Lab 2 - Keypad and Display Interface

Corrected version of September 29 1997.

Introduction

In this lab you will compile and run a C program to read the keypad and write to the 8-character ASCII display on the laboratory computer. You will use these two functions in a demonstration program.

The Laboratory Computer

Each station in the APSC 380 lab is equipped with a PC as well as a microcomputer based on the Motorola 68000 microprocessor (the "laboratory computer").

The lab computer includes a 'bus' which allows various plug-in printed circuit interface cards to be connected to the microprocessor. These interface cards provide input/output facilities for the laboratory apparatus. In this first lab you will use the keyboard/display card. This card will be used as a control panel and status display in the experiments.

The laboratory computers run a very simple operating system that does not support a compiler. However, it does accept commands to read and alter the contents of the lab computer's memory. Your software will run on the PC and will send commands to the lab computer over a serial interface. These commands will read and write specific memory locations which in turn will control the lab hardware.

The remainder of the laboratory computer system consists of a cabinet and a power supply.

Compiling your Program

Create a project file for your lab. Give this file the extension `.PRJ` (e.g. `LAB2.PRJ`). On the first line put the name of your C file (e.g. `LAB2.C`) and on the second line put the file name `IOLIB.LIB`. Save this file.

The project file will allow your program to use the `speek()` and `spoke()` routines described below. These two functions allow the PC to read and write the lab computer's memory by sending commands over the serial port.

Reading and Writing I/O Ports

The peripherals on the lab computer are controlled by reading and writing values from/to specific locations in the lab computer's memory. These special memory locations are often called "ports" or "registers".

The keypad/display card has a control port and a data port. The addresses (locations in memory) of the control and data ports are:

port	address
data port	208
control port	209

The following two functions are available to your Turbo C program to access the lab computer's memory:

```
int speek( int address ) ;
int spoke( int address, int value ) ;
```

The `speek()` function allows you to obtain the value of the memory at a given address and the the `spoke()` function allows you to set the value of a memory location. To use these functions you should include the line `#include <iolib.h>` at the start of your program.

Display

The display has 8 positions. Each position can display an upper-case letter or digit. To display a character on the display, you must write the ASCII value of the character to be displayed plus 128 to the data port. You must then write the following sequence of three values to the control port:

- the segment position (counting from the right starting with zero)
- the segment position plus 8
- the segment position

For example, to display an ‘A’ in the leftmost position, you would write the following sequence of values:

address	value
208	193
209	7
209	15
209	7

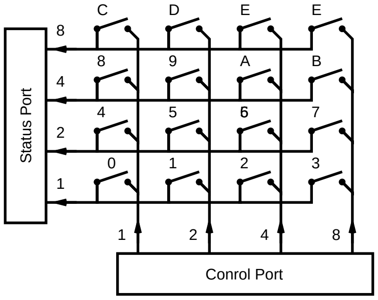
Display Handler Function

Write a function that displays the contents of an 8-character array on the lab computer’s LED display. The definition of this function will be as follows:

```
void display( char string[] )
{
   ...
}
```

Keypad

The diagram below shows a schematic diagram of the keypad:



The keypad consists of 16 switches arranged in a square matrix. Each pushbutton controls a switch which can make a connection between a wire running horizontally and a wire runing vertically.

The vertical wires are driven by the least significant four bits of the data port output while the horizontal wires are connected to the least significant four bits of the data port input. By writing and reading values to/from the data port it is possible to determine which key is currently pressed.

A data port bit will be read as a ’1’ only if the switch on that row is pressed *and* a ’1’ has been written to the data port bit for that switch’s column.

Keypad Handler Function

Write a function `int rdkbd (void)` that waits until a key is pressed and returns a number corresponding to the number pressed (e.g. if the ’3’ button is pressed the function should return the value 3).

Demonstration Program

Using the keypad and display handler functions, write a C program that lets the user enter a hexadecimal number from the keypad and displays it on the display unit. The program should do the following:

- Display `APSC 380` when started.
- When a key (0 to F) is pressed, scroll the display one position to the left and display the new number on the rightmost position.

Hints

Make sure you understand the description of the hardware interface in detail before you begin to write and debug your program. You may want to write one or more short test programs while developing your program to verify that the hardware indeed operates as you expect (“bottom-up implementation”).

Demonstration

Demonstrate the proper operation of the keypad/display demonstration program to the lab demonstrator.

Write-up

Submit a program listing for the keypad/display demonstration program. It should include comments as described in Lab 1.

Keep the keypad handler and display routines for use in the next lab.