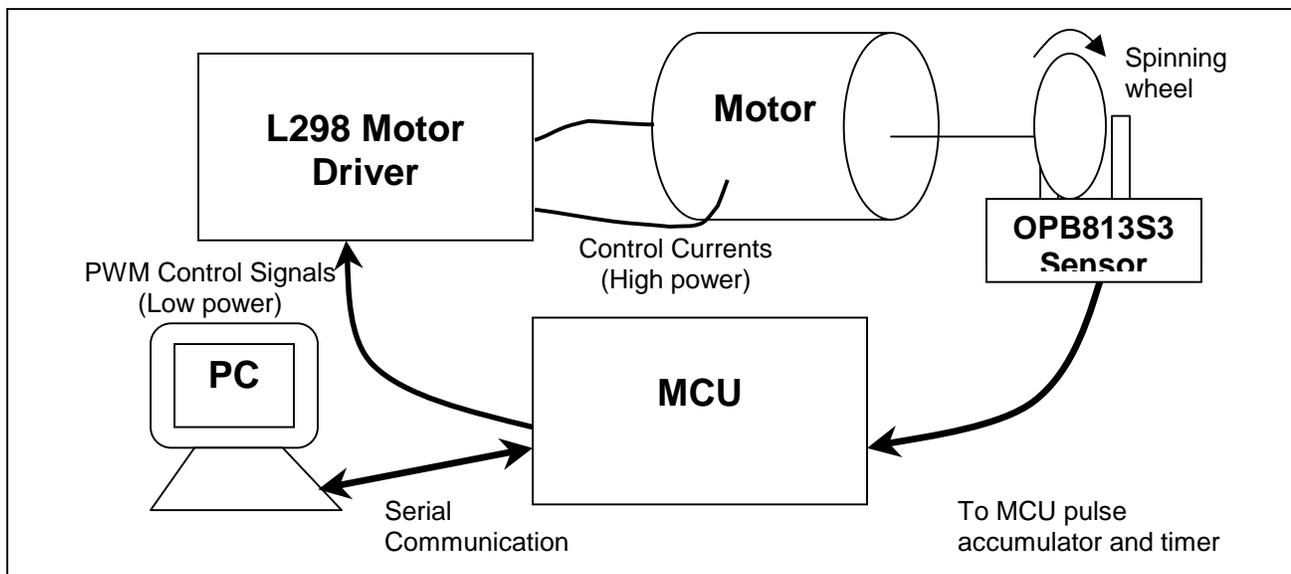
2. Requirement

Using the U (up), D (down), S (stop) keys on the PC keyboard, control the rotation speed and direction of the DC motor. The (i) Output PWM duty cycle, and (ii) Measured RPM information are sent to the terminal screen every second. The rotation direction of the motor is indicated by a pair of LEDs.

3. Material List

- Motorola HC11 EVB
- ST L298 motor driver kit
- DC servo motor kit
- Optek OPB813S3 Diode-Phototransistor optical switch
- LEDs and resistors

Figure 1: Bidirectional motor control using the L298 Motor Driver kit.



4. Useful Notes

L298 motor driver kit

The L298 Driver kit provides you the L298 chip, which is capable of controlling 2 motors independently, along with the associated flyback and freewheeling diodes, bypass capacitors, etc. Only one half of the kit is used. Figure 2 below shows one possible arrangement to connect the kit to a power supply (V_s) and use Pins C, D and V_{en} to control the operation of the motor. Make sure you understand how the L298 motor driver chip and the L298 kit work before you proceed any further. Connect Input 1(C), Input 2(D) and Enable (V_{en}) to the HC11 logic pins of your choice. Use the precision variable voltage supply available at each desk (set at +5V/0V for use in the voltage divider) to power to the kit.

The “U” (Up) and “D” (Down) keys on the keyboard should be used to vary the duty cycles of the PWM outputs (details below) in steps of 16 on Inputs 1 and 2 of the L298 motor driver kit (while Enable=High). Find the corresponding constant rpm to which the motor settles down to.

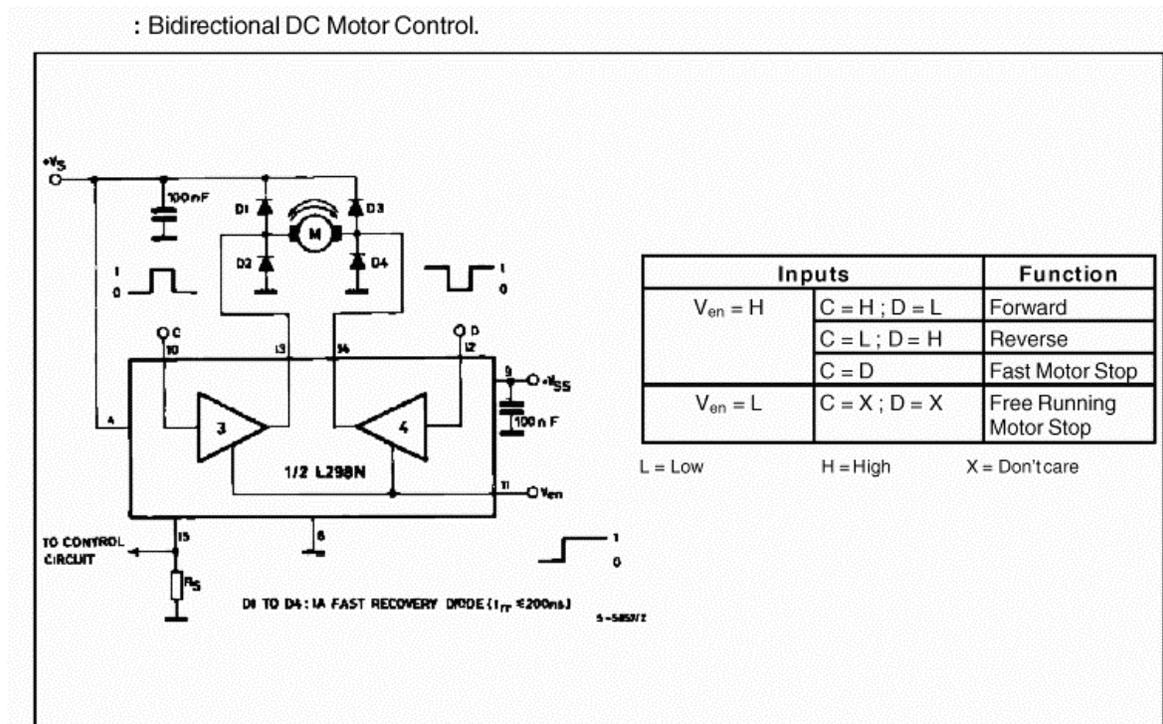


Figure 2: Bidirectional motor control using the L298 Motor Driver kit.

Pulse Width Modulation (PWM)

The PWM waveform will be used to control the motor driver chip, which in turn adjusts its output current and controls the motor. Choose an output pin on the MCU. Write a software loop that outputs a square waveform of alternating “1” and “0”. The waveform has a period of 256 clock cycles, in which it is HIGH for {0,16,32,48,...,or 240} clock cycles. The duration of output=HIGH in a period is called its *duty cycle*. The required system selects 1 of the 16 distinct {0,16,32,48,..., 240} duty cycles by using the “U”, “D” signals from the PC serial terminal. When Enable pin is HIGH, the Input 1 of the L298 kit controls the motor in the forward direction and Input 2 in the reverse. Each input require a PWM waveform. Please take care not to switch from outputting a non-zero duty cycle on Input 1 to a non-zero duty cycle on Input 2 *suddenly* to avoid unexpected results.

(i) Forward Rotation The system initializes with zero duty cycle at both forward and backward controls. Pressing the "U" key will increase the *forward PWM signal* to the next higher duty cycle (from 0 to 240 in steps of 16) until it reaches the maximum output and cause the motor to spin in the forward direction at a higher speed. The reverse PWM signal is a constant zero.

(i) Reverse Rotation Pressing the "D" key repeatedly to reduce the duty cycle of the forward PWM waveform until forward duty cycle=0. At this point, if the "D" key is pressed, the reverse PWM duty cycle=16 while the forward PWM is a constant zero, and the motor will rotate in the reverse direction. Further presses of the "D" key should cause an increase in the duty cycle of the reverse PWM output to the L298 Input 2 (from 0 to 240 in steps of 16) *making the motor to spin in the reverse direction* until the max duty cycle is reached. Finally pressing the "UP key" should reduce the duty cycle of the PWM output to bring the motor to a slow halt.

Finally, using the "fast motor stop" option of L298, implement an "S" keypress (an "EMERGENCY STOP" key) to bring the motor to a quick and complete halt regardless of the current power level.

Optek OPB813 S3 Optical Switch

1) This step looks at the construction of an optical encoder using the OPB813 S3 Slotted optical switch. Read through the details of the data sheet for the OPB815S3, which is (almost) identical to the OPB813S3 that you will be using. Follow the suggested testing procedure outlined in the next section to verify that your slotted optical switch is indeed functioning.

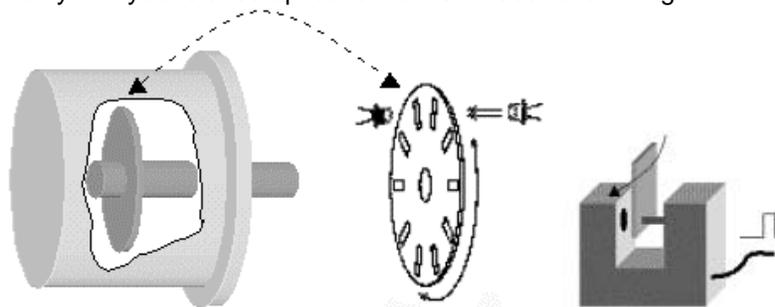


Figure 3: Use of a Slotted Optical Switch Infra-red LED Emitter/Photo-Transistor Pair for creating pulses.

The Optek OPB813S3 Infra-red LED Emitter / Photo-Transistor Pair provided consists of an infra-red light emitting diode (LED) and a phototransistor. The LED is a two terminal device with anode ("A") and cathode ("K") pins. The photo-transistor is a two-terminal device with collector ("C") and emitter ("E") pins. From the data sheet, determine the identities of the individual pins of the OPB813S3 and illustrate in your lab report, the bottom-view pin configuration, and precisely label each of the four pins.

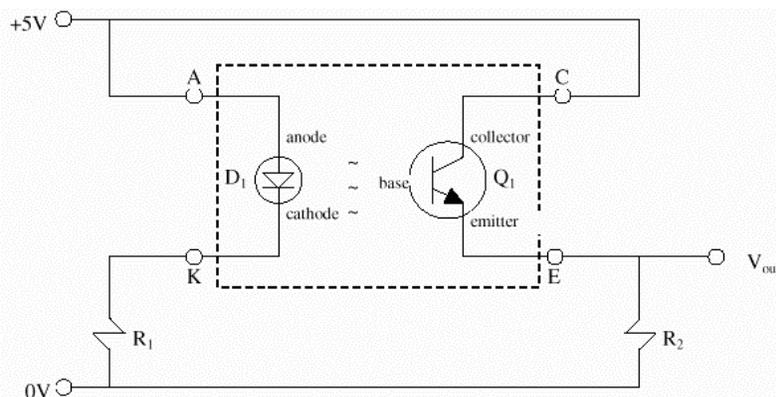


Figure 4: Circuit diagram for using the Optek OPB813S3 Slotted Optical Switch

Figure 4 shows a commonly used circuit for using this slotted optical switch. Use a resistor (R_1) in series with the LED to provide a current of about 20mA using the 5 Volt power supply from the workbench. Resistor $R_2=10K\Omega$ acts as a pull-down resistor when the phototransistor is OFF and creates the desired voltage drop when the phototransistor is turned ON. You can use your oscilloscope to measure and note the output voltage waveforms as an opaque object is moved in and out of the “slot” of the emitter/detector pair.

Position the disk with holes (provided with motor kit) and the optical switch so as to get a series of pulses when the motor shaft is rotated by hand. *Ensure that the clearances on all sides of the rotating disk are adequate.* Count the number of slots on the disc. This is the number of pulses created per revolution of the motor shaft. Use the MCU timer and pulse accumulator (PA) interrupt to measure the time taken for one pulse. Note that the timer only measures the duration of one pulse, and the PA only counts the number of pulses. Create a small loop to repeatedly time the arriving pulses (5 to 10 times in succession) and take the average value.

MCU Timer and Pulse Accumulator

Read the technical summary and find out all the registers related to the MCU timer and pulse accumulator. Here is a brief overview. Basically, the pulse accumulator is a event counter which triggers an interrupt (“pulse accumulator interrupt”) when a falling/rising edge is detected at a pin. It is capable of counting the *number* of occurrence of a event, but not the *durations* of it. Conversely, the timer (there are many timers in the MCU) is capable to measure the duration of *one* single event. There are 2 different timer modes, the *Input Capture (IC)*, and the *Output Compare (OC)*. The IC is intended for measuring a delay event, while the OC is intended for generating a delay event. Now, using the PA and the timer, it is possible to count and measure the output pulse from the optical sensor. There is a lot of excellent sources of information online for the HC1, such as:

<http://www.mcumaster.com/hc11/Block/PAC/pac.html>

LEDs and PC terminal Displays

On the terminal screen, print the following every second: (i) Output PWM duty cycle, and (ii) Measured RPM. Also, the Current Rotation direction is indicated by 2 LEDs as follow:

	Forward LED	Reverse LED
Motor Stopped	OFF	OFF
Motor Rotating Forward	ON	OFF
Motor Rotating Backward	OFF	ON

5. Suggested Workflow

Task Partitions

- PC terminal, LED, PWM
- Optical Sensor, MCU timer and pulse accumulator
- Motor Driver and Motor

Suggested Steps

- Observe sensor output pattern in the oscilloscope by (un)blocking diode-sensor disk manually.
- Observe sensor output pattern in the oscilloscope by turning the motor disc manually.
- Observe the PWM pattern on the oscilloscope and verify the changing duty cycle when “D” or “U” keys are pressed.

- Check the MCU timer and PA functionalities by directly connecting the PWM output to the PS input. Test your function by measuring some known PWM patterns.
- Connect the sensor output to MCU and measure the number and duration of the pulses using the MCU timer (hint: use *input capture* mode) and pulse accumulator. To achieve high quality measurements, accumulate 16 measurements and calculate the mean value.
- Finally, combine the different components and check if system is working as designed.

6. Lab Report

Provide a self-standing document, which could be used as an “Application Note” which describes and explains your system and would enable someone else to replicate your work. Specifically:

- Describe the implementation and operation of the two demonstrations & the intermediate calibration steps.
- Tabulate measurement results and plot an RPM vs Duty Cycle curve. For each "power level" corresponding to a particular PWM duty cycle, determine that the motor settles down to a fixed speed (rpm). How linear is the No-load RPM vs Duty Cycle curve? How accurate is the speed control?
- Document your system with circuit diagrams and list of components.
- Include your program code with concise comments