

Music Player

Final Project



OVERVIEW

For this project you will design a music player with the 68HC12 microcontroller. The music player will use all the peripherals you have learned in lab. You are tasked with implementing this design for each peripheral.

REQUIREMENTS

1. The music player shall have a login.
2. The music player shall have menu.
3. The music player shall play at least 3 songs.
4. The music player shall be able to pause and skip a song.
5. The music player shall keep a timer of the current song.
6. The music player shall have a battery.
7. The music player shall implement all peripherals listed below.

SPECIFICATIONS

- The LCD is used for the login, menu, song timer, and displaying the current song playing.
- If the music player finishes the last song it should reset and play the first song again. It should never stop playing music unless the user instructs it to.
- The battery is simulated using the DC Motor. If switch 7 is high the music player is considered plugged in and recharging. The battery should not lose charge while switch 7 is high, meaning the DC motor will constantly spin at max speed. Once switch 7 goes low the battery should start losing charge again (DC motor starts spinning slower).

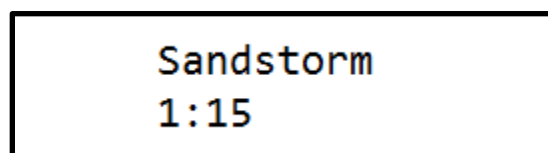
MILESTONES / PERIPHERALS

Login

The user should be able to login with a username and password. There should be three pre-programmed users. The user should be able to add new users to the system. If a user doesn't have an account, he/she should be able to create one from the login screen.

LCD

The LCD will be the menu for the music player. The menu should display menu options such as: selecting songs, adding new users, and logging out. When a song is playing the LCD should display the name and time of the song, like below.



Songs/Speaker

The music player should have at least three pre-programmed songs. You must implement Darude - Sandstorm. Link to song:

<https://www.youtube.com/watch?v=y6120QOIsfU>

LEDs

While in the login screen, the LEDs should flash from right to left then left to right with the RTI. When the user logs in the LEDs should stop flashing.

DC Motor

The DC Motor simulates the music player's rechargeable battery, which works similarly to a phone battery. When the battery is fully charged the motor should spin at max speed. As the battery loses charge the motor should spin slower, eventually stopping when the battery is dead. The battery should last about 5 minutes. When the battery dies the music player should log out the user and stop the music.

Switches

Switch 1 should be used to pause the song. When you pause a song, it should say "Paused" on the LCD. Switch 1 should be low to resume the song. Switch 7 should act as the "charger" for the battery. If switch 7 is high, then the battery has been charged and should return to full charge. After switch 7 has gone from high to low (on to off), the battery should start decreasing charge again.

Stepper Motor

The stepper motor should tick for each second in the song. One full revolution of the stepper motor should equal one minute. If the song is paused, then the stepper motor should pause as well.

Keypad

The hex keypad should be used to select menu items. The keypad should also be used for typing in usernames and hex passwords.

Potentiometer

The potentiometer should be used to scroll through menu items in a list like fashion. Kind of like on the original iPod, how the wheel was used to scroll through the menu.

IRQ Button

The IRQ is the power button for the music player. It should turn off the LCD screen (like an iPod), Pressing it again should turn the LCD back on. Music should continue to play after the IRQ has been pressed.

RTI

The timing should only be controlled by the RTI. The timer for the battery should be controlled by the RTI. There should be no delay loops used except too debounce the keypad.

Push Button

The push button should be used to skip the current song and go to the next one. If there are no more songs left, play the first song again.

Three Extra Items:

Your group should come up with three extra features, simple or complex.

10% OF THE GRADE FOR YOUR PROJECT WILL BE BASED ON THESE EXTRAS. YOU WILL CREATE EXTRAS ON YOUR OWN. YOU ARE ENCOURAGED TO BE CREATIVE - THE MORE COMPLEX THE EXTRA, THE MORE POINTS ARE EARNED.

Notes, Requirements, and Constraints:

The overall layout of your simulation should be easy to understand and make sense. A user should be able to operate the simulation with little to no training or explanation. If you are unsure if the layout of your system makes sense, ask one of your TAs or fellow students to try to move through your simulation.

No delay loops are allowed, you must utilize the Real Time Interrupt (RTI). DELAY LOOPS ARE ONLY ALLOWED FOR HEX KEYPAD DEBOUNCE.

You are encouraged to be creative and make this project your own. You can make reasonable assumptions in the development of this project, but keep in mind that the assumptions must make sense to the user (and to the lab TAs).

If you have any questions pertaining to this project, please discuss it with your Lab TAs as early as possible. You may use any C code provided through the labs without altering it, but all other code must be written in assembly.