ECGR 4101/5101, Fall 2006: Lab 5 (Version 1.1)

Serial I/O

Learning Objectives

This lab will introduce you to using polling to perform serial I/O available on the 30626P-SKP board, and new C programming concepts.

General Information

The general steps for this lab are:

1. Generate a new project. Name your new project Lab5.
2. Modify the main.c file and include the appropriate files. Include commenting along the way.
3. Program the lab. Don't forget the necessary include files to get the correct functionality.
4. Compile the code into an .x30 file, and load onto the board.
5. Test the program and repeat sets 4 and 5 until the program works as required.
6. Take measurements of the serial connection between your boards – include a screen shot of a transmission. You should capture an entire transmission (all bits of the transmission).
7. Write your lab report.
8. Demonstrate for a TA and turn in your report.

Prelab Activity

You may use the PCs in Woodward 203 or your own PC to do this lab experiment. The machines in Woodward 203 already have the software tools loaded.

1. What registers are used to set the baud rate on our processor?
2. What value will you write to these registers in order to get a baud rate of 115200 bps, what is the error?
3. What register will hold the received data?
4. Write the pseudo code for this lab (Homework 7)

Laboratory Assignments

In this lab you will be performing serial communications with polling. This lab will use the on-board UART to communicate between two boards. The LED's will be used for signaling and the LCD can be used to display debugging information. This lab must be demonstrated to the TA.

You will be expected to listen for several different commands. All valid commands will be transmitted in uppercase. The commands that are valid are R, Y, and G toggle the respective red, yellow, and green LEDs. All other characters should be considered invalid and ignored.

1. The program should always be polling for a new character.

2. If a character is received it should be checked for validity.
3. If the command is valid the program should act accordingly**, and return an acknowledgement character "A" (NOT "a")**.
4. Once the command has been processed the program should poll for the next command.

Steps

1. Follow the steps given in lab 2 for generating a new project.
2. Create the main.c file and include the appropriate files.
3. Build your program slowly, testing along the way. Perform compiles and solve each requirement one at a time.
4. Continue to build and test the program until all of the requirements have been met. Did we mention you should write your comments as you progress, not at the end?
5. If you run into problems, use the break point functionality of KD30 to step through the code until you find the problem.
6. Once all the requirements have been met, ensure that everything works.
7. Take measurements of the serial connection between your boards – include a screen shot of a transmission. You should capture an entire transmission (all bits of the transmission). Include the screen shot in your lab report. Use a logic analyzer or an oscilloscope. See the lab TA to use the logic analyzer or oscilloscope.
8. Finish lab write-up and demonstrate for a TA.

Requirements

Req. 1 – The code generated is written in C for the SKP16C62P.
Req. 2 – The code is well commented and easy to follow
Req. 3 – The serial communications should operate at 2400 baud, even parity, 8 data bits, one stop bit.
Req. 4 – The two student boards will be connected via the UART0 transmit, receive, and ground pins. The transmit pin on one board will be connected to the receive pin on the other board.
Req. 5 – The same code should be run on both boards, with the same functionality.
Req. 6 – Both boards will wait for a button press on its own board or a byte to be received from the other attached board.
Req. 7 – If "R" **(NOT "r")** is received from the other board, then the Red LED is inverted on the board.
Req. 8 – If "Y" **(NOT "y")** is received from the other board, then the Yellow LED is inverted on the board.
Req. 9 – If "G" **(NOT "g")** is received from the other board, then the Green LED is inverted on the board.
Req. 10 – If SW1 is pressed, then the Red LED is inverted on the board and the character "R" **(NOT "r")** is sent to the other board.
Req. 11 – If SW2 is pressed, then the Yellow LED is inverted on the board and the character "Y" **(NOT "y")** is sent to the other board.
Req. 12 – If SW3 is pressed, then the Green LED is inverted on the board and the character "G" **(NOT "g")** is sent to the other board.
Req. 13 – Once a command is processed the program returns to polling.
Req. 14 – The color of the LEDs on both boards must always match.

Req. 15: - If the command received by a board is valid the program should return an acknowledgement character "A" (NOT "a"). If the command received is not valid the program should return a not-acknowledgement character "N" (NOT "n").

Lab Report

Turn in a hard copy of the code you wrote and a printout of the map file. Also include in your lab report observations and procedure like the following:

The general learning objectives of this lab were . . .

The general steps needed to complete this lab were . . .

Some detailed steps to complete this lab were . . . .

1. *Step one*
2. *Step two*
3. *. . . .*

Code generated for this lab…

Some important observations while completing/testing this lab were . . .

In this lab we learned . . . .

Include the screen shot in your lab report. Use a logic analyzer (5 points extra credit) or an oscilloscope.