Lab #6

Mechatronic Systems Design
MECH 498 - 005

10/20/99

Subject:

Digital Circuitry: Encoder Interface

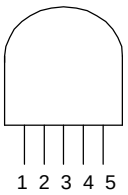
Objectives:

1) Understand the function of a quadrature optical encoder.
2) Understand the internal digital circuitry of an encoder interface chip (counter, latch, decoder, bus).
3) Use a micro-controller, encoder, and encoder interface chip to determine the position and velocity of a motor.

	Possible Points	Grade
1	25	
2	25	
3	50	
Total	100	

Section 1: Encoder

1) View the output of the encoder on the scope (both channel A and channel B). Set the 4-6.5 Volt power supply to 5 volts and use it to power the encoder. Use the 0-30 Volt power supply to power the motor.



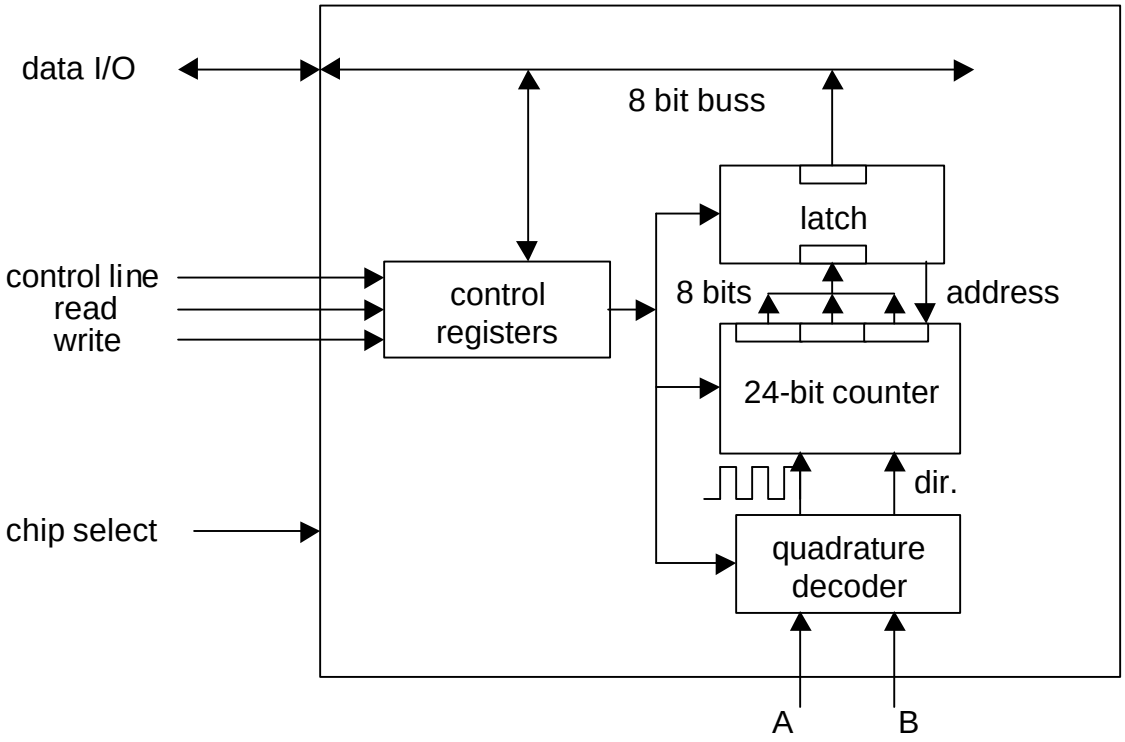
[1] GND
[2] NC
[3] Channel A
[4] +Vcc
[5] Channel B

Supply Voltage, Vcc……-.5 to 7 V
Output Voltage, Vo……-.5 to Vcc
Output Current per Channel…-1.0 mA to 5 mA

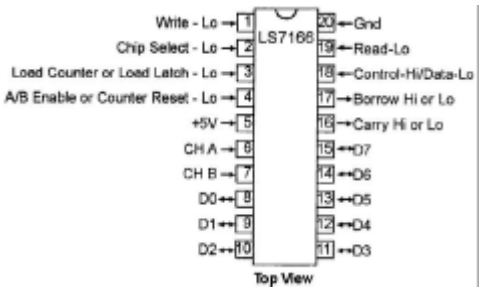
2) Rotate the motor at various speeds and in both directions.
3) Sketch the timing diagram showing both channels for positive and negative motor rotation.
4) Determine the speed of the motor by measuring the frequency of the quadrature wave. Take approximately 5 data points and create a function that relates output frequency to motor RPM.

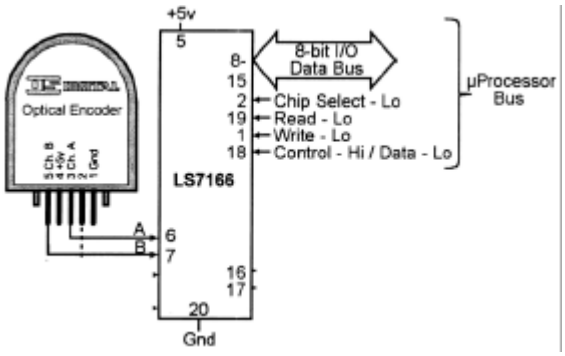
Section 2: Encoder Interface Chip

1) Discuss the function of the encoder interface chip within your group. Make sure you understand each component (I know this sounds corny, but I want you to understand what the chip is doing)



2) Discuss the pin out. Check this with the lab set-up.





3) Connect the encoder to the encoder interface chip. Use the power supply and ground from the BSII to power the encoder.

Section 3: Reading the Encoder with the BSII

1) Run the encoder-reading program provided.
2) Move the motor shaft with your hand (in both directions) and observe the output of the program. The program displays the decimal numbers corresponding to the 3 bytes of the 24-bit counter. The program also displays a decimal number that is constructed from the LSByte and the middle byte of the 24-bit counter.
3) Alter the program to display motor position in degrees, and motor velocity in RPM.
4) Write a lab report that explains what you have done.

Program:

```
CS          con   15              ' Chip select pin; 0 = active
READ_PIN  con   14              ' Pin for reading: 0 = reading
WRITE_PIN con   13              ' Pin for writing: 0 = writing
CNTRL     con   12              ' Pin for Control:

pos0  var   byte                'low byte of 24-bit counter
pos1  var   byte                'middle byte of 24-bit counter
pos2  var   byte                'high byte of 24-bit counter
position var word               'word with low and middle byte

output CNTRL
high WRITE_PIN
high READ_PIN

'********** MAIN PROGRAM *************
   low CS
   gosub initilize
again:                                                  ' Main loop.
   gosub read_position                                  ' read data from encoder chip
   pause 500                            ' Wait a half second.
goto again                                ' Endless loop.
'*****************************************

'********** INITIALIZE CHIP *************
initilize:
   high CNTRL
   high READ_PIN
   dirl=%11111111                       'declare as outputs
   outl=%00110101                       'master control register
   gosub execute_write
   debug CLS,"master control register:  ",bin inl,cr
   outl=%01001000                       'Input control register
   debug "input control register:  ",bin inl,cr
   gosub execute_write
   outl=%10000000                       'output control register
   debug "output control register:  ",bin inl,cr
   gosub execute_write
   outl=%11000001                       'quadrature control register
   debug "quadrature control register:  ",bin inl,cr
   gosub execute_write
return
'*****************************************
```

```
'********** READ COUNTER **************
read_position:
   dirl=%11111111                ' latch counter output
   high CNTRL                    ' latch counter output
   low 6                         ' latch counter output
   low 7                         ' latch counter output
   high 0                        ' latch counter output
   high 1                        ' latch counter output
   gosub execute_write           ' latch counter output

   low CNTRL
   dirl=%00000000
      low READ_PIN
   pos0=inl
   high READ_PIN
   debug CLS,"position0: ",dec pos0,"       "          ' Display data.
   low READ_PIN
   pos1=inl
   high READ_PIN
   debug "position1: ",dec pos1,"       "              ' Display data.
   low READ_PIN
   pos2=inl
   high READ_PIN
   debug "position2: ",dec pos2,cr                     ' Display data.
   high CNTRL

   position.lowbyte=pos0
   position.highbyte=pos1
   debug "position:  ",dec position,cr                 ' Display data.

return


execute_write:                                         'pulse write line
   low WRITE_PIN
   high WRITE_PIN
return
```