

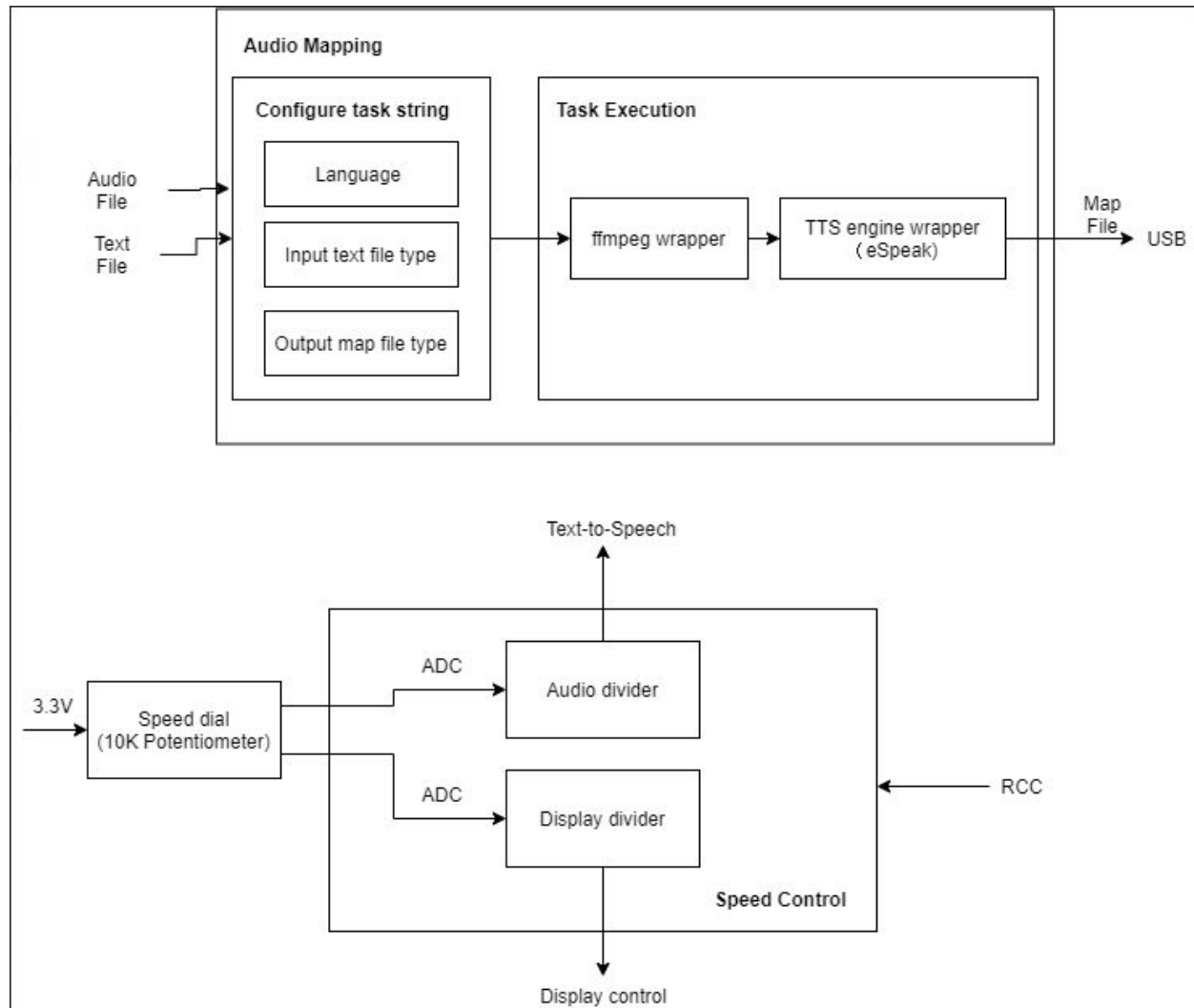
## **Subsystem 4: Speed Knob & Speed control & Audio Mapping**

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. The maximum prescaler for speed divider should be 16 and minimum should be 2.  
Justification: The speed divider should be diverse enough to notice the lighting frequency change of the LED on the board.
3. The audio file for e-book shall be in .mp3 or .wav format.  
Justification: The aeneas library can work on ffmpeg-supportive formatted file while ebooks often use .mp3.
4. The audio-to-text map file shall be in .xml format.  
Justification: Closed Caption File Format supports .xml format to allow further processing.
5. The potentiometer shall have a resistance range from 0-10k.  
Justification: This potentiometer shall provide sufficient resistance to support the speed range.

### 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 .xml format, which will contain all the mapped text-to-audio information, and this file will then be saved in the USB as one of the text-to-speech system input.

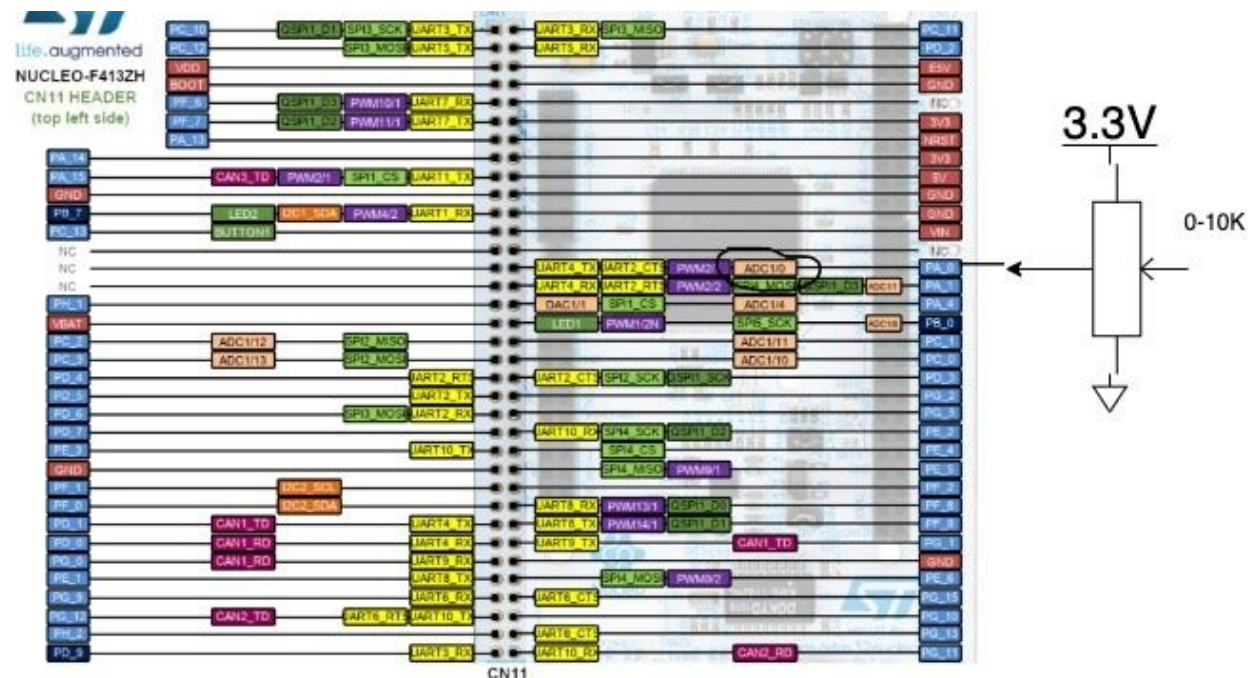
The power system supply 3.3V voltage to the 10K potentiometer (speed dial) and then by adjusting the resistance of potentiometer, the speed control will receive the resistance data and convert the resistance to both desired displaying speed and audio speed; after it will pass the audio speed data to the audio parsing system 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]
2. 1636.2-2010 - IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Maintenance Action Information via the Extensible Markup Language (XML)