



*Aguijón*

---

*Notas de aplicación*

## **Nota de aplicación 25:**

### *Read file from the usb drive*

#### **Descripción:**

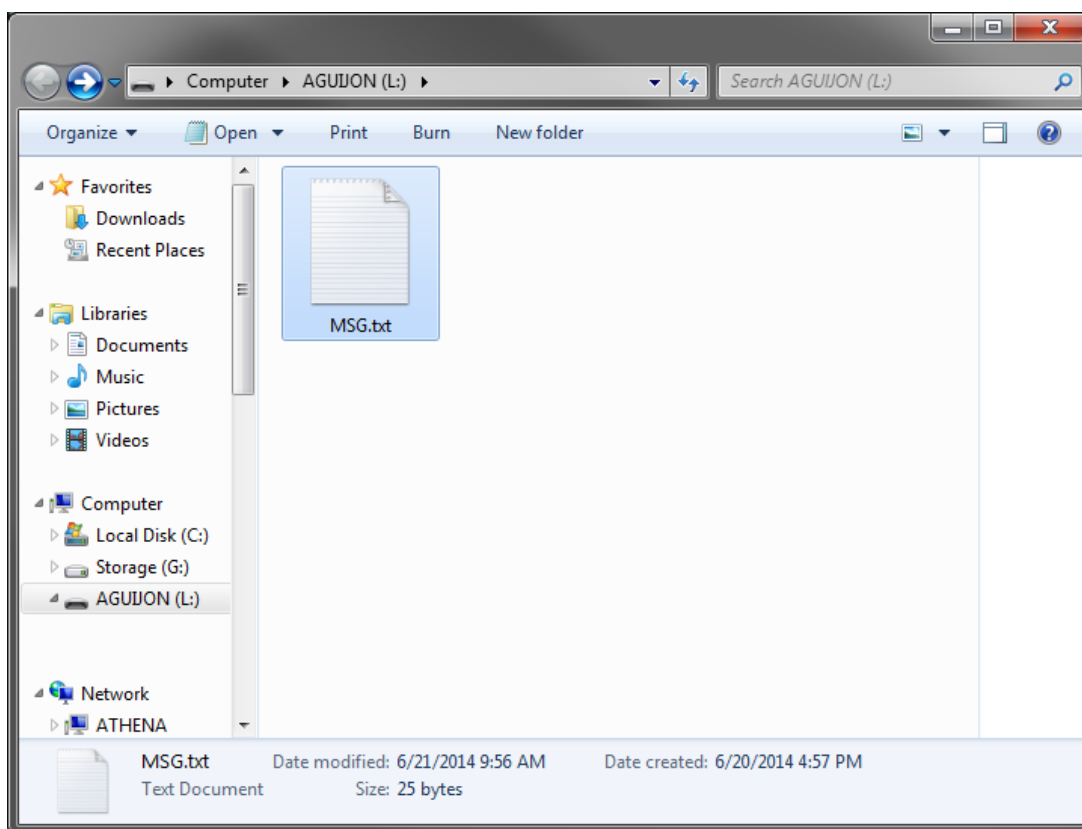
*Leer un mensaje contenido en un archivo de texto ubicado en una memoria USB*

#### **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.
6. Memoria usb 2.0

## Pasos:

1. Crear un archivo de texto en la memoria USB

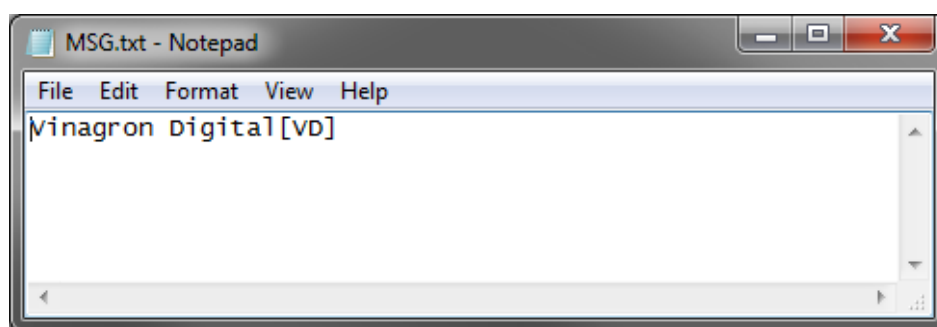


- Crear un archivo de texto con el nombre “MSG.txt”

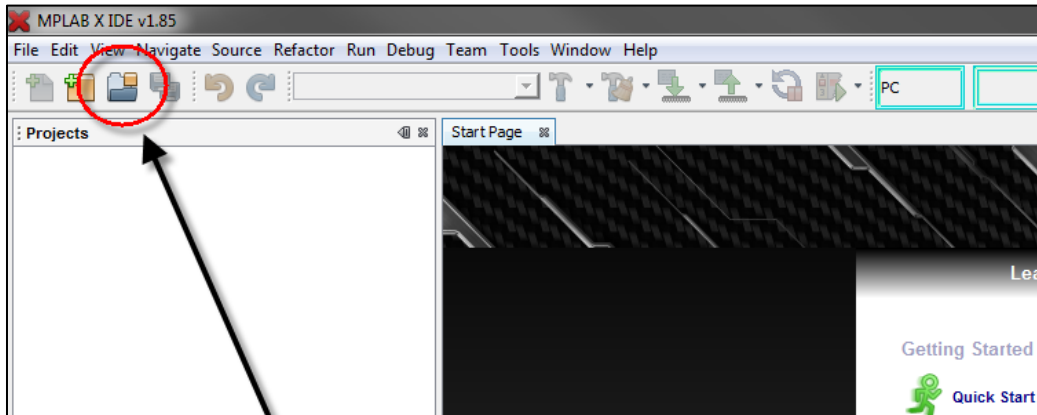
Este debe contener en su interior un mensaje de máximo 20 caracteres.

- Ejemplo:

- 

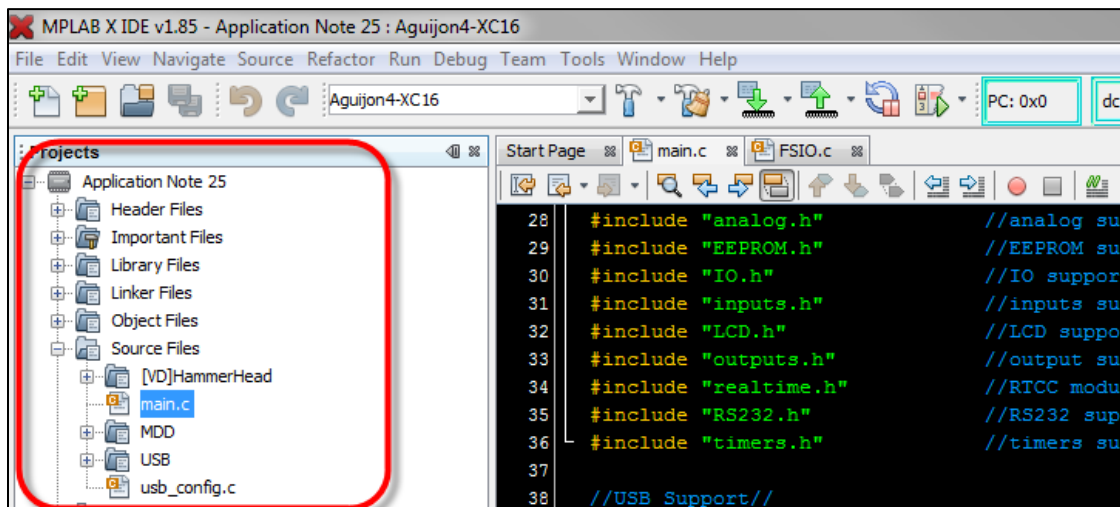


2. Abrir MPLAB X® y cargar el archivo del proyecto: **Application Note 25.X**



Haz 'clic' aquí y  
abre el proyecto

3. Abrir el archivo **main.c**



#### 4. Ir a la línea #133

Utilizaremos la siguiente función:

```
124 LCD_IntroAnimation();
125 LCD_PutStr(1,0,"Vinagron Digital",TRUE);
126 LCD_PutStr(2,0,"Application Note 25",FALSE);
127 delayms(1000);
128 LCD_PutStr(1,0,"Aguijon USB MSD",TRUE);
129 LCD_PutStr(2,0,"Insert flash drive",FALSE);
130
131 for(;;){
132     //USB stack process function
133     USBTasks();
134
135     //if thumbdrive is plugged in
136     if(USBHostMSDSCSIMediaDetect()){
137         InsertChime();
138         //now a device is attached
139     }
```

#### **USBTasks ();**

- Esta función es la herramienta necesaria para configurar e inicializar nuestro dispositivo USB, esta debe ser llamada antes que cualquier otra función relacionada con el puerto USB

5. Ir a la línea #136

Utilizaremos la siguiente función:

```
129 LCD_PutStr(2,0,"Insert flash drive",FALSE);
130
131 for(;;){
132     //USB stack process function
133     USBTasks();
134
135     //if thumbdrive is plugged in
136     if(USBHostMSDSCSIMediaDetect()){
137         InsertChime();
138         //now a device is attached
139
140         //See if the device is attached and in the right format
141         if(FSInit()){
142             //Opening a file in mode "r"
143             //The device must have the file with the same name
```

**USBHostMSDSCSIMediaDetect ( );**

- Función que determina si un dispositivo de almacenamiento masivo está conectado y listo para utilizar; donde:  
Devuelve un valor booleano dependiendo del estado del dispositivo  
(TRUE= MSD presente y listo, FALSE = MSD no está presente o no está listo).

6. Ir a la línea #141.

Utilizaremos la siguiente función:

```
134
135     //if thumbdrive is plugged in
136     if(USBHostMSDSCSIMediaDetect()){
137         InsertChime();
138         //now a device is attached
139
140         //See if the device is attached and in the right format
141         if(FSInit()){
142             //Opening a file in mode "r"
143             //The device must have the file with the same name
144             myFile = FSfopen("msg.txt","r");
145
146             //Read some data from the file
147             FSfread(lcdMSG,1,20,myFile);
148
```

### **FSInit ();**

- Esta función inicializa el dispositivo físico que debe ser conectado al microcontrolador ; donde:  
Devuelve un valor booleano dependiendo del estado de la inicialización  
(TRUE= Inicialización exitosa, FALSE = Inicialización no exitosa).

7. Ir a la línea #144

Utilizaremos la siguiente función:

```
137      InsertChime();
138      //now a device is attached
139
140      //See if the device is attached and in the right format
141      if(FSInit()){
142          //Opening a file in mode "r"
143          //The device must have the file with the same name
144          myFile = FSfopen("msg.txt","r");
145
146          //Read some data from the file
147          FSfread(lcdMSG,1,20,myFile);
148
149          //Create a strings for the LCD//
150          LCD_PutStr(1,0,lcdMSG,TRUE);
151          //Always make sure to close the file.
```

**FSfopen (char \* fileName, char \*mode);**

- Función que edita un archivo ya sea este nuevo o existente; donde:  
**Char \*FileName** = El nombre del archivo a abrir o crear. (Cadena de caracteres, incluyendo extensión del archivo.)  
**Char \*mode** = Modo del uso que se le desea dar al archivo; donde:
  - W:** Crea un archivo nuevo o reemplaza uno existente
  - R:** Lee datos de un archivo existente
  - A:** Agrega datos a un archivo existente
  - W+:** Crea un archivo nuevo o reemplaza uno existente (También habilita lectura)
  - R+:** Lee datos de un archivo existente (También habilita escritura)
  - A+:** Agrega datos a un archivo existente (También habilita lectura)



8. Ir a la línea #147

Utilizaremos la siguiente función:

```
140 | //See if the device is attached and in the right format
141 | if(FSInit()){
142 |     //Opening a file in mode "r"
143 |     //The device must have the file with the same name
144 |     myFile = FSfopen("msg.txt","r");
145 |
146 |     //Read some data from the file
147 |     FSfread(lcdMSG,1,20,myFile);
148 |
149 |     //Create a strings for the LCD//
150 |     LCD_PutStr(1,0,lcdMSG,TRUE);
151 |     //Always make sure to close the file.
152 |     FSfclose(myFile);
153 |     //Just sit here until the device is removed.
154 |     while(USBHostMSDSCSIMediaDetect()){
```

**FSfread (void \*ptr, unsigned long size, unsigned long n, FSFILE \*stream);**

- Función que lee datos de algún archivo específico; donde:  
**\*ptr** = Buffer de destino para los bytes leídos.  
**Size** = Tamaño de las unidades en bytes  
**n** = Número de unidades a transferir.  
**Stream** = Apuntador a la estructura del archivo

9. Ir a la línea #150

Utilizaremos la siguiente función:

```
143 //The device must have the file with the same name
144 myFile = FSfopen("msg.txt","r");
145
146 //Read some data from the file
147 FSfread(lcdMSG,1,20,myFile);
148
149 //Create a strings for the LCD//
150 LCD_PutStr(1,0,lcdMSG,TRUE);
151 //Always make sure to close the file.
152 FSfclose(myFile);
153 //Just sit here until the device is removed.
154 while (USBHostMSDSCSIMediaDetect()) {
155     USBTasks();
156 }
157 RemoveChime();
```

**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).

10. Ir a la línea #152

Utilizaremos la siguiente función:

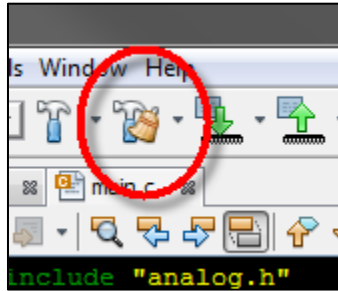
```
145
146      //Read some data from the file
147      FSfread(lcdMSG,1,20,myFile);
148
149      //Create a strings for the LCD//
150      LCD_PutStr(1,0,lcdMSG,TRUE);
151      //Always make sure to close the file.
152      FSfclose(myFile);
153      //Just sit here until the device is removed.
154      while(USBHostMSDSCSIMediaDetect()){
155          USBTasks();
156      }
157      RemoveChime();
158      //Create a strings for the LCD//
159      LCD_PutStr(1,0,"Aguijon USB MSD",TRUE);
```

**FSfclose (FSFILE \*fo);**

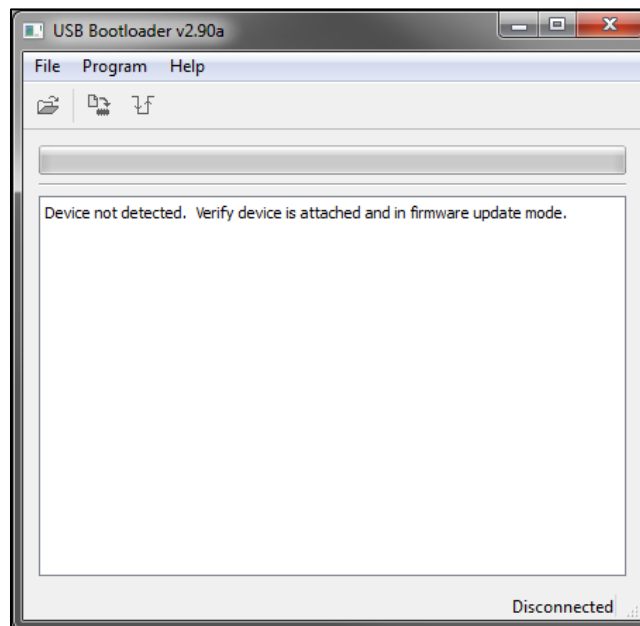
- Función que Actualiza la información y cierra el archivo seleccionado; donde:  
**FSFILE \*fo** = Apuntador a la estructura del archivo a cerrar.

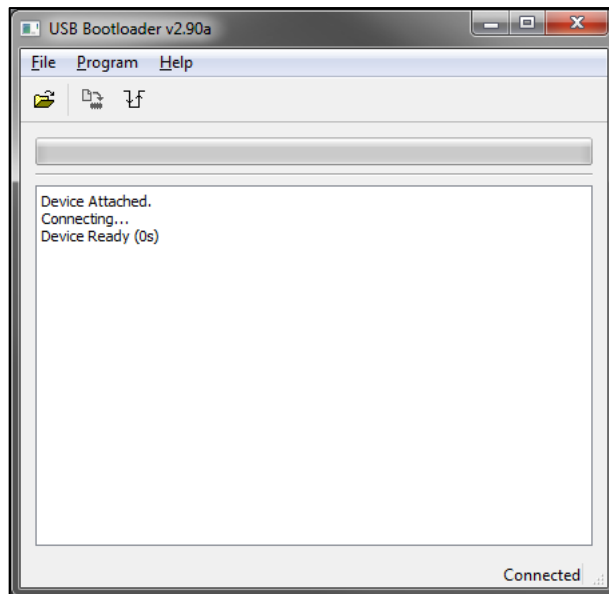
## 11. Compilar y programar

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



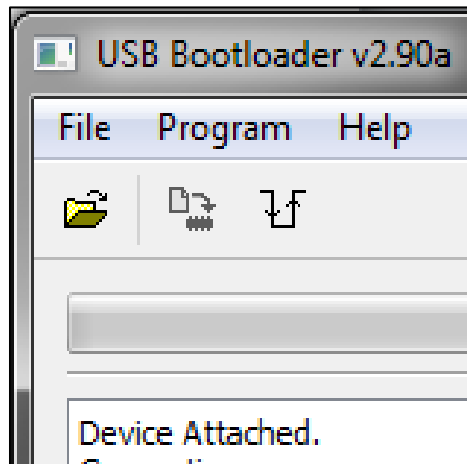
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





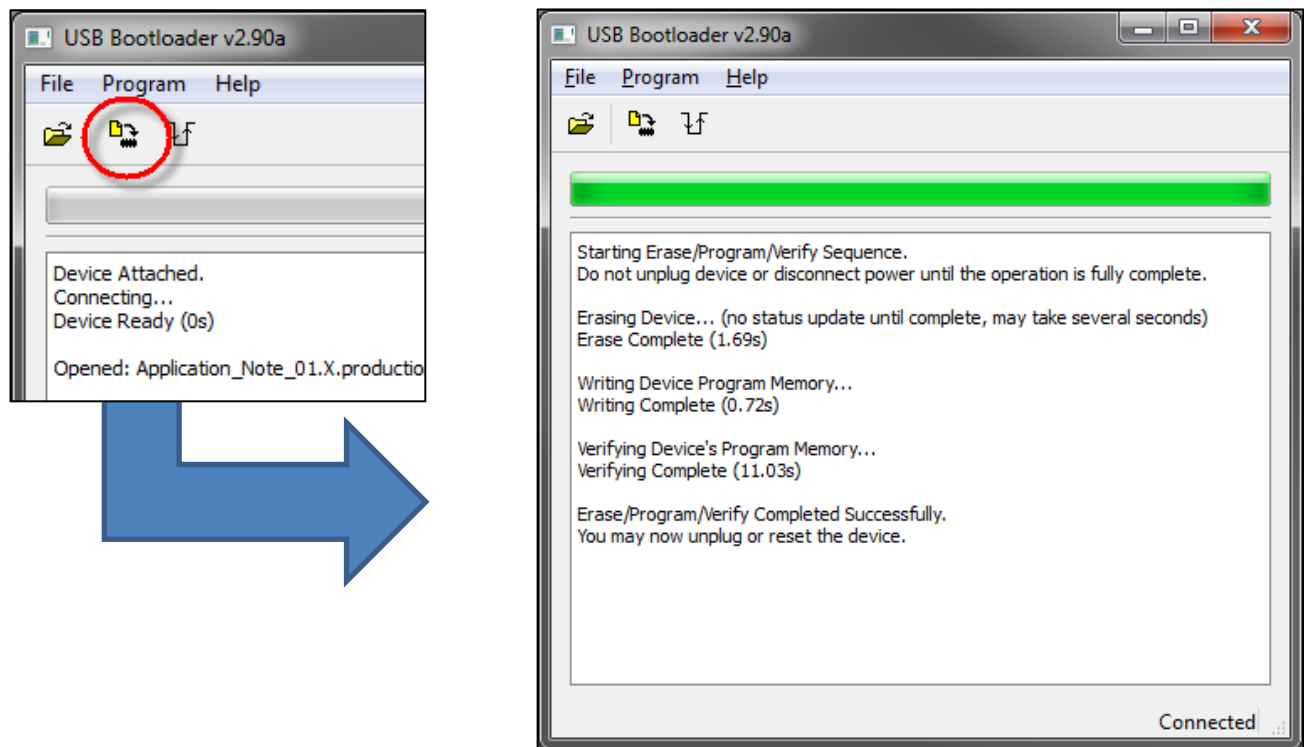
*La ventana del Bootloader indicará la conexión establecida con el aguijón:*

12. Hacer Clic en Abrir y Cargar el archivo **Application Note 25.X.production.hex**

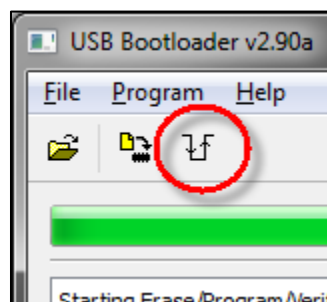


*El archivo, depende de la plataforma de hardware.*

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.

13. Para verificar el funcionamiento del programa verifique que al conectar un dispositivo de almacenamiento USB, el cual contenga un archivo de texto de nombre “MSG.txt”, y a su vez este contenga el mensaje “Vinagron Digital[VD]”, dicho mensaje se despliegue en nuestra pantalla LCD.