Multimedia Design Project Final Report

Danilo Zelenovic, 501032542

Department of Computer and Electrical Engineering

Toronto Metropolitan University

Toronto, Canada

1. Abstract

The multimedia device implemented on the MCB1700 board utilizes many concepts learned during the course. The multimedia center’s features include an image gallery, which is capable of displaying BMP files, an MP3 player, able to play audio tracks from the connected PC, and it also has the option of playing a game. Each of these options utilizes the joystick and the LCD built into the board. The MCB1700 board includes many features. The NXP board features a 100MHz ARM Cortex-M3 processor, along with 512KB Flash and 64KB of RAM. It also includes a QVGA TFT color LCD, USB 2.0, a 5-position Joystick and push-button, analog voltage control for ADC input, an amplifier and speaker, up to 70 GPIO and more, however, these are all the features looked into when designing and creating this project. The Keil MDK (Microcontroller Development Kit) is a software development environment created for usage for a range of Arm Cortex-M based microcontrollers. The software includes the uVision IDE, compiler, as well as all essential middleware. It includes ULINK debug adapters which allow users to program, debug and analyze embedded applications. Using the hardware and software together and learning how they interact with each other is crucial for the multimedia center.

1. Introduction

This is the final version of the multimedia design project with all features completed. The design includes an image gallery, an audio player, and a game, with functions which can be selected using either the joystick on the FPGA or using an external attached keyboard. An embedded system is a computer system built with a dedicated purpose, with real-time constraints, though it itself is not a computer. For the duration of this course, we have learned about the basics regarding the applications and theory behind these devices, along with getting hands-on experience ourselves. Embedded systems are used in almost every single electrical device, as they have a wide variety of uses, as demonstrated in this project. It was with the aid of studying the Cortex M3, as well as learning how each component worked over the course of a semester that it was made much easier to implement this multimedia center onto the MCB1700 board. The project aimed to utilize as many features of the MCB1700 board as possible, while using all of the information learned from the labs to create an efficient, proper product which showcases the potential of microcontrollers. Much was done during the previous phase, however, many additions and small fixes were done during the second portion of the allocated timeframe of the project.

1. Past Work

The main menu was completed by creating multiple versions of the screen. When the main menu is accessed, the cursor is defaulted to the image gallery option. From there, the user can input down on the joystick to enter the next screen, which appears identical to the previous one, with the only exception being that the cursor is now set to the next lower option, that being the audio player. From this screen, if up direction is pressed, the screen with the cursor on the image gallery option will be selected, and if the joystick is pressed down, the screen where the cursor is set to game will be displayed. It also has the option to scroll through all the options, meaning that if you are on the game option, and down is pressed, the cursor will instead re-appear at the top on the image gallery option, and the opposite is true as well. If the user has the cursor on the image gallery option and presses up, they can go directly to the screen where the cursor points to the game option. At any of these options, if the user presses the joystick, they will select that option and be brought to that screen.

If the image gallery option is selected, a screen will pop up with instructions on how the gallery works. From there, the first image will pop up. Now, the users have the option to either move to the next image by pressing right on the joystick, left or up on the joystick to go back to the main menu. For subsequent images that are not the first or last ones; right on the joystick will take users to the next image, while left will move them to the previous one, while up will take them to the main menu still. For the last image in the gallery, left will bring up the previous image, while up or right will bring the user back to the main menu.

If the audio player option is selected, a new screen will pop up, and the USB audio will be enabled in a similar fashion, with the use of the provided example code as a base. A small message or something of the sort may appear on screen to let users know that the audio is ready to play. Users are then able to play audio from the computer.

If time permits, a bar could be placed on the LCD as well, which will grow and shrink based on the volume of the audio playing, by using the potentiometer ADC readings. If the user presses left on the joystick, the USB audio will turn off, and the main menu will pop up. Note that the potentiometer is always tracked. However, this does not impact any other functionality if the audio player has not been selected, as the only purpose is the adjust volume, but the volume will not be allowed to play unless the audio player is enabled.

When the game option is selected, a new screen will appear with instructions. Then if they select left, they will return to the main menu. If instead, they press select on the joystick they will be entered into the game. Once the game is completed, a results screen will appear, and users will have the option to play once again by pressing select on the joystick or they will be able to press left on the joystick to return to the main menu.

1. Methodology

While information has been noted in the previous section, as well as on the interim report, it will be noted in further detail in this section. Functions were used to implement all portions. As the program is flashed, the user will encounter the main menu. There will be 3 options listed. The users will be able to list between the three by using up and down on the joystick, with a cursor letting them know in which location they are. If the user selects the image gallery, they will be moved into that screen and the LCD will display the settings for a short while, then after the delay counter is completed, the first image will pop up. From there, the users can move the joystick left or right to switch between images. They may also press up on the joystick at any time to return to the main menu. Also available, is if the user is on the first image and presses left, or on the last image and presses right, they will also return to the main menu. The images (PNG or JPEG) were converted to a version that is readable and can be displayed by the LCD first. The audio player allows the connection to the speakers once selected, and that method will be moved to. Once that section has been left (i.e. the user moves the joystick left), the audio will disconnect, and they will be back at the main menu. The game was also implemented. The game is a simple one, but a fun one, which can also be used for training purposes. It is a game of reaction speed. Images may appear on the screen, and users will need to move the joystick in a direction corresponding to that image as quickly as possible. After the game has been completed, the end screen will pop up with the user’s score, and they will have the option of either returning to the main menu or to play again. A keyboard option was attempted; however, it was ultimately decided against. The flowchart can be found below. The main note about the main menu is the fact that it operates, at its core, with the use of a variable tracking the position of the cursor, based on if users input up or down on the joystick. This enables it to activate the proper function within in an if statement nestled in the joystick position tracker which will move it the correct next screen.

A diagram of a flowchart

Description automatically generated

Figure 1. Flowchart of project

1. Design

Most of the program was created while using functions to execute tasks. The RTOS possibility was not taken. Using functions allowed the project creation to be much simpler. When the program is first started on the board, the main function is called. The menu will be cleared, and the option will then be displayed. The menu is in fact multiple different screens. The cursor initially starts on the image gallery, but if it is moved up or down, the next corresponding screen will then replace it, which will be identical, with the only difference being that the cursor will be in a different location, pointing to a different option. The variable is also adjusted, so that when the user makes their selection on an option, they will go to the next screen which belongs to that option, and the corresponding function will be called. The screen value is kept with a variable, which will decrease or increase, depending on if the joystick is pressed up or down. Once the select option is selected, it will execute a function, each corresponding to a certain value of the joystick variable, which will be used to determine which option was selected and bring users to the next correct screen.

As an example, if the image gallery is selected, the corresponding function is executed, based on the screen and joystick position tracking variable the user was on in the main menu. The LCD will refresh, and the following screen will be displayed. The screen that was designed to appear next was an instruction screen. This screen has the instructions displayed on it, which will give users all required information on how to browse the image gallery. After a short while, using a delay counter, the screen will be cleared, and the first image will appear. While in the image gallery, the options will always be available to go up to return to the main menu. The program will use a similar approach as before to keep track of which image is being displayed, so that it knows if it is either the first one or the last one. This way it knows if pressing either left or right should bring it to the next/previous image, or if it should bring the user back to the main menu by ending the loop where the image gallery is executed. The images were first converted to a format that could be displayed, while keeping in mind the size constraints of the LCD. As the hardware and software cannot display an image as a PNG or JPEG, the image first needed to be converted to a C-source file. This was completed by using the GIMP program. GIMP was used to export the raw image as a C-source file. Ensure that in order to get the preferred output, the options “use macros instead of struct” and “save as RGB565(16-bit)” must be selected. Once this is completed, and the images are added into the project folder, change the “guint8” variables into “unsigned char” type so that Keil software is able to understand it. The images codes were they included to the blinky main code as an extern file, allowing them to be used by the main portion of the code. Examples on what the image code is included in the appendix.

When the audio player option is selected, using the same approach as the image gallery, its function will be invoked instead, the screen will clear, and a new screen will pop up as an indication that the audio player is working. When this option is selected, the audio player will be allowed to play audio directly from the MCB1700 board. The potentiometer will also control the volume. The use of the potentiometer is not difficult to use, as this was provided as an example code. It can be used at any time, however, it will have no effect, unless the audio is playing, as otherwise, the speakers are not connected. It will only be allowed to play audio while in this option and will disconnect once the user clicks left on the joystick to return to the main menu. The example files were used as a reference for this portion, however, it had to be implemented only on 1 screen in the project, so this was the challenge here. As more things were added to this screen, minor issues kept arising such as the main menu turning black afterwards or flashing occurring on the LCD, so these changes were reverted to how they were initially. The example USB audio was used for this portion to allow the audio to be played. This feature was utilized by executing the code from a method only once that screen was accessed, so the joystick was entered as select, once the position-keeping variable was set to a specific value. Once this screen is exited, the method is exited, which runs as a while loop, and the main menu is displayed once again, while turning off the speaker, thereby not allowing audio to be played.

While on the main menu, if the game option is selected, (the joystick variable is the proper corresponding value and the select option is entered with the joystick), the function to launch the game by clearing the LCD, and displaying the next screen is executed. Instructions will appear on this screen, and users will have the chance to either return to the main menu, or to play the game, depending on their input on the joystick. The game is a reaction time tester. Different images will be displayed, and users must input the correct direction corresponding to that image as fast as possible, to keep their time as low as possible. The images will be displayed in no particular order. After the game is timed out, the end screen will display their score, and users will be given the choice to either replay or return to the main menu based on their input. The input will be read, and the proper function will be executed. The game, ending screen and the instructions are all included in a while loop. This while loop is broken once the users exit the game by pressing the joystick left on either the instruction screen, or on the end screen, which displays the user’s score. The game keeps track of the user’s score by keeping a tick timer, which increases every tick, but resets after every correct image direction input. These values are summed once the game terminates, and this value is displayed as the user’s score. Other games were considered, however, this game seemed very creative, and also had a use, as potentially helping users increase their reaction times slowly but steadily. On the end screen and the instruction screen, it uses if statements to note if the user inputs left to return to the menu by executing the menu() function, or if they input select, they will return to the game() function.

1. Experimental Results

The results obtained during from the project were successful. The main components of the multimedia center were set in accordance to the specifications, and although minor bugs have appeared with every adjustment, overall the project could be deemed to be successful, as all requirements were met in a timely fashion. While the decision not to use the provided RTOS by Kiel may affect the performance of the board and program slightly, it is not a difference what was found to be drastic, as the response times currently executing are not very large. The main menu UI is a simple one, but one that is easy to understand and functions nicely. The main menu appears as follows:

A close up of a device

Description automatically generated

Figure 2: Main menu with cursor on media player

A close up of a device

Description automatically generated

Figure 3: Main menu with cursor on image gallery

A close up of a device

Description automatically generated

Figure 4: Image gallery instruction screen

A close up of a circuit board

Description automatically generated

Figure 5: Example image 1

A close up of a computer chip

Description automatically generated

Figure 6: Example image 2

A close up of a computer chip

Description automatically generated

Figure 7: Example image 3

A close up of a circuit board

Description automatically generated

Figure 8. Audio player screen

The audio player is functional and displays a message to the LCD indicating it is functional and displays how to return to the main menu. When the left input is pressed with the joystick, within the music playing function, all connections turn false, disconnecting the LCP speaker.

A close up of a circuit board

Description automatically generated

Figure 9. Game instruction screen

A close up of a circuit board

Description automatically generated

Figure 10. Score screen, post-game

As seen from the figures above, the multimedia project was not only successful in accomplishing the required tasks using functional execution, but it also is quite intuitive, and appears very nice.

1. Conclusions

In Conclusion, the multimedia center has been implemented mostly correctly, with some minor hiccups. Another portion that could have been done better was execution time could have been lowered with more optimal code, however, for the short timeframe permitted for this project, this was a good job. The project utilized most functions of the board and used many concepts we learned throughout the course to complete. The project implemented an image gallery, audio player, and a game center all in one embedded system, showcasing its many potential uses, and why they are so popular today. This project can be called a success, and much was learned from it. There are also additional functionalities that could be added next time, such as a bar displaying how loud the volume is on the audio player, or the option to play multiple different games, or keep track of your high score. Minor issues encountered included the screen turning black instead of white after certain screens, so certain functions were adjusted to make it function properly, however, this made certain portions look/act slightly different than what was first planned. Overall, the multimedia center is able to use all features required, including the game, which randomly generates images and keeps track of your input speed to evaluate you and give you a score, though the scoring system sometimes may have bugs. The image gallery also functions as planned, with users being able to shuffle between multiple images back and forth, and able to exit back to the main menu at any moment they wish, as well as an audio player, which enables the speaker option on the board, using the potentiometer to increase or decrease the volume, based on the ADC converted value selected by users.

1. References

[1] <https://www.ecb.torontomu.ca/%7Ecourses/coe718/labs/Media-Center.pdf>

[2] Previous lab files

[3] Provided USB audio example files

[4] Keil provided example files for components such as the ADC conversion

[5] <https://www.keil.com/arm/mcb1700/>

[6] For the game, this code was used as a reference: <https://github.com/Matheesan2113/EmbeddedSystemMediaCentre>

1. Appendix

Blinky code:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Game code:

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

USBaudio:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

These were the main codes created or edited to meet requirements that were not used as-is from the provided versions.