

1 Introduction

In this lab, you will rewrite the LCD Demo program in Lab3 for MicroC/OS. This will involve a redesign of the LCD Demo tasks along with configuring and building a MicroC/OS-based project.

2 Program Requirements

The requirements for this lab are identical to Lab3 with the following changes:

The two tasks, ClockTask() and LCDDemoTask() must be rewritten so they are independent tasks designed for MicroC/OS. In addition, to meet long term accuracy requirements, a third clock task must be added.

Clock Requirements:

Long Term: E-Clock accuracy

Short Term: $\pm 30\text{ms}$

os_cfg.h – This header file controls the OS configuration. Edit this file to the minimal settings required to support this lab.

3 Program Design

Since two asynchronous tasks are writing to a shared resource, the LCD, you must use a mutex to control access. This means you also have to make sure that the two tasks always reset the cursor position before writing to the LCD and holds the mutex key for less than 30ms at a time. Figure 1 shows the suggested data flow diagram. The third task, TimeDispTask() must be added because of the OSMutexPend() required to access the display. Without this task, the clock task would lose long-term accuracy.

You will also need to rewrite the LCD module to use a mutex. LcdInit(), LcdDemoDisp(), and LcdDispTime() should be the only public functions from the LCD module.

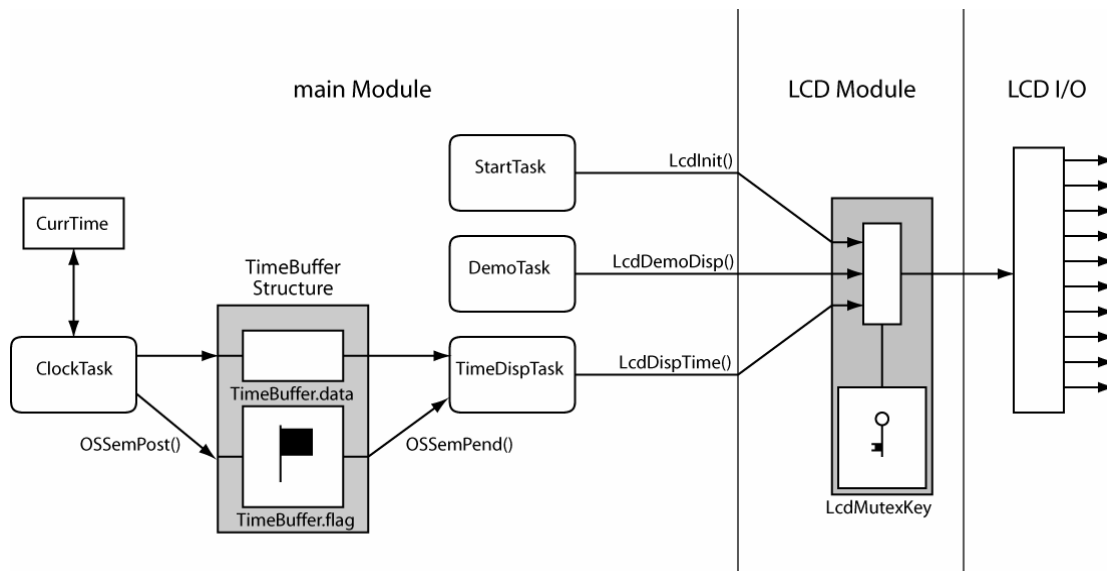


Figure 1 – Lab4 Dataflow Diagram

3.1 Converting Cooperative Task to MicroC/OS Task

In order to convert your tasks to MicroC/OS tasks, consider the following list of changes. Note that this list may not be complete depending on your original design.

- Convert the cooperative task to an independent task by inclosing the task code in an endless loop. You can also remove the state machine that was implemented to meet the timeslice requirements.
- Since the task, which is a function, never exits, your static local variables can now become regular locals. They will always persist because the code never falls through and exits the function.
- The tasks must use the OS timer in place of the OCDelay routines.

3.2 Debug Helpers

Debugging helper bits must be used to show task execution. They should be set up to go low while the task is pending and high while the task is running its task code. Debugging helper signals must be defined as follows:

Task	Debug Bit
StartTask()	PP7
DemoTask()	PP6
ClockTask()	PP5
TimeDispTask	PP4

4 Write-up

The source code for your project must be released by midnight on the due date. The write-up must include the following material:

Introduction

Program Description

Source Code and Header File(s) (hardcopy)

Comments and Conclusion including:

A comparison of Lab 3 and Lab 4 program design

A comparison of Lab 3 and Lab4 ROM and RAM usage

Due Dates. Source: February 26, 2008, Write-up: February 29, 2008