



Aguijón

Notas de aplicación

Nota de aplicación 11:

Read and Write EEPROM Memory

Descripción:

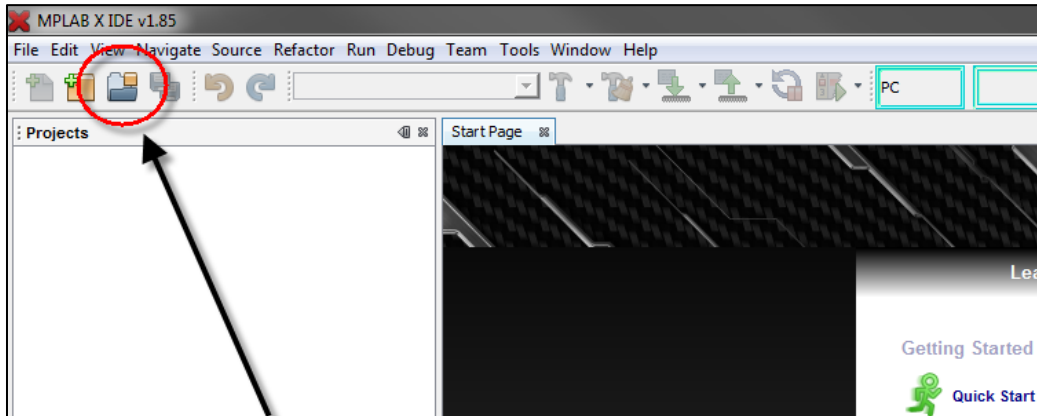
Leer y Escribir en una localidad de la Memoria EEPROM.

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 11.X**



Haz 'clic' aquí y
abre el proyecto

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



3. Ir a la línea #72

Utilizaremos la siguiente función:

```
66
67     HammerHead_Init(); //initialize [VD]HammerHead
68     LCD_IntroAnimation();
69     LCD_PutStr(1,0,"Vinagron Digital",TRUE);
70     LCD_PutStr(2,0,"Application Note 11",FALSE);
71
72     data=EEPROM_ReadByte(0x05); //Read address 0x05
73
74     for(;;){
75
76         sprintf(lcdMSG,"Conteo = %i",data); //create the string
77         LCD_PutStr(1,0,lcdMSG,TRUE); //Write lcdMSG
78
79         data++;
```

EEPROM_ReadByte (char address);

- Función que lee una dirección específica de la memoria EEPROM; donde:
Char addres = Dirección de la localidad específica a escribir (carácter del 0 al 127.)

4. Ir a la línea #76

Utilizaremos la siguiente función:

```
69 LCD_PutStr(1,0,"Vinagron Digital",TRUE);
70 LCD_PutStr(2,0,"Application Note 11",FALSE);
71
72 data=EEPROM_ReadByte(0x05);           //Read address 0x05
73
74 for(;;){
75
76     sprintf(lcdMSG,"Conteo = %i",data); //create the string
77     LCD_PutStr(1,0,lcdMSG,TRUE);       //Write lcdMSG
78
79     data++;
80     EEPROM_WriteByte(0x05,data);       //Write data in address
81     delayms(500);
82
83     if(data==100)
```

sprintf(char *, const char *, ...);

- Función que Genera una cadena de caracteres y lo asigna a una variable; Donde:
Char * = Variable a asignar cadena de caracteres (Variable de tipo char)
const char *, = Cadena de caracteres a asignar a la variable.

5. Ir a la línea #78

Utilizaremos la siguiente función:

```
71
72     data=EEPROM_ReadByte(0x05);           //Read address 0x05
73
74     for(;;){
75
76         sprintf(lcdMSG,"Conteo = %i",data); //create the string
77         LCD_PutStr(1,0,lcdMSG,TRUE);        //Write lcdMSG
78
79         data++;
80         EEPROM_WriteByte(0x05,data);       //Write data in address
81         delays(500);
82
83         if(data==100)
84         {
85             data=0;
```

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).

6. Ir a la línea #81

Utilizaremos la siguiente función:

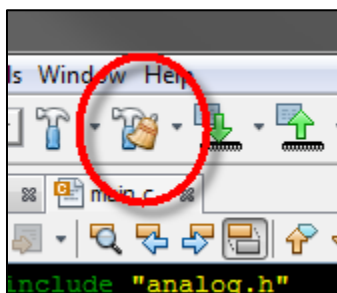
```
74     for(;;){
75
76         sprintf(lcdMSG,"Conteo = %i",data);           //create the string
77         LCD_PutStr(1,0,lcdMSG,TRUE);                 //Write lcdMSG
78
79         data++;
80         EEPROM_WriteByte(0x05,data);                 //Write data in address
81         delays(500);
82
83         if(data==100)
84         {
85             data=0;
86         }
87     }
88     return 0;
```

EEPROM_WriteByte (char address, char new_data);

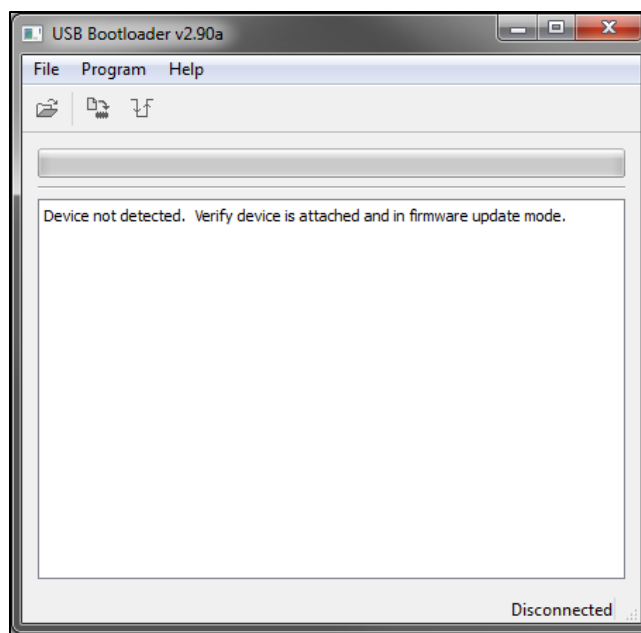
- Función que escribe datos a una función específica de la memoria EEPROM; donde:
char address = Dirección de la localidad específica a leer (carácter del 0 al 127.)
char new_data = Dato a almacenar en la memoria (carácter de máximo 1 byte)

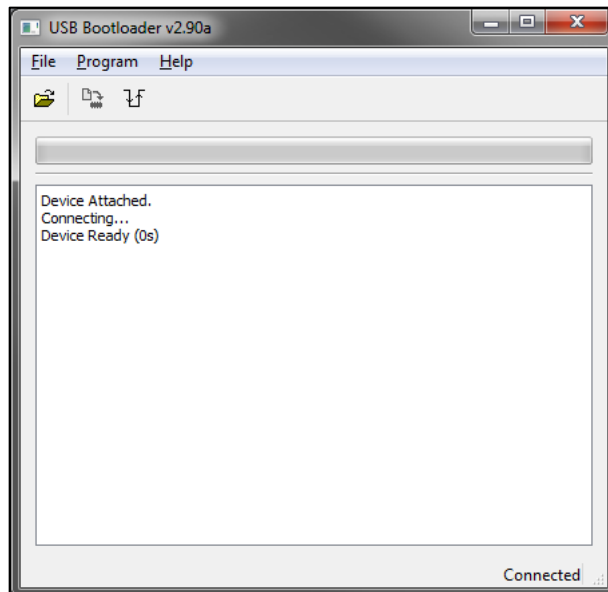
7. Compilar y programar

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



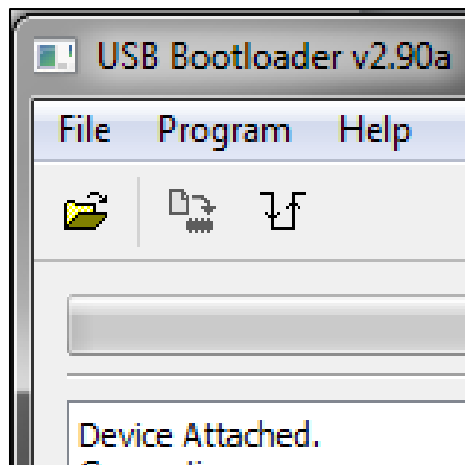
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





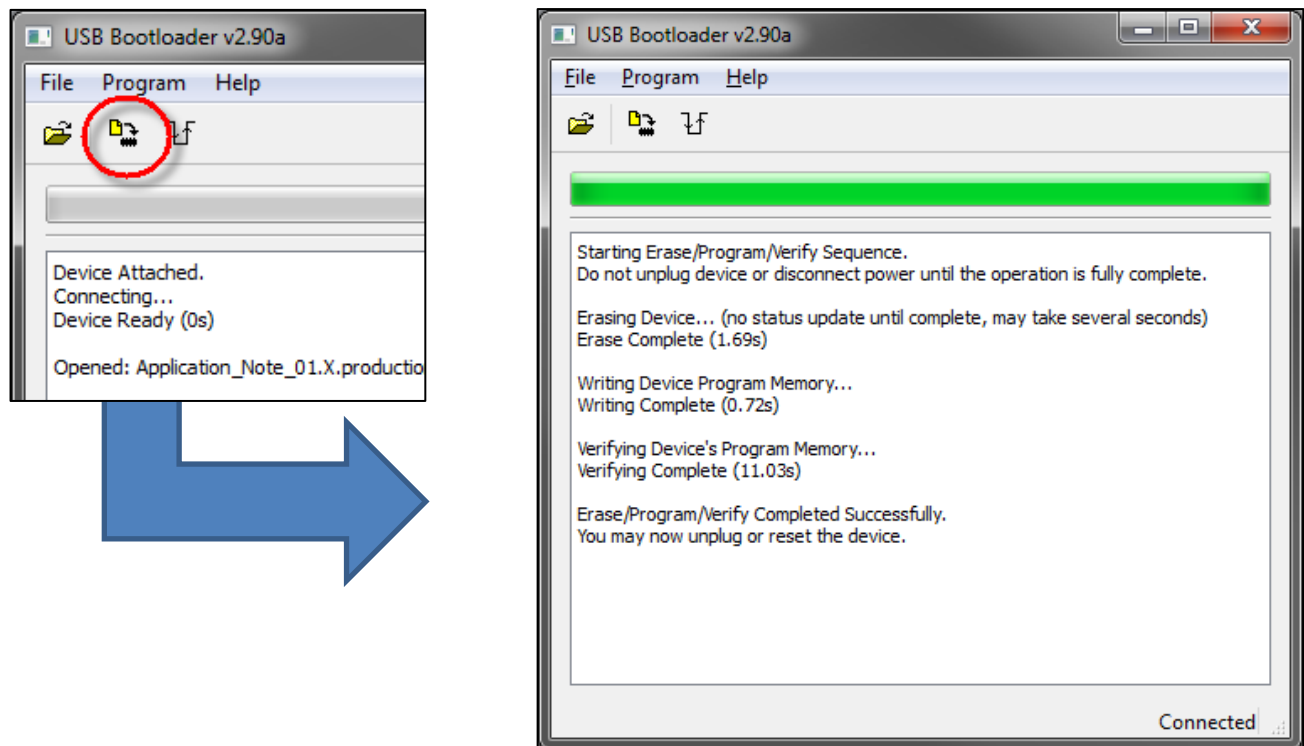
La ventana del Bootloader indicará la conexión establecida con el aguijón:

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

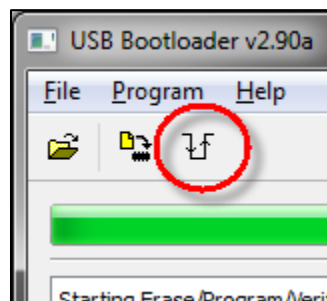


El archivo, depende de la plataforma de hardware.

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.

9. Para verificar el funcionamiento del programa verifique que el conteo en pantalla no se pierde aun y cuando se quita el suministro de energía, o se realiza un Reset.