

SCRIPT CONTROL SERVER.PY

la última versión en:
<https://github.com/AudioHumLab/FIRtro>

Índice de contenido

1 - Descripción.....	2
2 - Archivos de configuración.....	2
3 - Control de ganancias.....	2
3.1 - Comandos.....	3
4 - Ecualización de “preamplificador”	3
4.1 - Control de loudness.....	4
4.2 - Ecualización de sistema (room target curves).....	5
4.3 - Control de graves/agudos.....	5
4.4 - Cambios de Fs y de referencia de reloj en tarjetas de sonido “pro”	5
4.5 - Comandos.....	6
5 - Resto de funciones (a expandir).....	6

1 Descripción

El archivo server.py es un script de python, cuyo objetivo es controlar las funciones principales de audio. Una vez arrancado el programa se queda en segundo plano a la espera de los comandos que reciba mediante una conexión tipo socket, que puede ser establecida localmente o desde otro equipo de la lan. Se disponen de dos clientes que facilitan esta tarea:

client.py 'comando parametros'
control 'comando parametros'

Ambos hacen lo mismo, la diferencia es que control esta basado en la utilidad "netcat" y su ejecución es más rápida que invocar una instancia de python

Una vez procesada la orden, el server devuelve una lista formateada en JSON con las variables del sistema.

El conjunto se divide en una serie de modulos:

server.py:	Programa principal que escucha el puerto especificado en la configuración
server_input.py:	Realiza los cambios de entradas en jack
server_process:	Ejecuta las ordenes recibidas por el server
server_lcdproc:	Comunicación con el servidor lcdproc para manejar un display
peq_control:	Gestión del eq paramétrico
presets	Gestión de presets de configuración de altavoces definidos por el usuario

2 Archivos de configuración

- \$HOME/audio/config: Almacena los ajustes de funcionamiento del sistema. En particular loudspeaker almacena el nombre del directorio que aloja las configuraciones de altavoz.
- \$HOME/audio/status: Almacena los ajustes de estado de audio: ecualización, ganancia, entradas y filtrado.
- \$HOME/audio/inputs: Almacena los ajustes de las entradas: puertos de entrada, xover, ganancia...
- \$HOME\spk\<nombre_altavoz>\speaker: Almacena los ajustes de cada altavoz(curvas target, ref_level de calibrado)
- \$HOME\spk\<nombre_altavoz>\presets.ini: Almacena los presets de usuario (distintos filtrados de altavoz)

3 Control de ganancias

Vamos a trabajar con dos conceptos, gain y level. A modo de resumen:

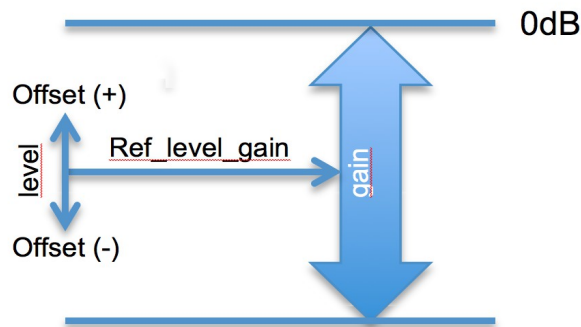
- **gain:** Es la atenuación real que se introduce a la señal. De modo que 0 es no atenuación.
- **ref_level_gain:** Es el valor de ganancia para un spl conocido en punto de escucha, y para el cual no es necesario corrección de loudness (Pendiente estudiar proceso de cómo calcular este valor). De este modo calibramos el sistema.

```
#speaker
[calibration]
ref_level_gain = -10
```

- **level:** Es el offset respecto a ref_level_gain, y que usaremos para establecer el volumen del sistema. **La idea es hacer aritmética solo en el dominio de level, que es la ganancia normalizada, y que sea el programa el que traduzca esos levels a gain real, que depende de la calibración concreta de cada instalación.**

```
#status
[level]
level = 6.0
```

En este caso estaríamos estableciendo un valor de 6dB por encima del nivel de referencia



- **input_gain:** Ganancia específica para cada entrada

```
#inputs
[a1]
in_ports:  system:capture_1 system:capture_2
clock:      card
gain:       0
xo:         lp
```

- **maxlevel:** El valor máximo de volumen que se puede establecer hasta llegar a los cero dB

$\text{maxlevel} = \text{ceil}(-\text{input_gain} - \text{ref_level_gain})$

En este ejemplo, si $\text{ref_level_gain} = -10$ e $\text{input_gain} = 0$ implica que $\text{maxlevel} = 10$. Efectivamente si mi nivel de referencia está 10dB por debajo del 0, no puedo tener un nivel (offset) mayor que 10, o tendría amplificación digital

¿Cómo se calcula la ganancia final?

Pues se puede cambiar directamente su valor (comando gain) hasta un máximo de cero (siempre que exista suficiente headroom), o mediante incrementos de level:

$$\text{gain} = \text{level} + \text{input_gain} + \text{ref_level_gain}$$

3.1 Comandos

control comando parametros

Comando	Parametros	Descripción
level	Valor float	Nivel calibrado en dB (relativo a ref_level_gain). Puede ser positivo o negativo.
level_add	Valor float	Incremento del nivel calibrado en dB. Puede ser positivo o negativo.
gain	Valor float	Ganancia en dB. Siempre negativo.
gain_add	Valor float	Incremento de la ganancia en dB. Puede ser positivo o negativo.
mute		Silencia los altavoces
unmute		Anula el silencio de los altavoces
toggle		Alterna entre mute y unmute
input_gain	Valor float	Ganancia de la entrada actual.

4 Ecualización de “preamplificador”

El ecualizador comprende las siguientes secciones:

- Loudness
- Ecualización de sistema (system_eq i.e. room target curves)
- Graves
- Agudos

El script usa el EQ que trae integrado brutefir para aplicar estas correcciones. Se usan las variables eq_mag y eq pha para en cada comando que se envíe ir calculando la EQ final, que será la suma de aquellas secciones que tengamos activas. Si al final del cálculo hay headroom suficiente (para evitar amplificación digital) se envía el resultado al EQ de brutefir, que lo renderiza en tiempo real.

```
eq_level = max(eq_mag)
level_headroom = -(level + input_gain + ref_level_gain)
si eq_level <= level_headroom se aplican los cambios
```

4.1 Control de loudness

Fichero de configuración:

```
#config
[equalizer]
;Nombres del archivo de frecuencias, curvas de loudness, tonos
frequencies = R20_ext-freq.dat
loudness_mag_curves = R20_ext-loudness_mag.dat
loudness pha_curves = R20_ext-loudness pha.dat
; Salto entre curvas de ecualización (dB)
; Ojo: No está implementado en el programa de control.
; Se da como indicación que debe asumirse.
step = 1
loudness_SPLref = 83
loudness_SPLmax = 90
loudness_SPLmin = 70
loudness_variation = 10
no_phase_xo = lp
```

En él se definen los archivos de frecuencias y curvas de loudness, a la vez que se establecen las referencias de SPL.

En este caso vemos que hay definidas 21 curvas de corrección (0-20), que corresponden desde los 90dB hasta los 70dB, siendo 83 la referencia donde la corrección es cero (que corresponde con level=0, según lo visto en el apartado anterior). Estos valores están registrados en los ficheros loudness_mag_curves y loudness pha_curves. Cada columna representa un valor de SPL (desde 90dB hasta 70db) y cada fila es un valor de frecuencia, cuya cabecera que se encuentra definida en el fichero frequencies.

Conceptos que se manejan:

- **loudness_ref**: Diferencia con el nivel de referencia, en dB.
- **loudness_variation**: Máximo offset que permitimos al parámetro loudness_ref
- **loudness_i**: es el índice que indica que columna del fichero loudness_mag_curves y loudness pha_curves se debe aplicar

$$\text{loudness_i} = \text{loudness_SPLmax} - (\text{level} + \text{loudness_SPLref} + \text{loudness_ref})$$

Por ejemplo, si suponemos que estamos en el nivel de referencia (level=0) y no hemos alterado el punto de referencia del loudness (loudness_ref=0)

$$\text{loudness_i} = 90 - (0+83+0) = 7$$

Si observamos el fichero loudness_mag_curves, vemos que columna 7 es la de valores cero, correspondiente a los 83dB de referencia. Cualquier incremento de level o loudness_ref mueve el índice para la izquierda, y si se decrementa, para la derecha.

- **loudness_max_i**: Son los límites del índice

$$\text{loudness_max_i} = \text{loudness_SPLmax} - \text{loudness_SPLmin}$$
$$\text{loudness_max_i} = 90 - 70 = 20 \text{ que es la última columna del fichero de curvas}$$

Si se manejan niveles que salgan fuera de los límites, se aplicará la curva máxima o mínima según sea el caso. Por ejemplo, si establezco level=-40

$$\text{loudness_i} = 90 - (-40+83+0) = 47$$

Como está fuera de rango, se aplicaría la curva 20

En resumen, la operativa es que si `loudness_track=true`, se calcula la curva correspondiente en función del level aplicado. Mediante el parámetro `loudness_ref` podemos variar la curva calculada (por ejemplo por variaciones de nivel de distintos programas musicales) en pasos de 1dB.

4.2 Ecualización de sistema (room target curves)

Los parámetros de la ecualización del sistema se define en el fichero de configuración del altavoz:

```
#speaker
[equalization]
system_eq = True
room_gain = 6
house_corner = 500
house_atten = 6
```

Esta ecualización es almacenada en el fichero

```
#config
[equalizer]
;Nombres del archivo de frecuencias, curvas de loudness, tonos
frequencies = R20_ext-freq.dat
syseq_mag_curve = R20-syseq_pha.dat
syseq_pha_curve = R20-syseq_pha.dat
no_phase_xo = lp
```

Y para calcularla se usa el script `syseq.py`, que usa las funciones definidas en `curves.py`. Este proceso se tiene que hacer la primera vez, o cuando los parámetros de la ecualización sean alterados.

Con `system_eq = True` o `system_eq = False` decidimos si queremos aplicar esta corrección o no.

Al cambiar de altavoz se carga el target apropiado. En cada arranque se calculan al vuelo las curvas `syseq`, o ecualización de sistema, y que son el target de ecualización general en sala, combinación del `room gain` y la `house curve`. Se hace efectivo el ajuste de dicha curva definido para cada altavoz en el archivo `~/lspk/<nom_altavoz>/speaker`.

Para el cambio al vuelo ya estaba escrita la rutina `syseq.py`, y simplemente se ha modificado `/usr/local/bin/initfirtro.py` para que lo ejecute en cada arranque. Ello ha arrastrado modificaciones de `/usr/local/bin/getconfig.py` y de `~/audio/config`.

4.3 Control de graves/agudos

Fichero de configuración:

```
#config
[equalizer]
;Nombres del archivo de frecuencias, curvas de loudness, tonos
frequencies = R20_ext-freq.dat
treble_mag_curves = R20_ext-treble_mag.dat
treble_pha_curves = R20_ext-treble_pha.dat
bass_mag_curves = R20_ext-bass_mag.dat
bass_pha_curves = R20_ext-bass_pha.dat
; Salto entre curvas de ecualización (dB)
; Ojo: No está implementado en el programa de control.
; Se da como indicación que debe asumirse.
step = 1
tone_variation = 6
no_phase_xo = lp
```

Al igual que antes, los valores se encuentran almacenados en los ficheros correspondientes. `tone_variation = 6` especifica el valor máximo del control, en este caso 6dB

4.4 Cambios de Fs y de referencia de reloj en tarjetas de sonido “pro”

Los cambios de Fs de FIRtro puede ser debidos a:

- ✓ La selección de una input que lo requiera implícitamente (opciones fs y clock en ~/audio/inputs)
- ✓ Un comando explícito `control clock nuevaFs [nuevaRef]`

La nuevaRef puede ser “card” o “spdif”

4.5 Comandos

`control comando parametros`

Comando	Parametros	Descripcion
loudness_ref	Valor float	Diferencia con el nivel de referencia, en dB. Puede ser positivo o negativo.
loudness_add	Valor float	Incremento de la diferencia con el nivel de referencia, en dB. Puede ser positivo o negativo.
loudness_track		Activa el seguimiento de loudness
loudness_track_off		Desactiva el seguimiento de loudness
bass	Valor float	Ganancia de graves, en dB (max 6dB).
bass_add	Valor float	Incremento de la ganancia de graves, en dB. Puede ser positivo o negativo.
treble	Valor float	Ganancia de agudos, en dB (max 6dB).
treble_add	Valor float	Incremento de la ganancia de agudos, en dB. Puede ser positivo o negativo.
flat		Ecualización plana. No escribe el archivo de configuración.
syseq		Activa la ecualización del sistema
syseq_off		Desactiva la ecualización del sistema
drc	Valor entero	Carga el conjunto de FIRs para drc, correspondientes a los drc-X de la carpeta del altavoz. 0 ajusta la etapa drc sin filtrado.
peq_defeat		Desactiva el eq paramétrico
peq_reload		Carga el eq paramétrico del preset actual
preset	etiqueta	Configura el altavoz con un preset de filtrado definido por el usuario (filtros de cruce, drc fir, eq parametrico, balance)
config		Restaura los valores almacenados en el archivo status
xover	lp mp toggle	Cambia el tipo de filtrado de cruce de vías (fase lineal - fase mínima)
input	etiqueta	Selecciona una de las entradas disponibles y ajusta su ganancia y su tipo de filtro de cruce de vías (lp/mp)
restore		Restaura la ultima entrada almacenada en status, la ganancia y el tipo de filtro asociado
mono	on off toggle	Conmutador mono/stereo
clock	fs(float) [clockRef(string)]	Fuerza el cambio de Fs en el sistema, con reinicio del audio y con la posibilidad de reconfigurar la referencia de reloj de una tarjeta “pro” que lo permita.

5 Resto de funciones (a expandir)

Comando	Parametros	Descripción
polarity	“+” o “-”	Cambia la polaridad (pendiente revisar)
exit		Salida del script (pendiente revisar)