

## EE 308 – LAB 8

## Using the HC12 Pulse Width Modulation Subsystem and the HC12 Pulse Accumulator

The speed of a motor can be adjusted by powering it with a pulse width modulated signal. Figure 1 shows how this can be done. The field effect transistor (FET), IRLD024, acts as a switch. When the signal on the gate of the FET is high the switch is closed, current flows through the motor, and the motor speeds up. When the signal is low, the switch is open, no current flows, and the motor slows down. With a high enough frequency PWM signal the amount the motor speeds up and slows down in one period is negligible, and the motor turns at a constant speed. By adjusting the duty cycle the speed of the motor can be controlled. For the motors you will be using in this lab, use a PWM frequency of about 5 kHz. Also, connect a power resistor of about  $10\ \Omega$  in parallel with the motor. (The FET switches we are using do not work well with low current draw. The  $10\ \Omega$  resistor increases the current through the FET so the switch will work as designed.)

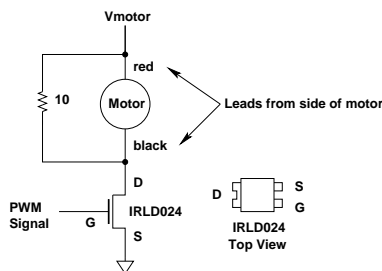


Figure 1: Using a pulse-width-modulated signal to adjust the speed of a motor.

To make a motor turn at a desired speed it is necessary to know how fast the motor is turning. The motors you will use in this lab have encoders which will generate 500 pulses in a single revolution of the motor. The encoder works as shown in Figure 2. A light emitting diode (LED) sends light through a slotted wheel attached to the shaft of the motor. An optical sensor is used to detect the light. When the light shines through a slot and hits the sensor, the output of the optical sensor is  $V_{CC}$ , a digital 1. When the slotted wheel blocks the light from reaching the sensor, the voltage goes to 0 V, a digital 0. Using this optical encoder you will be able to measure the speed of the motor.

In this lab you will use the PWM subsystem of the HC12 to adjust the speed of a small motor, and use the input capture subsystem to measure the speed of the motor.

For the **prelab** do the calculations required for Parts 3 and 4. Write the program required for Parts 5.

1. Connect the motor directly between 0 and 5 V. (Do not use the FET switch for this part.) Verify the the motor turns. Connect the red lead from the optical encoder to  $V_{CC}$ , and the black lead to ground. **Have a TA or Instructor check your wiring before turning the power on.** The motors are fairly expensive, and we do not want to burn up any of them. Use a logic probe to verify that the green lead from the optical encoder is pulsing, indicating that the encoder is working.
2. Program your HC12 to generate a PWM frequency of 5 kHz, with a PWM duty cycle of 50%. Connect the circuit shown in Figure 2. Look at the output of the optical sensor with a logic analyzer. Verify that the optical sensor is working, and record the period of the signal coming out of the optical encoder. What is the speed at which the motor is turning?

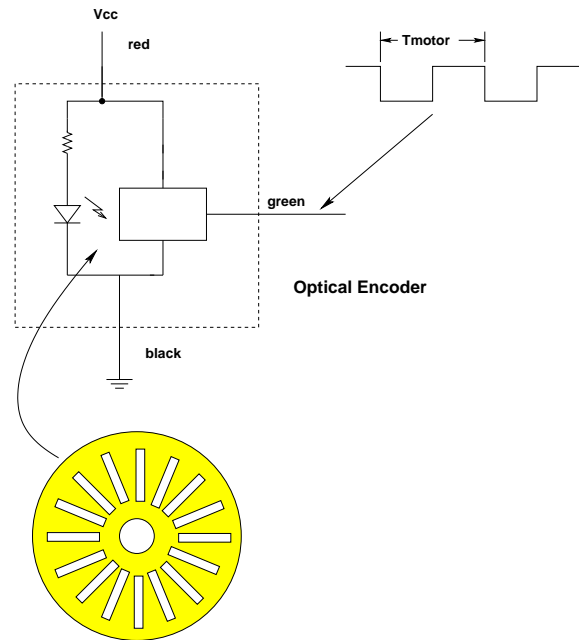


Figure 2: Optical encoder which generates 500 pulses per revolution of the motor.

3. Connect the output of the optical sensor to one of the input capture pins of the HC12. To your program from Part 2, add code to use the input capture function to measure the period of the signal from the optical encoder, and to print the value to the terminal. (Note you should use an interrupt to measure the period, and in the interrupt service routine set a global variable to the number of clock cycles between rising edges. You should print this number to the terminal in the main program loop, not in the ISR.)

When I tested my motor, found the maximum speed of the motor with a 5 V power supply to be about 1,500 RPM. The slowest it would reliably go before stalling was about 100 RPM.

- (a) With a 1,500 RPM motor speed, how much time will there be between successive rising edges from the optical sensor?
- (b) With a 100 RPM motor speed, how much time will there be between successive rising edges from the optical sensor?

Set the prescaler of the HC12 so the rollover time of the TCNT register is the smallest time which is larger than the above two numbers. What value did you use for the prescaler?

4. Modify your program to set the duty cycle based on the state of four of your DIP switches. The duty cycle of the signal should be determined by reading the four DIP switch values into the four least significant bits of Port B:

PB3:0	Duty Cycle	PB3:0	Duty Cycle
0000	6.25 %	1000	56.25%
0001	12.50%	1001	62.50%
0010	18.75%	1010	68.75%
0011	25.00%	1011	75.00%
0100	31.25%	1100	81.25%
0101	37.50%	1101	87.50%
0110	43.75%	1110	93.75%
0111	50.00%	1111	100.00%

Note that you should pre-calculate the duty cycles as the integer you will write to the duty cycle register which will give duty cycles closest to the values in the table. (Do this for part of your pre-lab.) Do not use floating point arithmetic in your C program.

5. Run the motor with each of the duty cycles from Part 4. Have the HC12 print the number of timer pulses between rising edges to the terminal. Record these numbers, and convert them to RPM. Record the speed in RPM for each duty cycle. Plot the motor speed as a function of duty cycle. How linear is it?
6. Compare your results (speed of the motor vs. duty cycle) to the results of at least two other groups. Do the motors all behave the same, or are there significant differences?