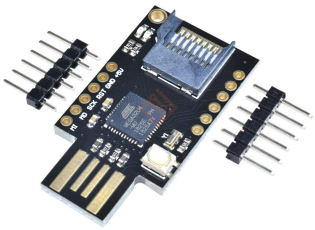
Adrian Cañadas

CJMCU Bad USB

Hardware Necesario:



- CJMCU Bad USB (https://es.aliexpress.com/item/1005002965725875.html)
- Tarjeta de memoria MicroSD, el tamaño es indiferente
- Lector para tarjetas MicroSD

Dependendencias previas

```
sudo pacman -Syu git base-devel libusb usbutils zip
```

Clonar el repositorio de Seytonic

```
git clone https://github.com/Seytonic/Duckduino-microSD
```

Instalación de Arduino IDE v2

```
git clone https://aur.archlinux.org/arduino-ide-bin.git
cd arduino-ide-bin
makepkg -si
cd
```

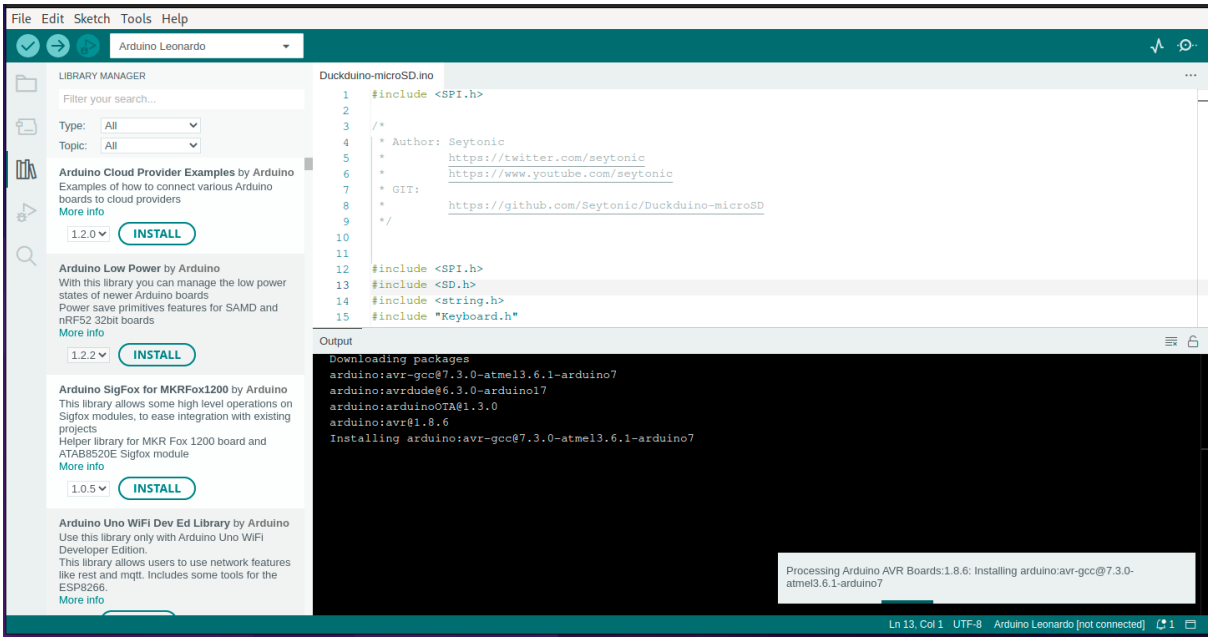
Agregamos nuestro usuario al grupo uucp y reiniciamos

```
sudo gpasswd -a $USER uucp
reboot
```

Iniciamos Arduino IDE v2

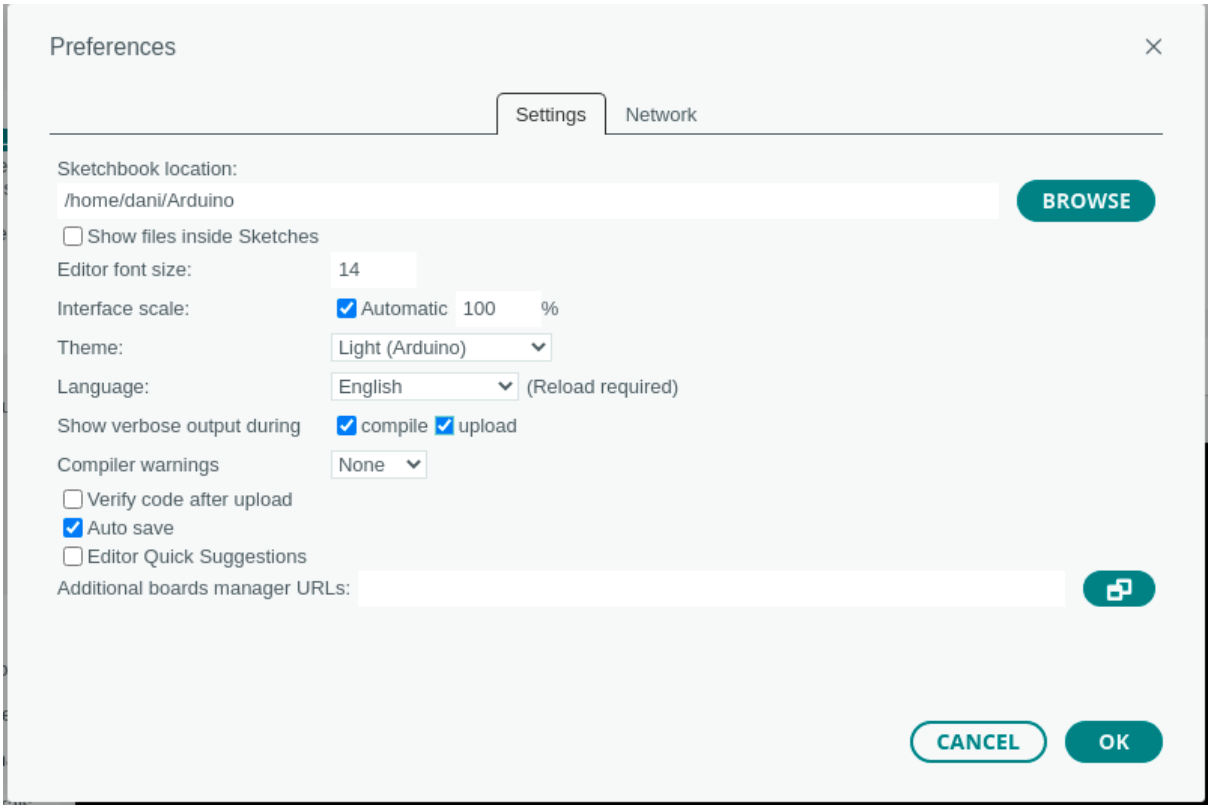
La primera vez se harán automáticamente las actualizaciones de paquetes pendientes

Adrian Cañadas



Cambiar preferencias

En el menu file -> Preferences activamos los checkbox de “Show verbose output during” compile y upload.

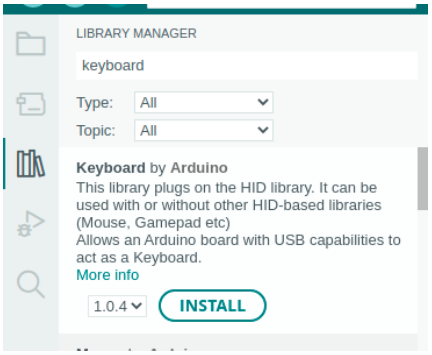


Esto nos ayudará a tener más información para poder cambiar la configuración de la librería.

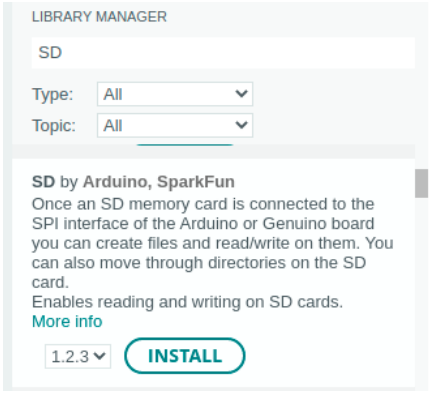
Adrian Cañadas

Librería arduino keyboard y SD

Hay que hacer click en el icono  para abrir el library manager.
Buscamos “Keyboard” e instalamos la última versión de la librería Keyboard



y luego buscamos “SD” y instalamos la libería SD by Arduino, SparkFun



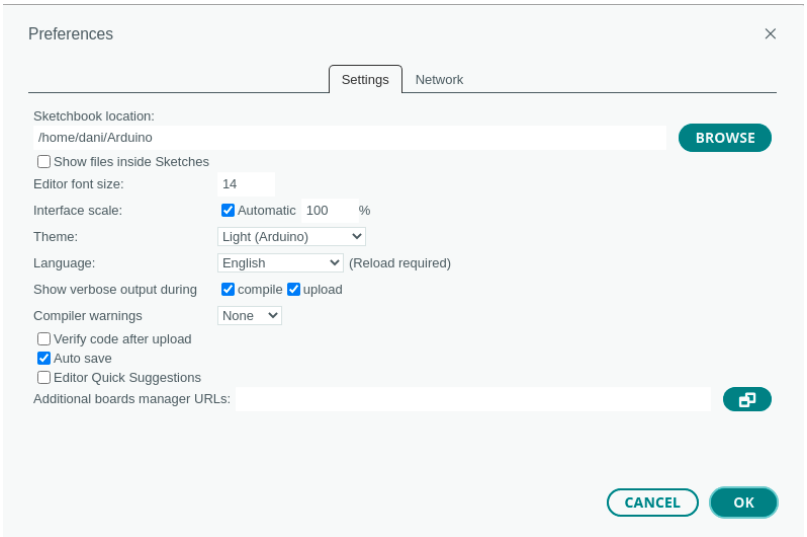
Abrimos el sketch en /home/$USER/ Duckduino-microSD/Duckduino-microSD/Duckduino-microSD.ino
y modificamos la línea 29, para pasarle el parámetro **KeyboardLayout_es_ES** a Keyboard.begin()

Conectar el CJMCU y subirle el sketch
Conectamos nuestro dispositivo (sin la memoria MicroSD).

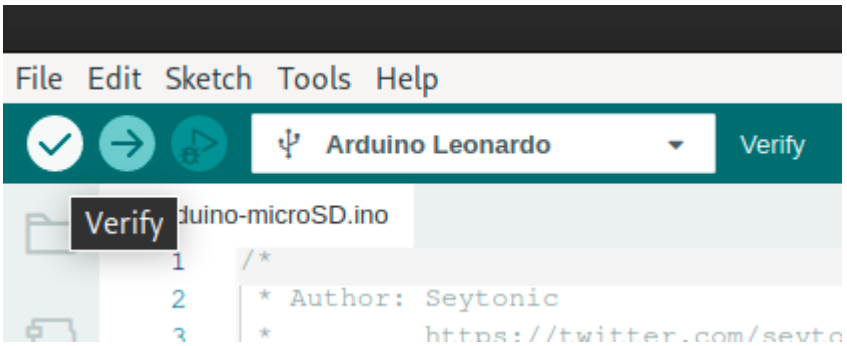
Adrian Cañadas

En el IDE de Arduino seleccionamo el tipo de placa Arduino Leonardo y el puerto.
Board: “Arduino Leonardo”
Port: “/dev/ttyACM0” (el número puede ser diferente en vuestro caso)

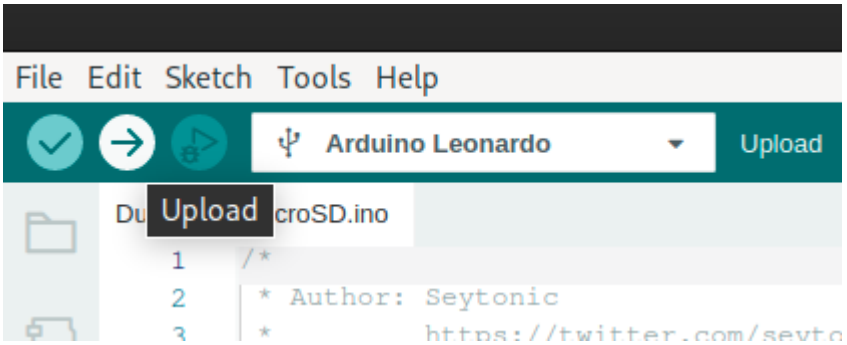
En el menú File -> Preferences marcamos las casillas “show verbose output during” compile y upload. Esto nos dará más información en caso de que algo falle.



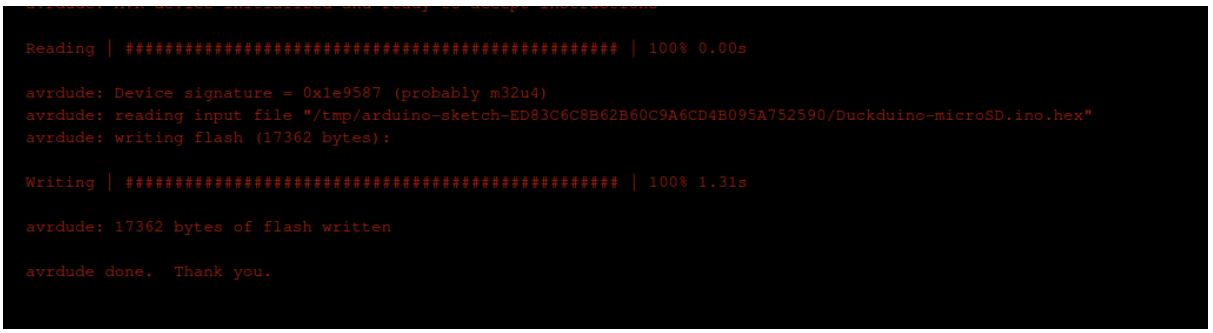
Pulsamos el botón Verify



y cuando termine la compilación “Upload”



Si todo ha ido bien en la ventana “output” debería salir algo parecido a esto:



Configurar payload

Se pueden descargar el payloads de https://www.ducktoolkit.com/

Descargamos el **duckycode.txt** y lo renombramos como **script.txt** y lo copiamos a la tarjeta microSD.

Metemos la tarjeta de memoria en el CJMCU Bad USB y ya estaríamos preparados para ir al PC de Windows.

Yo he preparado este script para la demo.

Mi script hace desactiva windows defender y crea un reverse shell hacia mi equipo con Arch Linux.

Escuchar mediante netcat (PC del atacante)
El comando netcat (nc) es también conocido como la navaja suiza de los administradores de red. Mediante netcat podemos “escuchar” (listening) por un puerto TCP de uno de los interfaces de red de nuestro equipo. Normalmente no es necesario instalar nc en arch, porque ya viene instalado por defecto en el paquete de Networking

```
nc -lnvp 8080 -s 192.168.1.41
```

En el PC Windows (PC del objetivo)
Hemos de introducir el CJMCU Bad USB cuando la sesión de windows esté iniciada

Cómo evitar los ataques con RubberDucky y similares en Windows

Todos estos ataques por teclado tienen en común un mismo principio, la elevación de permisos, que por defecto en Windows no nos pide contraseña. Por suerte para nosotros esto es muy fácil de remediar: Video: Elevación de permisos segura en Windows