To run any of the experiments in this lab, you must disable the Mouse on the Experiment computer. (To do this run c:\nomouse and reboot, to re-enable run c:\yesmouse and reboot. Don’t forget to detach/reattach the mouse).

Overview and Introduction

This lab constitutes a basic introduction to serial communication and use of a 16550A UART (the most common type of Universal Asynchronous Receiver/Transmitter in use today). The hardware and communication methodology are introduced, but the emphasis is on the programming interface. To accomplish these ends you will create a simply terminal program that can be used to communicate between 2 PCs using a NULL modem cable (which is simply a specially configured serial cable). The program will simply poll the UART to determine when data is ready to be received, and then retrieve the data and display it on screen.

For ample information on programming a UART go to www.beyondlogic.org. You will need to pay the most attention to the programming interface of the 16550A UART including the I/O port map and the function of each register. Pay special attention to the fact that which registers are available depend on the status of the DLAB bit (this is covered at the web site).

Pre - Lab

(Due at the beginning of the lab period.)

1) Draw a diagram showing all the connections between 2 computers when using a NULL modem cable (with full handshaking). For every signal, describe its function. For every crossed signal (signals not tied to the same signal on the other end) describe why the signal is crossed. You may use an appropriate reference to determine the connections, but your description must demonstrate understanding of the connections made. Under what circumstances would you be able to use a NULL-modem cable with no handshaking?

2) Write a program that:
   - Finds the I/O port address of the COM1 serial port. (See the attached code comadd.c)
   - Initializes the UART to
     1. Use a BAUD rate of 2400.
     2. Use No parity, one Stop Bit, and a Word length of 8.
     3. Set up the FIFO to a depth of 14 words (Bytes), turn off DMA, clear both the transmit and receive FIFOs, and enable the FIFOs.
     4. Enable DTR, RTS, and Output 2.
3) Write a program that:
   • Finds the I/O port address of the COM1 serial port.
   • Initializes the UART to
     1. Use a BAUD rate of 4800.
     2. Use Even Parity (do not use "sticky" parity), 1 stop bit, and a word length of 7 bits.
     3. Set up the FIFO to a depth of 14 Words, turn off DMA, clear both the transmit and receive FIFOs, and enable the FIFOs.
     4. Enable DTR, RTS, and Output2.

**UART Configuration Help**
The config. help is broken down according correspond to the step in the pre-lab directions. Therefore, hint 1 corresponds to step 1 of question 2 or 3.

1. To set the BAUD rate you will have access the Divisor Latch LSB (DLL) and Divisor Latch MSB (DLM) registers. These registers are only accessible with DLAB = 1.
2. You need to configure the LCR register.
3. You need to configure the FCR register.
4. You need to configure the MCR register.

Be sure and clear the FIFOs at the end of the UART configuration.

**Lab Procedure**

Using the programs from Pre-Lab questions 2 & 3 as a starting point to write a simple terminal program that will communicate with another terminal program on a PC connected with a NULL modem cable. NULL modem cables will be provided. This program will be a simple polling based program. In effect, this means the program will consist of a loop that continual checks the UART FIFO and the Keyboard for input, and then processes the appropriate input. The program should exit when the user hits CTRL-A. Pseudo Code for this program is:

```c
initialize_UART();

while (key_pressed_not_ALT-A) {
    if kb_pressed {
        getch();
        send_ch_to_UART();
    }
    if (FIFO_Data_Ready) {
        read_FIFO();
        printf("%c", read_in_char);
    }
}
```

You will have to #include the following headers in your program

```c
<dos.h>
<conio.h>
<stdio.h>
```
The following function calls will be of interest (use the Turbo C online help):
   kbhit() , getch()
   You will also need to re-use your inPortB() functions from previous labs, or use the I/O functions
to provide in <DOS.H>

Lab report

The report is due at the start of the following lab period. You will have to demonstrate the programs of the
Lab Procedure using both configurations (the 2400 baud, no-parity, and 4800 baud, even-parity
configurations.).