

2.737 Mechatronics

Laboratory Assignment 3:

Digital Electronics: An Encoder Interface

Assigned: Session 13

Reports due: Session 18 in checkoffs

Reading: Horowitz and Hill, Ch. 8; accompanying data sheets

Prelab exercises: The prelab is due in class on Wednesday, Session 16. For the prelab:

1. Understand the operation of the encoder and the circuit. Sketch the requested waveforms as the encoder is rotated clockwise and counter-clockwise.
2. Provide us with a schematic of the circuit with the pin numbers assigned, and explain how you assigned these. Be sure to include a schematic of your debounce circuit.
3. Build and test your debounce circuit. Include scope data showing that it operates properly.
4. Understand the operation of the commercial encoder interface chip and answer all of the questions in that section of the lab. Include sketches of relevant waveforms. How is this commercial device better or worse than the circuit you will build? What are the fundamental differences in operation? What about noise immunity?

1 Introduction

The purpose of this lab is to teach you how to build, understand, and design digital electronics. To this end, you will fabricate a circuit to interface with a rotary encoder. In the process you will also learn about how encoders work. These are an important class of sensors in mechatronic systems. In order to expedite the lab effort, you are given the circuit for the system. The circuits are to be built individually on your protoboard and demonstrated in the checkoff. The parts you need for this exercise are available in the lab.

2 Preparation

Before going into the lab, please take the time to understand how the circuit works. You will find the schematic diagram at the back of this handout. As part of the lab we have also supplied you with data sheets for all of the components. Take a look through these in order to learn how each of the components functions. We have also supplied you with the data sheet for a Hewlett-Packard integrated circuit which implements an encoder interface and provides ready interfacing with a microprocessor. Commercial systems would likely use such a chip. One section of the lab asks you questions about this chip.

1. First understand how an encoder works. To this end, the class lecture on encoders will be very helpful. Please also refer to the data sheet for the Bourns encoder used in the lab.
2. Look at the circuit and be sure you understand its operation. Sketch some expected waveforms through the circuit as the encoder is rotated through several states clockwise, and then through several states counter-clockwise.
3. Note that we have not assigned pin numbers to the logic elements. Read through the individual chip data sheets and use this information to assign pin numbers to all the elements in the schematic. You will need this information before you can wire the circuit. In the checkoff, we will ask you to show us waveforms in various parts of the circuit and to explain these waveforms to us. Thus you will need a copy of the schematic which has the pin numbers, and will need to be able to fairly rapidly find your way around the proto-board implementation.
4. Because we are using an encoder with mechanical switches, you will need to design a debounce circuit in order to eliminate switch bounce. A possible circuit is given in Figure 9.9 in Horowitz and Hill. For our needs, a time constant of a few milliseconds should suffice. If you make the time constant too long, you will not be able to count adequately fast rotational rates. For our purposes, the 'LS14 inverters are functionally identical to the 'HC14 inverters. Design a debounce circuit to fill in the block on the schematic. Take data showing the switch signals before and after the debouncing circuit, and convince us that it is working properly.
5. Prototype the circuit and test it. Take interesting scope data which shows the operation of the circuit.
6. Once you have your circuit working properly, we ask you to interface it to the dSPACE board, through its encoder inputs. These inputs are defined in the dSPACE manual and are labeled as Phi0 1 and Phi90 1. Since the board provides for differential encoder inputs (for noise immunity) and our circuit is single ended, you will need to provide a 1.5 V reference to the complementary inputs /Phi0 1 and /Phi90 1. See Fig. 9.1.1 on pg. 49 in the DS1102 User's Guide and also the information on pg. 57 in this guide. We have included a schematic for this connection in Figure 1. You will need to feed the *debounced* A and B inputs to the dSPACE board so that it counts properly.

The encoder inputs A and B are both referenced to digital ground. Be sure never to apply voltages outside the range of 0 to 5 volts to the encoder inputs as this is the limit for digital inputs. You will need to read the dSPACE documentation to determine how the encoder count is scaled. We ask you to then do two experiments with the dSPACE system. First, use Cockpit to display the encoder count on a numerical readout. Also display this count as a time waveform in Trace. Consult the dSPACE documentation for more details. Secondly, we want you to use the encoder to set a variable gain. Specifically, set up a block diagram which reads an A/D channel at

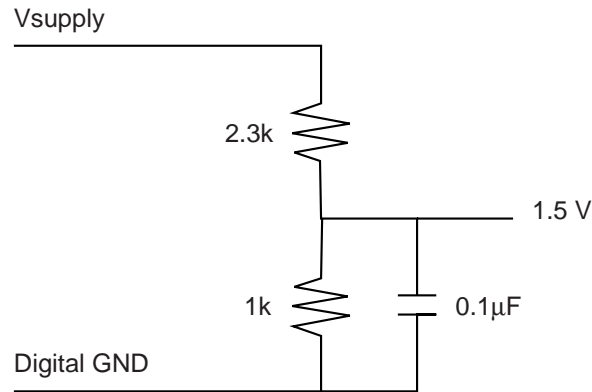


Figure 1: Circuit for generating a 1.5 V reference for the complementary inputs ϕ_0 and ϕ_{90} .

10 kHz, multiplies the signal by a gain variable from 0.1 to 10 and outputs this scaled signal on a D/A channel. Use the encoder as an input which allows you to vary the gain within the above range. (You need to think about how to implement this in the block diagram.) Show us data taken on the scope which indicates that the system operates as specified above.

3 Questions

1. Draw a timing diagram for relevant signals as the encoder goes up through states 00, 01, 11, 10, 00, and down through states 00, 10, 11, 01, 00. Be sure to show relevant waveforms in the field of OR gates which does the step detection. Record a few relevant operational waveforms on the scope. Save this data to disk, and then print it. In your report, compare with what you would predict.
2. What determines the width of the pulses on the UP and DN inputs of the 'LS192 counter? Record a representative pulse using the scope, and save this data to disk and print it. You will need to use sweep speeds on the order of 10 nanoseconds per division. At this rate, the lab scopes operate in a repetitive mode where successive sweeps are required to fill out the waveform. To see this process, go into the Trigger menu, and set up the scope to operate in the normal trigger mode, where waveform edges are required in order to initiate a sweep. You will also need to properly set the trigger level. Rotate the encoder and watch the process of filling in the waveforms on successive pulses. What determines the width of these pulses? How could you change their duration? To roughly 40 nsec? To 1 msec? Sketch circuits which could accomplish this. Also explore looking at these pulses on a 1msec/div time scale with the scope set in peak detect mode, and in sample mode (under the Acquire menu). Note the differences. You have to be careful not to miss such short events with a digital scope. For this a peak detect mode is essential.

- Now put the probes for Ch. 1 and Ch. 2 on the encoder A and B outputs, respectively. Look at these signals on the scope in normal y versus t mode. How do the states of the encoder relate to the lighting of the four LED's? Now put the scope into the x - y mode. Rotate the encoder and describe the resulting display. How does this display relate to the states of the encoder?
- Why are there inverters on the inputs UP and DN on the 'LS192? What happens if these are left out (as I did the first time I designed the circuit)? Try it and see. Why does this happen? (Read the 'LS192 data sheet carefully.)
- The 7-segment driver has open-collector outputs. Think of these as switches which close when the output is ON and are open when the output is OFF. When on, the LED's which make up the display drop about 1.5 volts. Given this information, what is the expected current in the LED?

4 Commercial encoder interface chip

In practice, one would be unlikely to build a circuit such as we have studied so far, since commercial IC's are available for this function. As one example, we have included a data sheet for an HP chip which does this. The chip uses a digital filter as shown in the data sheet. Explain the function of this filter. Sketch some representative waveforms which show the ability of this filter to reject spurious pulses. Such rejection is important, since encoders may likely be used in a high noise environment, e.g., attached to a motor driven by a switching power supply.

5 Phase interpolation

In class, we have discussed using an encoder which provides quadrature sine and cosine outputs in order to sub-resolve the cycles of the encoder. Suppose you are given an encoder which produces the output voltages

$$v_A = 10 \cos k\theta$$

and

$$v_B = 10 \sin k\theta.$$

For present purposes, suppose $k = 1000$, i.e., the encoder has 1000 cycles per revolution. Describe an algorithm to interpolate the angle θ . Suppose further that you use a digital processor to measure these voltages with an uncertainty of ± 0.1 volt. What is the worst angular error that you can make using your algorithm? Why?

6 Other random lab hints

- When using the scope probes to look at signals on the board, use short (30–50 mm) lengths of wire to connect both the ground lead and the scope probe to the board.

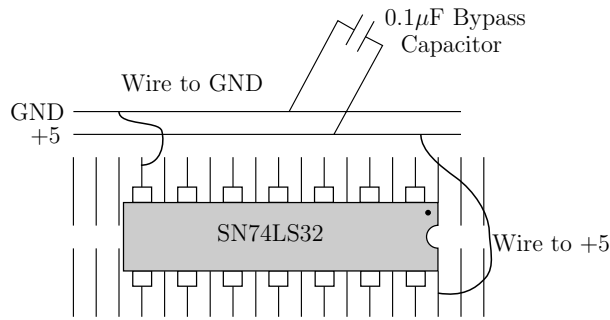


Figure 2: Connecting power supplies to TTL chips.

Do not connect the probe directly to the chips as this will tend to pull the chips out or bend the probe tip. An exception to the policy is the case where you need to look at very fast signals with high fidelity. Then you will need to minimize the lead length. See a staff member for hints on how to do this.

2. Take your time when breadboarding, and double-check connections before powering up. This may save some dead chips and/or extended debugging sessions. Watch out for blowing up LEDs, which happens if you inadvertently short-circuit the current limiting resistor from the 5 V supply to the LED.
3. Debug the circuit in sections! Do not interconnect more sections until each is working individually. For example, first test the encoder and debounce circuit alone. Then add the 'LS138 decoder, etc.

7 Lab Hints

- These notes make suggestions on practical matters associated with lab 3.
- The lab requires only 5V and GND. Please be sure not to use the $\pm 15V$ supplies as these voltages destroy logic chips. Check this carefully before powering up your circuit.
- When you wire logic, connect +5V and GND to power busses which are adjacent. You also need to include $0.1\mu F$ decoupling capacitors to provide a local low impedance to the chip power lines. This prevents spikes on the supply lines due to rapidly varying currents. Refer to Figure 2 for a schematic. Note that the notch and/or dot indicates pin 1 of the IC. Caution ! Wiring the supplies backwards will destroy the device.
- Connect power and ground to the chip and put in the bypass capacitors before you place any other wiring. This helps prevent forgetting these connections.
- When you cut wires, please be careful that pieces do not fall into the keyboard, computer etc. Try to have pieces of wire land on the floor.

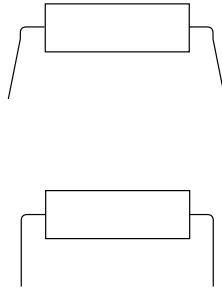


Figure 3: Bending the flared pins of an IC.

- I.C.'s come with slightly flared pins as shown in Figure 3. This is so that they can be readily handled by automatic insertion equipment. Before inserting them into the protoboard, gently bend the pins in with your fingers.
- When removing I.C.'s do not pull them out with your fingers. This will bend the pins, and perhaps impale them in your tender digits (fingers, not bits)! Use an IC puller, or a small screwdriver to slowly pry them out.
- Wire up small sections of the board and test each section as you go. Do not wire it all up and pray that it works! While tempting, this will likely result in a system with multiple problems. Rather, first figure out how each part works and sketch expected waveforms. Then, wire up this section and test. For instance, the encoder can be operated in isolation. Test it and be sure you understand what you are seeing. Then proceed to the next section etc.

