

Documentación tarea 2, elo 330

Pascal Sigel - Oscar Silva

Oct-10-2014

1 Introducción:

En el presente documento se comentarán los hechos más importantes de la tarea, se hará un análisis de la solución propuesta y además se explicarán los problemas que tomaron más tiempo y se explicará cómo fueron solucionados.

En la siguiente sección se mostrará un bosquejo de la solución que se propone.

2 Objetivo:

El Objetivo es crear un corrector de saturación de audio a través de polinomios de orden 4.

3 Solución:

Para simplificar el entendimiento del código éste fue dividido en los siguientes bloques de forma secuencial:

1. **Comprobacion de los datos de entrada:** Se analiza si la entrada con que el usuario ejecutó el programa es válida.
2. **CreacionOctave:** Se crea el proceso que representa a octave para su uso posterior en la graficación de las señales.
3. **ObtencionDatos:** Se obtienen los datos del archivo de audio para su posterior tratamiento
4. **suavizamiento de Saturacion:** Se suaviza la saturación de la señal y se guarda en un arreglo diferente.
5. **generacionGraficos:** Se crean los gráficos a través de octave.
6. **calculoDeError:** Se calcula el error entre la señal entrante y la suavizada.
7. **reproducir:** Si el usuario entró el parámetro p entonces se reproducen los audios.
8. **protocolo de Salida:** Se libera la memoria utilizada y se espera a que el usuario aprete enter.

4 Problemas que existieron:

En esta sección se expondrán los problemas que surgieron y sus soluciones.

Lectura y escritura binaria:

El archivo que es propuesto está en formato .raw el cual debe ser leído en forma binaria, este simple problema puede tomar mucho tiempo si uno no sabe.

Uso de int16_t:

Para lograr usar enteros de 16 bit se debe aprender a leerlos y a escribirlos, a diferencia de los otros tipos, éste tipo es diferente y debe ser tratado diferentemente. A continuación se muestra cómo debe ser escrito y leído un dato de tipo int16_t respectivamente:

```
printf("%" PRIu16, (int16_t) dato);  
scanf("%" SCNu16, (int16_t*)&dato);
```