Lab 6 Motor Control

Introduction

Microcontrollers such as our MSP432 operate at low voltages and low currents. Many real world devices and switches operate at much higher voltages and currents. For example, the motors on a typical wheelchair have a typical operating condition of 24V at 20Amps, yet can be controlled with only a 3.3V, 2mA signal. The amplification of the microcontroller signal using transistors or MOSFET's, or usage of relays is a solution for simple on-off circuits. Motor control can be more complex. For example, DC motors require continuous voltage variation while stepper motors require pulsed timing patterns. Further, both need not only speed control, but direction control. For the MSP432, TimerA0 and TimerA2 have the capability of mapping the timer output directly to pins that are connected to the motor controller (H-Bridge), which will allow driving PWM signals directly with low CPU overhead.

Files

https://kgcoe-git.rit.edu/CMPE_460/CMPE_460_Files
See Lab6

Objectives

For this exercise, you will understand how to amplify low power signals from the microcontroller to drive higher power devices. Further, this lab will explore how to drive motors using H-bridge circuits, servo motors with PWM signals, and stepper motors with Darlington transistors.

Prelab Activities

1. This lab will use Lynx motion GHM-01 gear head DC motors. The characterization data of the gearbox output is specified with the curves in Figure 6.1. The figures include the gear box. The gear reduction is 30:1. From the curves below, estimate:

    (a) Motor stall torque in oz-in and in kg-cm before reduction

    (b) Maximum current draw

    (c) Maximum motor turn speed before reduction

(d) Maximum torque in oz-in after gear reduction

(e) Maximum turn speed in kRPM after gear reduction

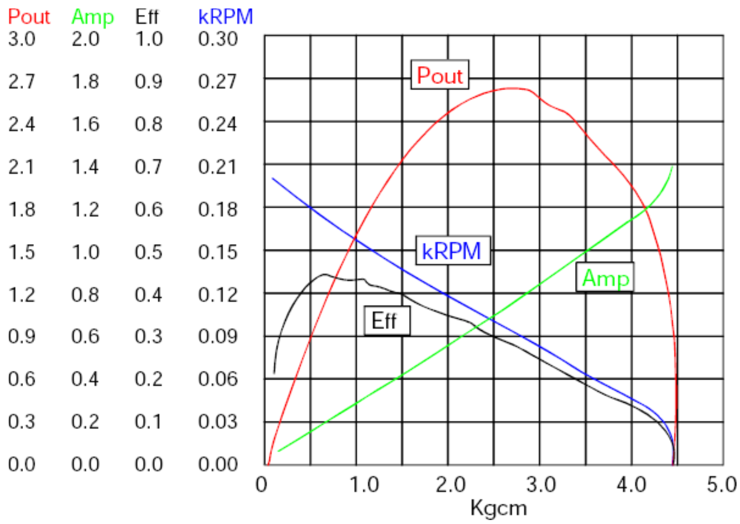

Figure 6.1: Motor Characterization Curves

2. A typical microcontroller can source or sink a few milliamps of current per IO pin. As such, if we were to generate a 5V, 2mA PWM signal from the MSP432, it would not provide sufficient power to turn our DC motor. To solve this problem, transistors, H-bridges, and motor controllers are used. Each of these amplify the output signal from the I/O port to the appropriate voltage and current necessary to drive the motor. For example, we could use NPN and PNP transistors to construct our own H-bridge. In the circuit in Figure 6.2, by having I/O signal A high and I/O signal B low, the motor turns in one direction, and swapping their polarities makes the motor turn in the other direction. Answer the following questions based on Figure 6.2.

   (a) What is the purpose of the diodes in this circuit?

   (b) What is the purpose of the capacitor in this circuit?

   (c) Why are the transistors in each side of the circuit in pairs?

   (d) Why use 2N3904/2N3906 for some transistors and TIP31/TIP42 for others?

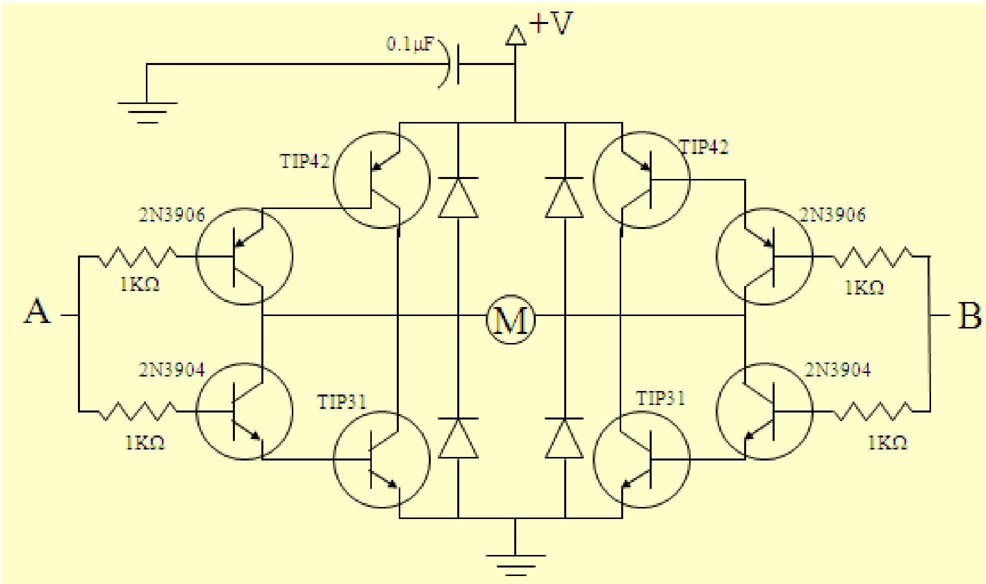

Figure 6.2: DC-Motor Control H-Bridge

Lab Procedure

1. Create a new project in Keil μVision 5.13.0

2. Copy your LED/uart code into the project and start with main template Lab6-DCMotor.c

   - Create a TimerA.c and TimerA.h file to handle the TimerA PWM (pulse-width-modulation) signals (including both TimerA0 and TimerA2). Also create a ServoMotor.c and ServoMotor.h to handle the servo signals. Finally, you should create a Motors.c and Motors.h to handle the DC motors. Create initialization functions for the Timers and dc motor and servo motor and any other functions that you might need.
   - NOTE: The Shield Pinout section of the Lab Manual has the Port and Pin information you need to connect your motors to the K64.
   -

Part I

3. Create code to generate a 20% duty cycle signal at 10KHz. using TimerA0. The PWM signals are generated on pins 2.4 and 2.7, consult the MSP432 Reference Manual to find the corresponding select settings for the MSP432 board. **Display on oscilloscope and demonstrate to your lab instructor or a TA to be checked off.**

4. Explain how the code works in your report and attach oscilloscope screenshot in your report.

5. The circuit shown in Figure 6.2 is offered as a prepackaged integrated circuit such as the L293NE/SN754410 which contains four half bridges. Two half bridges make a full H-bridge, so one L293NE/SN754410 IC can drive two DC motors. For simplicity, we'll only drive one motor, making connections as per the L293NE/SN754410 datasheet and class lecture slides. Connect the microcontroller generated PWM signals to input pins on the chip (use left side). Connect the enable pin and Vcc1 to +5V from the MSP432. Connect Vcc2 to +10V from power supply. Note: to drive the motor clockwise, one of the two signals is set to ground or 0% duty cycle; to change direction, swap which signal (input pin) is PWM and which is ground.

6. Before connecting your circuit to the motor, verify the signal on the output pins L293NE/SN754410 on the oscilloscope. Attach a screenshot showing the input 3.3V waveform and output 10V waveform on the oscilloscope. After verification, go ahead and connect to the DC motor.

7. In your report, **explain how to change speed and direction of turn of the DC motor.**

8. Drive the provided DC motor with a 10KHz, 20% duty cycle in one direction.

9. In the main() function, write an infinite while loop of code that does the following:

   - Generates a 0% to 100% duty cycle in forward direction
   - Generates a 100% down to 0% duty cycle in the forward direction
   - Generates a 0% to 100% duty cycle in reverse direction
   - Generates a 100% down to 0% duty cycle in the reverse direction

   Put a 10*ms* delay between each duty cycle and verify the code is working with an oscilloscope, then attach the H-bridge circuit

10. **Draw a wiring diagram showing the MSP432, the H-bridge chip, and the DC motor.**

11. **Demonstrate the above code that turns the motor at different speeds and direction to the lab instructor or a TA before continuing.**

12. The above method requires two PWM lines, one for forward, and one for reverse. **Describe in your lab report an alternate method in your report that uses only one PWM line and one GPIO line.**

Part II

13. Stepper motors are good for precise positioning and when there is a need for strong holding torque. You are provided with a sample stepper motor.

14. There are eight wires coming from the stepper motor. Note that some of the stepper motors have a gray ribbon cable, others have colored wires. Aside from that, the motors are identical. We could use the tips in the lecture notes to find each of the four coils, but, if you did, you would find we have 4 independent coils. This makes it difficult to figure out which are the A/C and which are the B/D coils. So, using Table 6.1, connect the stepper motor to the ULN2068B/ULN2083A Darlington array (noting that common wires of stepper motor need to be connected to Vss, not to ground). **Set Vss to +5V from external power supply.** Starting with the red wire (if gray ribbon wire, pick either end to start), the motor leads are in the order (NOTE: 'G' is NOT ground) **C G D G G A G B**, where all the wires labeled "G" which should go to Vss, and the lettered wires are driven from the microcontroller via Darlington transistor switches. Be very careful to have the order of the lettered wires correct. For example, the first C wire on the stepper gets connected to the line marked C (1b). Make sure ground of power supply is connected to ground of microcontroller. Use Pins [4.1 - 4.4] to drive the Darlington transistor switches.

15. Note that you will need a resistor between the MSP432 and the Darlington array. The datasheet for the ULN2068 has the information to calculate the resistor value. Use a value of 1K.

16. **Draw a wiring diagram showing the MSP432, the Darlington array, and the Stepper motor.**

Note: The stepper motor draws ~1A of current and the ULN2068 ULN2083A/ULN2083A chips are only designed to handle ~0.5A. Do not run the Stepper motor for more than 10-15 seconds at a time to prevent overheating and damage to the circuit components. The ULN2068/ULN2083A can get hot enough to burn to the touch, so be careful.

Table 6.1: Darlington Connectors

From Microcontroller	To Stepper
P4.1	A, 1a
P4.2	B, 2a

P4.3	C, 1b
P4.4	D, 2b

17. Modify your main function with sample code below, demonstrate turning the stepper motor at different speeds and directions to the lab instructor or a TA. Referring back to the class notes, **which stepping mode does this code use?**

```
//Configure the Signal Multiplexer for GPIO Pins

// Configure the GPIO Pins for Output

int forward = 1;
int phase = 0;

while(true){
    // Turn off all coils, Set GPIO pins to 0

    //Set one pin high at a time
    if(forward){
        if(phase == 0){/*turn on coil A*/; phase++;} //A, 1a
        else if(phase == 1){ /*turn on coil B*/; phase++;} //B,2a
        else if (phase == 2) { /*turn on coil C*/; phase++;} //C,1b
        else { /*turn on coil D*/; phase=0;} //D,2b
    }
    else    {//reverse
```

```
        if (phase == 0) { /*turn on coil D*/; phase++;} //D,2b
        else if (phase == 1) { /*turn on coil C*/; phase++;} //C,1b
        else if (phase == 2) { /*turn on coil B*/; phase++;} //B,2a
        else { /*turn on coil A*/; phase=0;} //A,1a
    }
//Note- you need to write your own delay function
    delay(10);  //smaller values=faster speed
}
```

Part III

18. Your TI IDE Cup Car also has a servo motor for controlling the steering.

19. **Draw a wiring diagram showing the MSP432, and the Servo motor**

20. Procedure**:**
- Create a new project which can drive the Servo motor using the TimerA2 timer.

    - Run the servo motor at 50Hz on Timer A2 . You will be using a pulse width to control the direction and position of the motor. Most servo motors use a standardized control signal. Google it). You will create a pulse between 1ms and 2ms. A pulse of 1.5ms should center the servo motor. Start by creating a 1.5ms pulse on a 50 Hz clock (20ms period).
        - **Display on oscilloscope and demonstrate to TA to be checked off before hooking up to the servo motor.**

    - NOTE: The servo motors will be in a box in the lab. You are only to use them in the lab and they are not to be removed. Thank you.
    - Modify your program to turn the servo from the CW position to the CCW position and back for signoff.

Part IV

21. Your TI IDE Cup Car has two DC motors and a servo.

NOTE: Use the power supply to power the car set at 6V, 2A

- Create a new project which can drive the DC motors using one timer module and the servo on a second timer module.
    - Run the DC motors at 10KHz using your code from the DC motor section of this lab.

    - Run the servo at 50Hz on Timer A2 using your code from the servo section of the lab.

        - **Display on oscilloscope and demonstrate to TA to be checked off before hooking up to the TI Car.**

    - Then display your TI Cup Car rear motors **and** servo being driven simultaneously. (It is okay if the two DC motors have the same duty cycle for this lab.) Use 30% for the duty cycle.

Shield Pinout

The adapter board sits on top of the MSP432 and fully covers all headers. The following table displays which pins on the motor shield connect to which pins on the MSP432.

A1, A2, B1, B2 are tied directly to the PWM input on the motor shield. EN A and EN B are tied to the H-Bridge enable pins on the motor shield. In order to use one of the motor outputs the respective enable pin must be set to logic 1. For example, to use H-bridge A, P3.6 must be in GPIO mode and set high. The enable pins should be initialized to a logic zero state upon power up.

Table 6.2: MSP432 Pinouts

Car Pin Mapping

OLED	**1306**
SDA	P1.6
SCL	P1.7

Camera		**Timer**
CLK	P5.4	SysTick
SI	P5.5	
AO	P4.7	

Motor		**Timer**
M1A	P2.4	TA0.1
M1B	P2.5	TA0.2
M2A	P2.6	TA0.3
M2B	P2.7	TA0.4
M1EN	P3.6	
M2EN	P3.7	

Servo		**Carpet Det**	
SIG	P5.6	SIG	P5.2

Lab Worksheet

Please submit one worksheet per group. The worksheet for this week should include:

1. Title of the Lab, your name, date the lab was performed, course name, TA's name and instructor's name. (Use template from GitLab).
2. Description of lab
3. Wiring diagrams
4. Neat, well annotated wiring diagrams (using a drawing program draw.io, Visio or other) with actual resistor values.

5. Answers to prelab questions
6. Oscilloscope captures
7. High level description of how code works
8. Answer to in-lab questions
9. Attached the completed Lab Sign Off Sheet to the back of the worksheet

Exercise 6: Motor Control

Student's Name:_________________________ Section:_________________________

PreLab		Point Value	Points Earned	Comments
PreLab	Motor Calculations	10		
	H-Bridge Questions	10		

Demo		Point Value	Points Earned	Date
Demo	20% Duty Cycle at 10kHz	10		
	DC Motor Functionality	5		
	Stepper Motor Functionality	5		
	Servo Motor Functionality	5		
	Simultaneous TI Car Motors-Servo Motor	15		

To receive any grading credit students must earn points for both the demonstration and the report.

Exercise 6: Motor Control

Report		Point Value	Points Earned	Comments
Exercise Description		5		
Circuit Schematics/Wiring diagrams		10		
Oscilloscope Captures	20% Duty Cycle	5		
	TI Car	5		
Code Explanation		5		
Questions		10		
Total for prelab, demo, and report		100		