**Programando el ATtiny85: Bitácora de errores y soluciones**

**Introducción**

Este tutorial documenta mi experiencia programando el **ATtiny85**, incluyendo el hardware utilizado, los errores que encontré y las soluciones que funcionaron. Además, acompaño con capturas de pantalla de configuraciones y programas exitosos.

**Hardware utilizado**

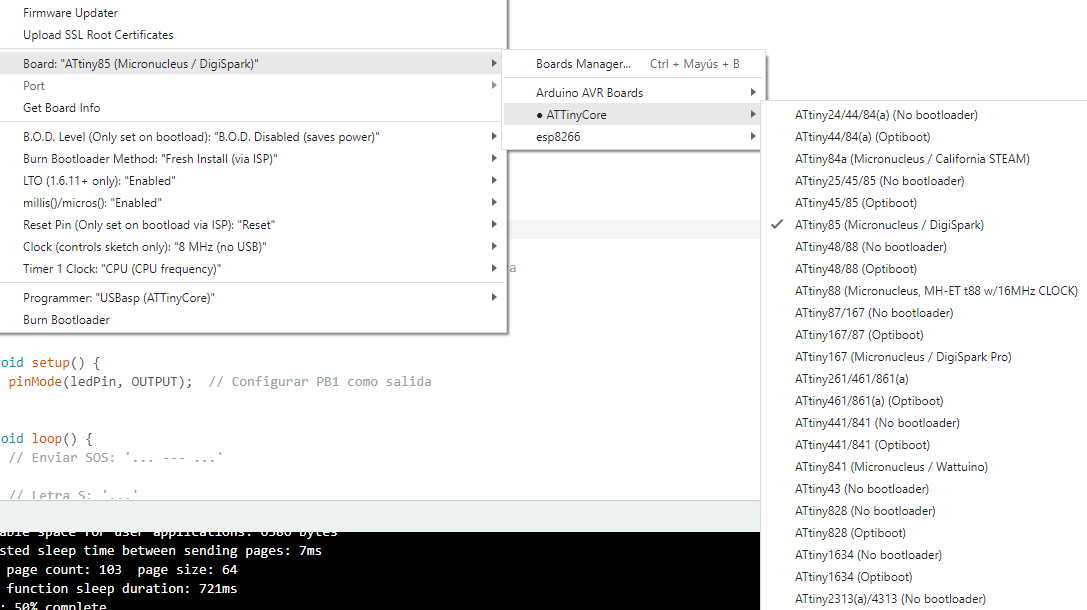
* **Microcontrolador:** ATtiny85 (DIP8)
* **Placa de desarrollo con chip ATtiny de 8 pines HW-26**0
* **Programador:** USBasp
* **Software:**
  + **Arduino IDE**
  + **AVRDUDESS** (Interfaz gráfica para avrdude)

**Paso 1: Configuración del Arduino IDE**

Para que el ATtiny85 aparezca en las opciones de placas del Arduino IDE, agregamos el siguiente **JSON** en las preferencias:

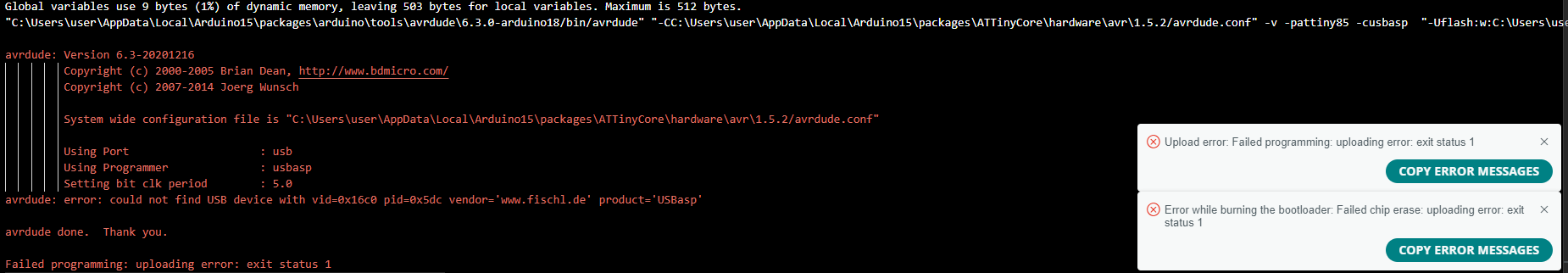
http://drazzy.com/package\_drazzy.com\_index.json

Luego, instalamos el paquete correspondiente en el **Gestor de Placas**.

*(**)*

**Paso 2: Cargar código sin bootloader**

Intenté cargar código en el ATtiny85 utilizando el **USBasp** directamente, sin bootloader. Desde arduino ide Sin embargo, encontré un problema:

1. **Arduino IDE compila, pero no logra conectar al microcontrolador.** **
2. Intenté ejecutar el comando de **avrdude** manualmente en la terminal (copiándolo del Arduino IDE), pero obtuve el mismo error.
3. Finalmente, utilicé **AVRDUDESS**, que sí detectó el USBasp y el ATtiny85 correctamente.

A pesar de haber logrado cargar programas con AVRDUDESS, resultaba incómodo copiar y pegar comandos entre programas. Por eso, decidí cargarle un **bootloader** para facilitar la programación.

**Paso 3: Quemar el bootloader**

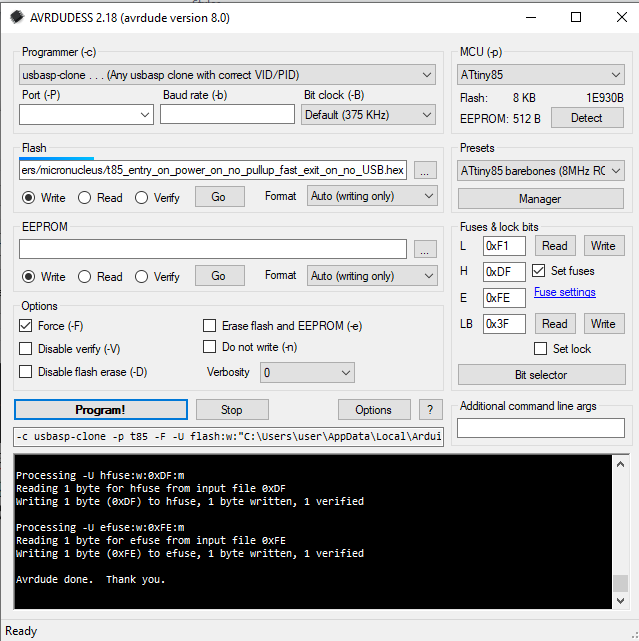
**En Arduino IDE**

1. Seleccionamos la placa **ATtiny85 (Micronucleus / DigiSpark)**.
2. Como programador, elegimos **USBasp**.
3. Intentamos quemar el bootloader, pero obtenemos un error.

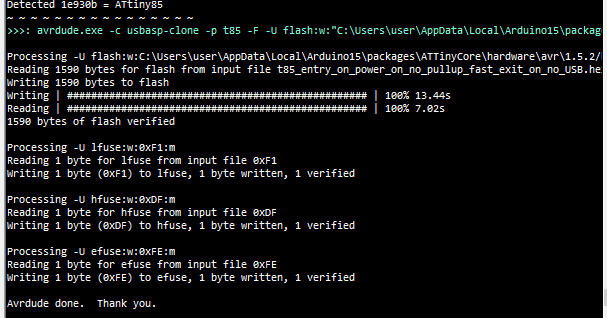
Para analizar el problema, copié el comando de **avrdude** generado por Arduino IDE:

avrdude -CC:\Users\user\AppData\Local\Arduino15\packages\ATTinyCore\hardware\avr\1.5.2/avrdude.conf -v -pattiny85 -cusbasp -e -Uefuse:w:0xFE:m -Uhfuse:w:0b11011111:m -Ulfuse:w:0xF1:m

Este comando define los **fuses** y carga el bootloader.

****En AVRDUDESS**

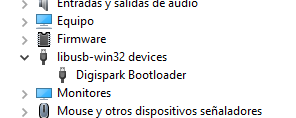
Ejecuté manualmente los mismos parámetros en **AVRDUDESS**, y esta vez el proceso finalizó correctamente.



**Paso 4: Conectar el ATtiny85 como DigiSpark**

Después de quemar el bootloader:

1. Desconectamos todo.
2. Insertamos el ATtiny85 directamente en el puerto **USB**.
3. El sistema lo detecta como **DigiSpark**.
4. Ahora podemos cargar programas directamente desde el **Arduino IDE**, sin necesidad del USBasp.

**

**Conclusión**

Este proceso me permitió programar el **ATtiny85** de dos maneras:

1. **Directamente con USBasp** y AVRDUDESS.
2. **Con bootloader (DigiSpark)** para cargar programas desde el Arduino IDE por USB.

Ambos métodos tienen ventajas y desventajas, pero usar el bootloader simplifica mucho la programación. ¡Espero que esta bitácora ayude a otros con el mismo problema!

