



Aguijón

Notas de aplicación

Nota de aplicación 04:

ADC and Blinking Led

Descripción:

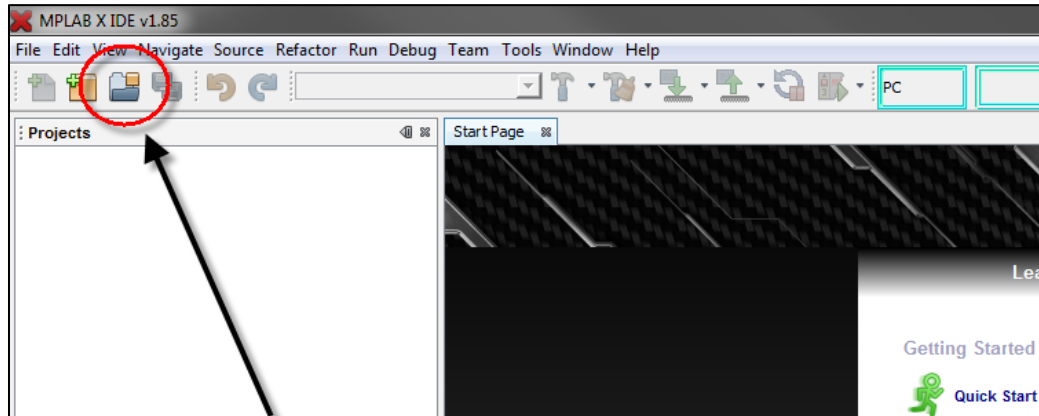
Conmutar un LED desde el puerto de entrada ADC.

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 05.X**



Haz 'clic' aquí y
abre el proyecto

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



3. Ir a la línea #73.

Utilizaremos la siguiente función:

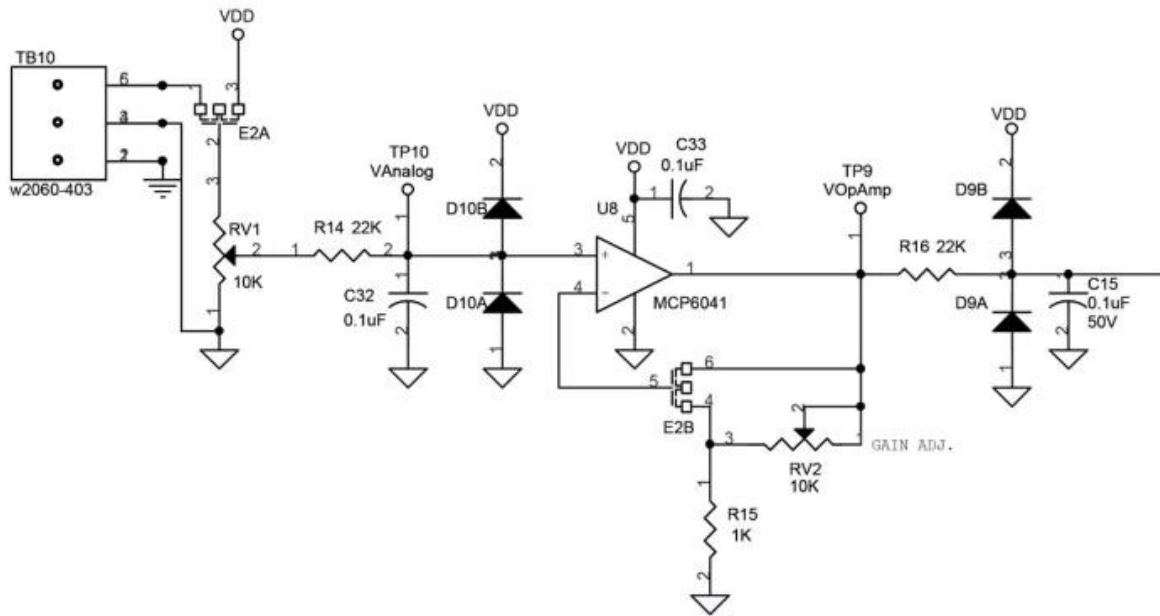
```
66     HammerHead_Init(); //initialize [VD]HammerHead
67     LCD_IntroAnimation();
68     LCD_PutStr(1,0,"Vinagron Digital",TRUE);
69     LCD_PutStr(2,0,"Application Note 05",FALSE);
70
71     for(;;){
72
73         value=ADC_Read(); //Read ADC
74
75         LED_Toggle(1); //Change logic state
76         delayms(value); //ADC delay value
77
78     }
79     return 0;
80 }
```

ADC_Read ();

- Esta función lee el puerto ADC;
Regresa un valor equivalente al Valor análogo introducido (entero del 0 al 1024)

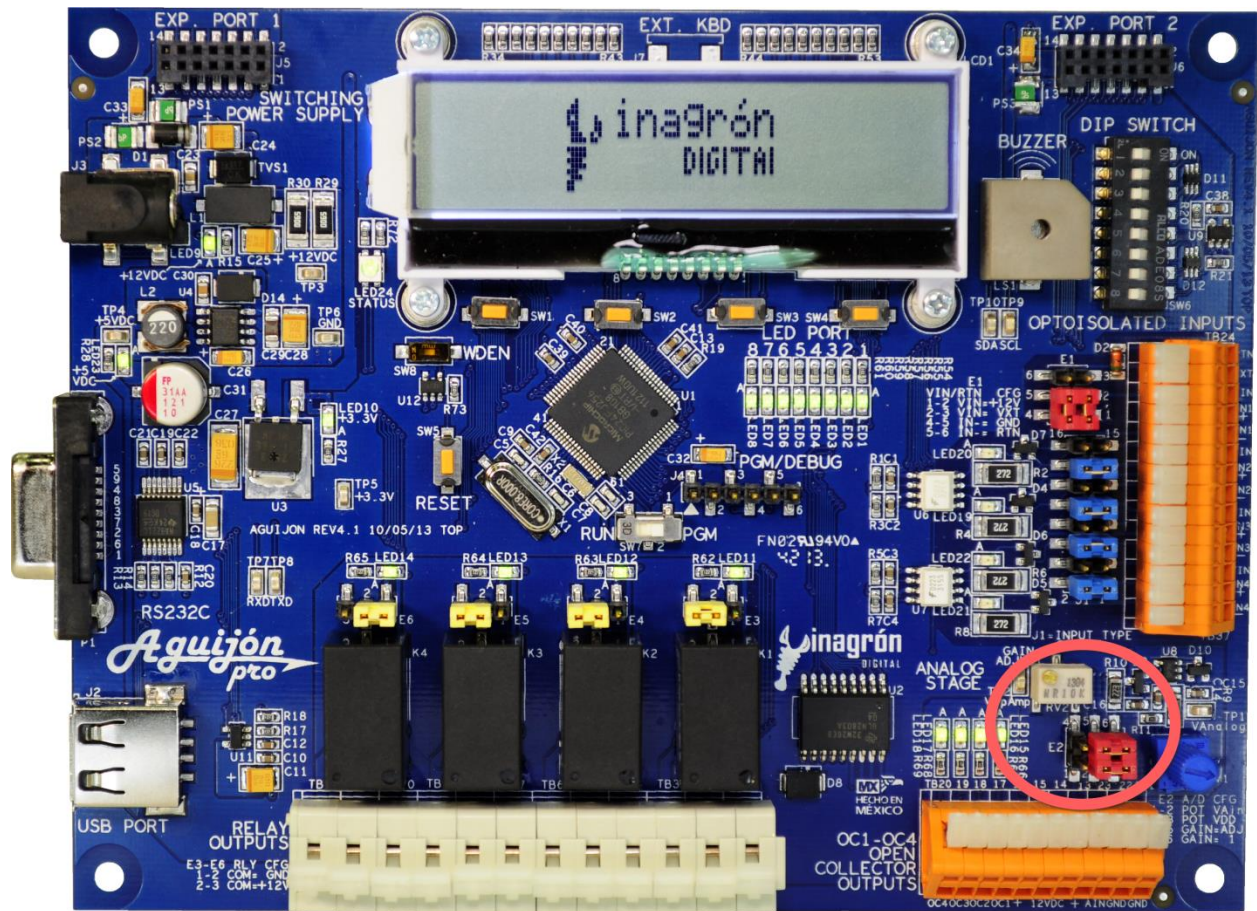
4. Aguijón ofrece una entrada analógica con:

- Ganancia unitaria o ajustable.
- Conversión digital de 10 bits, $v_{ref}=+3.3V$
- Protección contra sobre voltaje
- Potenciómetro en tablilla para atenuar $+3.3V$ o un voltaje externo.

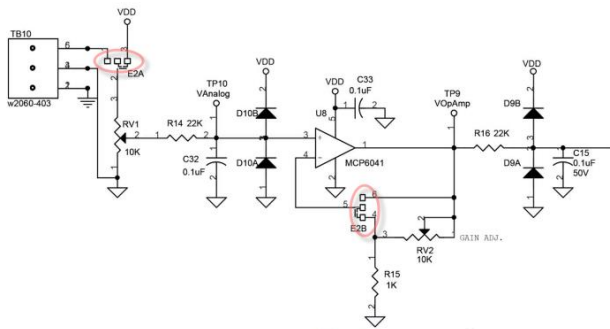


• TABLA DE CONFIGURACION

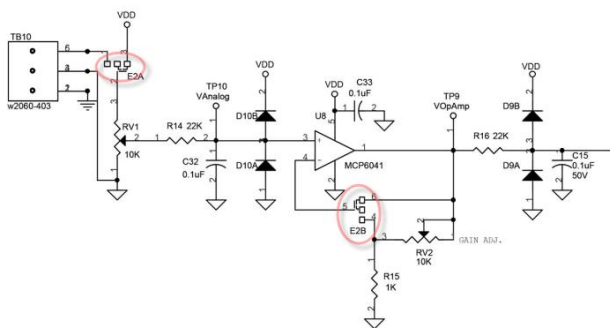
Jumper E2A	Jumper E2B	Ganancia	Entrada analógica	RV1
1-2	5-6	Unitaria	Externa	Atenúa señal externa
2-3	5-6		Interna (+3.3V)	Atenúa +3.3V
1-2	4-5	Ajustable con RV2	Externa	Atenúa señal externa
2-3	4-5		Interna (+3.3V)	Atenúa +3.3V



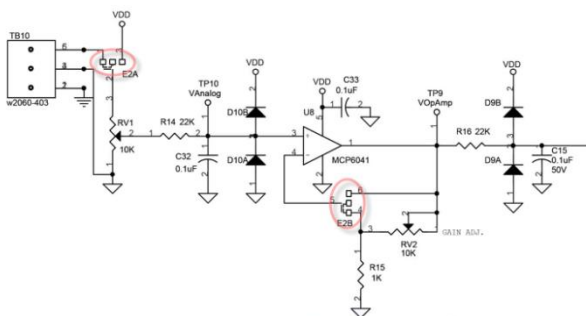
- **Configuración ganancia ajustable, entrada analógica interna.**



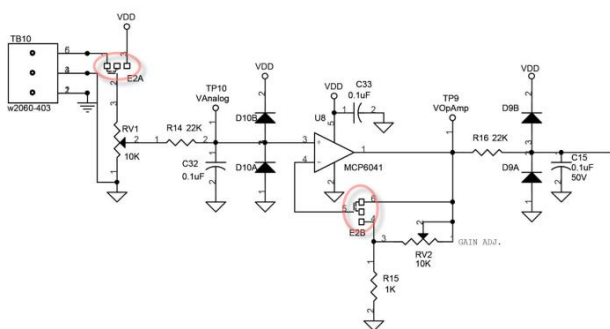
- **Configuración ganancia unitaria, entrada analógica interna.**



- **Configuración ganancia ajustable, entrada analógica externa**



- **Configuración ganancia unitaria, entrada analógica interna**



5. Ir a la línea #75

Utilizaremos la siguiente función:

```
68 LCD_PutStr(1,0,"Vinagron Digital",TRUE);
69 LCD_PutStr(2,0,"Application Note 05",FALSE);
70
71 for(;;){
72
73     value=ADC_Read();           //Read ADC
74
75     LED_Toggle(1);              //Change logic state
76     delayms(value);             //ADC delay value
77
78 }
79 return 0;
80 }
```

LED_Toogle (int num);

- Función que cambia el estado lógico de un LED; donde:
Int num = Número de OPEN COLLECTOR que queremos conmutar
(Valor entero del 1 al 8.)

6. Ir a la línea #76

Utilizaremos la siguiente función:

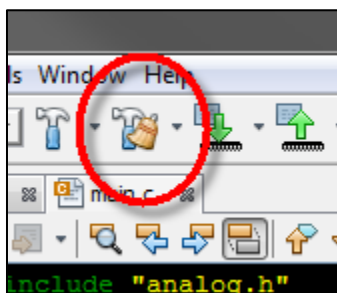
```
69     LCD_PutStr(2,0,"Application Note 05",FALSE);
70
71     for(;;) {
72
73         value=ADC_Read();           //Read ADC
74
75         LED_Toggle(1);              //Change logic state
76         delayms(value);             //ADC delay value
77
78     }
79     return 0;
80 }
```

Delayms (ms);

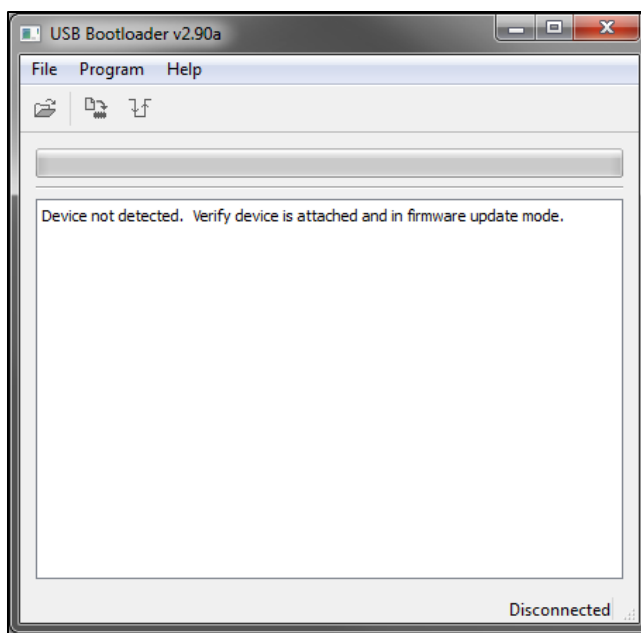
- Función que Pausa el programa por un tiempo determinado (en milisegundos); donde:
ms = el número de milisegundos que se desea pausar el programa.

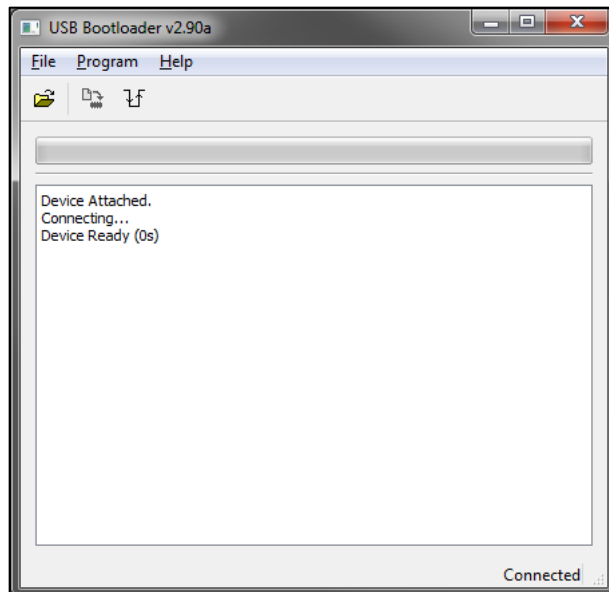
7. Compilar y programar

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



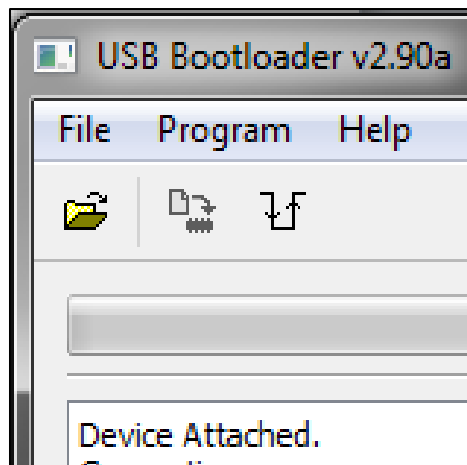
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.





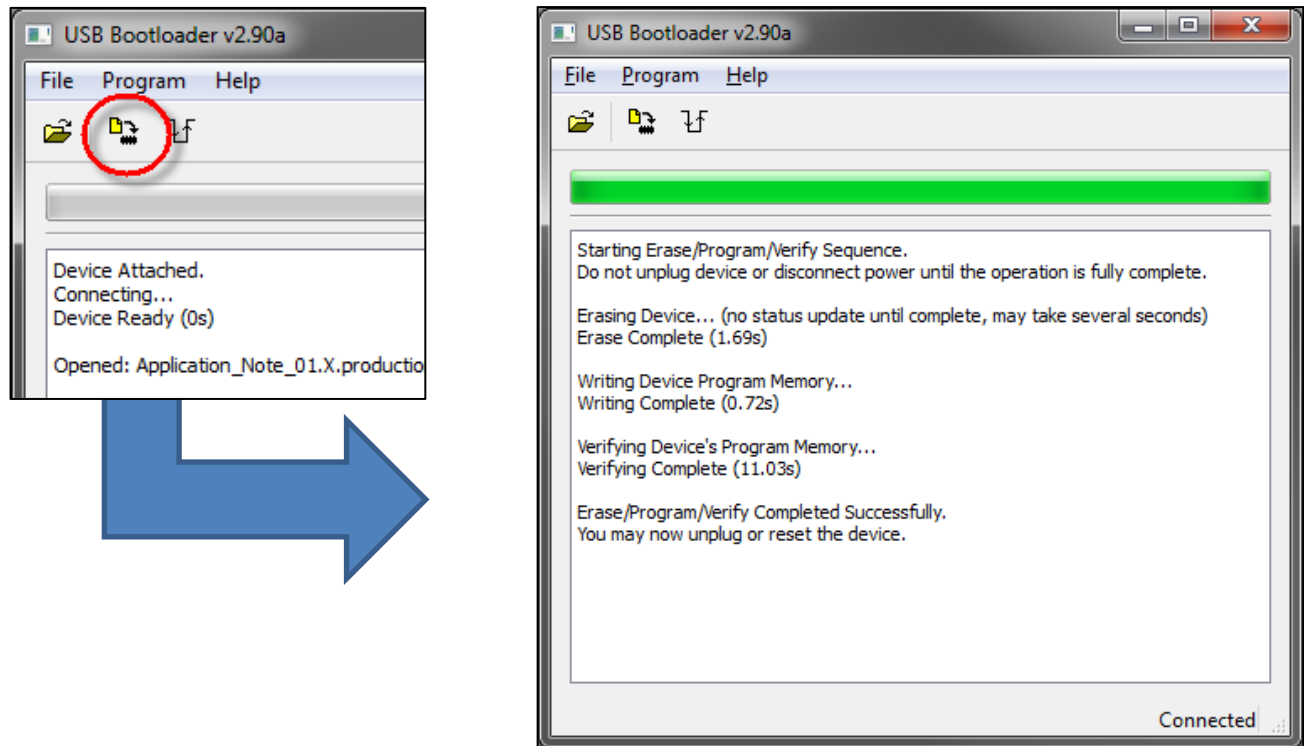
La ventana del Bootloader indicará la conexión establecida con el aguijón:

8. Hacer Clic en Abrir y Cargar el archivo **Application Note 05.X.production.hex**

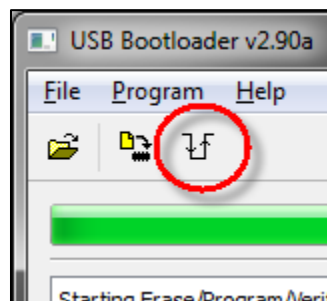


El archivo, depende de la plataforma de hardware.

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 LED 1 Conmute a una velocidad proporcional al valor del ADC.