Lab 12 Temperature Control

This laboratory assignment accompanies the book, Embedded Microcomputer Systems: Real Time Interfacing, by Jonathan W. Valvano, published by Brooks-Cole, copyright © 2000.

Goals
- Design a hardware interface to the DS1620 temperature sensor using the SPI module;
- Implement synchronous serial communication using the SPI module;
- Implement a simple bang-bang controller.

Review
- DS1620 data sheets
- Review Lab 6 about the DS1620,
- Valvano Section 3.4.8 about handshaking with the DS1620,
- Valvano Section 7.7 about the SPI port,
- Valvano Section 13.3.1 about bang-bang control systems,
- The chapter on the SPI in the Motorola Reference Manual.

Starter files
- DS1620B.C, DS1620.H, DSTESTB.C

Background

The overall goal of the control system is to maintain the room temperature at a desired value. The DS1620 will be used in "temperature sensor" mode to measure the room temperature. The desired temperature will be specified via numerical input from the PC via the SCI port. The two setpoint temperatures, T_{HIGH} and T_{LOW}, will be entered using the `InFDec()` routine you developed in Lab 1. You will not send these values to the DS1620, but rather use them in the 6812 software algorithm that implements the bang-bang controller. The output of the controller will be an LED, which will simulate the heater used to increase the room temperature. The LED state signifies whether or not heat should be applied to the room. An algorithm for bang-bang can be found in Lab 6 as Figure 6.3.

The SPI module automatically performs the tasks of generating the clock and shifting the data. The only trick in using the SPI for this lab is to perform the 9-bit data transfer by executing two SPI transmissions (it will actually send 16 bits.) The 6812 will be the master (responsible for generating the clock.) For the communications that start with sending the instruction followed by receiving back the data, you will have to change the SPI mode in the middle.

Preparation

Show the required hardware connections. Add an LED to a parallel port output pin. This LED will be used for the software implementation of the bang-bang controller. Label all hardware chips, pin numbers, and resistor values. Develop low-level software routines required to initialize the DS1620 and read the current temperature. The SPI port must be used. You do not need to implement all features of the DS1620, rather just those needed to run the software-based bang-bang controller. The foreground software will implement an interpreter with 2 commands, set T_{HIGH} and T_{LOW}. A periodic interrupt will perform the bang-bang control system in the background.

Procedure

Start with the lowest level routines and test your DS1620.C functions in small pieces. Again write a main program which performs the same low level operation over and over so that you can observe the synchronous serial communication on a dual channel scope. Running the same function over and over is called stabilizing, and it is an important debugging procedure.

Checkout

You should be able to demonstrate your debugging feature that allows you to execute each DS1620 function over and over. Connect the dual channel scope to CLK, DQ and explain the signals generated when running these debugging features. Choose high and low setpoints so that the bang-bang controller can be demonstrated to your TA.

Hints

1) Make sure the wires are securely attached to your board.
2) You can increase the temperature of the DS1620 with your finger, and decrease it with a fan. You could use one of those frozen cubes you put in your cooler, but I suggest you avoid using liquids (e.g., ice) in this lab.