

GE 423 Mechatronics Homework Assignment #3
Spring 2009, Due In Lecture March 11th. The Microcontroller Demonstration Check-Offs for Questions 5 and 6 are due by 3PM Tuesday March 10th.
Most answers should be typed. Graphs, etc. can be hand drawn if you wish.

- Review Chapters 5-6 in “Teach Yourself C”. Chapter 6 in “Introduction to Mechatronics”.
- Assume you have designed the touch sensor glue logic shown in Figure 1. This circuit is used in an industrial machine, where the switches are located strategically to indicate the different states of the machine.

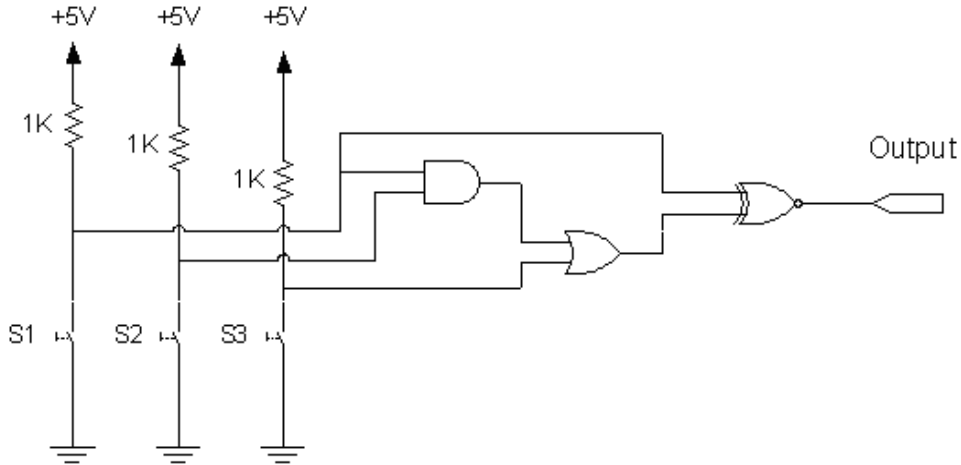


Figure 1.

Depending on the ON/OFF configuration of the switches, the circuit outputs a 0 or a 1 on the output line. Determine all the possible input/output combinations of your digital circuit. Note: The rightmost element is a logic XNOR gate.

- Assume you have three digital output lines coming from your DSP as show in Figure 2. These lines are used to enable and disable different sensors on a mobile robot. Design the glue logic required to map the lines [b0 b1 b2] to SENSOR #1 as shown in Table 1. See “Introduction to Mechatronics” Chapter 6 for help on this problem. Especially useful pages: 187-188 and 192-193.

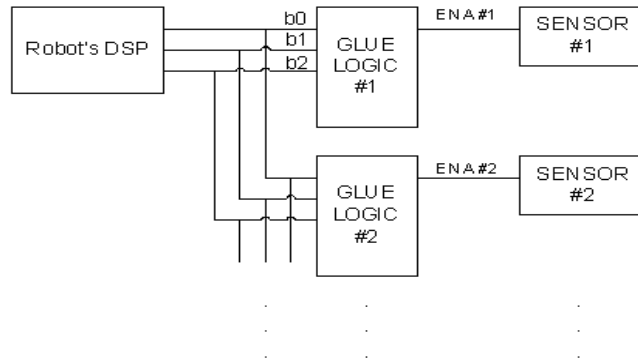


Figure 2.

b0	b1	b2	ENA #1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Table 1.

4. Using the schematic given in Figure 4 below, what C instruction should be used to read the 32-bit value stored at memory location 1150 in the SRAM chip? What C instruction will write 8423 to the SRAM memory location 0xdf2? What C instruction will command the LS7266's X and Y RLD registers to transfer the X and Y encoder counts (CNTR) to their output latches (OL) and reset their byte pointers? You will need to look at the LS7266 datasheet.

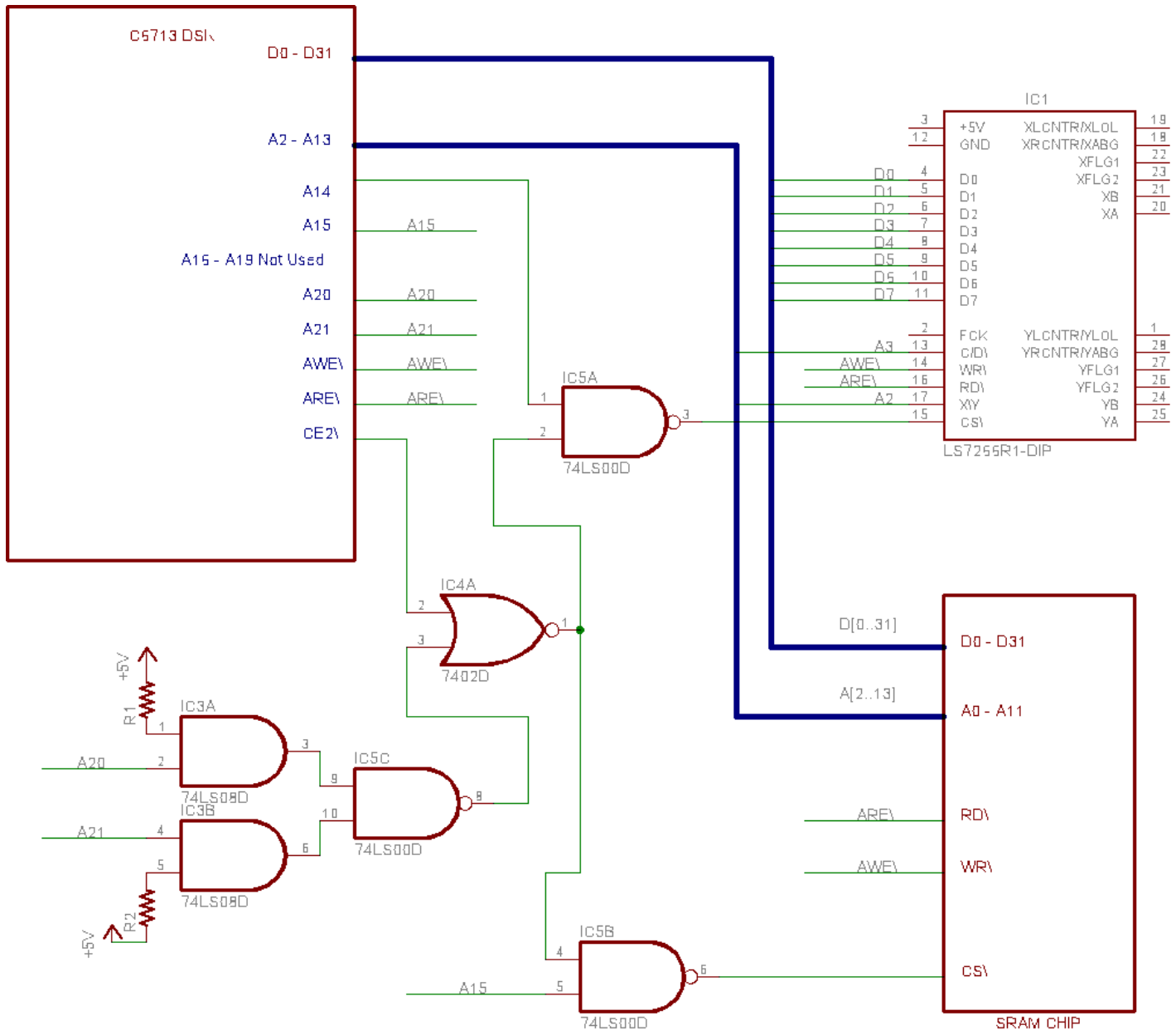


Figure 4:

5. Interface the MSP430F2012 microcontroller to the TLV5606 digital to analog converter given in your EZ430 kit http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/tlv5606.pdf. Notice that the EZ430 Break-Out board has the surface mount pattern for the TLV5606. As a first step solder the TLV5606 chip to the board. The solder traces on the board connect pin 8 (VDD) to +3.3V and pin 5 (AGND) to GND so you do not have to worry about powering the chip. Solder a 0.1 uF capacitor in the capacitor spot just above the DAC (TLV5606). This capacitor is just decoupling possible

noise on the power traces connecting the DAC. Pin 6 (REFIN) is connected to the output of the potentiometer pins located just to the right of the TLV5606. Solder a 10Kohm potentiometer in that POT section. Pin 1 of the pot is connected to +3.3V for you and Pin 3 is connected to GND. You will adjust this potentiometer later to produce the correct REFIN voltage.

To interface to the TLV5606 you will use the USI serial port in SPI mode. The TLV5606 has four pins to interface to a processor, \CS, FS, SCLK and SDO. There is no SDI pin on the TLV5606 because a DAC is just an output device. \CS and FS are both types of \SS (slave select) pins. \CS is used if we desire to communicate to more than one chip with the SPI port. In this exercise you will only be communicating to the one device so you will not use the \CS line except for wiring it to GND through a resistor so that the chip is always selected. Reading the TLV5606 datasheet, this is referred to as a three wire SPI interface. So, using a 2.2Kohm resistor, pull-down the \CS pin to GND on the break-out board. Note that GND pins are just above the \CS pins. When just using the three wire interface the \SS (slave select) signal should be wired to the FS input pin of the TLV5606. Using wires, solder P1.4 to FS (you may have to remove wires used in previous HWs), P1.5/SCLK to TLV5606's SCLK and P1.6/SDO to TLV5606's DIN.

Now you are ready to develop your C code to interface to the TLV5606. (Start with the code you developed for reading the photo-resistor's output because the last step asks you to wire the photo-resistor to A3's input). First power on the board and adjust the potentiometer to output approximately 1.024 volts to the REFIN pin. Do this by measuring the voltage at REFIN with a DMM on the benches. The C example http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/Cexamples/c/msp430x20x3_usi_02.c is a great start for this problem but it does not have all the appropriate settings for our interface. Study the TLV5606 datasheet to determine the polarity of the SCLK signal and the default state of the SDO pin. Also make sure not to use P1.7/SDI as a SPI pin. There are two reasons, first since we can only write to the TLV5606, SDI is not needed, and second we use P1.7 to communicate through the UART to HyperTerminal. P1.4, wired to FS, should be setup as a Digital Output and initially high. In your code when you want to write a new DAC value through the SPI make sure to first pull P1.4 low. Then when the data has been transmitted (i.e. you have received the transmit interrupt) pull P1.4 back high.

Write a program that ramps the DAC voltage from 0V to 2.048V. To do this increment an integer every 0.1 seconds and output this value to the DAC. Watch the DAC voltage increase on the Oscilloscope or the DMM.

As a final step, wire the photo-resistor's output into pin P1.3, ADC channel A3 (make sure P1.3 is not also connected to a LED). Sample the photo-resistor's value every .1 seconds and echo that value to the DAC. Scope the output on the oscilloscope.

Submit your code for this assignment and also demonstrate it working to your TA.

6. Write code to command both of your RC servo motors. **Note: you will not use the DAC to drive the RC servos. The code you developed for commanding the RC servo motors does not use the code you just finished in question 5.** To command an RC servo you need to create a continuous pulse train. The pulse's length of time determines what angle the RC servo motor will turn to. The pulse length is between 0.6 milliseconds (0°) and 2.8 milliseconds (180°). The pulse needs to be repeated every 20 milliseconds. You can think of this signal as a PWM signal with a carrier frequency of 50Hz and a duty cycle that ranges between 3% and 14%. For this assignment you will create two of these pulse trains (or PWM signals) in software using pins P1.1 and P1.2 to command both of your RC servo motors. In HW #2 you created a PWM signal using Timer A's CCR1 register. Unfortunately there is only one compare output that can easily be used for creating a PWM signal. Since you need to drive two motors separately, you will create the PWM signals for the RC servo motors in software. So keeping the sample rate of the default project at 9600 Hz, create two RC servo motor PWM signals by keeping track of pulse

on and off times in Timer A's interrupt service routine. Submit your code for this assignment and demonstrate it commanding the RC servo motors to different positions. **RC Servo Motor Wiring:** Pin1 (Black) → GND, Pin2 (Red) → +5V, and Pin3 (White or Yellow) → PWM Signal. Your TA will also help you with this wiring and show you where to connect the +5V source.

F2012 Play-Time: (These Items are not graded or required)

1. Experiment with how fast you can sample the ADC and echo its value to the DAC.
2. Wire the speaker to the DACs output. Write some code to produce a 0 to 50mV square wave. Vary the frequency of that square wave and you have just made a tone generator. Switch the tones at specific times and you can play some songs. Your TA can show you how to solder the speaker to the demo board.
3. Sample the microphone into a ADC10 channel and output (you possibly will need to scale the value) to the speaker. Turn on an LED for 1 second when a large noise is heard. See the datasheets for the microphone at http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/microphoneWM64_1.pdf and http://coecsl.ece.uiuc.edu/ge423/datasheets/MSP430Ref_Guides/microphoneWM64_2.pdf.