

ELEC 5260/6260/6266 Audio Recorder/Player Project

PART 1 – Rough Draft (15 points) Due date: To be posted

PART 2 – Final Project (25 points) Due date: To be posted

The goals of this exercise are to study:

1. Hierarchical design of embedded system software, including device driver libraries
2. Real-time data acquisition/control,
3. Audio input (MEMS microphone),
4. Audio output (Audio codec)
5. QSPI flash memory read/write
6. Serial device interfaces (SPI, I²C, SAI),

You are to design an audio recorder/player with button-selectable operations. Audio input is to be recorded from the MEMS microphone on the STM32L476G-Discovery board, stored in the QSPI flash memory, played via the audio codec and output via the audio mini jack (most standard speakers, “ear buds”, headphones, or similar listening device can be plugged into this jack.)

Project Specifications

1. **Idle Mode:** Enter this mode whenever the joystick UP button is pressed. This should halt any recording or playback and reset any elapsed time to 0.
2. **Record Mode:** Enter this mode when the joystick DOWN button is pressed. When in this mode, the joystick SEL button should start and stop recording audio data from the MP34DT01 MEMS microphone. For the demonstration, record at least 6 seconds of audio data.
3. **Play Mode:** Enter this mode when the joystick RIGHT button is pressed. When in this mode, the joystick SEL button should start and stop playing audio data play through the CS43L22 Audio Codec and the headphone jack. For the demonstration, play at least 6 seconds of audio data.
4. Audio data is to be written to the **Quad SPI (QSPI) flash** memory when in Record Mode, and read from that flash memory during Play Mode.
5. On the **LCD**, display a 3-letter code indicating the current operating mode (record, play, idle). Also display elapsed recording/play time in seconds (two digits). This should be 00 in Idle Mode.
6. To ensure that the processor can “keep up”, you may keep the audio sampling frequency relatively low (in AUDIO_IN_Init() and AUDIO_OUT_Init() function calls), but it should be high enough to produce “recognizable” audio.

You may find the following Board Support Package (BSP) library files to be useful:

```
stm32l476g_discovery.c
stm32l476g_discovery_glass_lcd.c
stm32l476g_discovery_qspi.c
stm32l476g_discovery_audio.c
```

Also from ../Drivers/BSP/Components

```
/cs43l22/cs43l22.c    (Audio Codec Driver)
/n25q128a/n25q128a.h    (QSPI flash header file)
```

Be sure to edit file **stm32l4xx_hal_conf.h** in your project to un-comment the #define *_ENABLED statements for the modules used in this project (DFSDM, LCD, QSPI, SAI, I2C, SPI, etc.)