



*Aguijón*

---

*Notas de aplicación*

## **Nota de aplicación 13:**

### **LCD Advanced (A)**

#### **Descripción:**

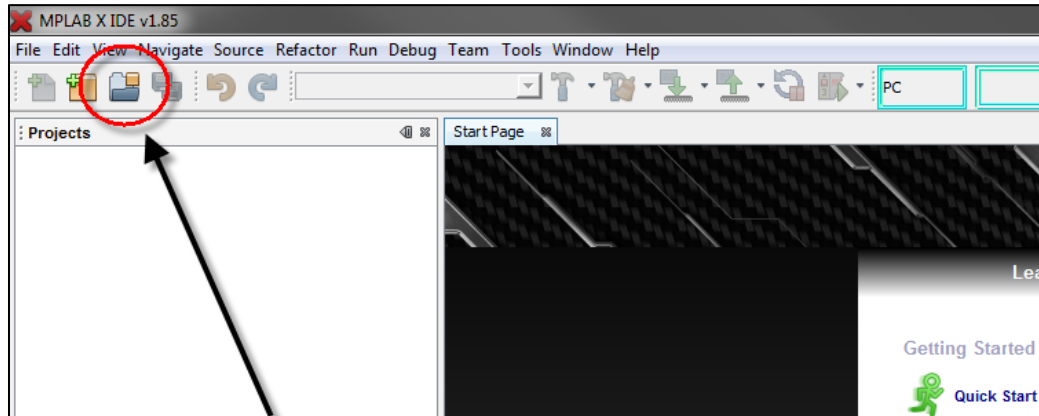
Familiarizarse con las funciones avanzadas para el LCD.

#### **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 13.X**



Haz 'clic' aquí y  
abre el proyecto

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



3. Ir a la línea #75.

Utilizaremos la siguiente función:

```
68 HammerHead_Init(); //initialize [VD]HammerHead
69 LCD_IntroAnimation();
70 LCD_PutStr(1,0,"Vinagron Digital",TRUE);
71 LCD_PutStr(2,0,"Application Note 13",FALSE);
72
73 for(;;){
74
75     keyboard=SW_Read(); //Read Keyboard
76
77     if(keyboard==1)
78     {
79         LCD_PutStr(1,0,"Blinking Cursor",TRUE); //Write "Blinkin
80         LCD_GotoYX(2,0); //LCD goto Y2, X
81         LCD_SetCursorBlink(TRUE); //LCD CursorBlin
82         delayms(1000);
```

**SW\_Read ();**

- Esta función lee el puerto de PUSH BUTTONS;  
Regresa un valor equivalente al Push-Button presionado (carácter del 1 al 4)

4. Ir a la línea #79

Utilizaremos la siguiente función:

```
72
73     for(;;){
74
75         keyboard=SW_Read();           //Read Keyboard
76
77         if(keyboard==1)
78         {
79             LCD_PutStr(1,0,"Blinking Cursor",TRUE);           //Write "Blinkin
80             LCD_GotoYX(2,0);           //LCD goto Y2, X
81             LCD_SetCursorBlink(TRUE);           //LCD CursorBlin
82             delayms(1000);
83
84             for(i=0 ; i<19 ; i++){
85                 LCD_Putc(lcdMSG[i]);           //Character i of
86                 delayms(80);
```

**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 20.)  
**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).

5. Ir a la línea #80

Utilizaremos la siguiente función:

```
73     for(;;){
74
75         keyboard=SW_Read();           //Read Keyboard
76
77         if(keyboard==1)
78         {
79             LCD_PutStr(1,0,"Blinking Cursor",TRUE);           //Write "Blinkin
80             LCD_GotoYX(2,0);           //LCD goto Y2, X
81             LCD_SetCursorBlink(TRUE);           //LCD CursorBlin
82             delayms(1000);
83
84             for(i=0 ; i<19 ; i++){
85                 LCD_Putc(lcdMSG[i]);           //Character i of
86                 delayms(80);
87             }
```

**LCD\_GotoYX (int Y, int X);**

- Función que envía el cursor del LCD a las coordenadas Y,X; donde:  
**Int y** = Coordenada en y (Valor entero del 1 al 2.)  
**Int X** = Coordenada en x (Valor entero del 0 al 20.)

6. Ir a la línea #81

Utilizaremos la siguiente función:

```
74
75     keyboard=SW_Read();           //Read Keyboard
76
77     if(keyboard==1)
78     {
79         LCD_PutStr(1,0,"Blinking Cursor",TRUE);           //Write "Blinkin
80         LCD_GotoYX(2,0);           //LCD goto Y2, X
81         LCD_SetCursorBlink(TRUE);           //LCD CursorBlin
82         delayms(1000);
83
84         for(i=0 ; i<19 ; i++){
85             LCD_Putc(lcdMSG[i]);           //Character i of
86             delayms(80);
87         }
88     }
```

**LCD\_SetCursorBlink (BOOL enabled);**

- Función que Activa o Desactiva el parpadeo del cursor; donde:  
**BOOL enabled** = Estado del Cursor (ON = Activado, OFF = Desactivado).

7. Ir a la línea #85

Utilizaremos la siguiente función:

```
78     {
79         LCD_PutStr(1,0,"Blinking Cursor",TRUE);           //Write "Blinkin
80         LCD_GotoYX(2,0);                                   //LCD goto Y2, X
81         LCD_SetCursorBlink(TRUE);                          //LCD CursorBlin
82         delayms(1000);
83
84         for(i=0 ; i<19 ; i++){
85             LCD_Putc(lcdMSG[i]);                          //Character i of
86             delayms(80);
87         }
88     }
89
90 }
91 return 0;
92 }
```

**LCD\_Putc (char letter);**

- Función que envía una letra a la LCD; donde:  
**Char letter** = Letra a enviar (Valor de la letra en su valor ASCII).



8. Ir a la línea #86.

Utilizaremos la siguiente función:

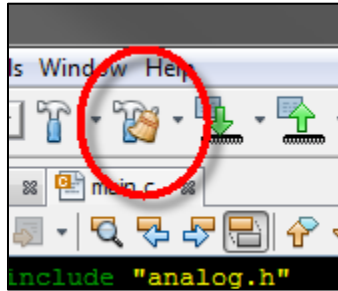
```
79         LCD_PutStr(1,0,"Blinking Cursor",TRUE);           //Write "Blinkin
80         LCD_GotoYX(2,0);                                   //LCD goto Y2, X
81         LCD_SetCursorBlink(TRUE);                          //LCD CursorBlin
82         delayms(1000);
83
84         for(i=0 ; i<19 ; i++){
85             LCD_Putc(lcdMSG[i]);                            //Character i of
86             delayms(80);
87         }
88     }
89
90 }
91 return 0;
92 }
```

### **Delayms (ms);**

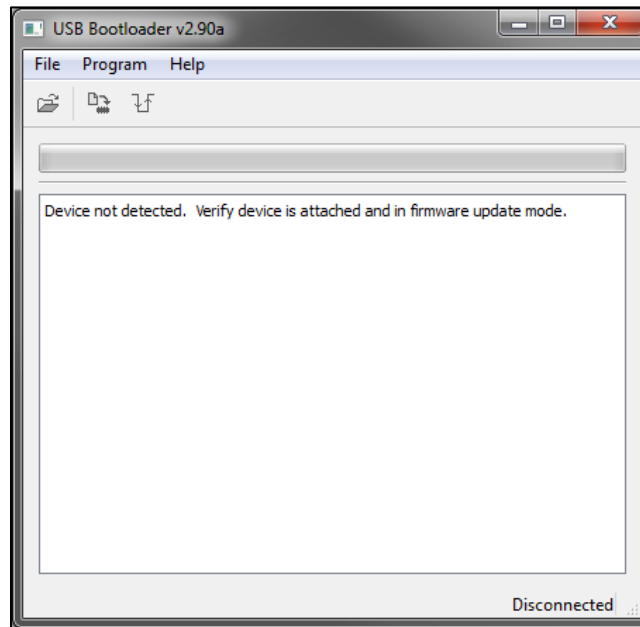
- Función que Pausa el programa por un tiempo determinado (en milisegundos); donde: **ms** = el número de milisegundos que se desea pausar el programa.

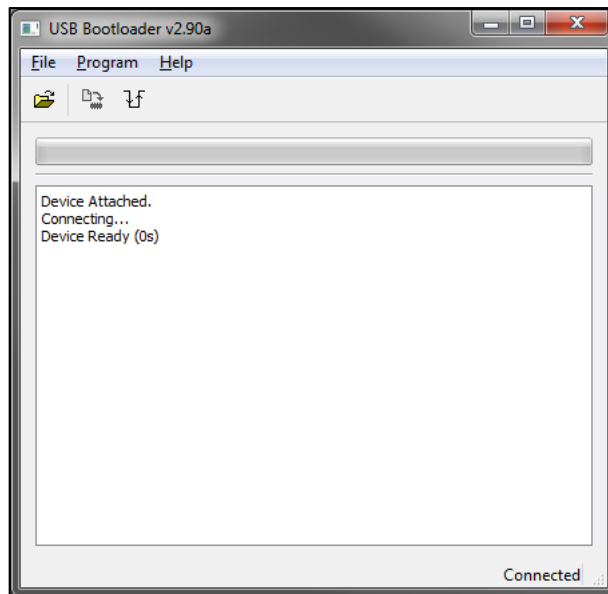
## 9. Compilar y programar

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



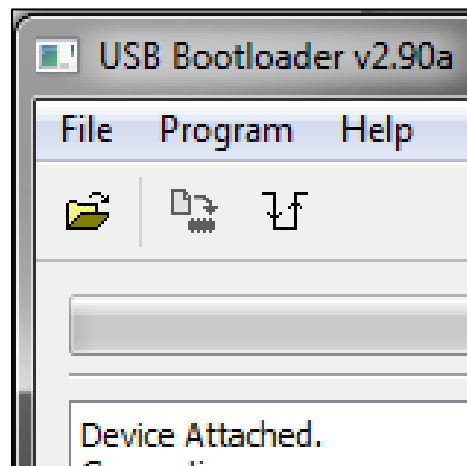
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





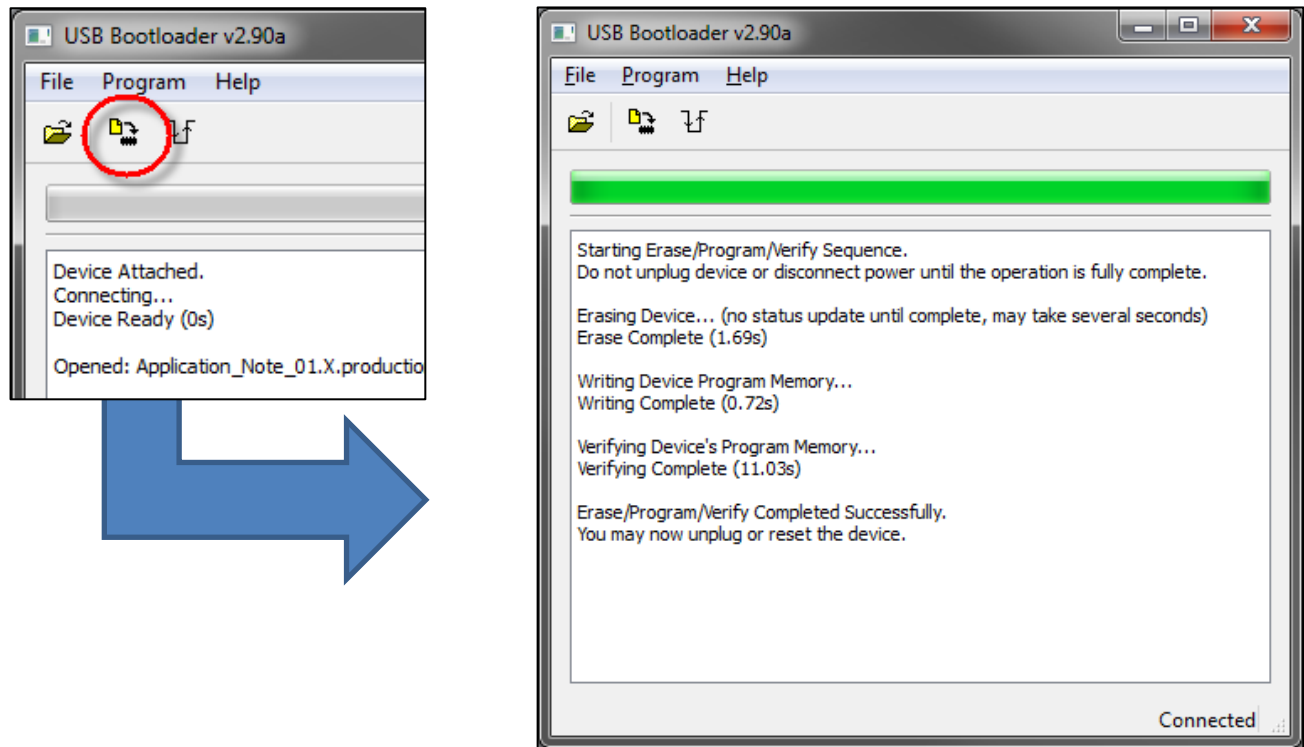
La ventana del Bootloader indicará la conexión establecida con el aguijón:

10. Hacer Clic en Abrir y Cargar el archivo **Application Note 13.X.production.hex**

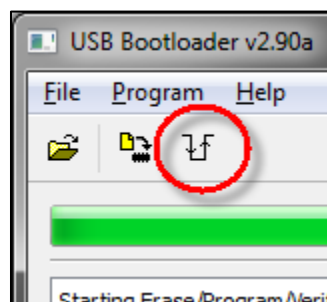


El archivo, depende de la plataforma de hardware.

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.

11. Para verificar el funcionamiento del programa verifique que al presionar el Push-Button 1 el cursor de la pantalla LCD comienza a parpadear mientras avanza desplegando el mensaje "Follows your text"