BIEN4220: Laboratory 4 – Servo and Stepper Motor

Introduction

This laboratory exercise further develops techniques in interfacing to the GPIO port on MSP430. The H-Bridge is introduced to implement bi-directional motor control.

Preparation:

1. Download the datasheet for the SN754410ne H-bridge and review it.

Part I: Generating Motion with RC Servo Motors on the MSP430

This part of the lab will require you to use two independent PWM signals to control two independent servo motors. Your first task is to play around with the servo to understand the input signals needed to drive it. Then you will develop a program to generate two PWM signals and confirm this with the scope. After your PWM signals match the specifications of the servos, you will then implement two buttons to control each of the servos. One you observe the desire behavior of your system, only then should you interface the MSP430 with the servo. **WARNING: Sending an inappropriate signal to the servo can damage it!** So, make sure you confirm the PWM on the scope first.

1. Let's first get to know the servo motors. Hook one of them to power and ground of the DC power supply. Set the DC output to 5V. Next, use the Function Generator to make a PWM signal and interface it with the control line on the servo; ensure the ground of the FG is shared with the DC supply ground. Ensure the square wave on the FG ranges from 0-3.3V with the appropriate period and duty cycle (as seen in lecture). Some servos have different position limits – you should find yours. Most servos will want pulse widths around 1-2ms. Modulate your pulse widths using the FG to find the limits of your servo motor – **make very small increments to avoid damaging your servo!** You will want to document these limits so that your MSP430 only generates pulse widths within this range.
2. Obtain either a Pan/Tilt or robotic gripper servo mechanism from your instructor. Create a new assembly program to implement a dual-channel PWM controller using Timer A (seen in lecture). You will use these two PWM signals to control the position of two servo motors. You should be able to pause your code, modify the duty cycle of your PWM, resume your code, and observe the PWM changing duty cycle. You should confirm your outputs are as desired by looking at them on the scope.
3. Add the ability for your program to take the input of *two* buttons. Each button must be used to control a servo motor in some way – be creative! You can use interrupts or polling methods for button conditioning.
4. After your system behaves as desired and your PWM matches servo specs, you can then interface your MSP430 with the servo motors. Demonstrate your working system to an instructor:

   Evaluator Signature: _____

Part II: Bipolar Stepper Motor Control with the MSP430: Interfacing with the H-Bridge

This part of the lab is more difficult (due to more code needed). Here, you will control a stepper motor using the MSP430 with the aid of an H-Bridge chip. The SN754410 actually has two H-Bridges in it – you can use one bridge for one winding of the stepper motor, and the other bridge for the second winding. Your MSP430 will have to generate the correct sequence of phases to move the motor.

1. Create a new assembly project. This project will support the use of the H-bridge to implement bidirectional control of the bipolar stepper motor.
2. Write an assembly program to continuously cycle through a **half-step** phase table of a hypothetical stepper motor. Simply running through the phase table as fast as possible might be too fast for the rotor and stater to magnetize together (imagine a track team performing a baton handoff except the person behind you shoots the baton out of a cannon – you probably won't catch it). To solve this, implement a timer ISR with an appropriate period which updates the index of a phase state table.
3. Once you have all 4 output pins modulated according to a phase table, confirm their existence and timing using the scope. The scope only has two probes so you will have to test two signals at a time relative to one another.
4. One you are happy with your output, obtain a bipolar stepper motor and an SN754410 H-bridge chip from your instructor and interface the H-Bridge with the MSP430. Use the class lecture notes for guidance on the circuit connections. Verify that the output of the H-Bridge shares identical timing characteristics as the digital pins earlier.
5. Interface your stepper motor with the H-Bridge and confirm the motor rotates as desired. It is not uncommon for your stepper motor hardware to not match your phase table. You may have to rearrange your table entries to be in a new order or change the wiring of the stepper motor to the H-Bridge. Show your rotating stepper motor to your instructor.

   Evaluator Signature: _____

6. Next, implement the ability to control the *direction* of rotation using a button (or simple wires). Use good coding practice. Demonstrate your working system to your instructor.

   Evaluator Signature: _____

   Extra Credit:
   Implement a second button to cycle through different speeds of the stepper motor.

   Evaluator Signature: _____

Upload this signed handout and the code for the servo and stepper motor to D2L.