

```

/*****
*
*           Librería de control de Altavoz
*
*   Autor:      Ángel Hernández Mejías (Mif)
*   Contacto:   angeldpe@hotmail.com
*   Fecha:      22/08/2007
*   Versión:    1.0
*
*****/

*
*           Licencia
*
*   Esta librería está publicada bajo la licencia Creative
*   Commons en la versión de Attribution and Share Alike. Es decir,
*   se permite copiar, modificar y distribuir este material
*   siempre que se reconozca al autor y se mantenga la misma
*   licencia.
*
*****/

*
*   Esta librería sirve para Manejar de un modo sencillo y limpio
*   un altavoz típico de PC generando notas musicales "afinadas"
*   así como algunas melodías sencillas.
*
*   Es recomendable usar el sistema Skypic para evitar problemas
*   en la electrónica al hacer montajes básicos.
*
*   Equivalencia entre Notas y Frecuencias
*
* -----
* | Nombre Do  Do# Re  Re# Mi  Fa  Fa# Sol Sol# La  Si b Si  Do  |
* | Hz        261 277 294 311 330 349 370 392 415  440 466  494 523 |
* -----
*
* Requisitos:
* -El Altavoz se colocará entre RB5 y Masa
*
*****/

```

```

#define Altavoz    PIN_B5           //Si se desea Cambiar el altavoz de pin
                                   //deberá modificarse esta línea

```

```

//*** Tono de 1Khz ***

```

```

void Altavoz_1Khz()
{
    long int i;
    for (i=0;i<=1000;++i)
    {
        Output_bit(Altavoz, 1);
        delay_ms(1);
        Output_bit(Altavoz, 0);
        delay_ms(1);
    }
}

```

```

// ***** Galería de Notas *****
// *   A cada Función se le debe pasar un valor de duracion que
// *   estará entre 0 y 255
// *****

```

```

void Altavoz_Do(int Duracion)      // Do es un tono de 261Hz
{
    // 1seg / 261Hz = 3831us
    int i;                          // TH = TL = 3831us / 2 = 1916
    int j;                          // Tiempo de sonido = 261 / 11 = 24
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=24;++i)
        {
            Output_bit(Altavoz, 1);

```

```

        delay_us(1916);
        Output_bit(Altavoz, 0);
        delay_us(1916);
    }
}

void Altavoz_Re(int Duracion) // Re es un tono de 294Hz
{ // 1seg / 294Hz = 3401us
    int i; // TH = TL = 3401us / 2 = 1700
    int j; // Tiempo de sonido = 294Hz / 11 = 27
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=27;++i)
        {
            Output_bit(Altavoz, 1);
            delay_us(1700);
            Output_bit(Altavoz, 0);
            delay_us(1700);
        }
    }
}

void Altavoz_Mi(int Duracion) // Mi es un tono de 330Hz
{ // 1seg / 330Hz = 3030us
    int i; // TH = TL = 3030us / 2 = 1515
    int j; // Tiempo de sonido = 330Hz / 11 = 30
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=30;++i)
        {
            Output_bit(Altavoz, 1);
            delay_us(1515);
            Output_bit(Altavoz, 0);
            delay_us(1515);
        }
    }
}

void Altavoz_Fa(int Duracion) // Fa es un tono de 349Hz
{ // 1seg / 349Hz = 2865us
    int i; // TH = TL = 2865us / 2 = 1433
    int j; // Tiempo de sonido = 349Hz / 11 = 32
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=32;++i)
        {
            Output_bit(Altavoz, 1);
            delay_us(1432);
            Output_bit(Altavoz, 0);
            delay_us(1432);
        }
    }
}

void Altavoz_Sol(int Duracion) // Sol es un tono de 392Hz
{ // 1seg / 392Hz = 2551us
    int i; // TH = TL = 2865us / 2 = 1275
    int j; // Tiempo de sonido = 392Hz / 11 = 36
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=36;++i)
        {
            Output_bit(Altavoz, 1);
            delay_us(1275);
            Output_bit(Altavoz, 0);
            delay_us(1275);
        }
    }
}

```

```

    }
}

void Altavoz_La(int Duracion) // La es un tono de 440Hz
{ // 1seg / 440Hz = 2272us
    int i; // TH = TL = 2272us / 2 = 1136
    int j; // Tiempo de sonido = 440Hz / 11 = 40
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=40;++i)
        {
            Output_bit(Altavoz, 1);
            delay_us(1136);
            Output_bit(Altavoz, 0);
            delay_us(1136);
        }
    }
}

void Altavoz_Si(int Duracion) // Si es un tono de 494Hz
{ // 1seg / 494Hz = 2024us
    int i; // TH = TL = 2024us / 2 = 1012
    int j; // Tiempo de sonido = 494Hz / 11 = 45
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=45;++i)
        {
            Output_bit(Altavoz, 1);
            delay_us(1012);
            Output_bit(Altavoz, 0);
            delay_us(1012);
        }
    }
}

void Altavoz_DoM(int Duracion) // Sol es un tono de 523Hz
{ // 1seg / 523Hz = 1912us
    int i; // TH = TL = 1912us / 2 = 956
    int j; // Tiempo de sonido = 392Hz / 11 = 47
    for (j=0; j<=Duracion; j++)
    {
        for (i = 0;i<=47;++i)
        {
            Output_bit(Altavoz, 1);
            delay_us(956);
            Output_bit(Altavoz, 0);
            delay_us(956);
        }
    }
}

// *****
// ***** Galería de Melodías*****
// *****

// *** Cumpleaños Feliz ***
void Altavoz_Cumple()
{
    Altavoz_Do(2);
    delay_ms(1);
    Altavoz_Do(1);
    delay_ms(1);
    Altavoz_Re(2);
    delay_ms(1);
    Altavoz_Do(2);
    delay_ms(1);
}

```

```

Altavoz_Fa(2);
delay_ms(1);
Altavoz_Mi(2);
delay_ms(2);

Altavoz_Do(2);
delay_ms(1);
Altavoz_Do(1);
delay_ms(1);
Altavoz_Re(2);
delay_ms(1);
Altavoz_Do(2);
delay_ms(1);
Altavoz_Sol(2);
delay_ms(1);
Altavoz_Fa(2);
delay_ms(2);

Altavoz_Do(2);
delay_ms(1);
Altavoz_Do(1);
delay_ms(1);
Altavoz_DoM(2);
delay_ms(1);
Altavoz_La(2);
delay_ms(1);
Altavoz_Fa(2);
delay_ms(1);
Altavoz_Mi(2);
delay_ms(1);
Altavoz_Re(3);
delay_ms(2);

Altavoz_Si(1);
delay_ms(1);
Altavoz_Si(1);
delay_ms(1);
Altavoz_La(2);
delay_ms(1);
Altavoz_Fa(2);
delay_ms(1);
Altavoz_Sol(2);
delay_ms(1);
Altavoz_Fa(4);
}

// *** Escala Musical Ascendente ***
// * Se debe pasar como parámetro *
// * la duración de las notas *
// *****
void Altavoz_Escala_Asc(int Duracion)
{
    Altavoz_Do(Duracion);
    Delay_ms(50);
    Altavoz_Re(Duracion);
    Delay_ms(50);
    Altavoz_Mi(Duracion);
    Delay_ms(50);
    Altavoz_Fa(Duracion);
    Delay_ms(50);
    Altavoz_Sol(Duracion);
    Delay_ms(50);
    Altavoz_La(Duracion);
    Delay_ms(50);
    Altavoz_Si(Duracion);
    Delay_ms(50);
    Altavoz_DoM(Duracion);
}

```

```
    Delay_ms(50);
}

// ***** Beep *****
// * Se debe pasar como parámetro *
// * el número de repeticiones *
// *****
void Altavoz_Beep(int Repeticion)
{
    int i;
    for (i=0; i < Repeticion; i++)
    {
        Altavoz_La(0);
        delay_ms(10);
    }
}

// *** SMS en Morse (Nokia) ***
void Altavoz_SMS_Nokia()
{
    Altavoz_Si(1);
    delay_ms(30);
    Altavoz_Si(1);
    delay_ms(30);
    Altavoz_Si(1);
    delay_ms(120);

    Altavoz_Si(5);
    delay_ms(40);
    Altavoz_Si(5);
    delay_ms(120);

    Altavoz_Si(1);
    delay_ms(30);
    Altavoz_Si(1);
    delay_ms(30);
    Altavoz_Si(1);
    delay_ms(30);
}

// *** Escala ascendente rápida sin esperas ***
void Altavoz_Sube()
{
    Altavoz_Do(0);
    Altavoz_Re(0);
    Altavoz_Mi(0);
    Altavoz_Fa(0);
    Altavoz_Sol(0);
    Altavoz_La(0);
    Altavoz_Si(0);
    Altavoz_DoM(0);
}

// *** Escala descendente rápida sin esperas ***
void Altavoz_Baja()
{
    Altavoz_DoM(0);
    Altavoz_Si(0);
    Altavoz_La(0);
    Altavoz_Sol(0);
    Altavoz_Fa(0);
    Altavoz_Mi(0);
    Altavoz_Re(0);
    Altavoz_Do(0);
}
```

