Media Center

COE718: Embedded System Design

Final Project

1. Objectives

The purpose of this final project is to create a media center using the MCB1700 board, uVision and all the programming concepts you have learned throughout the semester. The media center's features include a photo gallery capable of displaying various bmp files, an mp3 player that plays streaming audio/mp3 tracks from the PC, and a game center with one or more different games the user can play. More details are provided below.

2. Media Center Specifications

2.1. Overall Requirements and Functionality

Upon launching the media center, a graphical interface should appear consisting of a menu of options based on the functionalities/tasks of the media center. The graphical interface should be implemented on the LCD, with its functionality including:

- A Photo Gallery
- An mp3 Player
- Game(s)
- Any additional functionality you (as the programmer and designer) wish to implement

The menu should be navigated with the MCB1700's joystick. You may also wish to add LEDs during navigation and/or execution, and embed a keyboard via USB for game play.

The user should be able to select any of the menu options using the joystick. Once selected, the hardware must execute the task, being interactive with the user where necessary (i.e. user uses the joystick to browse through the photo gallery etc), and return to the main menu once the task or user is finished with the selection.

The next subsections describes these functionalities in detail.

2.2. Photo Gallery Specifications

The photo gallery should be able to display various bitmap pictures, one at a time, on the LCD. It is your choice how to present these various pictures, i.e. through another menu of options listing the .bmp file names, a simple carousel view of the pictures which can be manoeuvred by the joystick, etc. Be creative.

It is worth noting however that a .bmp cannot be directly displayed on the LCD. Rather, the .bmp file must first be converted to a C file, which is then read in to the program as an array of unsigned chars. You may find the program GIMP useful for this type of file conversion. GIMP is capable of exporting raw image data necessary to create a C file from the original .bmp. To make the conversion, open the image with GIMP and 'export' the image as a C-source file. Enter a prefixed name (or the default one given). Ensure that you check the options "Use macros instead of struct" and "Save as RGB565(16-bit)".

Press "Export". The C-file should now be created and present in the folder you selected. When opened, there will be header information in the file, describing your image. Ensure that all "guint8" variable types are changed manually to "unsigned char" type as Keil will not be able to understand this type. This header information will also be of use when integrating the pictures into your main code.

To get you started, you may wish to refer to the example project provided to you in the course directory under "U:\\coe718\project\examples\LCD_Blinky". Understand what the code is trying to implement, and thereafter try executing the .axf file on the board. When integrating your C-file picture data into the main file, note that the file generated by GIMP is of type static unlike the example provided to you in LCD_Blinky. Therefore you can not refer to this file as "extern" as done in the sample code, but rather you must use #include "filename.c" the picture into your main code.

2.3. mp3 Player Specifications

The media center's mp3 player is responsible for streaming audio via USB from the PC to the dev board. Therefore the USB should be able to connect to the PC when selected, and disconnect when signalled to exit from the player (using the joystick etc). The potentiometer on the board should be able to adjust the volume of the on-board speaker, located underneath the LCD screen. Counter clockwise rotation of the pot should increases the volume, whereas clockwise rotations to the pot should decrease the volume. When enabling the mp3 player, ensure that you play an mp3 or Youtube video on the PC to be streamed to the dev board to hear the audio playing on the dev board. It is also recommended to display a splash screen on the LCD when the mp3 player is selected and playing the audio.

To get you started with playing audio from the PC to the dev board, you may wish to refer to the example provided to you in the course directory "U:\\coe718\project\examples\USBAudio". In this folder you will notice many support files used for USB implementation. Understand the generalities of each of the files and what the main file accomplishes in correlation to the support files. Also evaluate the importance of the interrupts in USB execution. The pdf file lpc17xx.keil.usb.audio provided to you in the main coe718\project directory may also be of use.

2.4 Game Selection

Implementation of the game section is solely up to the programmer. It will require the integration and interaction of the MCB1700's LCD and joystick, with an option of using the keyboard. You may implement any game desired including Tetris, snake, pong/paddle, Pacman etc. Be creative. A one game minimum is expected when working individually, and a minimum of two games is required when working in partners in addition to keyboard integration.

An example of the media center's menu selection screen and its options is provided in Figure 1.

Fig. 1: Sample Media Center Menu Screen

3. Hints for Getting Started

It is recommended to use the files provided to you in the project directory U:\\coe718\project\common. Copy and paste all the items in this folder into your project directory. Use the system_LPC17xx.c file provided to you in /common in your 'Startup Code' section as you have for all labs. Add the \common\inc folder to your include paths, i.e. Project >> Options for Target 'LPC1700' >> C/C++ >> Include Paths.

Now browse through the files in \common\inc and \src. Notice that the LPC17xx.h and system_LPC17xx.c files are included and re-defined in both \common and \common\inc. Once again, double check that all of these files/folders have been copied into your project folder as they contain different parts of essential code pertaining to the project.

When evaluating and executing the examples provided to you in U:\\coe718\project\examples, also be sure to make note of their "Options for Target 'LPC1700' settings, including any scatter files and flash initialization files necessary for target execution.

4. Project Deliverables

This project is due in <u>week12</u> at the beginning of your lab session. You are expected to deliver the following:

- You are to write a report on the media center's implementation in an IEEE paper format. The final report should be 10 - 15 pages, **not** including the code and appendices. The following specifications should also be followed:
    - o Avoid cutting and pasting of Figures. Only use pictures and diagrams of your own, always labelled accordingly.
    - o Ensure that you reference as necessary using the IEEE format
    - o Use a suitable font - preferably Times New Roman, size 11 or 12 points with single line spacing. Single or double column is optional.
    - o The pages must be letter size, with 1.0'" top, bottom, left and right margins
    - o The report must include the following sections:
        - Abstract
        - Introduction
        - Past Work/Review - *include relevant information on your prior work, along with previous and relevant work on the subject you are implementing*

- Methodology - *describe the method you used to design your project, from a top-level and general perspective*
- Design - *give details based on your methodology for the various components, modules, functions etc you implemented in your project*
- Experimental Results - *describe your results in words, diagrams etc as necessary*
- Conclusion
- References
- Appendix - *including all your c files, any files you have adjusted given to you in any of the Keil examples etc and anything you deem relevant to the project*

- Ensure that you include the Ryerson University title page, dated and signed, with your report attached.

- You (and your partner if applicable) are to present a working demo of the media center, displaying all functionalities as specified. You may also be quizzed during the demo to test your knowledge of the topics covered and your implementation methods.