ECE 3710 LAB III

Due Date: Week of October 19th 2015 (20 Points)

Objectives

The student should write, download, and test a program that uses timers, UART, and the RS-232 module to transmit serial data between the microcontroller and the PC.

Overview

For this lab we were going to create a Magic 8-Ball® program but Mattel threatened to sue, so we'll implement a Spiteful 8-Ball program, instead. Upon being queried by the user (either a button press or a character sent on the serial port) your microcontroller-based 8-ball program will transmit an answer via the serial port.

Preparation

1. Read over UART and RS-232 in your book (V1.Ch8.2 and V2.Ch7.2).
2. Read over the UART setup in the TM4C123GH6PM manual.
3. Read through the requirements of the lab and create a diagram/pseudo-code/how you will implement your 8- Ball
4. Check out a Serial cable from the ECE store.

Hardware

A UART to RS-232 module is available in the lab pack. They are also available for purchase from the ECE store. You will have to connect this module to the UART pins on your board.

Requirements

Your program should accomplish the following:

1. Upon Reset the micro-controller should be waiting for input either the computer or a button.
2. The serial port needs to be configured for 9600 baud with 8 data bits and one stop bit. Immediately after reset, a carriage returns (ASCII 0x0D) and a line feed (ASCII 0x0A) need to be sent via the serial port.
3. After a button is pressed or any character is received on the serial port, exactly one of the following messages should appear on the console window:
   - Nope
   - You are doomed
   - Concentrate you fool
   - What a rubbish question
   - Only in your dreams

- Yes now leave me alone
- Heh you wish
- Oh yeah that will happen
- Stop bothering me
- Not if you were the last person on earth

Each of the above messages needs to be terminated by a carriage return and a line feed and selected for transmission at random.

Connecting to PC

Attach the 9 pin D-Sub to the serial port on the back of the Lab PC, and the other end to the RS232 module.

To indicate that you want a question answered, and to receive a response from the program, you'll need to setup a terminal emulator to communicate with the microcontroller over a serial interface. To do this, connect the RS-232 module on your evaluation board and to the serial port on your computer. Start a terminal emulator, such as HyperTerminal or Putty, on your computer and configure it for 9600 baud, 8 data bits, one stop bit, and no parity bit.

Documentation

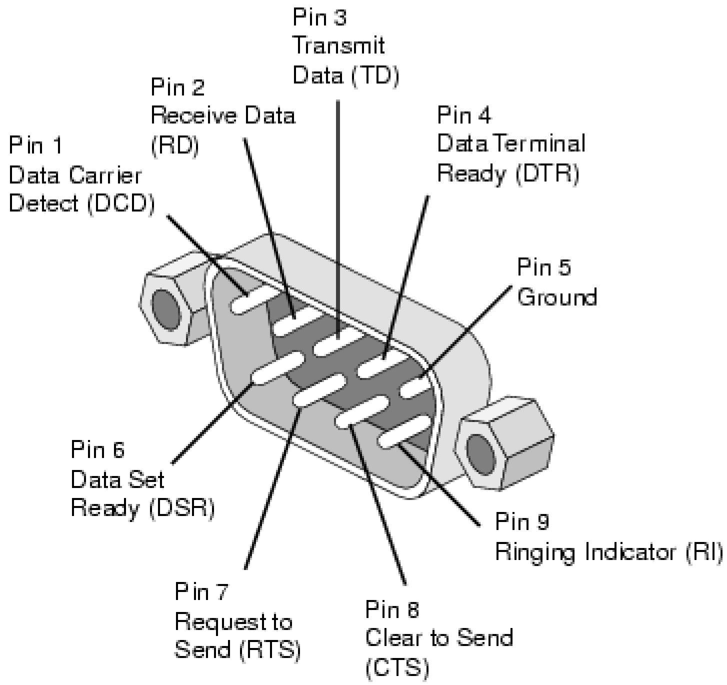
Your lab report should demonstrate that your program works as stated in the requirements by including:

1. Output from the logic analyzer that shows:
    - That the carriage return and line feed are transmitted immediately after the program starts
    - The duration of the first start bit is 1/9600 seconds (104 µs)
    - That 8/9600 seconds (833 µs) elapse between the end of the start bit and the beginning of the stop bit.
    - Include screenshots of both the event table, and the Bus displaying one of the messages.
2. A screenshot of the terminal emulator that shows you sending a character to the board and receiving a response. The screen shot should contain multiple queries/responses.
3. Remember to put give enough explanations with your screen shots convey what you learned in the lab.
4. A simple circuit diagram of the MAX-232 chip interfacing with your board. Use a circuit design program, such as Eagle, Orcad, or LTSpice – Microsoft Paint is not an acceptable tool. Indicate what pin the chip is connected on the micro-controller and what pin it is connected to in the RS-232 Sub D connector (Use your book as a resource).

Hints and Tips

- The GPIO pins have multiple functions (UART, SPI, ADC, Analog Comparator, timer outputs, PWM, etc.), and the pin can't have each peripheral claim it at once. The user must select which peripheral can claim the GPIO pin. This configuration is done using the GPIOAFSEL and GPIOPCTL registers. Read Valvano V1.Ch4.1.3 for more information on pin muxing.

- The logic analyzer is a useful tool to assist with debugging. Refer to Section “Setting Up a Serial or Parallel Bus” (p. 67) of the oscilloscope user manual to help you do this.
- In order to use the Bus decoder function of the logic analyzer, you must first send the message through the RS 232 module. Use the pinout below for the DB9 connector to connect the logic analyzer to the output of the RS232 module.
- Be careful about which cable you use. If you get a null modem cable, it will switch the RX and TX lines. Correct this by connecting two null modem cables together.


Pin 3
Transmit Data (TD)
Pin 2
Receive Data (RD)
Pin 4
Data Terminal Ready (DTR)
Pin 1
Data Carrier Detect (DCD)
Pin 5
Ground
Pin 6
Data Set Ready (DSR)
Pin 9
Ringing Indicator (RI)
Pin 7
Request to Send (RTS)
Pin 8
Clear to Send (CTS)