



Aguijón

Notas de aplicación

Nota de aplicación 24:

Create text file on the usb

Descripción:

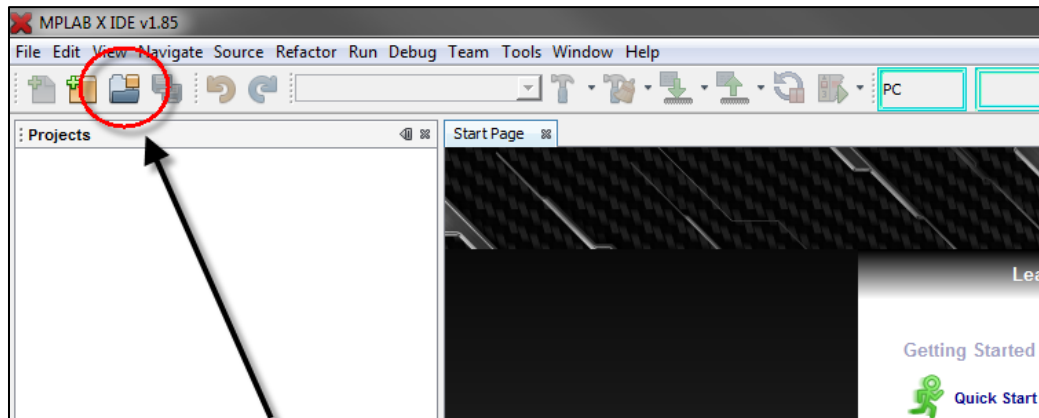
Generar y guardar un archivo de texto en una memoria USB

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

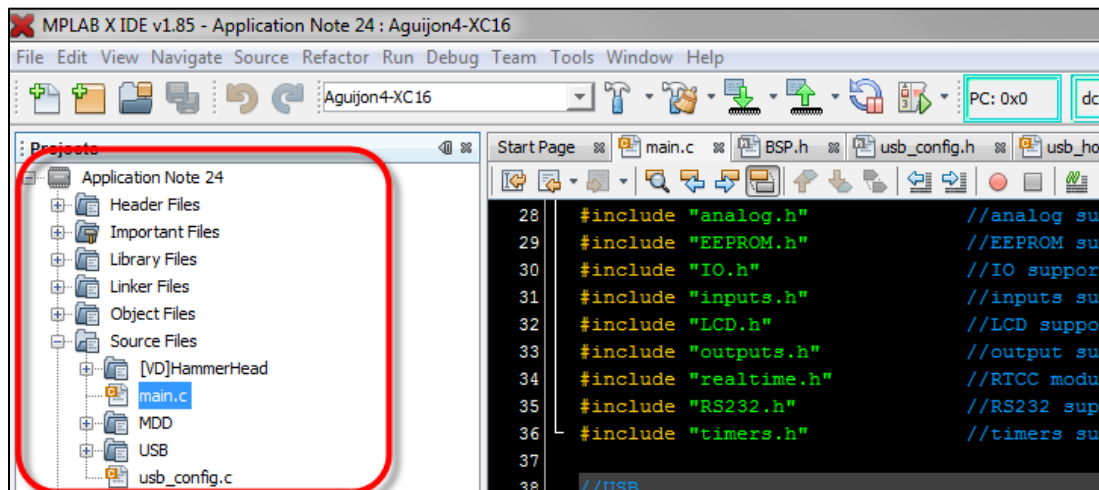
Pasos:

1. Abrir MPLAB X® y cargar el archivo del proyecto: **Application Note 24.X**



Haz 'clic' aquí y
abre el proyecto

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



3. Ir a la línea #131.

Utilizaremos la siguiente función:

```
124 LCD_PutStr(2,0,"Application Note 24",FALSE);
125 delayms(1000);
126 LCD_PutStr(1,0,"Aguijon USB MSD",TRUE);
127 LCD_PutStr(2,0,"Insert flash drive",FALSE);
128
129 for(;;){
130     //USB stack process function
131     USBTasks();
132
133     //if thumbdrive is plugged in
134     if(USBHostMSDSCSIMediaDetect()){
135         InsertChime();
136         //now a device is attached
137
138         //See if the device is attached and in the right format
139         if(FSInit()){
```

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

4. Ir a la línea #134

Utilizaremos la siguiente función:

```
127 LCD_PutStr(2,0,"Insert flash drive",FALSE);
128
129 for(;;){
130     //USB stack process function
131     USBTasks();
132
133     //if thumbdrive is plugged in
134     if(USBHostMSDSCSIMediaDetect()){
135         InsertChime();
136         //now a device is attached
137
138         //See if the device is attached and in the right format
139         if(FSInit()){
140             LCD_PutStr(1,0,"Open file 'test.txt'",TRUE);
141             //Opening a file in mode "w" will create the file if it does
```

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).

5. Ir a la línea #139.

Utilizaremos la siguiente función:

```
132
133     //if thumbdrive is plugged in
134     if(USBHostMSDSCSIMediaDetect()){
135         InsertChime();
136         //now a device is attached
137
138         //See if the device is attached and in the right format
139         if(FSInit()){
140             LCD_PutStr(1,0,"Open file 'test.txt',TRUE);
141             //Opening a file in mode "w" will create the file if it does
142             // exist. If the file does exist it will delete the old fi
143             // and create a new one that is blank.
144             myFile = FSfopen("test.txt","w");
145
146             //Write some data to the new file.
```

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).

6. Ir a la línea #140

Utilizaremos la siguiente función:

```
133         //if thumbdrive is plugged in
134         if (USBHostMSDSCSIMediaDetect()) {
135             InsertChime();
136             //now a device is attached
137
138             //See if the device is attached and in the right format
139             if (FSInit()) {
140                 LCD_PutStr(1,0,"Open file 'test.txt',TRUE);
141                 //Opening a file in mode "w" will create the file if it does
142                 // exist. If the file does exist it will delete the old fi
143                 // and create a new one that is blank.
144                 myFile = FSfopen("test.txt","w");
145
146                 //Write some data to the new file.
147                 FSfwrite("Welcome to [VD]HammerHead. "HAMMERHEAD_DEVICE,1,41
```

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).

7. Ir a la línea #144

Utilizaremos la siguiente función:

```
137
138     //See if the device is attached and in the right format
139     if(FSInit()){
140         LCD_PutStr(1,0,"Open file 'test.txt',TRUE);
141         //Opening a file in mode "w" will create the file if it does
142         // exist. If the file does exist it will delete the old fi
143         // and create a new one that is blank.
144         myFile = FSfopen("test.txt","w");
145
146         //Write some data to the new file.
147         FSfwrite("Welcome to [VD]HammerHead. "HAMMERHEAD_DEVICE,1,41
148
149         //Always make sure to close the file so that the data gets
150         // written to the drive.
151         LCD_PutStr(1,0,"Close file 'test.txt',TRUE);
```

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 LCD_PutStr(1,0,"Open file 'test.txt',TRUE);
141 //Opening a file in mode "w" will create the file if it doesn't
142 // exist. If the file does exist it will delete the old file
143 // and create a new one that is blank.
144 myFile = FSfopen("test.txt","w");
145
146 //Write some data to the new file.
147 FSfwrite("Welcome to [VD]HammerHead. "HAMMERHEAD_DEVICE,1,41 ,myFile);
148
149 //Always make sure to close the file so that the data gets
150 // written to the drive.
151 LCD_PutStr(1,0,"Close file 'test.txt',TRUE);
152 FSfclose(myFile);
153
154 //Just sit here until the device is removed.
```

FSfwrite (void *data_to_write, unsigned long size, unsigned long n, FSFILE *stream);

- Función que escribe datos en algún archivo; donde:
data_to_write = Apuntador a la cadena de caracteres a escribir.
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 #152

Utilizaremos la siguiente función:

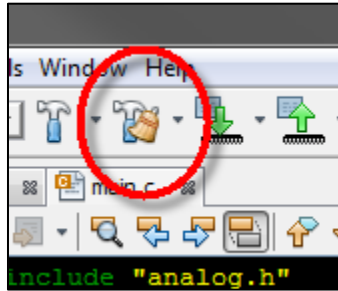
```
145
146         //Write some data to the new file.
147         FSfwrite("Welcome to [VD]HammerHead. "HAMMERHEAD_DEVICE,1,41
148
149         //Always make sure to close the file so that the data gets
150         // written to the drive.
151         LCD_PutStr(1,0,"Close file 'test.txt'",TRUE);
152         FSfclose(myFile);
153
154         //Just sit here until the device is removed.
155         LCD_PutStr(1,0,"DONE!",TRUE);
156         LCD_PutStr(2,0,"Safe to remove USB",FALSE);
157         while(USBHostMSDSCSIMediaDetect()){
158             USBTasks();
159         }
```

FSfclose (FSFILE *fo);

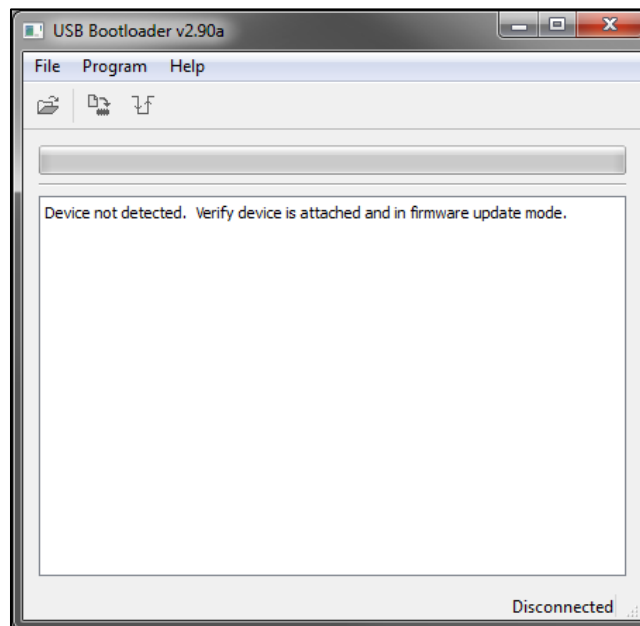
- Función que Actualiza la información y cierra el archivo seleccionado; donde:
FSFILE *fo = Apuntador a la estructura del archivo a cerrar.

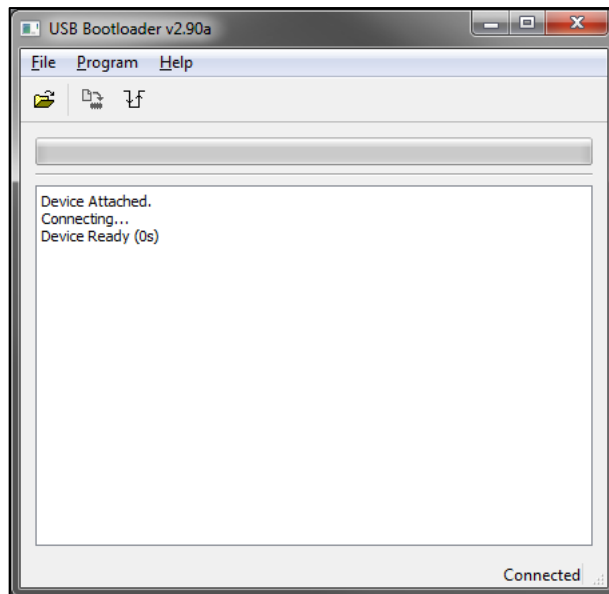
10. Compilar y programar

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



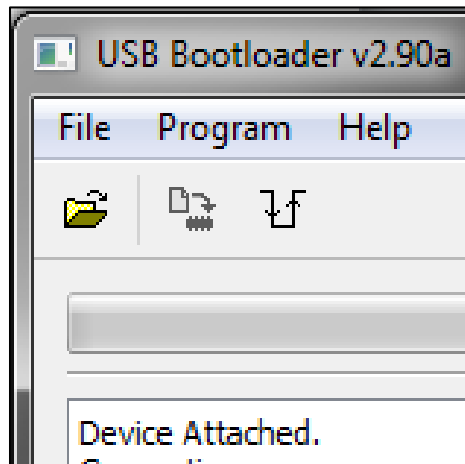
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





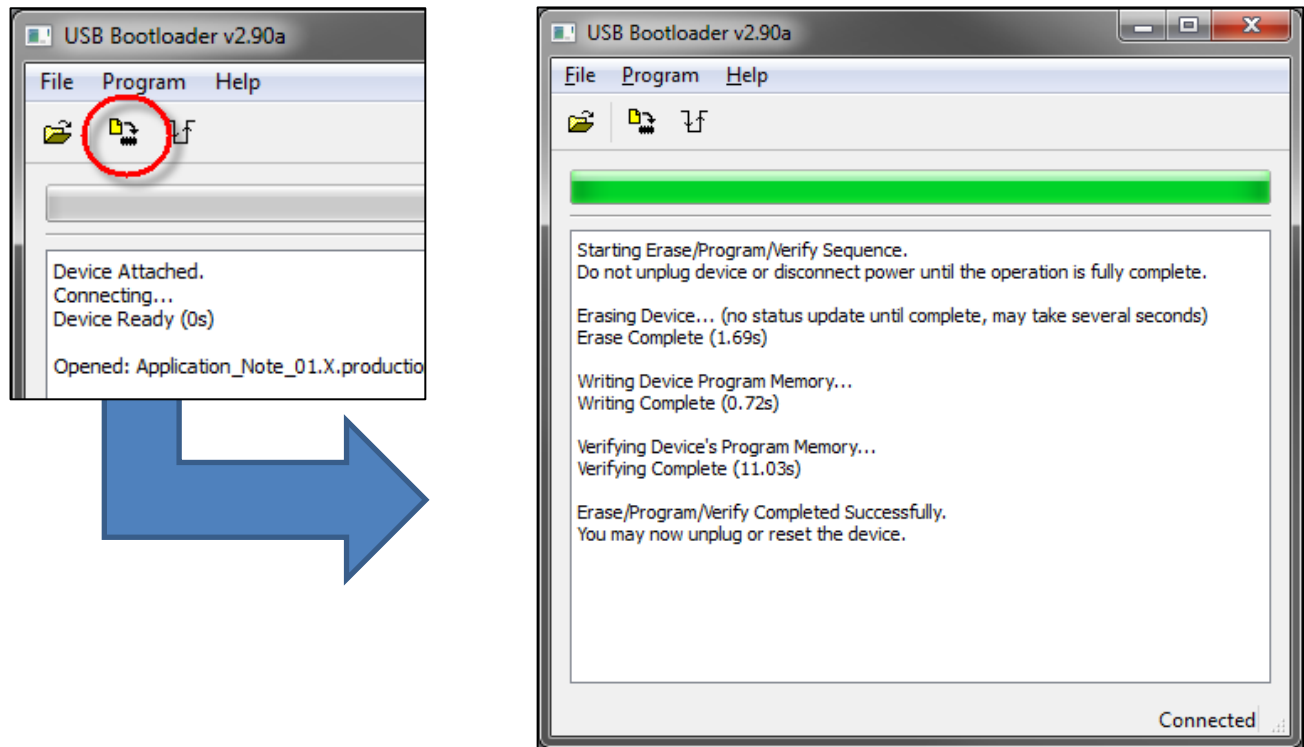
La ventana del Bootloader indicará la conexión establecida con el aguijón:

11. Hacer Clic en Abrir y Cargar el archivo **Application Note 24.X.production.hex**

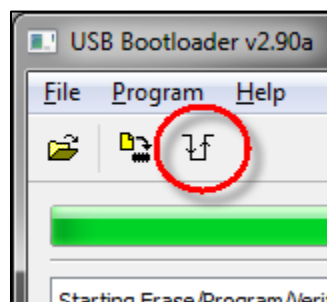


El archivo, depende de la plataforma de hardware.

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 al conectar un dispositivo de almacenamiento USB, en este se genere un archivo con el nombre 'TEST.txt', y que este a su vez contenga el mensaje "Welcome to [VD] HammerHead. Aguijon revx.x"