



*Aguijón*

---

*Notas de aplicación*

## **Nota de aplicación 17:**

### Numerical systems

#### **Descripción:**

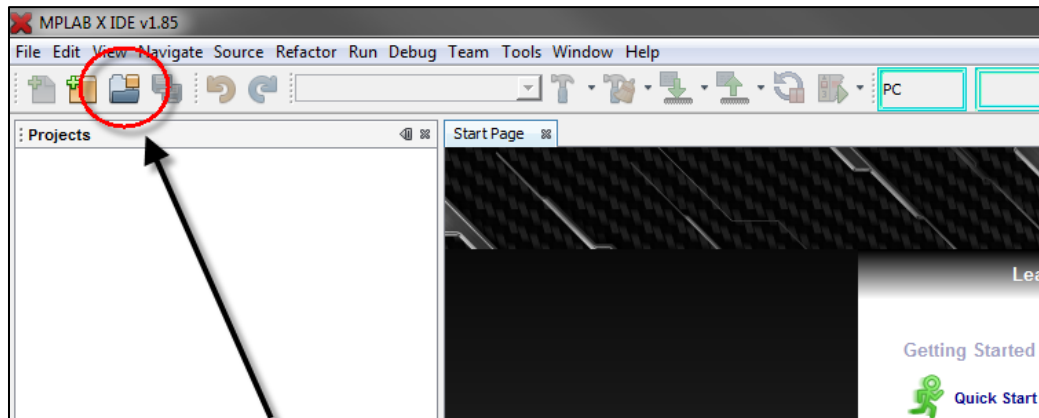
*Mostrar distintos tipos de sistemas numéricos en la pantalla LCD.*

#### **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 17.X**



Haz 'clic' aquí y  
abre el proyecto

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



3. Ir a la línea #90.

Utilizaremos la siguiente función:

```
83 LCD_PutStr(1,0,"Vinagron Digital",TRUE);
84 LCD_PutStr(2,0,"Application Note 17",FALSE);
85 delayms(1000);
86
87 for(;;){
88
89     //read DIPsw and store it in DIPvalue var
90     DIPvalue = DIPSW_Read();
91
92     //create a string with sprintf() and displayit in Y1 X0
93     sprintf(lcdMSG,"DEC: %u, HEX: %.2X",DIPvalue,DIPvalue);
94     LCD_PutStr(1,0,lcdMSG,TRUE);    //display it on Y1X0
95
96     //Using sprintf() and byte_to_binary(),create a string for the LCD
97     sprintf(lcdMSG,"BIN: %s",byte_to_binary(DIPvalue));
```

#### **DIPSW\_Read ();**

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

#### 4. Ir a la línea #93

Utilizaremos la siguiente función:

```
86
87     for(;;){
88
89         //read DIPsw and store it in DIPvalue var
90         DIPvalue = DIPSW_Read();
91
92         //create a string with sprintf() and displayit in Y1 X0
93         sprintf(lcdMSG,"DEC: %u, HEX: %.2X",DIPvalue,DIPvalue);
94         LCD_PutStr(1,0,lcdMSG,TRUE);    //display it on Y1X0
95
96         //Using sprintf() and byte_to_binary(),create a string for the LCD
97         sprintf(lcdMSG,"BIN: %s",byte_to_binary(DIPvalue));
98         LCD_PutStr(2,0,lcdMSG,FALSE);
99
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.

**Const char**, si se desea incluir una variable es necesario incluir los especificadores de formato, y estos deben ser los adecuados para cada tipo de variable.

- **Especificadores de formato**

Especificador	Descripción
%c	Carácter
%d	Número entero(int)
%i	Número entero(int)
%D	Número entero long(o también %ld)
%f	Punto flotante(float)
%e	Notación científica con e minúscula
%E	Notación científica con E mayúscula
%g	Formato para tipo punto flotante(float)
%G	Formato para tipo punto flotante(float)
%o	Número octal sin signo
%s	Cadena de texto
%u	Entero sin signo
%U	Entero sin signo long(o también %lu)
%x	Hexadecimal sin signo con minúsculas
%X	Hexadecimal sin signo con mayúsculas
%p	Puntero, dirección de memoria
%n	Número de caracteres
%o	Formato entero octal
%O	Formato entero octal long(o también %lo)
%lf	Formato double
%LF	Formato long double
%l	Formato double
%h	Formato double
%L	Formato long double

5. Ir a la línea #98

Utilizaremos la siguiente función:

```
91
92     //create a string with sprintf() and displayit in Y1 X0
93     sprintf(lcdMSG,"DEC: %u, HEX: %.2X",DIPvalue,DIPvalue);
94     LCD_PutStr(1,0,lcdMSG,TRUE);    //display it on Y1X0
95
96     //Using sprintf() and byte_to_binary(),create a string for the LCD
97     sprintf(lcdMSG,"BIN: %s",byte_to_binary(DIPvalue));
98     LCD_PutStr(2,0,lcdMSG,FALSE);
99
100
101     LEDport_Set(DIPvalue,ON);    //set the LEDport
102
103     delayms(150);    //Delay 150 milliseconds
104
105 }
```

**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 #101

Utilizaremos la siguiente función:

```
94     LCD_PutStr(1,0,lcdMSG,TRUE);    //display it on Y1X0
95
96     //Using sprintf() and byte_to_binary(),create a string for the LCD
97     sprintf(lcdMSG,"BIN: %s",byte_to_binary(DIPvalue));
98     LCD_PutStr(2,0,lcdMSG,FALSE);
99
100
101     LEDport_Set(DIPvalue,ON);        //set the LEDport
102
103     delayms(150);                    //Delay 150 milliseconds
104
105 }
106 return 0;
107 }
```

**LEDport\_Set (char value, BOOL invert);**

- Función que enciende y apaga el puerto de LEDs; donde:  
**Char value** = Número que queremos Encender (Valor entero del 1 al 255.)  
**BOOL invert** = Estado del puerto de LEDs (ON = Encendido, OFF = apagado).



7. Ir a la línea #103

Utilizaremos la siguiente función:

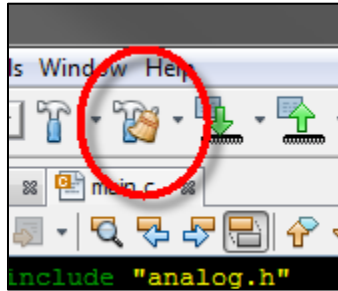
```
96      //Using sprintf() and byte_to_binary(),create a string for the LCD
97      sprintf(lcdMSG,"BIN: %s",byte_to_binary(DIPvalue));
98      LCD_PutStr(2,0,lcdMSG,FALSE);
99
100
101      LEDport_Set(DIPvalue,ON);      //set the LEDport
102
103      delayms(150);                  //Delay 150 milliseconds
104
105  }
106  return 0;
107 }
```

**Delayms (ms);**

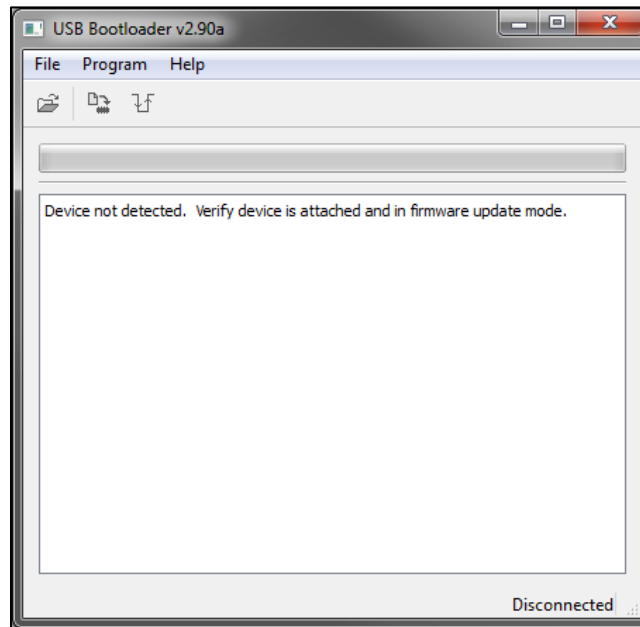
- Función que Pausa el programa por un tiempo determinado (en milisegundos); donde:  
**ms** = el número de milisegundos que se desea pausar el programa.

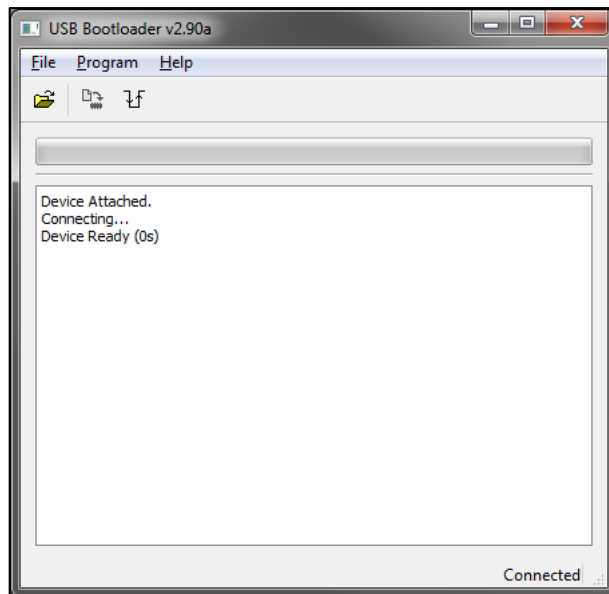
## 8. Compilar y programar

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



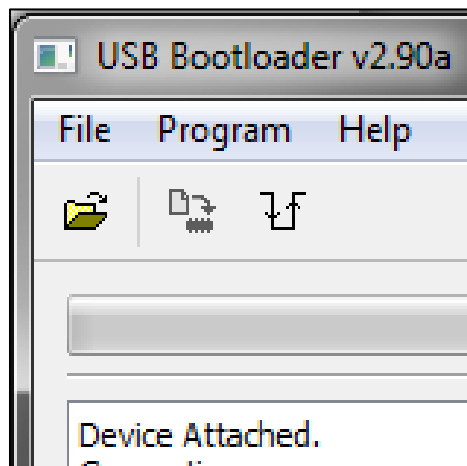
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





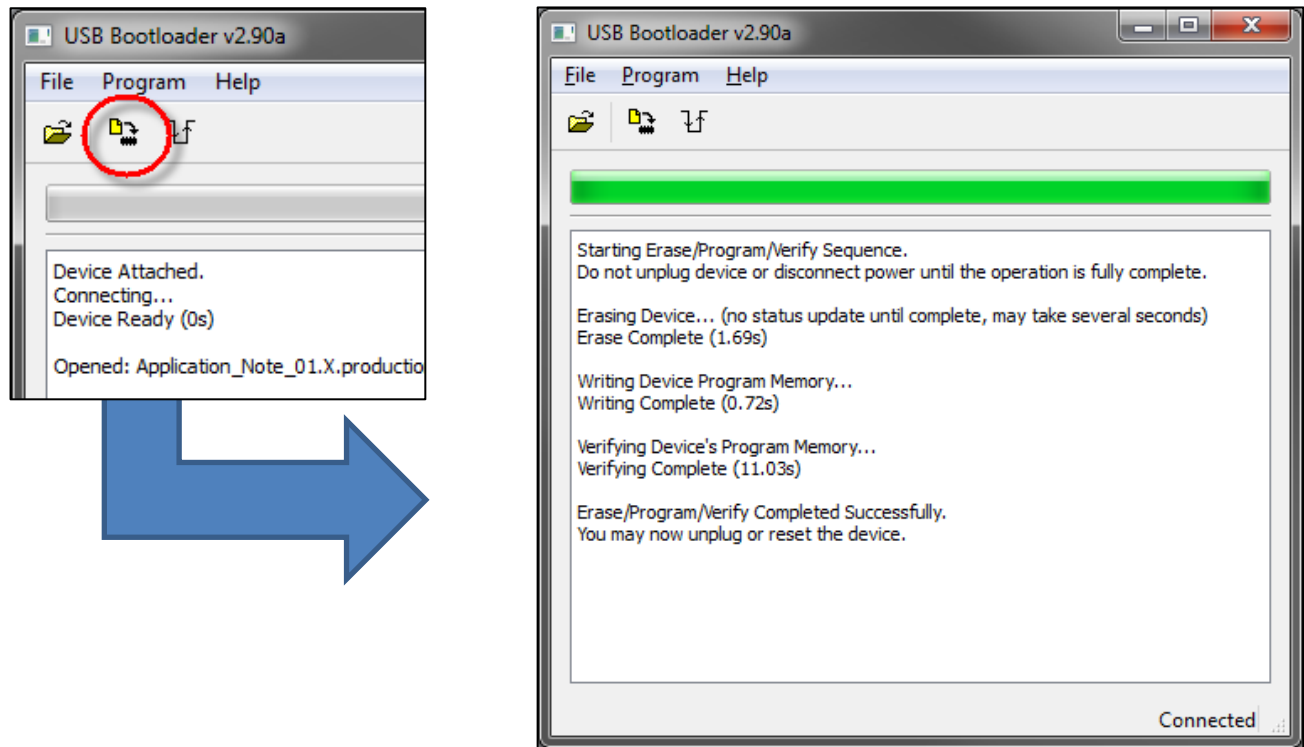
La ventana del Bootloader indicará la conexión establecida con el aguijón:

9. Hacer Clic en Abrir y Cargar el archivo **Application Note 17.X.production.hex**

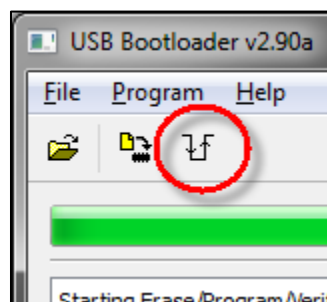


El archivo, depende de la plataforma de hardware.

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.

10. Para verificar el funcionamiento del programa ingrese un numero en el puerto DIP-Switch y verifique que la pantalla LCD lo muestre en formato, decimal, hexadecimal y binario.