

```
1 #define HwDataReadTimesPerSec 10
2
3 /**
4  * @brief Lee los datos de micrófono, líneas y los niveles de hardware
5  * @param count número de muestras a tomar
6  * @param mic espectro de onda del micrófono
7  * @param left espectro de onda de la linea izquierda
8  * @param right espectro de onda de la linea derecha
9  * @param lLevel nivel de hardware. Canal izquierdo
10 * @param rLevel nivel de hardware. Canal derecho
11 * @param status objeto de acceso a memoria compartida
12 * @return (void)
13 */
14 void readAudioData(int count, Spectrum &mic, Spectrum &left, Spectrum &right,
15                   float &lLevel, float &rLevel, HardwareStatus *status)
16 {
17     ChannelInformation m, m2, l, r;
18     Spectrum mAc, lAc, rAc;
19
20     for (int i = 0; i < count; i++)
21     {
22         for (int j = 0; j < HwDataReadTimesPerSec; j++)
23         {
24             usleep(UnSegundo / HwDataReadTimesPerSec);
25             lLevel += status->inputPressure[0];
26             rLevel += status->inputPressure[1];
27         };
28         loadSlowChannels(m, m2, l, r);
29         mAc.sum(m.spectrum);
30         lAc.sum(l.spectrum);
31         rAc.sum(r.spectrum);
32     };
33
34     mAc.divide(count);
35     lAc.divide(count);
36     rAc.divide(count);
37
38     mic = mAc;
39     left = lAc;
40     right = rAc;
41
42     lLevel /= count * HwDataReadTimesPerSec;
43     rLevel /= count * HwDataReadTimesPerSec;
44 };
45
46 /**
47  * @brief Carga las calibraciones
48  * @param micCal
49  * @param leftCal
50  * @param rightCal
51  * @return (void)
52 */
53 void loadCalibrations(Calibracion &micCal, Calibracion &leftCal, Calibracion
&rightCal)
54 {
55     micCal.leeCalibracion(0);
56     leftCal.leeCalibracion(1);
57     rightCal.leeCalibracion(2);
58 };
59
```

```
60 /**
61  * @brief Guarda las calibraciones
62  * @param leftCal
63  * @param rightCal
64  * @return (void)
65  */
66 void saveCalibrations(Calibracion &leftCal, Calibracion &rightCal)
67 {
68     leftCal.guardaCalibracion(1);
69     rightCal.guardaCalibracion(2);
70 }
71
72 /**
73  * @brief Programa principal
74  * @param argc
75  * @param argv
76  * @return
77  */
78 int main(int argc, char **argv)
79 {
80     Spectrum mic, left, right;
81     Calibracion micCal, leftCal, rightCal;
82     HardwareStatus *status;
83     float lLevel, rLevel;
84     float micBandValue;
85
86     ...
87
88     loadCalibrations(micCal, leftCal, rightCal);
89     int count = 10;
90
91     ...
92
93     printf("----- CALIBRADOR -----\\n");
94     printf("----- Leyendo Lineas -----\\n");
95     readAudioData(count, mic, left, right, lLevel, rLevel, status);
96
97     float Mic1K = mic.dB[Spectrum_1Khz_BandIndex] -
micCal.internalEqualization[Spectrum_1Khz_BandIndex];
98
99     // Referencia
100    leftCal.dBRef = Mic1K;
101    leftCal.ref = left.energy[Spectrum_1Khz_BandIndex];
102
103    rightCal.dBRef = Mic1K;
104    rightCal.ref = right.energy[Spectrum_1Khz_BandIndex];
105
106    // Limpiar ecualización interna
107    for (int i = 0; i < Spectrum_BandCount; i++)
108        leftCal.internalEqualization[i] = rightCal.internalEqualization[i] =
0;
109
110    // Ajustamos la banda de 1K como central
111    left.calibrate(leftCal);
112    right.calibrate(rightCal);
113
114    // Calculamos la ecualización interna
115    for (int i = 0; i < Spectrum_BandCount; i++)
116    {
117        micBandValue = mic.dB[i] - micCal.internalEqualization[i];
```

```
118         leftCal.internalEqualization[i] = micBandValue - left.dB[i];
119         rightCal.internalEqualization[i] = micBandValue - right.dB[i];
120     }
121
122     ...
123
124     // Finishing
125     printf("----- Linea Left -----\\n");
126     leftCal.fullPrint();
127     printf("----- Linea Right -----\\n");
128     rightCal.fullPrint();
129
130     saveCalibrations(leftCal, rightCal);
131     return 0;
132 };
```