



Aguijón

Notas de aplicación

Nota de aplicación 10:

Musical Notes

Descripción:

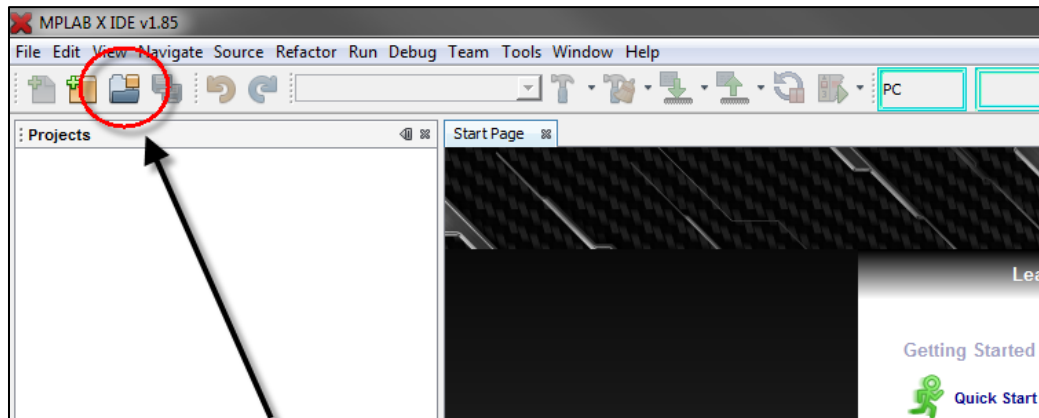
Controlar el Buzzer integrado en nuestra tablilla, variando su Frecuencia y por tanto generando diferentes tonos.

Herramientas:

1. Aguijón 3.0, Aguijón 4.0 ó Aguijón 4.1
2. MPLAB X®
3. Aguijón HID bootloader
4. Cable USB 'A' to 'A'
5. Librerías HammerHead.

Pasos:

1. Abrir MPLAB X® y cargar el archivo del proyecto: **Application Note 10.X**



Haz 'clic' aquí y
abre el proyecto

2. Abrir el archivo **main.c**



3. Ir a la línea #34.

```
34 const long MusicalNotes[101] = {
35     //c    c#    d        d#    e        f        f#    g        g#    a        a#    b        //octave
36     61162,58737,54496,51414,48544,45809,43253,40816,38521,36364,34317,32394, // 0
37     30581,28860,28241,25714,24282,22910,21622,20408,19264,18182,17161,16197, // 1
38     15288,14430,13620,12857,12134,11453,10811,10204,9631 ,9091 ,8581 ,8099 , // 2
39     7645 ,8216 ,6811 ,6428 ,6068 ,5827 ,5405 ,5102 ,4816 ,4545 ,4290 ,4050 , // 3
40     3822 ,3608 ,3405 ,3214 ,3034 ,2863 ,2703 ,2551 ,2408 ,2273 ,2145 ,2025 , // 4
41     1911 ,1804 ,1703 ,1607 ,1517 ,1432 ,1351 ,1276 ,1204 ,1136 ,1073 ,1012 , // 5
42     956 ,902 ,851 ,804 ,758 ,716 ,776 ,638 ,602 ,568 ,536 ,506 , // 6
43     478 ,451 ,426 ,402 ,379 ,358 ,338 ,319 ,301 ,284 ,268 ,253 , // 7
44     239 ,225 ,213 ,201 ,0
45 }
```

En el código, se incluye un arreglo de valores que representa el periodo correspondiente de cada nota musical.

- Cada columna representa las notas musicales, mientras que, cada renglón representa la octava de la nota.

4. Ir a la línea #88.

Vamos a utilizar la siguiente función:

```
81     delayms (500);
82     LCD_PutStr(2,0," DO   RE   MI   FA",FALSE);
83
84
85
86     for(;;){
87
88         keyboard=SW_Read();           //Read keyboard
89
90         if(keyboard)                  //If keyboard is pressed..
91         {
92
93             switch (keyboard)
94             {
95                 case 1:
```

SW_Read ();

- Función que lee el puerto de PUSH BUTTONS;
Devuelve un valor equivalente al Push-Button presionado (carácter del 1 al 4)

5. Ir a la línea #96

Utilizaremos la siguiente función:

```
89
90     if (keyboard)                //If keyboard is pressed..
91     {
92
93     switch (keyboard)
94     {
95         case 1:
96             Buzzer_On (MusicalNotes[60],50);    //play a 5th octave c
97             LED_Set (8,ON);                      //Turn led 8 and 7 ON
98             LED_Set (7,ON);
99             break;
100        case 2:
101            Buzzer_On (MusicalNotes[62],50);    //play a 5th octave d
102            LED_Set (6,ON);                      //Turn led 6 and 5 ON
103            LED_Set (5,ON);
```

Buzzer_On (int period, int duty_cycle);

- Función que establece el periodo y ciclo de trabajo del BUZZER; donde:
Int period = Periodo de oscilación (Valor entero de 0 a 8000.)
Int duty_cycle = Ciclo de trabajo en %porcentaje (valor entero de 0 a 100)

6. Ir a la línea #67.

Utilizaremos la siguiente función:

```
101         Buzzer_On(MusicalNotes[62],50);           //play a 5th octave d
102         LED_Set(6,ON);                             //Turn led 6 and 5 ON
103         LED_Set(5,ON);
104         break;
105     case 3:
106         Buzzer_On(MusicalNotes[64],50);           //play a 5th octave e
107         LED_Set(4,ON);                             //Turn led 4 and 3 ON
108         LED_Set(3,ON);
109         break;
110     case 4:
111         Buzzer_On(MusicalNotes[65],50);           //play a 5th octave f
112         LED_Set(2,ON);                             //Turn led 2 and 1 ON
113         LED_Set(1,ON);
114         break;
115     }
```

LED_Set (int num, BOOL state);

- Función que enciende y apaga un LED; donde:
Int num = Numero de LED que queremos conmutar (valor entero del 1 al 8)
BOOL state = Estado del LED (ON=Encendido, OFF = Apagado)

7. Ir a la línea #117.

Utilizaremos la siguiente función:

```
110         case 4:
111             Buzzer_On(MusicalNotes[65],50);    //play a 5th octave f
112             LED_Set(2,ON);                      //Turn led 2 and 1 ON
113             LED_Set(1,ON);
114             break;
115     }
116
117     while(SW_ReadPort()); //wait until no SW is pressed
118     Buzzer_Off();          //turn buzzer off
119     LEDport_Set(0xff,OFF); //clear LEDport
120 }
121 }
122 return 0;
123 }
```

SW_ReadPort ();

- Esta función lee el puerto de PUSH BUTTONS;
Regresa un valor equivalente al valor del puerto de Push Buttons
(Carácter del 1 al 15 (4 Bits))

8. Ir a la línea #118

Utilizaremos la siguiente función:

```
111         Buzzer_On(MusicalNotes[65],50);    //play a 5th octave f
112         LED_Set(2,ON);                      //Turn led 2 and 1 ON
113         LED_Set(1,ON);
114         break;
115     }
116
117     while(SW_ReadPort()); //wait until no SW is pressed
118     Buzzer_Off();          //turn buzzer off
119     LEDport_Set(0xff,OFF); //clear LEDport
120 }
121 }
122 return 0;
123 }
```

Buzzer_off ();

- Función que apaga el BUZZER;
Si el Buzzer es activado es necesario utilizar esta función para apagarlo.

9. Ir a la línea #119

Utilizaremos la siguiente función:

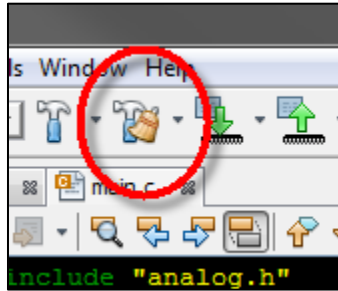
```
112         LED_Set(2,ON);           //Turn led 2 and 1 ON
113         LED_Set(1,ON);
114         break;
115     }
116
117     while(SW_ReadPort()); //wait until no SW is pressed
118     Buzzer_Off();          //turn buzzer off
119     LEDport_Set(0xff,OFF); //clear LEDport
120 }
121 }
122 return 0;
123 }
```

LEDport_Set (char value, BOOL invert);

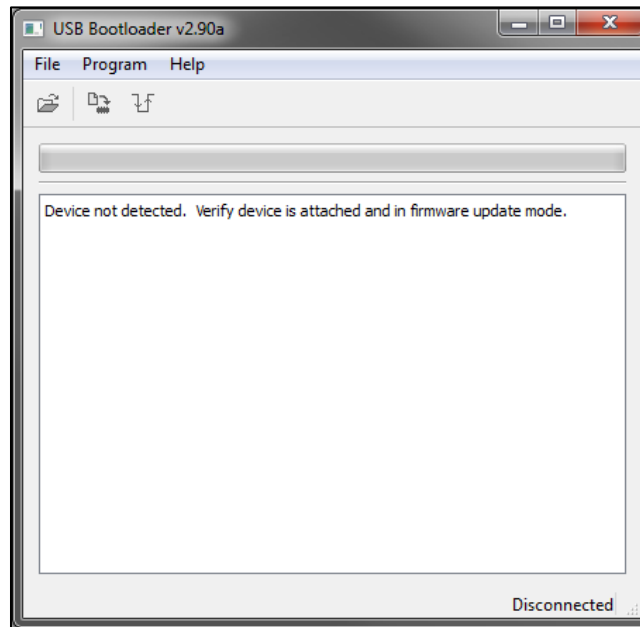
- Función que enciende y apaga el puerto de LEDs; donde:
Char value = Número que queremos Encender (Valor entero del 1 al 255.)
BOOL invert = Estado del puerto de LEDs (ON = Encendido, OFF = apagado).

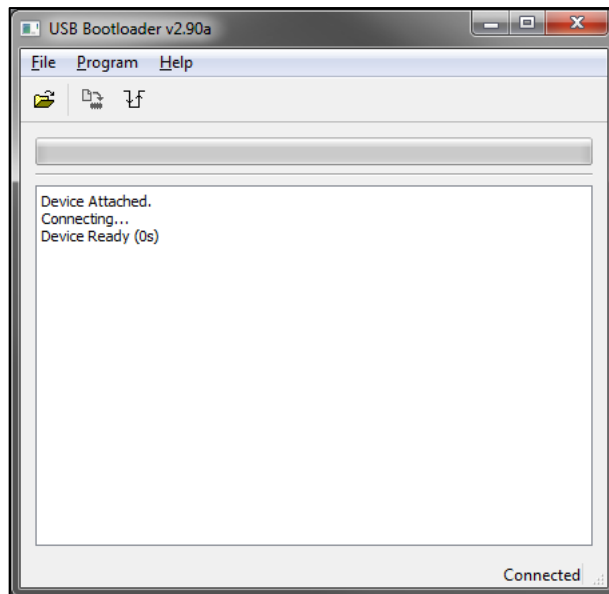
10. Compilar y programar

Al hacer clic en el ícono de compilar, y si no hay errores de compilación, el bootloader será cargado automáticamente.



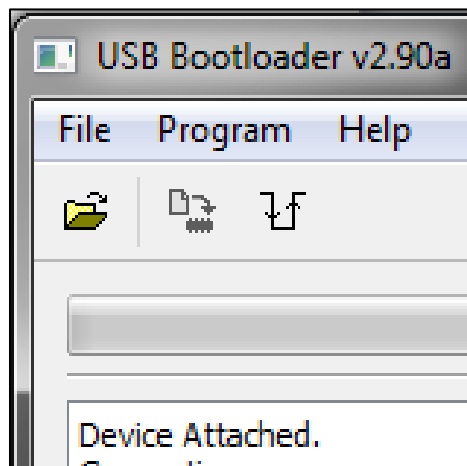
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





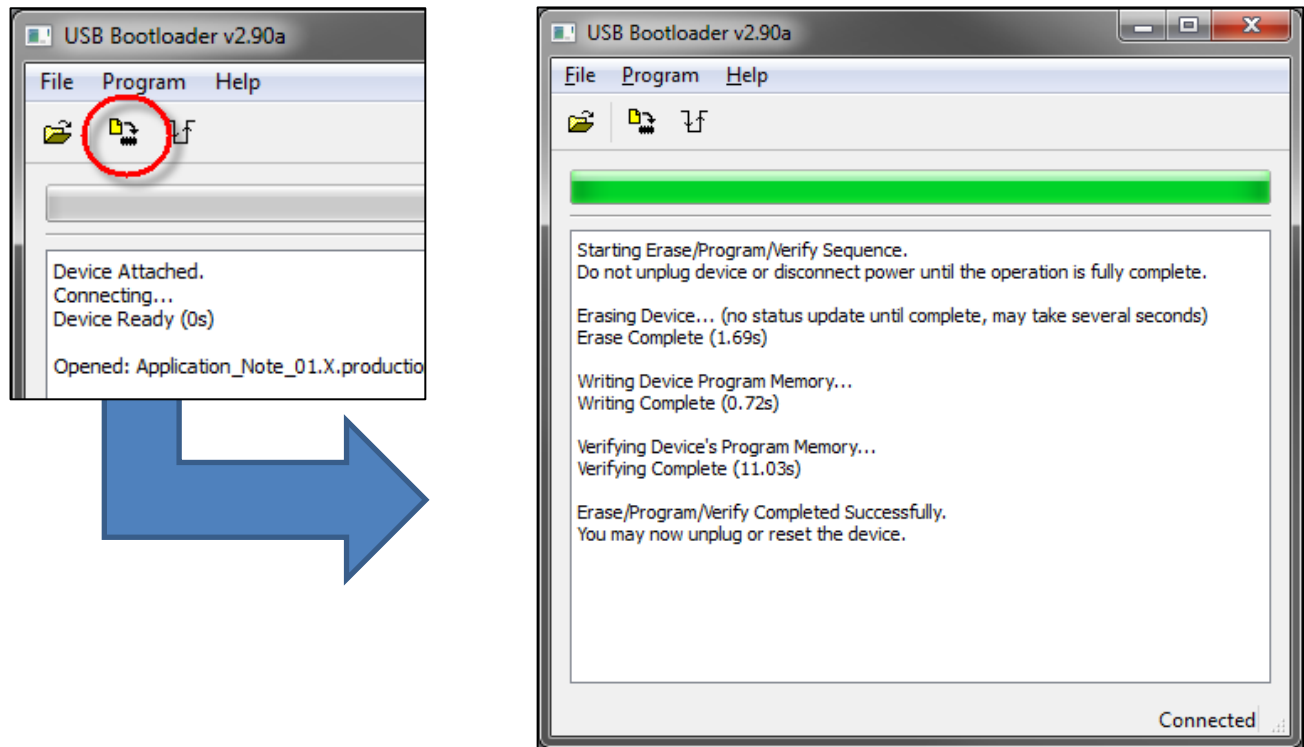
La ventana del Bootloader indicará la conexión establecida con el aguijón:

11. Hacer Clic en Abrir y Cargar el archivo **Application Note 10.X.production.hex**

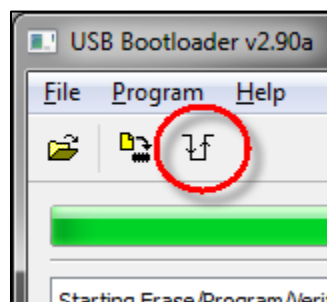


El archivo, depende de la plataforma de hardware.

Una vez cargado el archivo, hacer clic en el ícono de programar.



Hacer clic en el ícono de Reset cuando en la ventana del Bootloader se indique que se terminó de programar con éxito.



Una vez programado podemos verificar el programa corriendo en la tarjeta.

12. Para verificar el funcionamiento del programa verifique que el BUZZER produzca diferentes notas al presionar los PUSH-BUTTONS.