TUTORIAL 2 - Digital Input/Output

MTRN3500 Computing Applications in Mechatronic Systems

Dr Jay Katupitiya
The University of New South Wales

1 Introduction

- PC104 Addresses. We use only four ports for these exercises.

  1. Digital Input Port:These ports are located at address 0x303 and 0x30B. The ports are connected to two switch banks with 8 switches in each bank. By changing the switches' on/off positions, you can change the data read by the computer. A bit corresponding to a certain switch that is turned on will be read in as '1' and those corresponding to switches that are turned off will be read as '0's.
  2. Digital Output Port: The two digital output ports are located at addresses 0x303 and 0x30B. The ports are connected to two sets of 8 LEDs. When you write data to these ports, the lights corresponding to bits with the value '1' will light up and those corresponding to the value '0' will remain off.

2 Problems

1. Write a program to make the LEDs alternate between the following two bit patterns. Ensure a time gap of 0.5 seconds between two consecutive changes. The two bit patterns are

   - 1010101010101010
   - 0101010101010101

   The program should terminate itself when bit-0 of digital input low byte is turned off. Note: For your program to start ensure this bit is set to 1.

2. Write a program that make a lit up LED to appear at LED 1 and travels up to LED 16 with a time gap of 0.5 seconds between each hop. This is called an 'LED walk' program. Terminate the program after completing a full walk to LED 16 and then lighting up all 16 LEDs for one second. Turn off all LEDs before exiting.

3. Write a program that make a lit up LED to appear at LED 1 and travels up to LED 16 and back to LED 1 with a time gap of 0.5 seconds between each hop. Terminate the program when the lit up LED has returned to LED 1.

4. Write a program that lights up LEDs according to the following bit pattern.

   - 0111011101110111

   Make provision for a user to chose a bit location by entering a number between 1 and 16 and then toggle the chosen bit. Make provision to terminate the program if the number entered is 0.

5. Write a program that reads in a 16-bit binary bit pattern you type and lights up the 16 LEDs accordingly. If the binary bit pattern you typed is 0000000000000000, respond to that bit pattern normally and then terminate the program.

6. Write a program that reads 8, 8-bit binary patterns you have typed in a file and display each bit pattern on a bank of 8 LEDs 0.5 seconds apart. Show your demonstrator how the lighting pattern changes when you change the data file. Typical file read code is given below:

```
\\Read file
#include <iostream>
#include <fstream>
...
using namespace std;

int main()
{
    ifstream is("yourfile.dat");
    unsigned int N[8];
    ...
    for(int i; i< 8, i++)
        is >> N[i];
    is.close();
```

```
        ...
    }
```

This code reads the binary bit patterns as integers into the array `N`.

7. Write a program that repeatedly displays a 16-bit pattern you would enter and its complement half a second apart on a bank of 16 LEDs. Terminate the program when the most significant bit of digital input high byte is set to 1 by turning the corresponding switch on. Note: To enable your program to start, set this bit to 0, i.e. turn the switch off.

8. Write a program to continuously display on the screen the 16 switch status as a binary bit pattern until the difference between the low byte and high byte switch patterns is 0x0F. The program should then exit.

9. For a random switch pattern, light up the LEDs of digital outputs according to the following bit pattern:

   - 0111100010100110

   Detect changes in the switches and toggle the corresponding LED on the digital output lines. When switches are changed to turn off all LEDs the program should terminate.