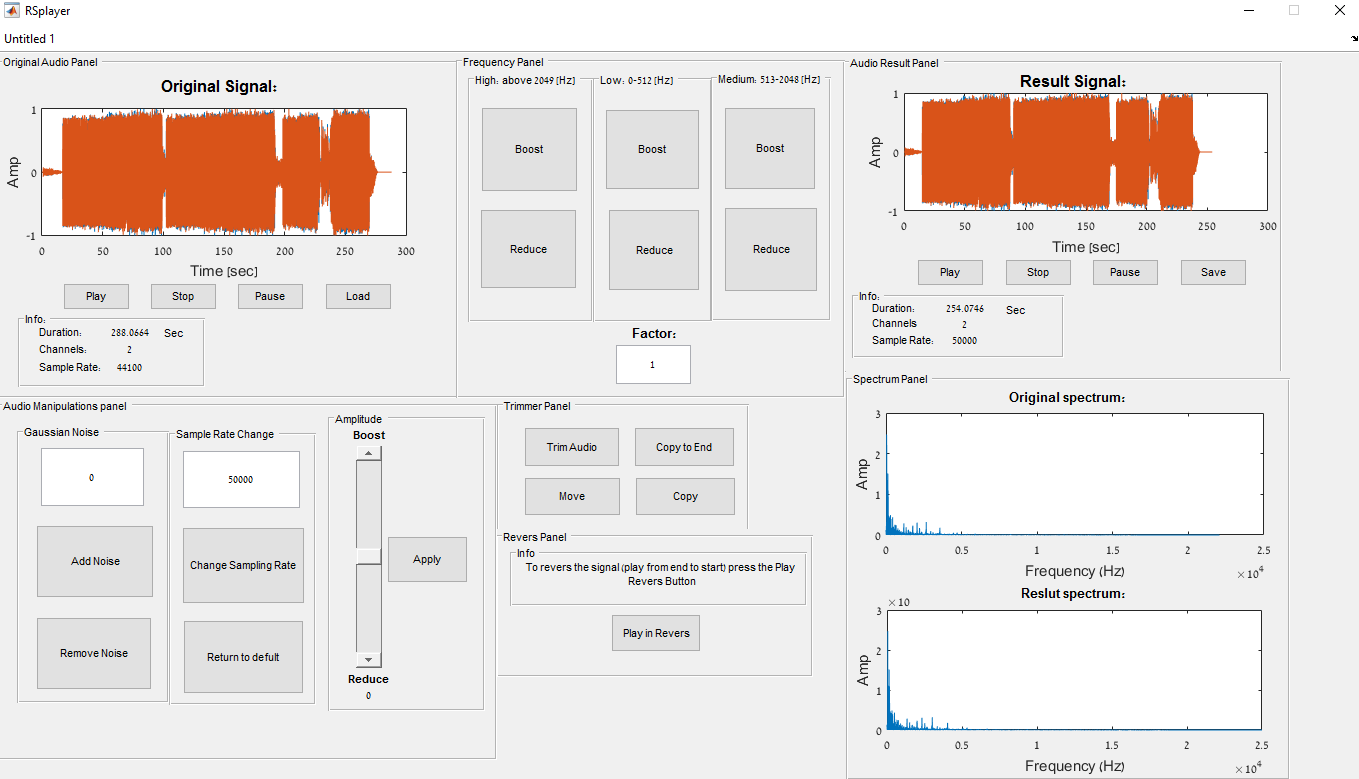
**RSplayer Info:**

**GUI screenshot:**  
  
  
**Note:** after every option that the user selects, the result will be displayed in the result Panel via: the audio graph, some info about the new audio file such as: the Duration, the number of channels (stereo vs mono) and the sample rate (Fs) and the user will have the options to: Play the new audio file, Pause the new audio file, Stop the new audio file and to Save the new audio file with a the name he choose and the location he choose for the new file.  
The new audio file will also be presented in a Spectrum graph.

1. **Loading, Playing, Pausing and Stopping the Original audio file:**we cannot do manipulation on the original audio file so the audio file after each manipulation is saved in the variable: handles.data.\*\*\*Res  
   The original audio file has the following criteria:

|  |  |
| --- | --- |
| Loading a new audio file to the program | Load |
| Playing the selected (original or result) audio file | Play |
| Stopping the selected (original or result) audio file | Stop |
| Pausing the selected (original or result) audio file | Pause |
| Displaying the original or result audio file Sample Rate (Fs) | Sample rate |
| Displaying the original or result audio file duration (in seconds) | Duration |
| Displaying how many channels the original or result audio file has (stereo vs mono) | Channels |
| Saving the new audio file (the result audio file) | Save |

1. **Adding a Gaussian noise to the audio file (the noise factor is via the user selection):**The user can add Gaussian noise to the audio file with a factor which he selects.  
    **Note:** the noise function has a pretest to check if the user has entered a noise factor. If he hasn't entered a noise factor => an error message will pop stating that a noise factor has not been entered and will exit the function without adding noise to the audio file.  
   Otherwise => the noise (with the noise factor) will be added to the audio file.  
     
   The pretest will use a predefined variable that has the Boolean value of True and will change its value to False when a noise Factor is entered.
2. **Filtering the audio file from the Gaussian noise:**We will use a lowpass Butter filter. Since we cannot remove all the Gaussian noise from the audio file, we will filter the High frequencies of the audio file because the noise is most noticeable in those frequencies. So we want to filter those frequencies and avoid changing the low frequencies.
3. **Boosting and reducing the low, medium and high frequencies (by a factor from the user):**We used an fft convolution to switch to the frequency domain. After we did that, we used an fftshift function to organize the audio file frequencies such that the zero frequencies will be at the middle.  
     
   Then the user insert the boost of reduce factor and then the user can boost or reduce the **frequencies of the audio file in the 3 ranges listed below:**

|  |  |  |
| --- | --- | --- |
| Basses and depth frequencies | 0-512 | Low [ Hz] |
| Normal audio files hearing frequencies range | 513-2048 | Medium [Hz] |
| Hearing pain threshold | 2049 and above | High [Hz] |

We used the information located in lecture 7 in the course material

1. **Changing the Sample Rate (Fs) of the audio signal:**The user can change the Sample Rate (Fs) of the audio file by a factor he choose.  
   The user enter a Sample Rate factor and then press the button to change the audio file sample rate.  
     
   The user will have the option to return the audio file sample rate to its original default sample rate.
2. **Trimming a part of the signal via the user selection:**The user can trim a part of the signal by selecting a starting point and an ending point for the trim part. **The trimming action is preformed according to this steps:**
   1. After the user press the Trim Audio button. A new graph is displayed with the original audio file. Then the user asked to select the new trim audio file starting point.
   2. With the ginput function, the user select the new starting point for the trim audio file.
   3. The user is asked to select the new trim audio file end point.
   4. With the ginput function, the user select the new end point for the trim audio file.
   5. The new trim audio file in displayed via a graph with all its new data in the result Panel.

Note: the ginput function returns 2 values: the X coordinates and the Y coordinates.  
We only need the X coordinates so we will write the ginput function as follow:  
where the sign ~ means that the value that is given in his location is not saved (it’s the Y coordinate which we don't need).

1. **Copying a part of the audio file and placing it at the end of the signal (Copy To End):**The user can select a part of the signal (as shown in section 6) and the selected part will be located at the end of the audio signal.  
   As a result, the length of the new audio signal will increase (due to the selected part that been added to the end of the signal).
2. **Copying a part of the audio file and placing it at a selected position in the signal (Copy To End):**The user can select a part of the signal (as shown in section 6) and then inserting it at a selected position in the audio signal.  
     
   **Note:** the section of the signal which the selected part is placed will be removed during the insertion part of the function. The new audio signal will have the same length as the original signal.
3. **Move a part of the audio file to a position in the signal:**The user can select a part of the signal (as shown in section 6) and then moving it to a selected location which the user select in the signal.  
     
   **Note:** in this option the selected part is moved to a different place in the signal but it is not been removed. Thus the length of the new audio signal is not been change.
4. **Amplification and attenuation of the signal:**We used a slider to select the volume factor of the signal ( in the range: 0÷2).  
   The default volume factor is set to 1 and the user can select with the slider, a new volume factor in the defined range.  
   Then the user press the Apply button and then select if he wants to apply the new volume to the original signal or to apply it to the result signal.
5. **Additional function of our selection:**In this function the user can reverse the original signal such that the new signal will be the original signal from back to front.  
   The reversed signal will be displayed in the result panel via a graph and the signal information. It will also be displayed in the result spectrum graph.