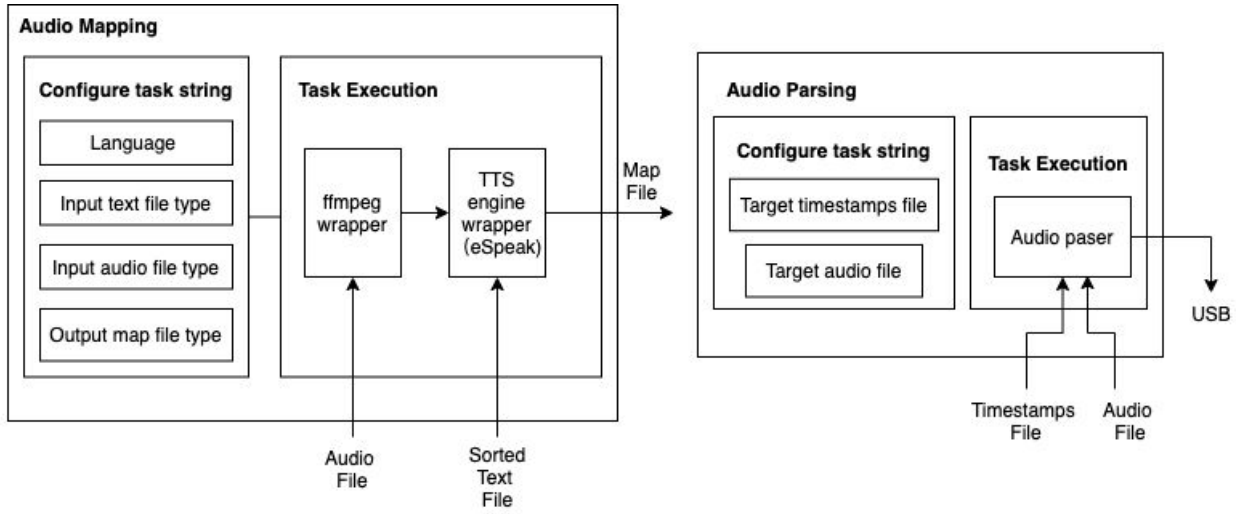
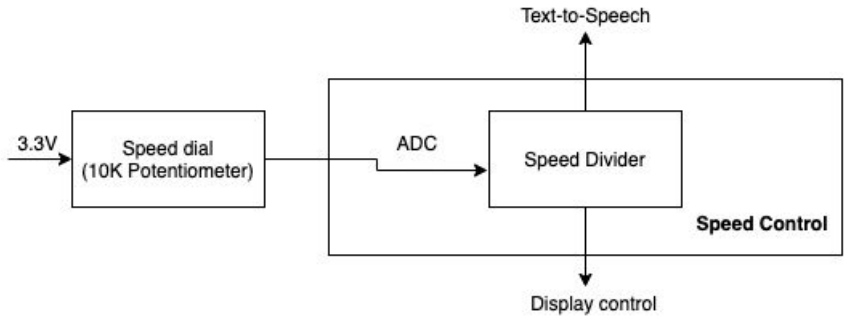
Subsystem 4: Speed Knob & Speed control & Audio Mapping & Audio Parsing
Subsystem Owner: Qi Dai

Specifications

1. The voltage feeding into the speed knob shall be 3.3V (DC).
   Justification: The 3.3V voltage shall be sufficient enough to support the speed dial to function properly.
2. ADC conversion resolution should be 8 bit (max:255 min:0).
   Justification: The resolution should support high speed conversion for analog voltage.
3. ADC conversion time should be less than 5ms and delay less than 100 ms.
   Justification: ADC sampling time should be short enough to provide fast conversion.
4. The pre-scaler range should be 800 to 8000.
   Justification: Speed dividers should be diverse enough to create differentiable reading/audio speed for readers.
5. The audio file should contain audio matching the text in the sorted ebook file. .
   Justification: gTTS should convert sorted text file into exact audio file in .mp3 format.
6. The audio file for e-book shall be in .mp3 format.
   Justification: The aeneas library can work on ffmpeg-supportive formated file while ebooks often use .mp3.
7. The audio-to-text map file shall be in .csv format.
   Justification: CSV format could be imported into excel for timestamps file formatting
8. The potentiometer shall have a resistance range from 0-10k.
   Justification: This potentiometer shall provide sufficient resistance to support the speed range.
9. The audio file should be parsed by timestamps with accuracy of 95%.
   Justification: Each audio file should match the timestamps set by map file with high accuracy in preparation for text-to-speech output.
10. The text file should be sorted by commas.
    Justification: Text should be sorted to make convenience for audio parsing.

Sub-system diagrams





Sub-system interactions

The audio mapping system will retrieve data from the assigned audio file and text file, mark segments of audio frames with their corresponding text based on specifications entered and separated audio fragments by punctuation and blanks. The output sync map will be in .csv format, which will contain all the timestamps and corresponding line info. And this file would be imported to excel to extract timestamps and exported as .txt for audio parsing. And parsed audio files would be saved to USB for text-to-speech output.

The power system supply 3.3V voltage to the 10K potentiometer (speed dial) and then by adjusting the resistance of potentiometer, the speed control will receive the voltage data and convert the voltage to desired speed divider; after it will pass the audio speed data to the text-to-speech output while displaying speed will be passed to display control system.

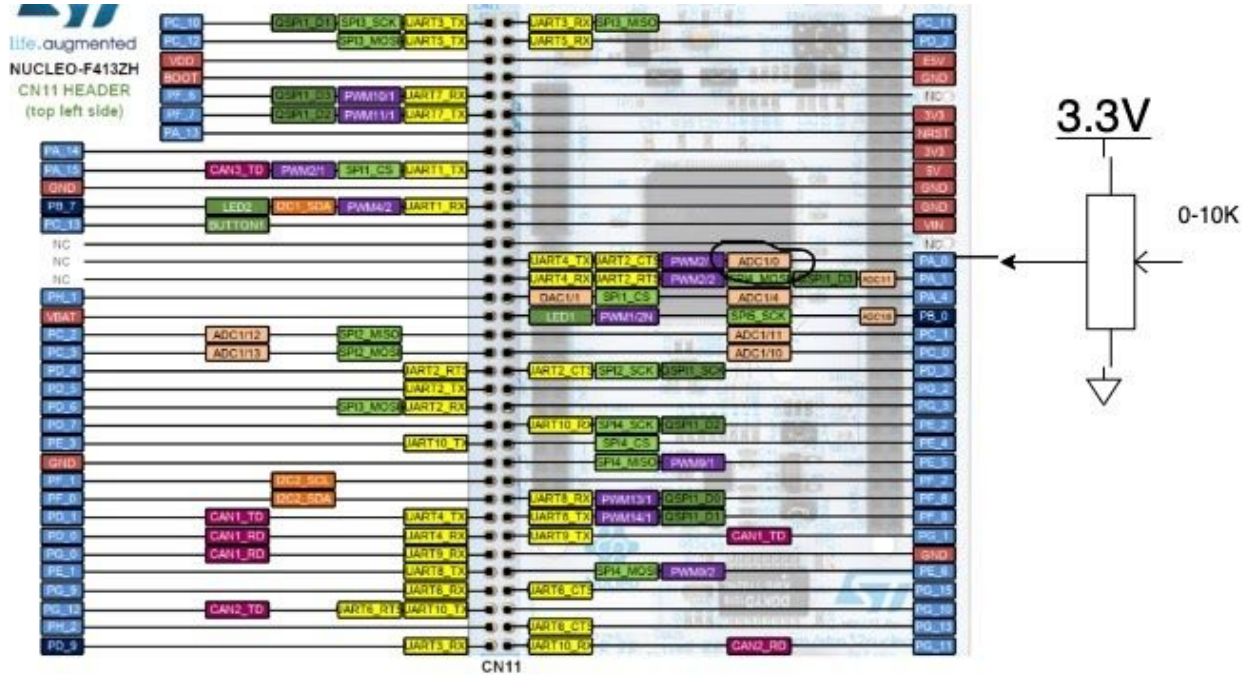
Core ECE design tasks

- Audio mapping design: how to markup audio fragments with text file
    - ECE 438 Digital Signal Processing With Applications
- Speed control design: how does clock work, how does divider work, how to pass signal within microcontroller, how to control sample time

    - ECE 36200 - Microprocessor Systems and Interfacing
    - ECE 26400 - Advanced C Programming
- Circuit design: how does potentiometer work, how to connect potentiometer to microcontroller
    - ECE 20700 - Electronic Measurement Techniques

Schematics / parts / algorithm

Schematic:



Software suits: Eclipse, Pycharm

Libraries: Aeneas, stdio.h, stm32f4xx.h

Programming language: C, Python

Algorithms:

1. Will write algorithms to map frames of audio file with lines of text.
   Library reference: https://www.readbeyond.it/aeneas/
2. Will write ADC algorithms to convert analogue signal to digital input.
   ADC tutorial: https://visualgdb.com/tutorials/arm/stm32/adc/

Standards

1. IEEE 1003.1i-1995 - Standard for Information Technology - Portable Operating System Interface (POSIX(R)) - Part 1: System Application Program Interface (API) - Amendment: Technical Corrigenda to Realtime Extension [C Language]