



Aguijón

Notas de aplicación

Nota de aplicación 19:

Temperature Control

Descripción:

Simular un sistema de control de temperatura de una habitación.

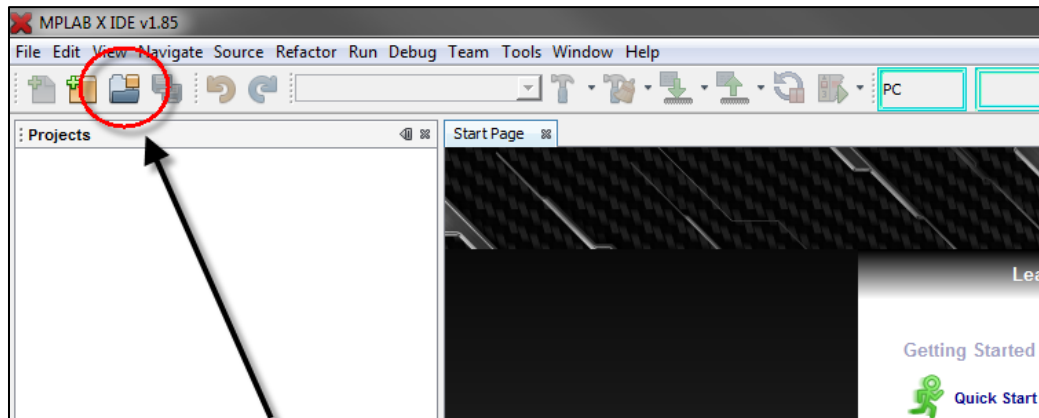
Para una simulación más didáctica, conecte un sensor de temperatura a la entrada del puerto ADC, esta a su vez estará siendo monitoreada continuamente, mientras este dentro de un límite superior e inferior, si esta se sale del rango activara una salida open collector, o un relay, es en estos donde se puede conectar un ventilador y un foco, para simular los estados de temperatura.

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 19.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  HammerHead_Init(); //initialize [VD]HammerHead
71  LCD_IntroAnimation();
72  LCD_PutStr(1,0,"Vinagron Digital",TRUE);
73  LCD_PutStr(2,0,"Application Note 19",FALSE);
74
75  for(;;){
76
77      temp=ADC_Range(-20,60); //Read ADC input
78
79      if(temp>=max_temp) //If temp is greater than max te
80      { //Then..
81          RLY_Set(1,ON); //Turn it ON
82      }
83      else
84      {
```

ADC_Range (int min_val, int max_val);

- Esta función lee el puerto ADC y devuelve un valor proporcional al Rango establecido;
Donde:
Int min_val = valor mínimo del rango (Entero de 0 a 1023)
Int max_val = valor máximo del rango (Entero de 0 a 1023)
Devuelve un valor proporcional al rango (Entero de min_val a max_val).

4. Ir a la línea #81

Utilizaremos la siguiente función:

```
74
75     for(;;){
76
77         temp=ADC_Range(-20,60);           //Read ADC input
78
79         if(temp>=max_temp)                 //If temp is greater than max te
80         {                                  //Then..
81             RLY_Set(1,ON);                 //Turn it ON
82         }
83         else
84         {
85             RLY_Set(1,OFF);                //Turn it OFF
86         }
87         if(temp<=min_temp)                 //If temp is less than max temp
88         {                                  //Then..
```

RLY_Set (int num, BOOL state);

- Función que enciende y apaga un RELAY; donde:
Int num = Número de RELAY que queremos conmutar (Valor entero del 1 al 4.)
BOOL state = Estado del RELAY (ON = Encendido, OFF = apagado).

5. Ir a la línea #89

Utilizaremos la siguiente función:

```
82     }
83     else
84     {
85         RLY_Set(1,OFF);           //Turn it OFF
86     }
87     if(temp<=min_temp)           //If temp is less than max temp
88     {                             //Then..
89         OC_Set(1,ON);             //Turn it ON
90     }
91     else
92     {
93         OC_Set(1,OFF);           //Turn it OFF
94     }
95
96     //Create a strings for the LCD
```

OC_Set (int num, BOOL state);

- Función que enciende y apaga un OPEN COLLECTOR; donde:
Int num = Número de OPEN COLLECTOR que queremos conmutar
(Valor entero del 1 al 4.)
BOOL state = Estado del OPEN COLECTOR (ON = Encendido, OFF = apagado).

6. Ir a la línea #97

Utilizaremos la siguiente función:

```
90     }
91     else
92     {
93         OC_Set(1,OFF);           //Turn it OFF
94     }
95
96     //Create a strings for the LCD
97     sprintf(lcdMSG1,"Temperatura =%li",temp);
98     sprintf(lcdMSG2,"Temp Max %li Min %li",max_temp,min_temp);
99
100    //Display it in (y1,x0) and (y2,x0)
101    LCD_PutStr(1,0,lcdMSG1,TRUE);
102    LCD_PutStr(2,0,lcdMSG2,FALSE);
103
104    delayms(150);                //Delay 150 milliseconds
```

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.

7. Ir a la línea #101

Utilizaremos la siguiente función:

```
94     }
95
96     //Create a strings for the LCD
97     sprintf(lcdMSG1,"Temperatura =%li",temp);
98     sprintf(lcdMSG2,"Temp Max %li Min %li",max_temp,min_temp);
99
100    //Display it in (y1,x0) and (y2,x0)
101    LCD_PutStr(1,0,lcdMSG1,TRUE);
102    LCD_PutStr(2,0,lcdMSG2,FALSE);
103
104    delayms(150); //Delay 150 milliseconds
105 }
106 return 0;
107 }
```

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).

8. Ir a la línea #104.

Utilizaremos la siguiente función:

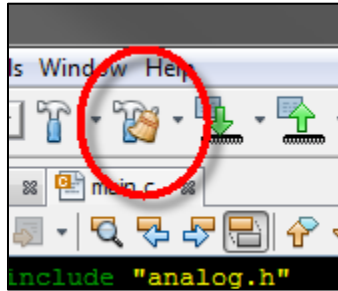
```
93         OC_Set(1,OFF);           //Turn it OFF
94     }
95
96     //Create a strings for the LCD
97     sprintf(lcdMSG1,"Temperatura =%li",temp);
98     sprintf(lcdMSG2,"Temp Max %li Min %li",max_temp,min_temp);
99
100    //Display it in (y1,x0) and (y2,x0)
101    LCD_PutStr(1,0,lcdMSG1,TRUE);
102    LCD_PutStr(2,0,lcdMSG2,FALSE);
103
104    delayms(150);                  //Delay 150 milliseconds
105 }
106 return 0;
107 }
```

Delayms (ms);

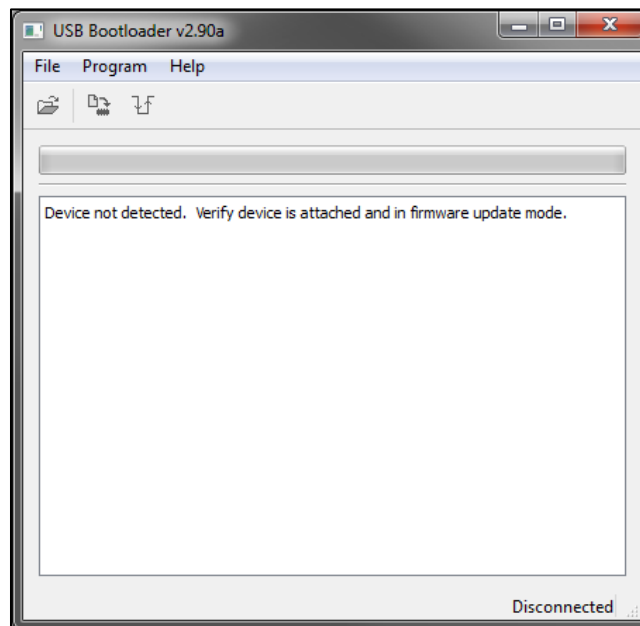
- Función que Pausa el programa por un tiempo determinado (en milisegundos); donde:
ms = el número de milisegundos que se desea pausar el programa.

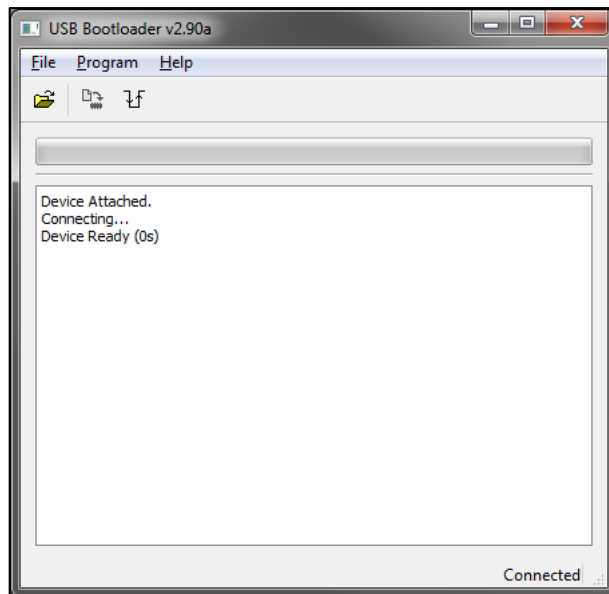
9. Compilar y programar

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



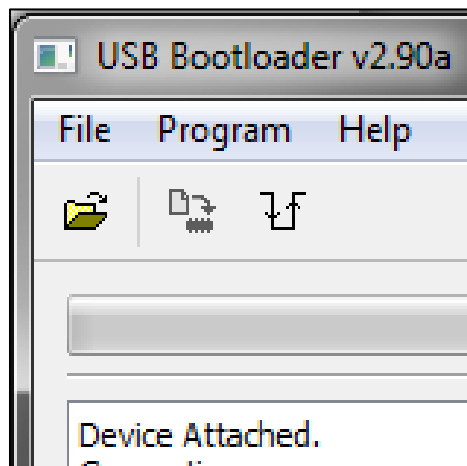
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





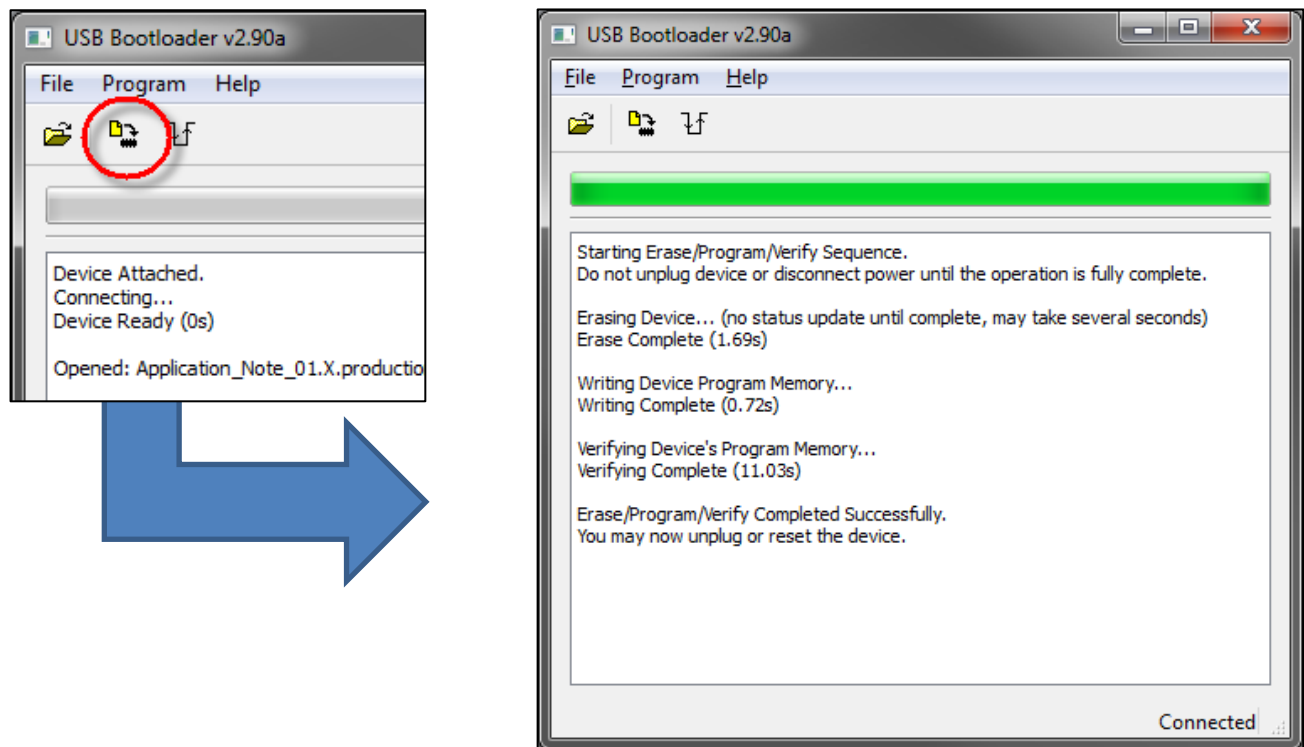
La ventana del Bootloader indicará la conexión establecida con el aguijón:

10. Hacer Clic en Abrir y Cargar el archivo **Application Note 19.X.production.hex**

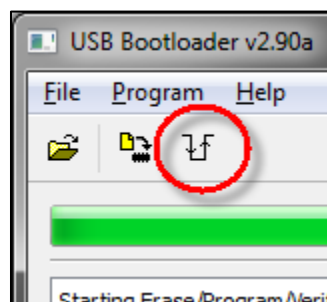


El archivo, depende de la plataforma de hardware.

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 en la pantalla LCD se muestre continuamente el valor de la temperatura, así mismo al salir esta de sus límites active el periférico indicado (máximo = Relay 1 = Ventilador) ,(mínimo = OC 1 = Foco).