Jonathan W. Valvano

(4) Question 1. Write a C function that outputs (transmits) one byte using the SCI port.

```
void SCI_OutChar(unsigned char data){
  while((SCISR1&0x80)==0){};  // wait for TDRE to be 1
  SCIDRL = data;              // start output
}
```

(4) Question 2. Give an example C code that has a ***dropout*** error.

```
Result = 1234*(Input/1000);  // multiply by 1.234
```

(4) Question 3. Give the four **inequalities** that must be true in order for the interface to operate properly.

$V_{OH} \geq V_{IH}$	$I_{OH} \geq I_{IH}$
$V_{OL} \leq V_{IL}$	$I_{OL} \geq I_{IL}$

(4) Question 4. Write a C function that outputs (transmits) one byte using the SPI port.

```
void SPI_OutChar(unsigned char data){ unsigned char dummy;
  while((SPISR&0x20)==0){};  // wait for SPTEF to be 1
  SPIDR = data;              // start output
  while((SPISR&0x80)==0){};  // wait for SPIF to be 1
  dummy = SPIDR;             // clear SPIF
}
```

(4) Question 5. Use **`long`** type (signed 32 bit) because we need to store -100000 to +100000.

(4) Question 6. Give a mathematical equation, defining sampling jitter, δt in terms of f_s t_{i-1} and t_i.

$$\delta t_i = (t_i - t_{i-1}) - 1/f_s$$

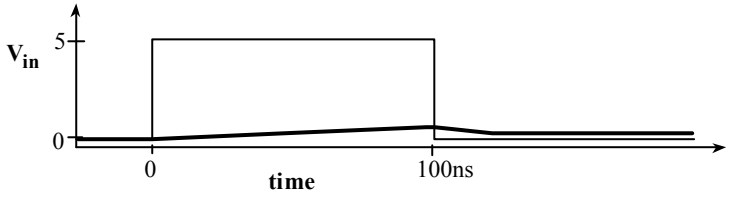
To be real time, we must be able to place an upper bound, **k**, on the time-jitter.

$$-k \leq \delta t_i \leq +k \quad \text{for all } i$$

(8) Question 7. Review your EE411 before going on the interview trail.

R*C= 1μs. From 0 to 100ns, $V_{out} = 5\text{-}5e^{-t/RC} = 5\text{-}5e^{-t/1000ns}$

At 100ns, $V_{out} = 5\text{-}5e^{-100ns/1000ns} = 0.48V$



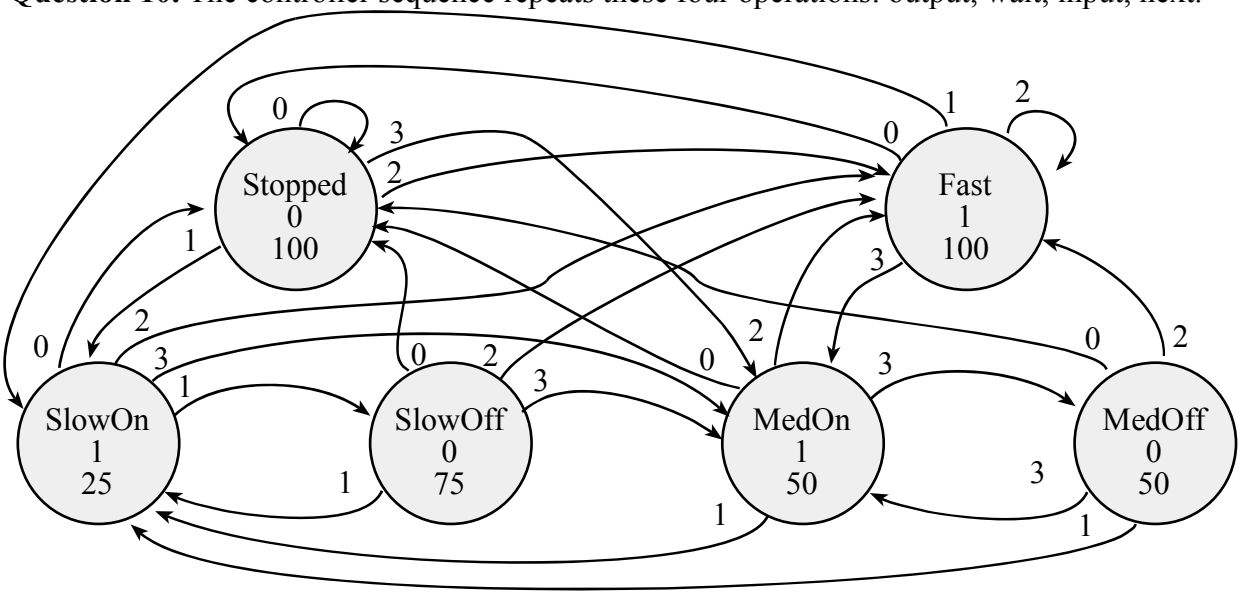
(4) Question 8. This one captures the data in the least amount of time

A) **`if(n<100){BufT[n]= TCNT; BufX[n]=x; n++;}`**

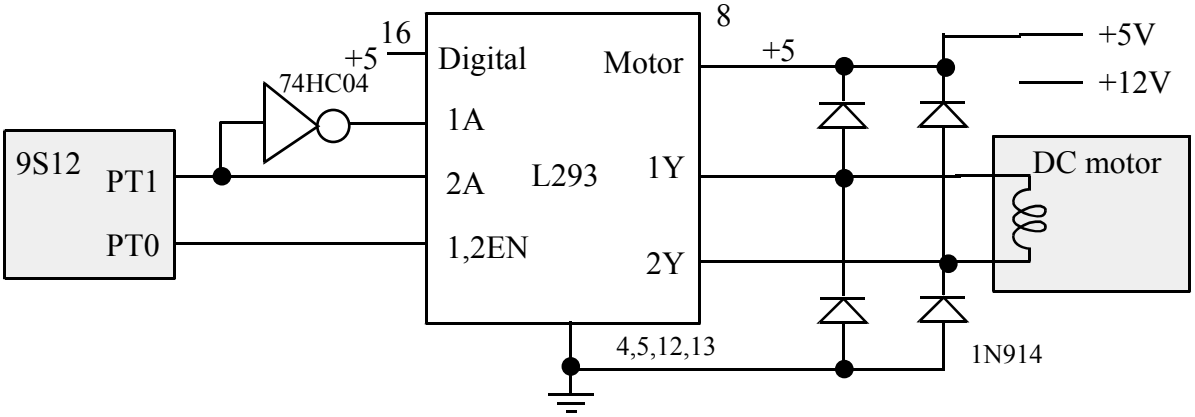
(4) Question 9. Rationalizations an engineer might have for being unethical.

1) Too expensive to do the right thing; lawsuits and jail time are more expensive.
2) Everyone else is doing it; no, everyone else is not doing it.
3) Won't get caught; do the right thing because it is the right thing.
4) Didn't know it was wrong; you are responsible, it is your job to know.
5) My boss told me to do it; hit men for the mob go to jail too.
6) Arrogance, I am right, so I could not have made a mistake; mistakes happen.
7) It all averages out; you can not balance a little bit of evil with a lot of good. Right and wrong are absolute. Search the morality associated with the expression "color of money".

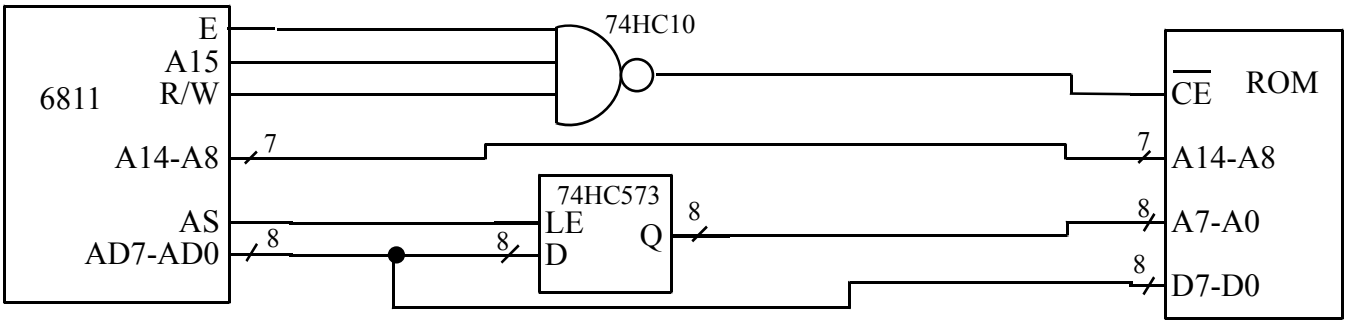
(10) Question 10. The controller sequence repeats these four operations: output, wait, input, next.



(10) Question 11. PT0 controls the On/Off and PT1 controls the direction of rotation.



(15) Question 12. Address decoder at A15 must be 1, activate second half of the cycle E must be 1, activate for read cycle R/W must be 1. Negative logic output. Part a)



Part b) **Read Data Available** = [500+20+120, 1000+10+10] = [640,1020]

(10) Question 13. You will use a Sharp GP2Y0A21YK0F infrared object detector to measure distance. These sensors are $10 on digikey.com

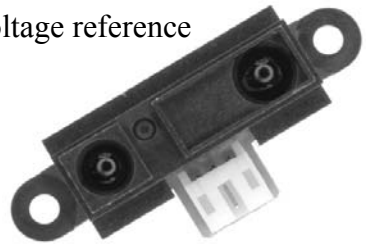
Part a) Map the 0.5 to 3V range of the transducer into the 0 to +5V range of the ADC.

V2 = 2***V1** – 1 0.5→0, 3→5

Part b) **Vref** = 2.50V, **Vg** = 0V

V2 = 2***V1** – 0.4***Vref** offset terms generated with voltage reference

V2 = 2***V1** – 0.4***Vref** – 0.6***Vg** sum of gains equals 1

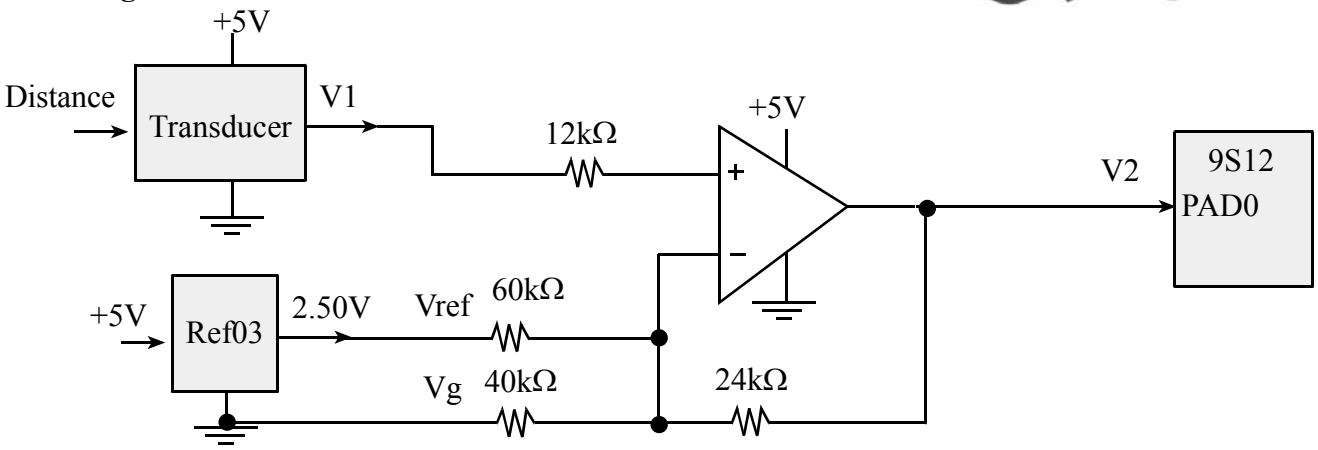


Common multiple of 2, 0.4, 0.6 = 24 kΩ

V2 Gain of 2 is 24 kΩ /12 kΩ

Vref Gain of 0.4 is 24 kΩ /60 kΩ

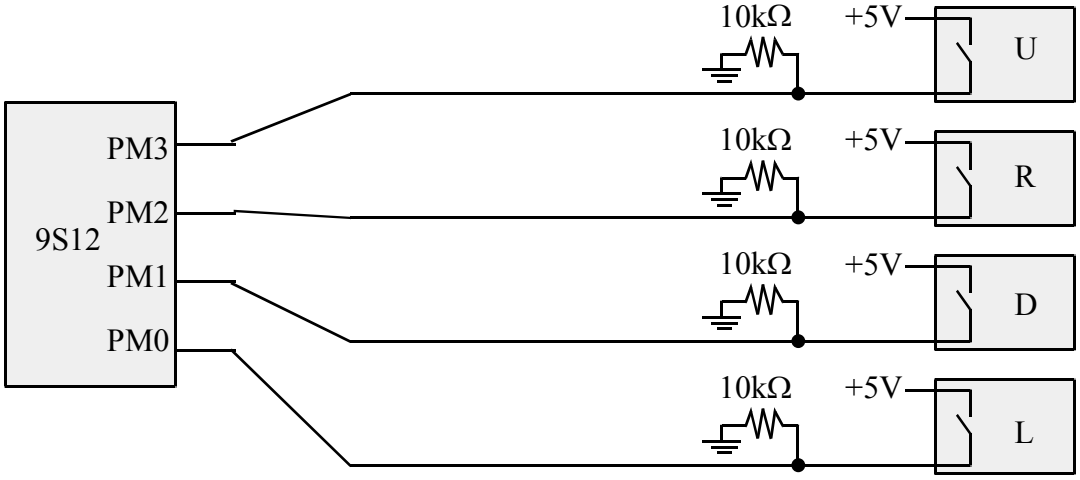
Vg Gain of 0.6 is 24 kΩ /40 kΩ



(15) Question 14. Add the other 7 switch patterns to the table, then sort by **PTM** value. This joystick is $12 (part number JS-5) on http://www.allelectronics.com.

Joystick position	Switch U	Switch R	Switch D	Switch L	**PTM**	**code**
Center	Off	Off	Off	Off	0	0
Left	Off	Off	Off	On	1	7
Down	Off	Off	On	Off	2	5
Down and Left	Off	Off	On	On	3	6
Right	Off	On	Off	Off	4	3
Bad					5	255
Down and Right	Off	On	On	Off	6	4
Bad					7	255
Up	On	Off	Off	Off	8	1
Up and Left	On	Off	Off	On	9	8
Bad					10	255
Bad					11	255
Up and Right	On	On	Off	Off	12	2
Bad					13	255
Bad					14	255
Bad					15	255

Part a) Simple positive logic switch interfaces



Part b) Show the initialization code for the system, enable and arm output compare 7. Initialize Port M.

```
void JoyStick_Init(void){
  DDRM &= ~0x0F;          // PM3-PM0 are inputs from joystick
  Fifo_Init();
  TIOS |= 0x80;           // channel 7 is output compare
  TSCR1 = 0x80;           // activate timer
  TSCR2 = 0x05;           // TCNT at 4MHz/32 = 125 kHz
  TIE |= 0x80;            // arm channel 5
  TC7 = TCNT+50;          // interrupt right away
  asm cli                 // enable
  }
}
```

Part c) Show the output compare interrupt 7 ISR. Use the last entry in table to map **`PTM`** into **`Code`**.

```
const unsigned char CodeTable[16]= {
0,7,5,6,3,255,4,255,1,8,255,255,2,255,255,255};
interrupt 15 void TC7handler(void){ // executes at 10 Hz, 100ms
unsigned char in;         // 0 to 15 input from PTM
unsigned char code;       // 0 to 8 code, specifying joystick position
  TFLG1 = 0x80;           // acknowledge OC7
  TC7 = TC7+12500;        // 100 ms (4,000,000/32/12500= 10 Hz)
  in = PTM&0x0F;          // switch input, 0 to 15
  code = CodeTable[in]; // current position of joystick, 0 to 8
  Fifo_Put(code);         // send to foreground
}
```