EE-3306 HC6811 Lab #6

DC Motor Control Project

The purpose of this lab is to synthesize the programming concepts learned in the previous 68HC11 labs along with some circuit design into a working controller for a DC Motor with user input from the LCD display. The PWM modulation technique learned in the third lab will be used to control the speed of the motor while pulse detection will also be implemented to detect the RPM of the motor. Lastly, the approximation types will be used to approximate the curve of the motor speed with respect to the duty cycle of the PWM.

Objectives

- Understand how to code the microcontroller to run a DC motor.
- Understand how to use the 68HC11 to detect pulses.
- Understand the circuitry used to drive a DC Motor using the 68HC11.
- Apply Pulse Width Modulation (PWM) to the application of controlling the speed of a DC Motor.

Prelab:
Answer the question given in the file Prelab_motor and also design the circuit.

References

Equipment for this lab:
- 68HC11 trainer kit, to include 68HC11 EVBU and prototyping strips
- IBM compatible PC to connect to the trainer kit via an RS-232 serial cable
- Agilent E3649A DC power supply
- Agilent 54621D oscilloscope
- Floppy disk provided by the student

<table>
<thead>
<tr>
<th>Part Description</th>
<th>Part Number</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photo Emitter</td>
<td>160-1063-ND</td>
<td>1</td>
</tr>
<tr>
<td>Photo Detector</td>
<td>160-1065-ND</td>
<td>1</td>
</tr>
<tr>
<td>Mounted DC Motor with Slotted Disk</td>
<td>FE-260-18130</td>
<td>1</td>
</tr>
<tr>
<td>NPN General Transistor</td>
<td>2N222A</td>
<td>≤2</td>
</tr>
<tr>
<td>Rectifier Diode</td>
<td>1N4004</td>
<td>1</td>
</tr>
<tr>
<td>Resistors and Capacitors (Varying Sizes as Needed)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Laboratory Exercise

Timing
It is anticipated that this lab will take two weeks to complete.

Notes
• In this lab, all numbers such as addresses and data are given in hexadecimal format (“hex”) unless otherwise indicated. In completing the lab, record all information in hexadecimal unless directed differently.
• Make sure your development EVBU board is connected to power(green LED on board ON) and the serial port of the EVBU is connected to the serial port of your development PC containing the AXIDE software, configured to the correct port at the correct baud rate, etc.
• Be sure to bring your RTF file from the Timers lab. We have provided you with a starting point named stud_version1_dc_lab.RTF. Download and modify this file according to the specifications below. The final code you submit should be named DCMotor.RTF.
• The motor’s speed should not exceed 6000 RPM. If the motor is shaking enough to move around the table, this is a good indication that the motor is moving too quickly.
• IMPORTANT: Do not touch the slotted disk while it is spinning. Doing so could result in bodily harm.

Part 1—Fully Functional DC Motor Control

In this lab you will create a control interface and use the knowledge from lab 3 to create a PWM square wave to drive a DC motor. To accomplish this, you will need to write a code that includes tasks like increment, decrement and doubling the speed of the motor. Another requirement is to include a part in the code which enables the user to control the speed of the motor manually. This is done by altering the duty cycle of the PWM wave proportional to the speed entered by the user. (the equation used will be dictated by which approximation type is selected currently). Also need to accept pulses generated by an emitter-detector pair, calculate the motor’s RPM, and display it on the screen. Below is a list with the specifications for the interface program. Follow the hints to write the code to create an interface for the DC motor. The TA will have a working example of one implementation.

When in “Display Mode”:

Increment Speed by 100………………………………… …….When key ‘A’ is pressed.
Decrement Speed by 100……………………………….............When key ‘B’ is pressed
Double Speed………………………………………………….. When  key ‘C’ is pressed
Reset Speed and Approximation Type to Defaults…………… When key ‘*’ is pressed
Enter “Speed Manual Entry Mode”……………………………..When key ‘#’ is pressed
Enter “Acceleration Type Entry Mode”……………………….. When key ‘D’ is pressed
Update the Display with the Current RPM…………………..…When key ‘0’ is pressed

When in “Speed Manual Entry Mode”:
Enter Speed and Return to “Display Mode”…………………..#

When in “Approximation Type Entry Mode”:
Use Linear Approximation………………………………………0
Use Positive Quadratic Approximation…………………………1
In order to complete Part 1, you will need the basic circuits which you designed in your prelab. IMPORTANT: Your TA must verify your circuit before supplying power to them.

You will need to modify your program to control the duty cycle of the output wave from the 68HC11 to spin the motor according to (1) the speed entered and (2) the current approximation type:

- Linear: Duty Cycle = A*Speed + B
- Quadratic: Duty Cycle = A*Speed^2 + B*Speed + C

In these equations, A, B, and C are constants. Also, the positive quadratic approximation will be concave up while the negative quadratic approximation will be concave down. Your program will ONLY need to handle POSITIVE speeds and should be limited so that the maximum RPM of the motor does not exceed 6000. For convenience you may implement the speed so that it is initialized and resets to 100. The diode in Figure 1 is included because of the use of Pulse Width Modulation. During the period of time at which the 68HC11 output is logically low, it creates an alternate path for any small current that can drive the motor. However, the addition of the diode allows current to continue to flow through an alternate path.

Figure 1: Motor Control Circuit
Part 2
In the second part you will be adding code to the previous part to calculate the RPM of the motor. In addition to the code you will need a circuit to calculate the RPM.
The circuit that you will need will be a pulse detector circuit using the photo emitter-detector pair provided. The disk attached to the motor will spin between the emitter and detector, blocking the light except when the slot passes by them. The emitter of this pair is simply a light emitting diode (LED). When the light from the emitter strikes the detector, a voltage is created across the terminals of the transistor (the light basically takes the place of a base current). Your program must then use the output of the detector as input for the 68HC11 to read in the generated pulses. These pulses can then be used to calculate the RPM of the motor and display it on the LCD display. The general circuit is shown in Figure 2. Please enter the values of R1 and R2 used in your circuit below (you must design the circuit so that $I_F = 20 \text{ mA}$).

![RPM Detector Circuit](image_url)

Figure 2: RPM Detector Circuit

R1: ___________
R2: ___________
Note: Since speed is incremented by 100 and multiplied by 100 in the manual entry mode, the result of (speed/100) is always an integer.

Requirements of the program:

1. When in the “Display Mode”:
   • The LCD should display the current approximation type (default should be 0 (linear))
   • The LCD should display the current speed (default should be 0)
   • The program should respond appropriately to the keys specified on p.2 (and display an error message for others)

2. When in the “Speed Manual Entry Mode”:
   • The LCD should display a line requesting the user for a new speed (e.g. “Please enter the speed”)
   • The LCD should display the numbers for the speed as they are entered
   • Once a number is validly entered, the speed will update to 100 times the number entered in the “Display Mode”

3. When in the “Acceleration Type Entry Mode”, the LCD should display:
   • The options for the different acceleration types (0 = linear, 1 = pos quadratic, 2 = neg quadratic)
   • If an intermediate mode is used for direction entry, the direction options should be displayed

4. Error Detection:
   • If the user presses a character from the keypad that is not specified as an input for a given mode, the LCD should briefly display an error message and then return to the previous display (you may use a single error message for all types of errors).

5. Display the RPM:
   • The LCD should display the current RPM of the motor (default should be 0)
   • The program should respond appropriately to the keys specified on p.6 (and display an error message for others)—IMPORTANT: note the addition of the ‘0’ key.

For the approximation types, the equation is designed so that the duty cycle is 25% for a speed input of 5000 and 0% for 0 (note, your program doesn’t need to take 0 as an input—this is just for the equation you will use).

6. For the speed of the motor and RPM display:
   • IMPORTANT: The output duty cycle must be at least 1% to hold the switch low.
   • The speed should increase and decrease immediately after the increment or decrement keys are pressed.
   • The speed should immediately adjust upon entering of a new speed or approximation type.
   • The RPM should update immediately any time there is an update of the “Display Mode” (the displayed RPM should be within +/- 100 of the actual RPM).

7. The time display with the time that the program has been running should update any time the “Display Mode” is updated.
Demonstrate to the T. A. that your program is running. T. A. Initials: ______________________

Suggestions
The second part will need the following as a part of the earlier code:

- A new subroutine to display the RPM (or a modification of the subroutine that displays the speed and approximation type).
- A new portion of code within the interrupt subroutines that counts the number of pulses accumulated over a known period of time.

Part 3—Questions
When you turn in the lab report, please answer the following questions. Provide flow charts and/or assembly language programs as necessary to explain your answers.

1. What did you find easiest about this lab? What was most challenging?
2. Take at least 5 points of data for varying speeds for each of approximation types and graph the user-entered speed versus the actual speed (from the oscilloscope). Which curve best approximates the speed curve of the motor? Name one way of more accurately controlling the speed of the motor so it reflects the speed entered by the user.
3. Take at least 10 points of data for the RPM calculated by your program and the RPM calculated from the oscilloscope. What could cause a difference between the actual and computed RPM? Draw a circuit that could aid in the accurate counting of pulses.
4. The circuit designed in the prelab was only one way of stepping up the current to drive the motor. Design an alternate circuit and name at least 2 advantages and 2 disadvantages of your new circuit as compared to the one designed in the prelab (hint, you can use online resources such as www.digikey.com to determine parts, prices, electrical characteristics, etc.)