CG2271 Real Time Operating Systems

Tutorial 3

You are only allowed to write bare-metal programs for this tutorial.

Question 1

We are all familiar with the binary number system, and in this question we will use the Arduino Uno has a (very expensive) binary to binary-coded-decimal (BCD) converter. The mapping between binary bits b3-b0 and BCD B4-B0 is shown in the table below:

b3	b2	b1	b0	B4	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	1	0	0
1	1	1	1	1	0	1	0	1

a. Design a circuit for the Arduino Uno with four switches (not push-buttons) connected to digital pins 0, 1, 2 and 3, and five LEDs connected to digital pins 5, 6, 9, 10 and 11. In addition there is a push-button switch connected to digital pin 4. You can use Fritzing (http://fritzing.org) to draw your circuits, in which case use the push-button for the switch at pin 4, and the toggle switches for pins 0-3.

b. Write a program that, when the push-button is depressed, causes the Uno to read in the binary patterns on the four switches, and outputs the BCD equivalent on the LEDs. There is no need to do debouncing for any of the switches.

Question 2

a. Modify your circuit from Q1 by adding a potentiometer to analog pin 1.

b. Add in code for your program in Q1 to read in the potentiometer, using interrupt driven ADC. Store the read value in a variable called “adc_in”. The potentiometer is read every time the push-button at pin 4 is pressed.

The variable adc_in will be used in the next tutorial to control the brightness of the LEDs.

Question 3

Modify your program in Q2 so that the program reads the potentiometer at analog input 1 and the switches at pins 0 to 3 every 5 ms instead of each time the push-button is pressed.

Question 4

Supposed we are given a system where the PWM encoding is to be read from/written to a port with the following specification:

Port Data Rate:	512 bps
PWM Encoding Lenth:	16 bits

a. What is the length of the PWM cycle, in milliseconds?
b. Given that the analog voltage range is ±5v, what is the resolution of the A/D and D/A converters, in volts per bit?
c. Suggest the encoding for -5v, 0v and 5v. Give the encoding for -3.5v, 2.2v, 4.5v. Is it possible to get exact encodings for these voltages? Why or why not?