

# ECGR 3/5/6090, Fall 2003: Lab 4

Introduction to serial I/O of the MSV30262-SKP

## Learning Objectives

This lab will introduce you to using polling to perform serial I/O available on the MSV30262-SKP board, and new C programming concepts.

## General Information

The general steps for this lab are:

1. Create a new folder for lab 4. Copy your files from lab3 into the new folder.
2. Generate a new project using the files you just copied. Name your new project Lab4.
3. Open and edit your main.c file to perform the lab functions.
4. Program the lab. Don't forget the necessary include files to get the correct functionality.
5. Compile the code into an .x30 file, and load onto the board.
6. Test the program and repeat sets 4, 5, and 6 until the program works as required.
7. Write your lab report.
8. Demonstrate for a TA and turn in your report.

## Prelab Activity

You may use the PCs in Smith 347 or your own PC to do this lab experiment. The machines in Smith 347 already have the software tools loaded.

Inspect the files supplied for lab 3.

1. What registers are used to set the baud rate on the 30262?
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.

## Laboratory Assignments

In this lab you will be performing serial communications with polling. This lab will use the on-board UART to communicate with a PC via Hyperteminal. The LED's will be used for signaling and the LCD can be used to display debugging information. This lab will be tested by the TA's using test software loaded on a development board.

You will be expected to listen for several different commands. All valid commands will be transmitted in uppercase, and should be echoed back to the PC in lowercase. You must check for the validity of the command before processing. If a command is invalid, all LED's should be turned off and a lower case "i" should be transmitted. The other commands that are valid are R, Y, and G toggle the respective red, yellow, and green LEDs. You will also be expected to handle a D to display both partners unity ID on the LCD. This is the only command that requires the LCD to display anything. All other characters should be considered invalid and should trigger the invalid response.

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.
4. If the command is invalid the board should return an “i” to the PC and turn off all LEDs.
5. 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. Modify the main.c file and include the appropriate files. Include commenting along the way.
3. Build your program slowly, testing along the way. Perform compiles and solve each requirement one at a time. Make sure comments are written as you progress.
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 Lab TA.
8. Include your main.c and .map file files in the lab write-up.

## Requirements

- Req. 1 – The code generated is written in C for the MSV30262-SKP
- Req. 2 – The code is well commented and easy to follow
- Req. 3 – The serial communications should operate at 57600 baud
- Req. 4 – The program handle all invalid commands
- Req. 5 – If “R” is received then an “r” should be transmitted and the Red LED inverted
- Req. 6 – If “Y” is received then a “y” should be transmitted and the Yellow LED inverted
- Req. 7 – If “G” is received then a “g” should be transmitted and the Green LED inverted
- Req. 8 – If “D” is received then a “d” should be transmitted and unity ID’s displayed
- Req. 9 – Once a command is processed the program returns to polling

## Lab Report

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 . . . .*