ELEC334 Project3

December 24, 2020

Objective

Main objective of this project/final is to create **a digital voice recorder** that can record your voice, playback a selected voice recording, and delete single or all recording data. During this project, you will use various modules such as Timer, PWM, ADC, and External Interrupts.

This project will get you to incorporate all the modules that you implemented individually into one project. As you have seen previously, getting all of them working correctly is not that easy, and if not carefully designed, can cause a lot of problems. So I would strongly recommend you tackle problems indivudually while applying the **approach** section.

Approach

1. **Draw a detailed flowchart** of the design. Determine interrupts, draw the flow diagram of each interrupt routine. Define globally shared variables among different interrupts.
    - Utilize interrupts for all functionality.
    - Probably your main loop should be empty.
2. **Draw a state transition diagram** of the states. This will help you with implementing the design and keeping things organized.
3. **Determine tasks for the project.** These tasks should be as basic as possible, and once completed, can be checkmarked easily. Requirements items should be considered as part of these tasks.
4. Gradually complete the tasks and incorporate them into your project. **Don't do everything at once.** Your flowchart here will help you a lot.

Requirements for the project

> *Make sure to read and understand the requirements for each section since your grade will depend on it.*

Technical requirements

- Written in C. No HAL or equivalent libraries. (Using CMSIS, and `stm32xxx.h` kind of headers is fine.)
- Connect a microphone to record your voice. Keep in mind that if this microphone does not have an on-board amplifier, you will need to build one yourself.
- Build an amplifier and connect a speaker with variable pot to playback the recordings.
- Connect 2 x 24LC512 EEPROMs on the same I2C bus. Keep in mindm when wiring the bus will require pull-up resistors on both lines, and each of these devices need different address to communicate.
- You should be able to **at least** record 4 tracks with 5 seconds each. 5 seconds should be fixed, but if you can fit more tracks that is fine.
    - Calculate the maximum datasize for two EEPROMs for keeping your data and create a table of how many seconds can be recorded with different data rates. Pick one that will fit the requirement.
- A keypad should be attached to operate the device.

    - Assign a key for recording a voice. The recording will go for 5 seconds and automatically stop/save it. After the track is played, it will stop and go back to IDLE state.
        - Pressing any other button should not have any effect.
    - Assign first 4+ number keys for track select when not recording. For example pressing 1 will select the first track, pressing 2 will select the second track, etc. This key press will not do anything else.
    - Assign a key for playing/pausing the selected track when not recording. After the track is played, it will stop and go back to IDLE state.
        - Pressing any other button should not have any effect.
    - Assign a key for deleting the selected track. After the track is deleted, it will go back to IDLE state.
    - Assign a key for seeing the track status. After the key is pressed, 7SD shows the number of available tracks.
- A 7SD should be attached to display the operations and status.
- You should have multiple states, some of which include:
    - START state which only happens when the board powers up 7SD should show your ID (first 2 and last 2 digits)
    - IDLE state which displays `IdLE` on the 7SD and does not do anything else. (waiting for track select or record start)
    - FULL state which displays `FuLL` on the 7SD and prevents going into RECORD state.
    - RECORD state where the 7SD shows `rcd` and a count down from 5 seconds indicating the recording. (i.e. `rcd3`, `rcd2`)
    - PLAYBACK state where the 7SD shows `PLb` and the track being played back (i.e. `PLb2`, `PLb1`)
    - STATUS state where the 7SD shows `Ava` the number of available tracks. (i.e. `Ava3`, `Ava0`)
- If no button is pressed for 10 seconds, the device should go back to IDLE state.

Quality requirements

- No bouncing on the buttons
- No considerable delay with button presses
- No flickering on the displays **AND no brightness difference on the 7SDs**
- Recording should be undersdandable
- Code should be properly commented with your name / school ID added in the beginning

Submission requirements

1. A *1-minute video* of your project demonstration
    - Record the video explaining your code briefly, and show the demonstration.
    - Preferably hold the camera still.
    - Upload the video to wherever you want (youtube / onedrive / stream) and give a link.
2. A well-written report in PDF format
    - Cover page
    - Block diagram as well as connection diagram
    - **State transition diagram**
    - Flow chart
    - Parts list (w/ prices)
    - Project setup w/ picture
    - Task list and their completition status (checkmark / x should be suffice)
    - Methodology for any numerical work
    - References
    - Properly formatted code list in Appendix. Your code list should have a fontsize of 10 and single spaced with monospace font.
    - Any missing parts of the project and explanation why they are missing
    - Any challenges that you faced and how you resolved them.

- Conclusion about what you learned from the project

3. Your submission should be a zip file with the following name
   `yourname.lastname.project3.zip`
   The zip file should have the following folder structure

```
yourname.lastname.project3/
  report/
    yourname.lastname.project3.report.pdf
  code/
    project3.c
    project3.h
    anyextrafiles.c/h
  video/
    video_link.txt
```