ECGR 4101/5101, Fall 2006: Lab 5

Serial I/O with Queues and Interrupts

Learning Objectives

This lab will introduce you to using polling to perform serial I/O available on the 30626P-SKP board, queues, interrupts, 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 steps 3 and 4 until the program works as required.
6. Write your lab report.
7. 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 is the maximum size of the queue you need?
2. Where should you handle reading the receive queue and why?
3. What register will hold the transmitted byte?

Laboratory Assignments

In this lab you will be performing serial communications with queuing. This lab will use the on-board UART to communicate between your board and a PC. 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 command strings. All valid commands will be transmitted in mixed case. The commands that are valid are Red!, Yellow!, and Green! and toggle the respective red, yellow, and green LEDs. All other strings should be considered invalid and ignored.

1. The program should rely on interrupts.
2. If a string is received it should be checked for validity.
3. If the command string is valid the program should act accordingly, and return an acknowledgement. Invalid strings should be rejected.

Steps

1. Follow the steps given in lab 1 and 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. 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 300 baud, even parity, 8 data bits, one stop bit.

Req. 4 – The student board will be connected to a PC via the UART0 transmit, receive, and ground pins and a MAX232 chip. The transmit pin on board will be connected to the receive pin on the PC.

Req. 5 – The board will use interrupts to wait for a character to be sent from the PC. Received characters will be put in a receive character queue.

Req. 6 – The board will not process the queue until either the “!” character is received or the queue is full.

Req. 7 – Processing the queue should not occur in the ISR.

Req. 8 – If “Red!” is received from the PC, then the Red LED is inverted on the board.

Req. 9 – If “Yellow!” is received from the PC, then the Yellow LED is inverted on the board.

Req. 10 – If “Green!” is received from the PC, then the Green LED is inverted on the board.

Req. 11 – If the command received by a board is valid the program should return an acknowledgement of the form “Green ON” or “Green OFF”, depending on the state of the LED and the color change request.

Req. 12 – If the command received is not valid the program should return the not-acknowledgement string “Bad input”.

Req. 13 – Each string sent to the PC should use a queue and interrupts. Each string should end with a CR and LF.

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 . . . .