Tune Studio

A mono audio tone melody editor software for Windows.
Compose your own tone melody, save, playback and output to C/C++ header format.

Revision 1.0.4

by Ricky Gai

MAY 30, 2021

Nexuz Innovation, Malaysia.
(MA0255412-M)

Introduction

About this Book

This is a book about the hands-on information of the Tune Studio software for Microsoft Windows.

In the learning Arduino platform especially to the tone melody creation, you need a tone editor to allow you to compose own melody and play it on the physical 8Ω 2W speaker, active and passive buzzer.

You cannot just rely on downloaded third party notes or melodies because it may subject to copyright limitation upon product commercializing later.

Therefore, Tone Studio software is created to allow you to compose tone melody, save it, playback and output to C/C++ header format that can be used as part of Arduino sketch compilation.

Tone Studio is a proprietary software, it is not perfect but you can create, commercialize own composed tone melodies or musics at royalty free and it costs just a cup of coffee :o)

You may get Tone Studio from Microsoft Store.

All the necessary help documentations and tutorial videos are available via Github at:
https://github.com/rickygai/tonestudio

For any errors found, suggestions and questions, please do email to:
support@nexuzinnovation.com

About the Author

Ricky Gai
The founder and technical director of Nexuz Innovation, a small R&D IT company established in Kuala Lumpur, Malaysia.

He earned a distinction of Oxford Computer Engineering certification discipline in 1992 and mostly exposed to C/C++ programming in his career from the earlier days of MS-DOS until Microsoft Windows environment today.

Nonetheless, much time had been devoted to the electronics studies further before entering into the world of Arduino because the understanding of analog, digital electronics and PCB design are so important to build a stable circuitry beside software programming alone merely.

Thank you for supporting Tone Studio software, I hope you find something useful here.

DISCLAIMER

Abbrevation	Descriptions
NEXUZ INNOVATION / AUTHOR	refers to the author, Ricky Gai.
READER / READER(S) / READER'S	refers to the person who read or experimented the information from the contents of this book.
COMPONENTS / EQUIPMENTS	refers to electronics components, tools, materials that used as part of the circuitry setup.
CONTENTS	Information described in this book, including software source code and hardware circuit designs.
SOFTWARE / PROGRAM / SKETCH	refers to software created by the author.
IP / INTELLECTUAL PROPERTY / COPYRIGHT / PERMISSION	refers to the copyrighted materials (eg. Photo, Diagram, Source Code, Links) that owned by other creators.

The information contained in this book is intended for general reference purposes only and may share portion of extracted information from other resources as part of the illustration usage. Any copyright infringement, please do contact the author to exclude from this book.

While the author endeavour to keep the information up to date and correct, the specification, quality and availability of electronics components may change in time, therefore the author make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the information, vendors, products, services, or related graphics contained in the book for any purpose.

Any reliance you place on such information is therefore strictly at your own risk.

In no event will the author be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from loss of data or profits arising out of, or in connection with, the use of information derived from this book and the software.

Prerequisition

Tone Studio - Setup

Software & Hardware requirements

Below were the basic requirements to run the Tone Studio software.

Software Requirements:

- Microsoft Windows x86/x64

  min version: 10.0.17763.0
  max version: 10.0.19041.0

- Windows Store.
- .NET framework v4.7.2 or above.
- Tone Studio software.

Hardware Requirements:

- Intel PC i3, i5, i7 or above.
- 2-4 GB RAM.
- 1-2MB Harddisk space.
- Intel/Nvidia standard graphics card.
- Keyboard and Mouse.
- Built-in motherboard sound card (eg. Realtek) or built-in internal PC speaker.
- Direct audio jack cable connected to motherboard audio socket..
- None direct audio jack connected speaker is not supported. eg. USB speakers, Bluetooth speakers and other wireless speakers are not supported.

The Overview of Tone Studio control functions.

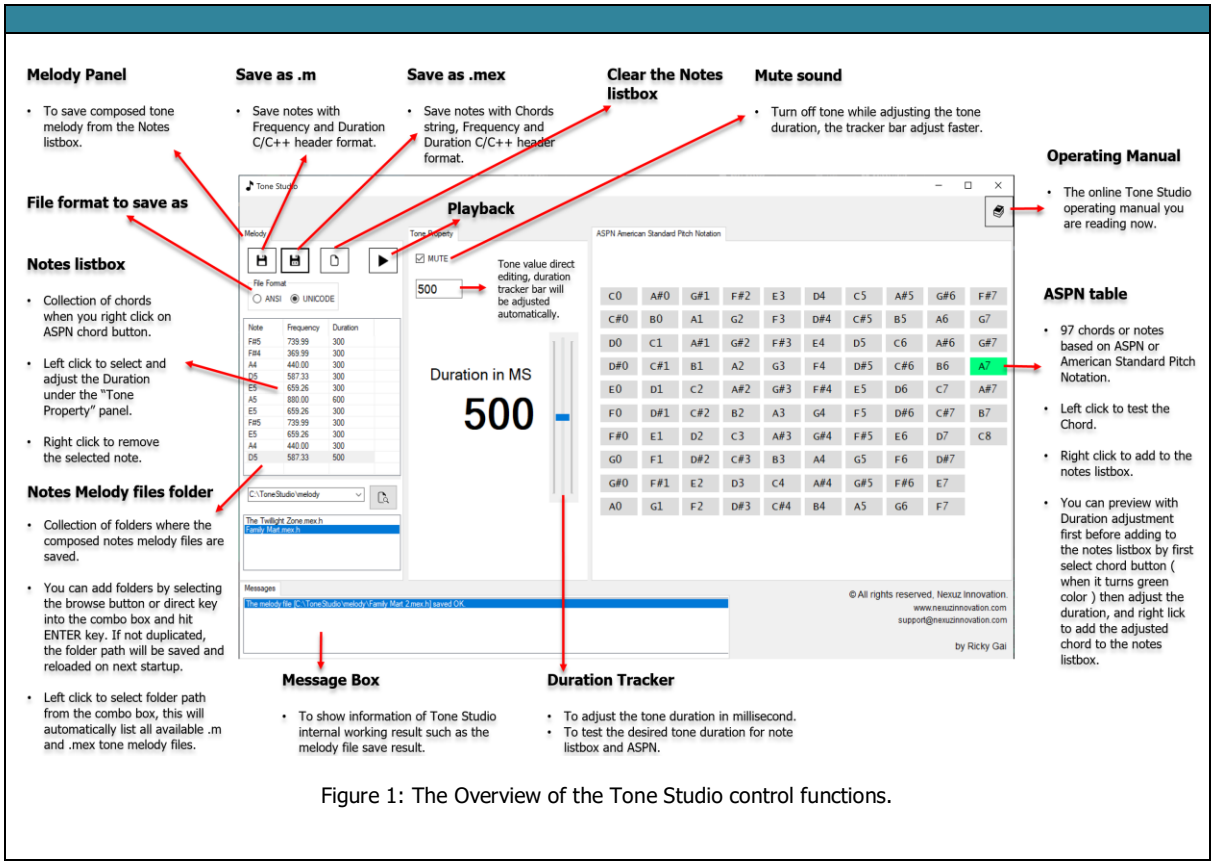
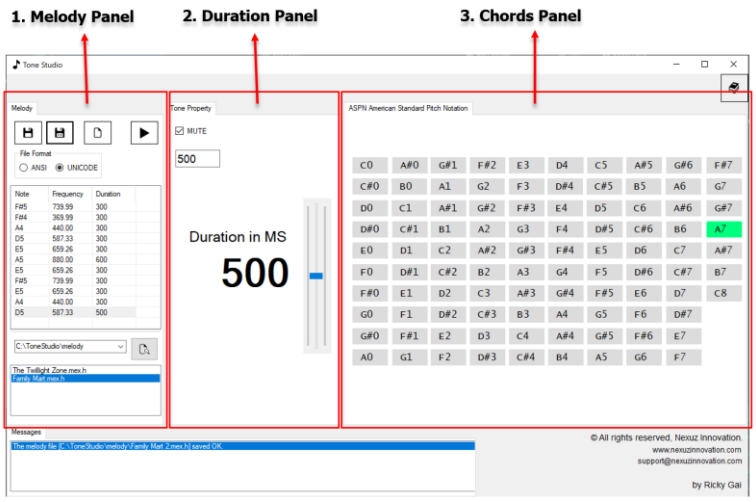


Figure 1: The Overview of the Tone Studio control functions.

Operating Guidelines

How to play a tone ?

Diagrams	Descriptions & References
 Figure 2: The three major control panels	The Tone Studio software is built with three basic control panel interfaces. **Melody Panel** This is where you compose the notes melody, save it, playbank and output to C/C++ header format. **Duration Panel** The Tone Property panel allows you to adjust the duration of a note. **Chords Panel** Contains of ASPN / American Standard Pitch Notation with its 97 chords or notes with predefined frequencies respectively. **How to play a tone ?** 1. Firstly you navigate to the Chords Panel. 2. When the mouse over the chord button, it turns to green color, indicating that chord is selected. 3. Left click on it to produce a tone sound.

The ASPN chords or notes control panel

How to compose a tone melody ?

Diagrams	Descriptions & References

Figure 3: The Chords Panel

Before you can compose a tone melody, you have to select the closest chord pitch to simulate the melody desired.

4. Goto the Chords Panel and left click the chord buttons to find the wanted chord.

5. In this case, I am going to compose a theme music from the classics TV series "The Twillight Zone".

6. So I found the chords that are matching as shown below:

   C5 C#5 D5 G6 F#6 G#4 A4 A#4

7. Right click on the chord buttons above to add to the notes listbox as shown on Figure 4 below:

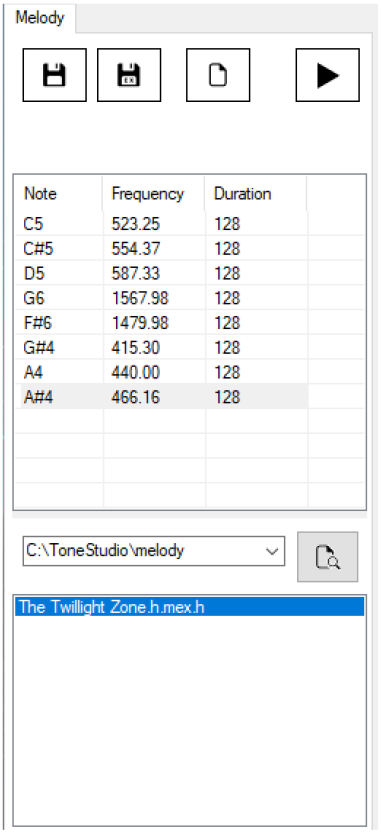


Figure 4: The Notes listboxl

The Melody control panel

How to adjust the tone duration ?

Diagrams	Descriptions & References

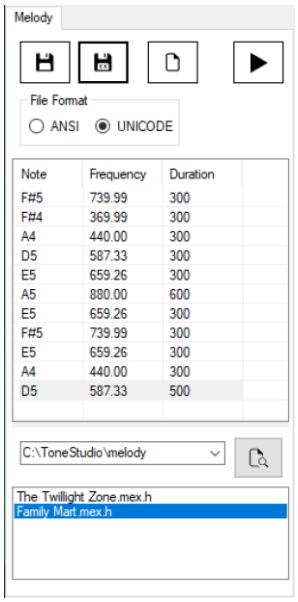


Figure 5: The Melody Panel

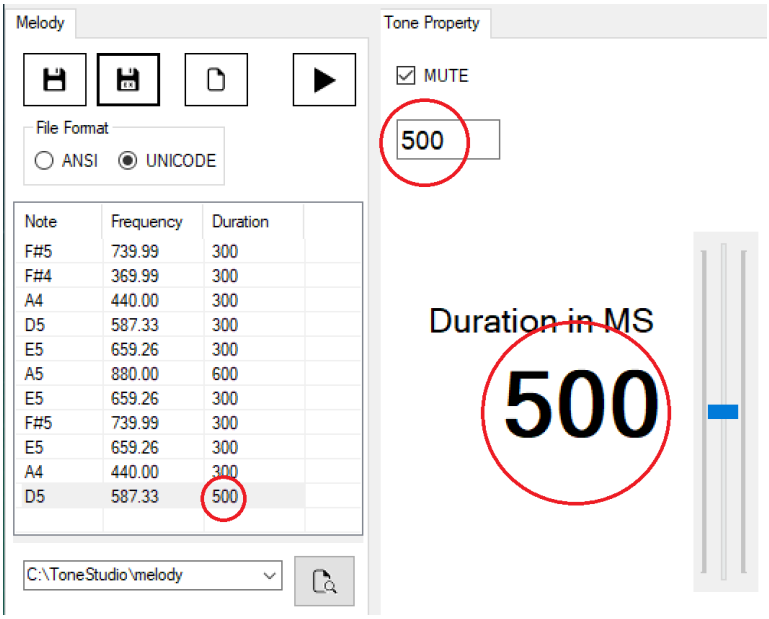


Figure 6: Adjusting the note duration with Duration Traker.

8. Now, you have all the required chords added and you can adjust the Duration of each individual chord if you want to.

9. To adjust a tone duration, left click on one of the note at the notes listbox.

10. Drag the Duration tracker at the "Tone Property" panel such as below

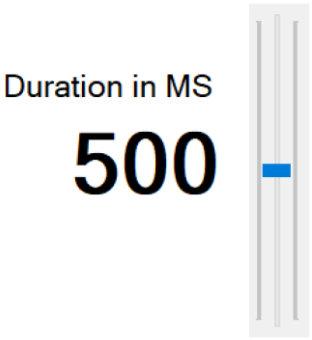


Figure 7: The Duration Tracker.

11. You will notice the Duration under the notes listbox and the Duration textbox will change as shown on Figure 6.

12. Now, you click on the ▶ button to preview the melody.

13. Once the melody is finalized and confirmed, you may save this composed melody which will be described on the next page.

How to save the composed melody ?

Diagrams	Descriptions & References

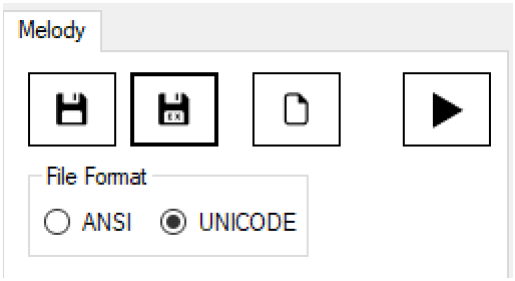


Figure 8: The Melody Panel

```
#ifndef The_Twillight_Zone_H_
#define The_Twillight_Zone_H_
#include "pitch.h"
_MELODY The_Twillight_Zone[]=
{
{523.25, 128},{554.37, 128},{587.33, 128},
{1567.98, 128},{1479.98, 128},{415.30, 128},
{440.00, 128},{466.16, 128}
};
#endif // The_Twillight_Zone_H_
```

Figure 9: The .m file extension format.

```
#ifndef The_Twillight_Zone_H_
#define The_Twillight_Zone_H_
#include "pitch.h"
_MELODYEX The_Twillight_Zone[]=
{
{"C5", 523.25, 128},{"C#5", 554.37, 128},{"D5", 587.33, 128},
{"G6", 1567.98, 128},{"F#6", 1479.98, 128},{"G#4", 415.30, 128},
{"A4", 440.00, 128},{"A#4", 466.16, 128}
};
#endif // The_Twillight_Zone_H_
```

Figure 10: The .mex file extension format.

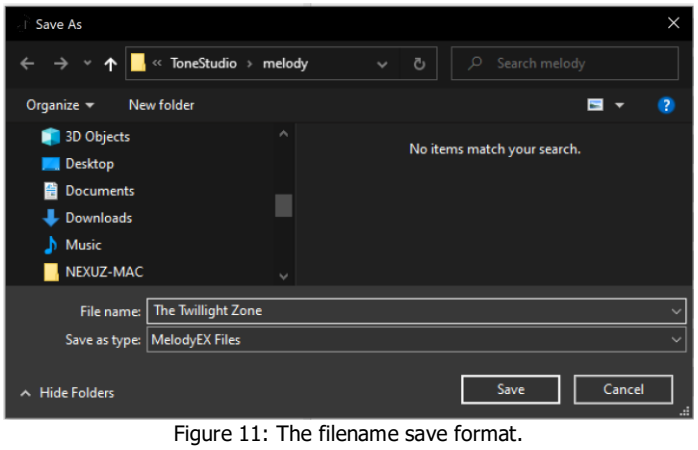


Figure 11: The filename save format.

14. Tone Studio software allows to save the composed melody into C/C++ header format.

15. If to compile with Arduino sketch, click on the "ANSI" file format option before file exporting.

16. If to save the filename under other languages, click on the UNICODE file format before file exporting.

17. There are two file extensions:

    .m extension

    only Frequency and Duration data are saved.

    .mex extension

    Chord string, Frequency and Duration are saved.

18. Figure 12 shown the way to save the filename format as "filename", the Tone Studio software will auto append "**.mex.h**" to the filename and the final filename will be:

    filename.**mex.h**

19. Here are the files saved under \melody folder:

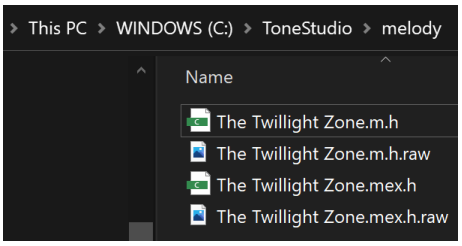
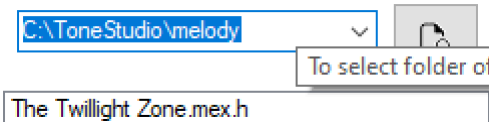
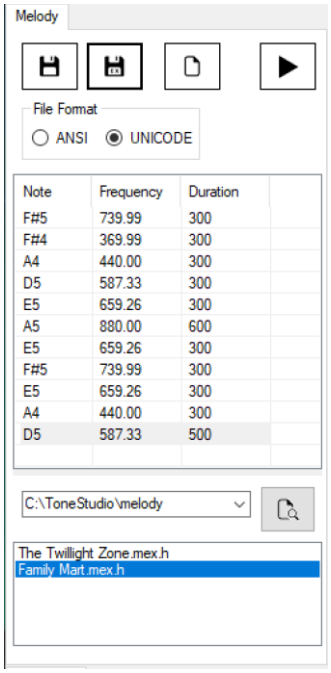


Figure 12: The melody folder saved files.

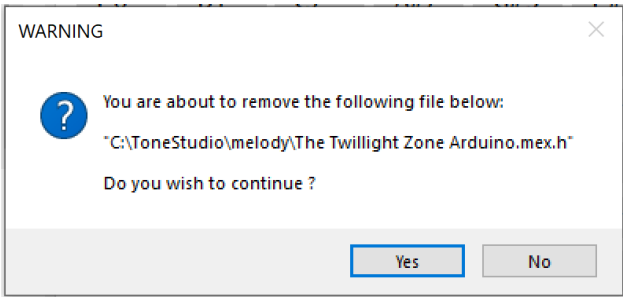
20. The file with extension ".raw" is generated automatically so Tone Studio software can directly load the contents with reference to ".mex.h" filename into the notes listbox.

21. Next, I will show you how to load the saved melody file (where internally the Tone Studio is loading the .raw file).

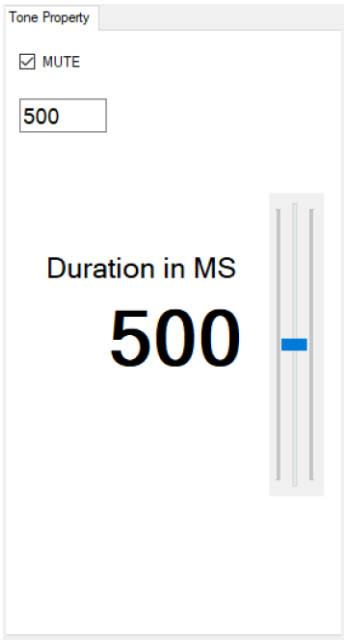
How to open the saved melody files ?

Diagrams	Descriptions & References
 Figure 13: Selecting the default "melody" folder from the combo box.  Figure 14: Left click the file to load the saved melody notes to notes listbox.	22. When the Tone Studio software is first time running, it will create a default folder and listed under the combo box as shown on Figure 14. 23. Click on the combo box and select the listed folder path. In this case, it is "C:\ToneStudio\melody". 24. This will auto list all ".mex.h" melody files found on that folder to the listbox as shown on Figure 14. 25. To open the saved melody file, in this case where the melody files are: The Twillight Zone.mex.h The Twillight Zone Arduino.mex.h 26. Left click on the "The Twillight Zone Arduino.mex.h", it will open and load the saved contents into the notes listbox as shown on Figure 14. 27. To save to other folder location, click the  button and browse for the targeted folder. 28. Once the new targeted folder is selected, it will auto add the new folder to the melody path combo box and auto load all saved folders on next startup.

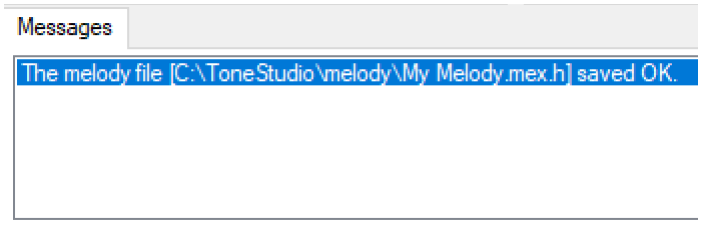
How to delete or remove unwanted melody files ?

Diagrams	Descriptions & References
 Figure 15: Confirmation dialog to delete a melody file.	29. To remove unwanted melody file, right click on the target file listed on the melody file listbox. 30. This will pop up the confirmation dialog to delete the file permanenly such as Figure 15.

The Tone Property (Duration) control panel

Diagrams	Descriptions & References
 Figure 16: The Tone Property or Note Duration adjust panel.	31. As explained earlier, the tone duration tracker provides two functions: • Adjusting the selected note from the notes listbox. • Adjusting the selected chord button to preview before adding to the notes listbox for melody composing purpose. 32. To adjust the duration tracker faster without tone sound, click on the "MUTE" checkbox. 33. Alternatively, you may key in the duration value directly. This will auto update the duration tracker and the selected note from the notes listbox.

The Messages Display Panel

Diagrams	Descriptions & References
 Figure 17: The Message Display Panel.	34. The messages display panel is to display the Tone Studio software internal process result as shown on the Figure ? where the new file "My Melody" is saved as "My Melody.mex.h" in this case. 35. DONE.

Programming Guidelines

How to compile Tone Studio generated melody header ?

Tone Studio C/C++ melody header demo source - tone.ino

```
#include "pitch.h"
#include "The Twillight Zone.mex.h"

#define melody The_Twillight_Zone

const int max = sizeof( melody ) / sizeof( _MELODYEX );
int n {};
char buf[100]{};

void setup()
{
}

const int speaker = 5;
const int active_buzzer = 6;
const int passive_buzzer = 9;
int pin {};
int cnt{};
int spk {-1};

void loop()
{
   unsigned int frequency {};
   unsigned long duration {};

   if( spk + 1 < 3 )
     spk++;
   else
     spk = 0;

   switch( spk )
   {
   case 0: // 8 ohm speaker
      pin = speaker;
      break;

   case 1: // active buzzer
      pin = active_buzzer;
      break;

   case 2:
      pin = passive_buzzer;
   }

   for( n=0; n<max; n++ )
   {
      frequency = (unsigned int)melody[n].frequency;
      duration = (unsigned long)melody[n].duration;

      tone( pin, frequency );
      delay( duration );
      noTone( pin );
   }

   delay( 3000 );
}
```

Descriptions & References

The Tone Studio software sample source code tone.ino is available at GitHub.

Here, we use Arduino IDE to compile with the melody header such as:

The Twillight Zone.mex.h

Source code on the left is the simplified version for easy understanding.

The function is to playback the tone melody "**The_Twillight_Zone**" through out the Arduino Nano pin 5,6,9 from the flowing sequence repeatedly:

8Ω 2W speaker -> Active Buzzer -> Passive Buzzer

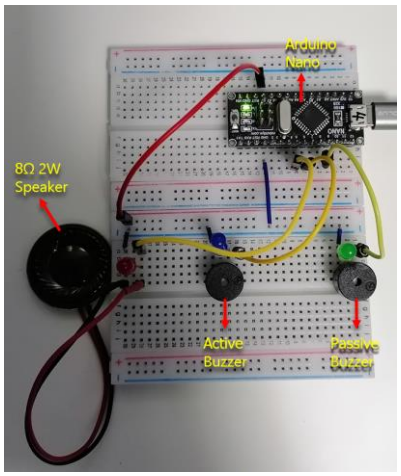


Figure 18: The Tone Studio Arduino tone project.

The file "**pitch.h**" contains the followings:

```
typedef struct
{
   float frequency;
   int   duration;
}
_MELODY;

typedef struct
{
   char  chord[5];
   float frequency;
   int   duration;
}
_MELODYEX;
```

To save memory without chord string, you may use _MELODY but for display or debug purpose, we use _MELODYEX to show the chord string.

Thus, "**The Twillight Zone.mex.h**" was adopted in this section.

