



Aguijón

Notas de aplicación

Nota de aplicación 04:

LED Port & Input Read

Descripción:

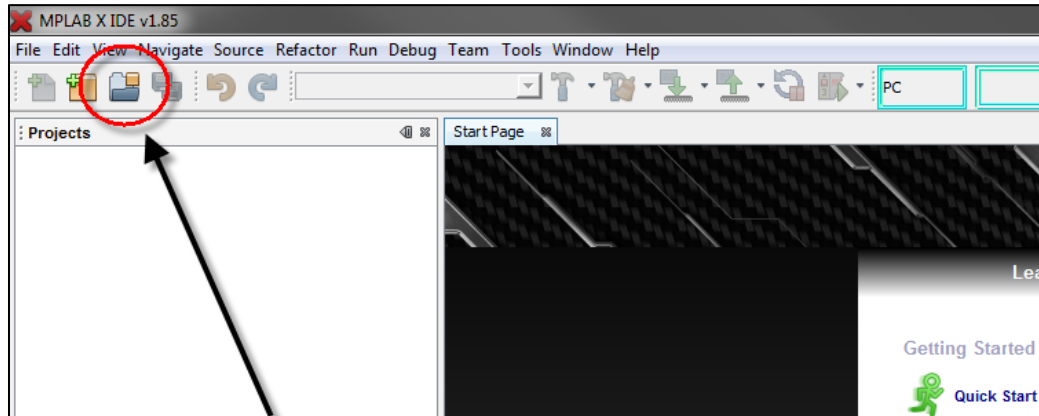
Generar un corrimiento en el puerto de Leds, al hacer coincidir un valor fijo con el valor introducido en el puerto de entradas Opto Aisladas.

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 21.X**



Haz 'clic' aquí y
abre el proyecto

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



3. Ir a la línea #91

Utilizaremos la siguiente función:

```
84 LCD_PutStr(2,0,"Application Note 21",FALSE);
85 delayms(500);
86 LCD_Clear();
87
88 for(;;){
89
90     //Create a strings for the LCD//
91     LCD_PutStr(1,0,"DESACTIVADO",FALSE);
92
93     while (IN_ReadPort()==Active)           //While IN_Read Port is 9 (10
94     {
95         //Create a strings for the LCD//
96         LCD_PutStr(1,0,"ACTIVADO",TRUE);
97         LED_sequence();                       // Initialize LED _sequence
98         LEDport_Set(i,ON);                   //Show i value in LEDport
```

LCD_PutStr (int y, int x, char *msg, BOOL clear);

- Función que muestra una cadena de caracteres en la pantalla LCD; donde:
Int y = Coordenada en y (Valor entero del 1 al 2.)
Int X = Coordenada en x(Valor entero del 0 al 19.)
Char *msg = Cadena de caracteres (De 0 a 20 caracteres)
BOOL clear = Determina si se borra la pantalla antes de escribir
(TRUE =Borrar, FALSE =No borrar).

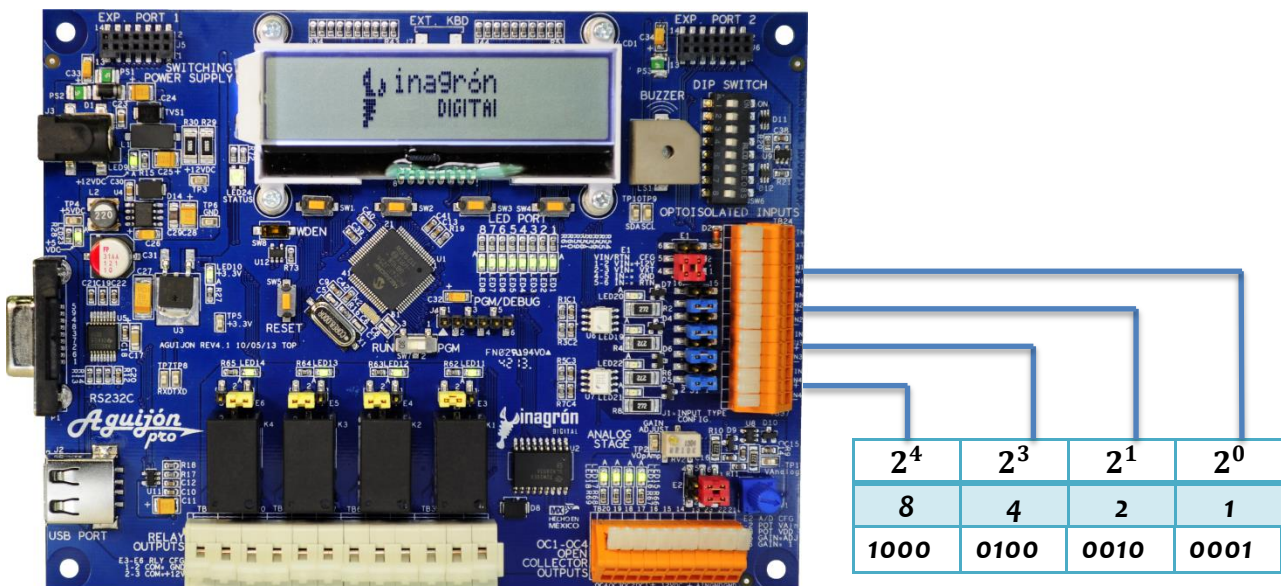
4. Ir a la línea #93.

Utilizaremos la siguiente función:

```
86 LCD_Clear();
87
88 for(;;){
89
90     //Create a strings for the LCD//
91     LCD_PutStr(1,0,"DESACTIVADO",FALSE);
92
93     while (IN_ReadPort()==Active)           //While IN_Read Port is 9 (10
94     {
95         //Create a strings for the LCD//
96         LCD_PutStr(1,0,"ACTIVADO",TRUE);
97         LED_sequence();                       // Initialize LED_sequence
98         LEDport_Set(i,ON);                   //Show i value in LEDport
99         delaysms(250);                       //Delay 250 milliseconds
100     }
```

IN_ReadPort ();

- Esta función lee el valor del puerto de entradas Opto-Aisladas;
Regresa un valor equivalente al del puerto de entrada (carácter del 1 al 15 (4 Bits))



5. Ir a la línea #97

Utilizaremos la siguiente función:

```
93     while (IN_ReadPort()==Active)           //While IN_Read Port is 9 (10
94     {
95         //Create a strings for the LCD//
96         LCD_PutStr(1,0,"ACTIVADO",TRUE);
97         LED_sequence();                       // Initialize LED _sequence
98         LEDport_Set(i,ON);                   //Show i value in LEDport
99         delaysms(250);                       //Delay 250 milliseconds
100     }
101 }
```

LED-sequence ();

- Función que genera un corrimiento tanto ascendente como descendente en el puerto de leds. Dicha función está estructurada en el mismo archivo en la línea #37.

```
33
34     int i=128;
35     BOOL Direction=ON;
36
37     void LED_sequence(void)
38     {
39         if(Direction)
40         {
41             i>>=1;
42             Direction=(i==1) ? OFF : ON;
43         }
44         else
45         {
46             i<<=1;
47             Direction=(i==128) ? ON: OFF;
48         }
49     }
```

6. Ir a la línea #98

Utilizaremos la siguiente función:

```
91     LCD_PutStr(1,0,"DESACTIVADO",FALSE);
92
93     while (IN_ReadPort()==Active)           //While IN_Read Port is 9 (10
94     {
95         //Create a strings for the LCD//
96         LCD_PutStr(1,0,"ACTIVADO",TRUE);
97         LED_sequence();                      // Initialize LED _sequence
98         LEDport_Set(i,ON);                   //Show i value in LEDport
99         delayms(250);                        //Delay 250 milliseconds
100     }
101 }
102 return 0;
103 }
```

LEDport_Set (char value, BOOL invert);

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

7. Ir a la línea #99.

Utilizaremos la siguiente función:

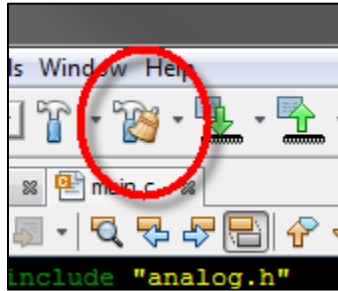
```
90      //Create a strings for the LCD//
91      LCD_PutStr(1,0,"DESACTIVADO",FALSE);
92
93      while (IN_ReadPort()==Active)          //While IN_Read Port is 9 (10
94      {
95          //Create a strings for the LCD//
96          LCD_PutStr(1,0,"ACTIVADO",TRUE);
97          LED_sequence();                     // Initialize LED _sequence
98          LEDport_Set(i,ON);                  //Show i value in LEDport
99          delayms(250);                       //Delay 250 milliseconds
100     }
101 }
102 return 0;
103 }
```

Delayms (ms);

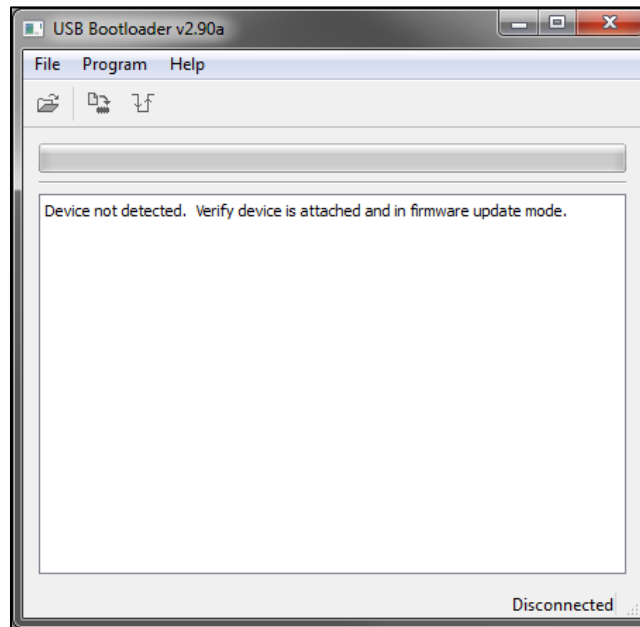
- Función que Pausa el programa por un tiempo determinado (en milisegundos); donde:
ms = el número de milisegundos que se desea pausar el programa.

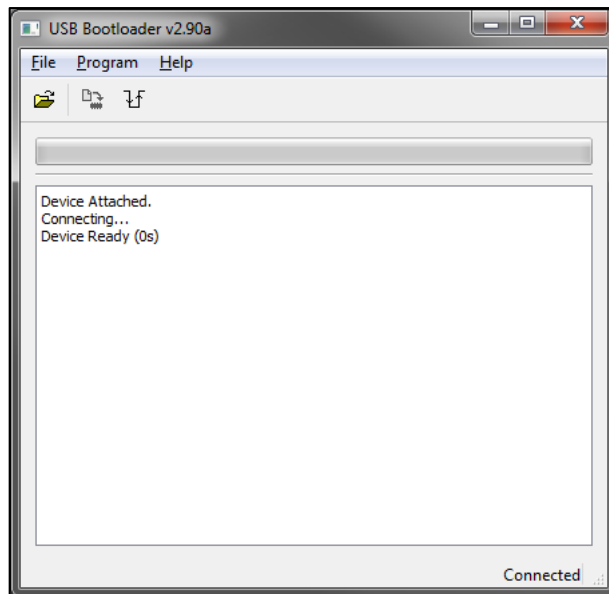
8. Compilar y programar

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



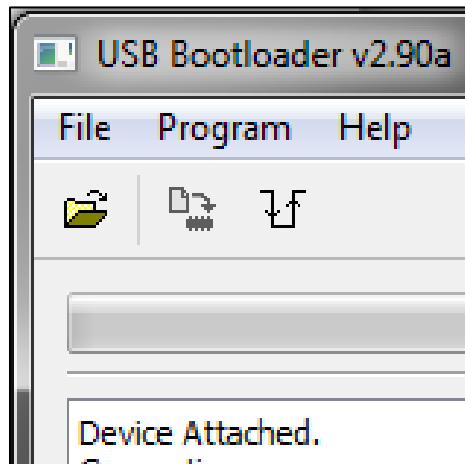
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





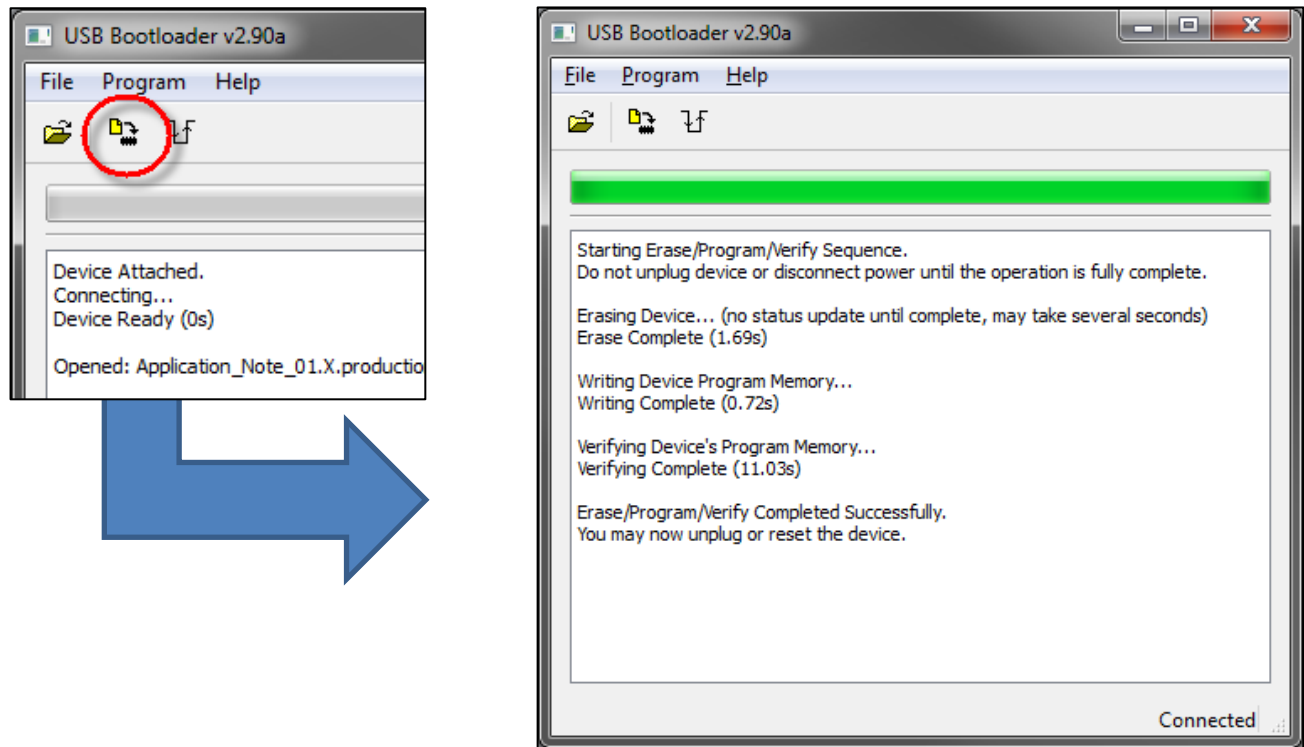
La ventana del Bootloader indicará la conexión establecida con el aguijón:

9. Hacer Clic en Abrir y Cargar el archivo **Application Note 04.X.production.hex**

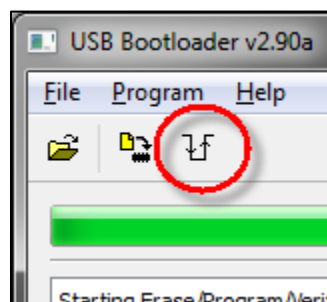


El archivo, depende de la plataforma de hardware.

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.

10. Para verificar el funcionamiento del programa verifique que mientras se introduzca un valor de 9 en el puerto de entradas Opto Aisladas, es decir activar la entrada 1 y 4 (1001), Se genere un corrimiento en el puerto de leds.