# Phys 232 Final Project Documentation – Pennywhistle group

Author : Jack, Alnis, Aureliano, Young

## Software/Hardware Coding :

**Classes and objects** : We defined midi232 class which has 3 attributes: *time, instrument, tone.* Time is the timestamp which signals the pennywhistle when to play the following notes. instrument signals the type of the instrument. In this case, we created the csv file with all pennywhistle cases. Tone signals the note that the pennywhistle need to play at the given timestamp. We added *"stop"* to the tone attribute as well so that the pennywhistle will stop playing all notes.

**Song CSV file :**
In our local directory, we stored the song's csv file. We used Python's *with open* command to open the file, and read each line parsed by comma.
The song that we played is "marry has a little lamb" , and here is the CSV file that we created to meet our requirement of API:

*0, pennywhistle,stop*
*1.1, pennywhistle ,A*
*2, pennywhistle ,G*
*2.5, pennywhistle ,F*
*3, pennywhistle ,G*
*3.5, pennywhistle ,A*
*4, pennywhistle ,A*
*4.5, pennywhistle ,A*
*5, pennywhistle ,stop*
*5.5, pennywhistle ,G*
*6, pennywhistle ,G*
*6.5, pennywhistle ,G*
*7, pennywhistle ,stop*
*7.5, pennywhistle ,A*
*8, pennywhistle ,C*
*8.5, pennywhistle ,C*
*9, pennywhistle ,stop*
*9.8, pennywhistle ,A*
*10.5,pennywhistle ,G*
*11 ,pennywhistle ,F*
*11.5,pennywhistle ,G*

*12, pennywhistle ,A*
*12.5,pennywhistle ,A*
*13, pennywhistle ,A*
*13.5,pennywhistle ,stop*
*13.8,pennywhistle ,A*
*14.3,pennywhistle ,G*
*15, pennywhistle ,G*
*15.5,pennywhistle ,A*
*16, pennywhistle ,G*
*16.5,pennywhistle ,F*
*17.3,pennywhistle ,stop*

For each line of csv file, we added the command into a midi232 class object, and stored them in a list.

**Time Management:**
We used the Python's module : time.monotinic() function to keep track of the time. While the input list is still non-empty, we will keep playing the notes from the input list. After the note is played, we popped the midi232 object out of the list. So when all notes from the input are played, program will stop.

**API Design and Implementation:**
We created functions that map the notes from the command to the specific holes of the pennywhistle that need to be closed to perform such notes. Since the note has already been analyzed before the functions were called, the functions do not have input parameters. For example, there is a function called "play_c()" which asks the corresponding motors to close all 6 holes on the pennywhistle to perform such a note. Other functions will perform different notes by sending different instructions to the motor to close required holes.
Also we stop wind blows in each call of the play note function. It will blow the wind after "fingers" are in the correct position. Since there are 6 motors in our design and our capacitor did not work as expected, we add some time.sleep to make sure there are only 2 motors moving at the same time.

**Motor Design and Implementation :**
"Helper method " for play_note.py to help to control motor operation.
For each hole on the pennywhistle, we used a motor to control both the opening and closing of this hole. We also have another motor to control the wind blowing. We calibrated the angle of these motors so that they can close the hole without pushing too hard onto the pennywhistle nor not being sealed. API will call the functions to either close or open some of the holes in order to perform the given note, so the functions for motors here is just to respond with API's function call and rotate the corresponding motors per instructions.