



Aguijón

Notas de aplicación

Nota de aplicación 20:

Watchdog Timer

Descripción:

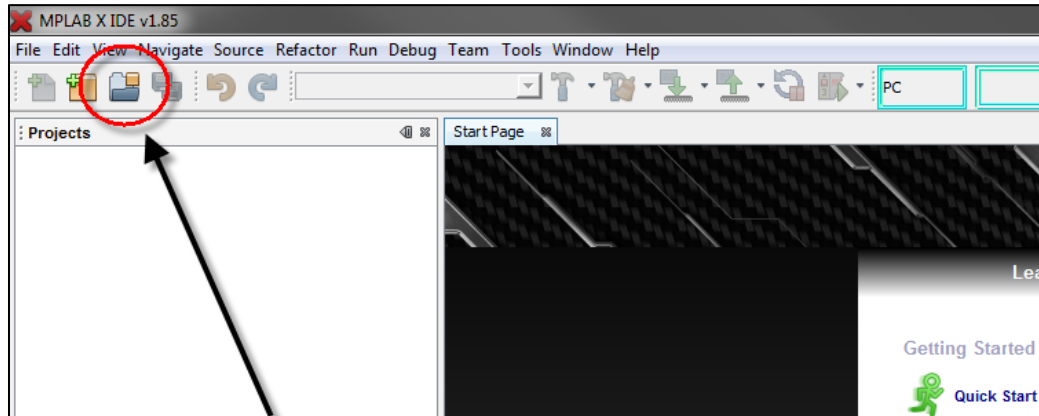
Provoca un reset del sistema desde el programa utilizando el Watchdog Timer. Así como habilitar, deshabilitar, y atender a el mismo.

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.

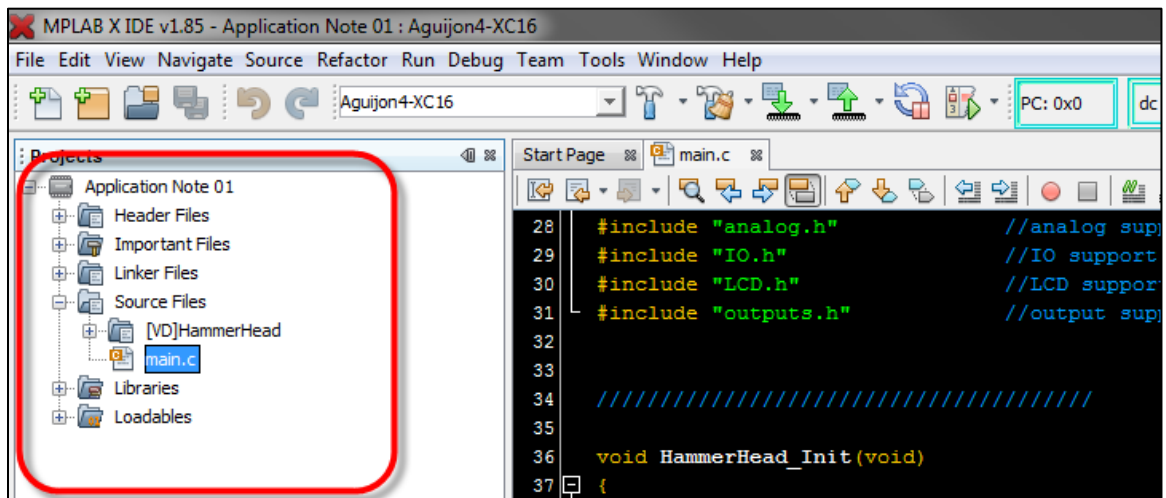
Pasos:

1. Abrir MPLAB X® y cargar el archivo del proyecto: **Application Note 20.X**



Haz 'clic' aquí y
abre el proyecto

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



3. Ir a la línea #77

Utilizaremos la siguiente función:

```
70 LCD_PutStr(2,0,"Application Note 20",OFF);
71 delayms(500);
72 LCD_Clear();
73
74 for(;;)
75 {
76     //Create a strings for the LCD//
77     LCD_PutStr(1,0,"Presione un boton",FALSE);
78     WDOG_SoftEnable();           //WDOG Timer Enable
79
80     if(DIPSW_Read()==1)         //If DIP SW is 1
81     {                             //Then...
82         WDOG_SoftDisable();      //WDOG Timer Disable
83     }
84 }
```

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).

4. Ir a la línea #78.

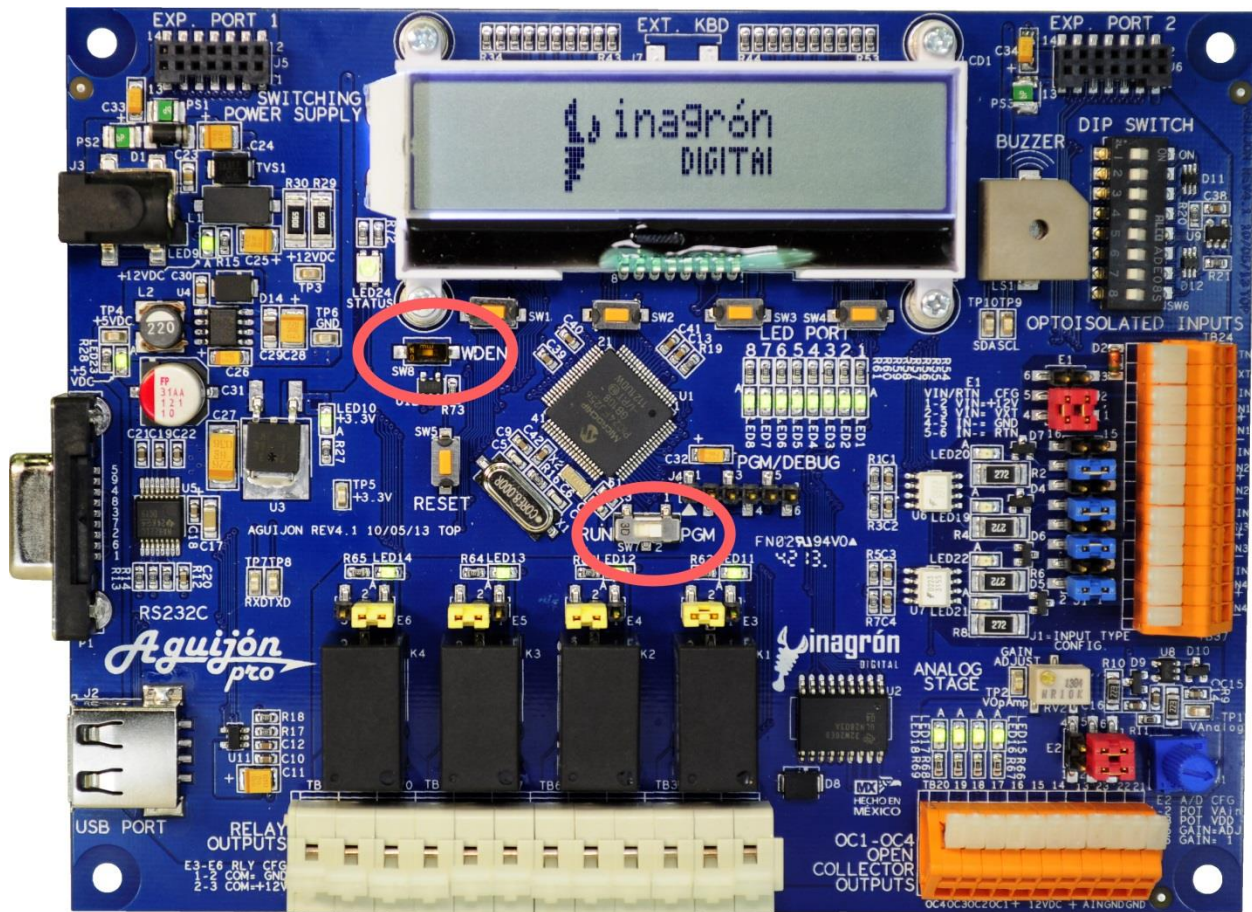
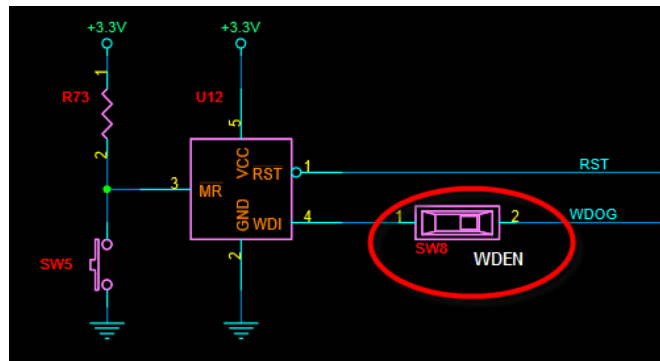
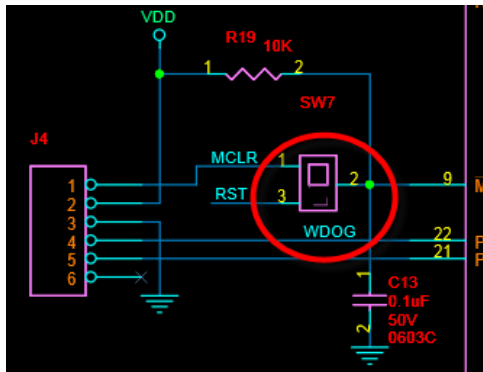
Utilizaremos la siguiente función:

```
71     delayms(500);
72     LCD_Clear();
73
74     for(;;)
75     {
76         //Create a strings for the LCD//
77         LCD_PutStr(1,0,"Presione un boton",FALSE);
78         WDOG_SoftEnable(); //WDOG Timer Enable
79
80         if(DIPSW_Read()==1) //If DIP SW is 1
81         { //Then...
82             WDOG_SoftDisable(); //WDOG Timer Disable
83         }
84
85         if(SW_ReadPort()) //If SW Port is true.
```

WDOG_SoftEnable ();

- Esta función habilita la interrupción por desbordamiento del Watchdog Timer y este a su vez genera un reset en el sistema.

Para que el reset generado al desbordamiento del Watchdog timer sea efectivo es necesario posicionar el **WS8** en la posición **WDEN**, así como el **SW7** en la posición **RUN**, esto en nuestra tablilla Aguijón



5. Ir a la línea #80.

Utilizaremos la siguiente función:

```
73
74     for(;;)
75     {
76         //Create a strings for the LCD//
77         LCD_PutStr(1,0,"Presione un boton",FALSE);
78         WDOG_SoftEnable();           //WDOG Timer Enable
79
80         if(DIPSW_Read()==1)         //If DIP SW is 1
81         {                             //Then...
82             WDOG_SoftDisable();       //WDOG Timer Disable
83         }
84
85         if(SW_ReadPort())            //If SW Port is true.
86         {                             //Then...
87             for(i=0;i<=10;i++)        //Counting from 0 to 10
```

DIPSW_Read ();

- Esta función lee el puerto de DIP-SWITCH;
Regresa un valor equivalente al valor del DIP-SWITCH (carácter del 1 al 255)

6. Ir a la línea #82.

Utilizaremos la siguiente función:

```
75     {
76         //Create a strings for the LCD//
77         LCD_PutStr(1,0,"Presione un boton",FALSE);
78         WDOG_SoftEnable();           //WDOG Timer Enable
79
80         if(DIPSW_Read()==1)         //If DIP SW is 1
81         {                           //Then...
82             WDOG_SoftDisable();      //WDOG Timer Disable
83         }
84
85         if(SW_ReadPort())           //If SW Port is true.
86         {                           //Then...
87             for(i=0;i<=10;i++)      //Counting from 0 to 10
88             {
89                 WDOG_Pet();         //Refresh the watchdog c
```

WDOG_SoftDisable ();

- Esta función deshabilita la interrupción por desbordamiento del Watchdog Timer.

7. Ir a la línea #85.

Utilizaremos la siguiente función:

```
78     WDOG_SoftEnable();           //WDOG Timer Enable
79
80     if (DIPSW_Read()==1)         //If DIP SW is 1
81     {                             //Then...
82         WDOG_SoftDisable();       //WDOG Timer Disable
83     }
84
85     if (SW_ReadPort())            //If SW Port is true.
86     {                             //Then...
87         for (i=0;i<=10;i++)        //Counting from 0 to 10
88         {
89             WDOG_Pet();            //Refresh the watchdog c
90             //Create a strings for the LCD//
91             sprintf(lcdMSG,"TIME = %i",i);
92             LCD_PutStr(1,0,lcdMSG,TRUE);
```

SW_ReadPort ();

- Esta función lee el puerto de PUSH BUTTONS;
Regresa un valor equivalente al valor del puerto de Push Buttons
(Carácter del 1 al 15 (4 Bits))

8. Ir a la línea #89.

Utilizaremos la siguiente función:

```
82         WDOG_SoftDisable();           //WDOG Timer Disable
83     }
84
85     if(SW_ReadPort())                 //If SW Port is true.
86     {                                 //Then...
87         for(i=0;i<=10;i++)           //Counting from 0 to 10
88         {
89             WDOG_Pet();               //Refresh the watchdog
90             //Create a strings for the LCD//
91             sprintf(lcdMSG,"TIME = %i",i);
92             LCD_PutStr(1,0,lcdMSG,TRUE);
93             delayms(250);              //Delay 250 millisecond
94         }
95     }
96
```

WDOG_Pet();

- Esta función reinicia el conteo del Watchdog Timer, de esta manera al ser llamada continuamente evitara un reset provocado por el desbordamiento del mismo, al estar este activado.

9. Ir a la línea #91

Utilizaremos la siguiente función:

```
84
85     if(SW_ReadPort())                //If SW Port is true.
86     {                                //Then...
87         for(i=0;i<=10;i++)            //Counting from 0 to 10
88         {
89             WDOG_Pet();                //Refresh the watchdog c
90             //Create a strings for the LCD//
91             sprintf(lcdMSG,"TIME = %i",i);
92             LCD_PutStr(1,0,lcdMSG,TRUE);
93             delayms(250);                //Delay 250 milliseconds
94         }
95     }
96
97 }
98 return 0;
```

`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.

10. Ir a la línea #93.

Utilizaremos la siguiente función:

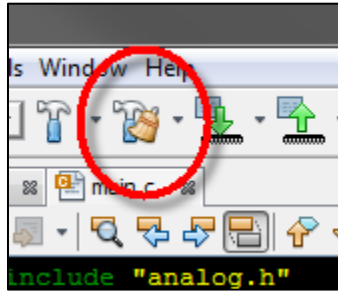
```
86     {                                     //Then...
87         for(i=0;i<=10;i++)               //Counting from 0 to 10
88     {
89         WDOG_Pet();                       //Refresh the watchdog c
90         //Create a strings for the LCD//
91         sprintf(lcdMSG,"TIME = %i",i);
92         LCD_PutStr(1,0,lcdMSG,TRUE);
93         delayms(250);                     //Delay 250 milliseconds
94     }
95 }
96
97 }
98 return 0;
99 }
```

Delayms (ms);

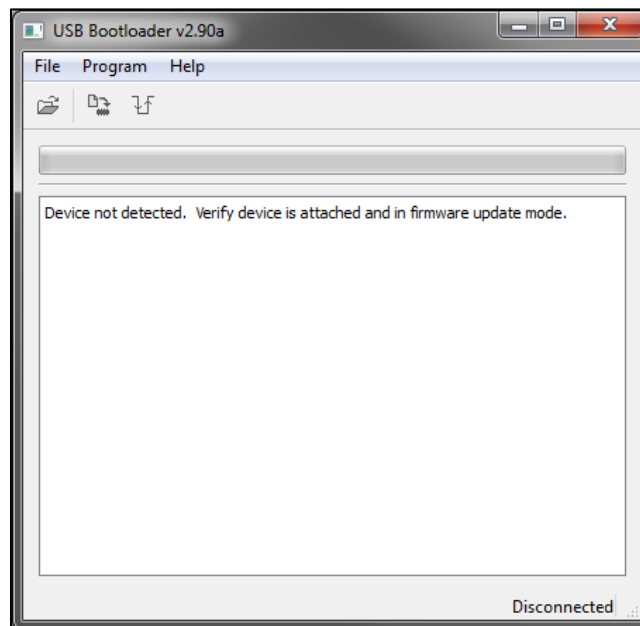
- Función que Pausa el programa por un tiempo determinado (en milisegundos); donde:
ms = el número de milisegundos que se desea pausar el programa.

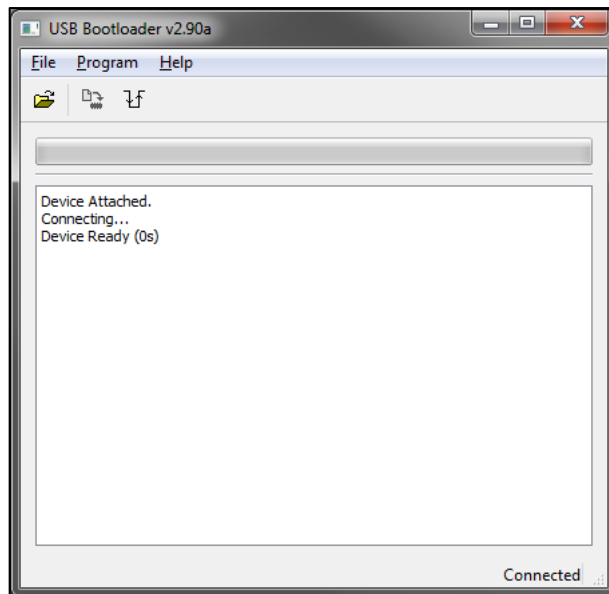
11. Compilar y programar

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



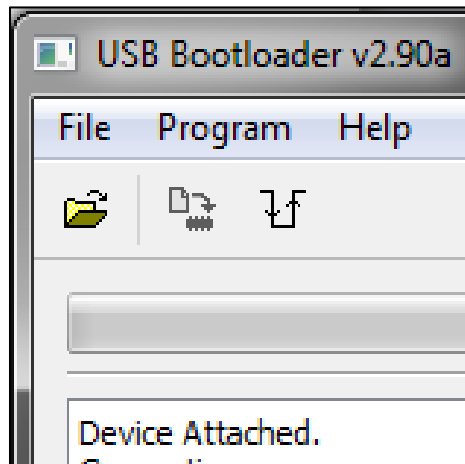
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





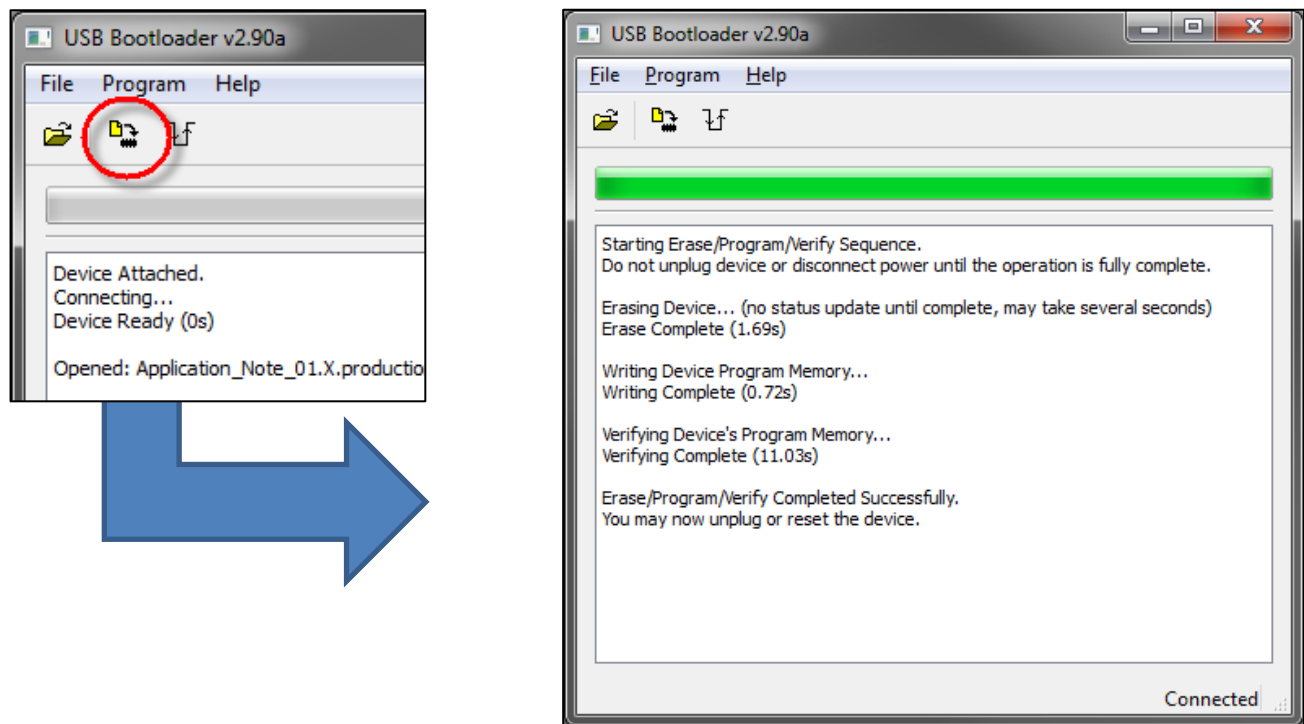
La ventana del Bootloader indicará la conexión establecida con el aguijón:

12. Hacer Clic en Abrir y Cargar el archivo **Application Note 20.X.production.hex**

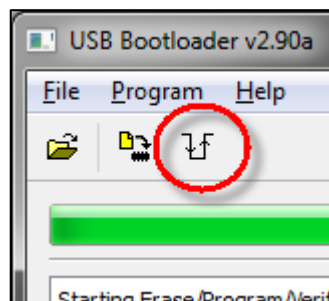


El archivo, depende de la plataforma de hardware.

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 aparecer la leyenda “presione un botón”, si este no es presionado en un corto tiempo, se generara un reset gracias al Watchdog timer, para evitar este reset es necesario activar el primer bit del puerto del Dip Switch.

Si el botón es presionado antes del desbordamiento del Watchdog se generara un conteo ascendente del 0 al 10, para luego comenzar de nuevo nuestro programa.