```
1 void calibrador(bool calibrarLineas = false, bool sinConfirmar = false)
 2 {
 3
       int ctr = 0;
       Spectrum avgMic, avgLeft, avgRight;
 4
 5
       EstadoDelLimitador status;
 6
       Calibracion cal;
 7
 8
       cal.leeCalibracion(1);
       printf("Line 1\n");
 9
10
       cal.print();
11
12
       cal.leeCalibracion(2);
13
       printf("Line 2\n");
14
       cal.print();
15
16
17
18
       while (true)
19
       {
20
           sleep(1);
21
22
           status.actualiza(); //Si el sistema no está funcionando sale
23
24
           avgMic.sum(status.mic);
25
           avgLeft.sum(status.left);
26
           avgRight.sum(status.right);
27
28
           printf(" . ");
29
           ctr++;
30
31
           status.mic.show();
           status.left.showStereo(status.right);
32
33
34
           printf("Mic : %8.1f dBA\tLeft : %8.1f dBA\tRight : %8.1f\n",
35
               status.mic.globalAWeighted,
36
               status.left.globalAWeighted,
37
               status.right.globalAWeighted
38
           );
39
40
           if (ctr >= 10)
41
               break;
42
       }
43
44
       if (calibrarLineas)
45
       {
           puts("Lineas de entrada calibradas");
46
47
48
           . . .
49
50
           avgMic.divide(ctr);
51
           avgLeft.divide(ctr);
52
           avgRight.divide(ctr);
53
           //Guardar calibracion izquierda
54
55
           cal.dBRef = avgMic.dB[Spectrum 1Khz BandIndex];
56
           cal.ref = avgLeft.energy[Spectrum_1Khz_BandIndex];
57
           cal.ruido = 0:
58
           for (int i = 0; i < Spectrum BandCount; i++)</pre>
59
               cal.equalization[i] = 0; //avgMic.dB[i]-avgLeft[i]
           cal.guardaCalibracion(1);
```

localhost:4649/?mode=clike

```
10/5/2021
                                            ServidorSerie .cpp
61
62
            cal.print();
63
64
            //Guardar calibracion derecha
            cal.dBRef = avgMic.dB[Spectrum 1Khz BandIndex];
65
            cal.ref = avgRight.energy[Spectrum_IKhz_BandIndex];
66
            cal.ruido = 0;
67
            for (int i = 0; i < Spectrum_BandCount; i++)</pre>
68
                cal.equalization[i] = 0;
69
70
            cal.guardaCalibracion(2);
71
            cal.print();
72
73
        }
74 }
```

localhost:4649/?mode=clike 2/2