

ESPJoy

MANUAL



Índice

Introducción	1
Puertos y botones	2
Uso	3
Configuración	3
Selección del mapa de teclas	3
Actualización de firmware	5
Firmware como código fuente (formato INO)	6
Arduino IDE	6
Actualización	6
Firmware en formato HEX	8
AVRDUDE	8
Actualización	8
Notas Técnicas	10
Sobre la programación de Arduino Leonardo	10
Referencias	11

Introducción

ESPjoy es un placa adaptadora con puertos [DE-9](#) para mandos [norma Atari](#) o [Sega Mega Drive \(Genesis\)](#) y que convierte pulsaciones de sus botones a señales equivalentes de teclado.

Está diseñada originalmente para usarse con [ESPECTRUM](#), pero puede utilizarse con otros dispositivos, ya que es capaz de convertir, según se deseé, a dos protocolos distintos de teclado (USB y PS/2), usando un interruptor incorporado.

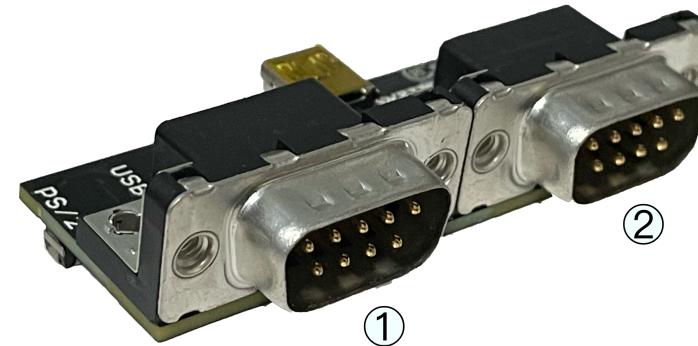
Esta traducción a teclado es muy útil para emuladores o para aquellos dispositivos basados en FPGA, como la familia [ZX-Uno](#) y sus derivados o [MiSTER](#), que no tengan soporte para conectores de joystick. Por ejemplo, para el uso en cores de ordenadores cuando no se desea un teclado conectado permanentemente.

El programa (firmware) grabado en la placa envía pulsaciones cuando se pulsa un botón de algún mando conectado y es capaz de utilizar dos mandos de manera simultánea.

Son compatibles los mandos y joystick de uno o dos botones (norma Atari), así como mandos Sega de tres o de 6 botones.

Puertos y botones

1	Primer Puerto DE-9
2	Segundo Puerto DE-9



3	Puerto USB Mini-B
4	Interruptor de selección de protocolo

Uso

Configuración

Para utilizarlo en ESPpectrum se ha de seleccionar la opción PS/2 del interruptor. La opción USB se puede utilizar con MiSTer u otro sistema compatible (PC, Mac, etc.). Con el firmware original diseñado por David Carrión se emula un teclado HID, por lo que se puede usar sin problema en MiSTer con cores de ordenadores.

La detección de la conexión se realiza en los primeros segundos del arranque cuando la placa se conecta a un puerto USB o PS/2 y recibe alimentación. Si la placa está conectada a un dispositivo como un PC o similar durante el arranque del mismo, es muy probable que falle la detección USB, porque el sistema operativo no esté aún preparado cuando se alcanza el límite de tiempo de espera de detección de la placa. En ese caso, se ha de desconectar y volver a conectar una vez que el dispositivo esté preparado para detectar la conexión.



El interruptor de selección del modo PS/2 o USB se ha de cambiar siempre antes de conectar la placa a la alimentación, o no funcionará correctamente.

Selección del mapa de teclas

Durante el arranque, si se mantiene pulsado alguno de los botones especificados, según las tablas que hay a continuación, se pueden seleccionar distintas traducciones de teclado .



El comportamiento en la asignación por defecto (cuando no se pulsa ningún botón en el arranque) varía según el firmware que haya instalado en el ESPJoy. Cuando se indique la opción (Original) se estará hablando el firmware no adaptado para ESPpectrum. En el otro caso (ESPpectrum), se refiere al firmware adaptado para ESPpectrum.



La asignación por defecto del firmware de ESPpectrum al usar el puerto PS/2 lo hace incompatible con otros dispositivos al usar una conexión PS/2. Por tanto, en ese caso, se ha de pulsar obligatoriamente un botón que active una configuración distinta de teclado durante el arranque.



Existen algunos mandos clónicos que pueden enviar señales extrañas en algunas direcciones. Debido a esto, falla la comprobación en el código, y no funciona correctamente la activación en el arranque, por ejemplo, al pulsar izquierda en el mando.

Puerto 1 (PS/2)

Pulsación Arranque	Arriba	Abajo	Izquierda	Derecha	A	B	C	Start	X	Y	Z	Mode
Ninguna (Original)	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Alt Dcha.	F1	X	Y	Z	M
Ninguna (ESPectrum)	Arriba	Abajo	Izquierda	Derecha	A	B	C	Start	X	Y	Z	Mode
Arriba	Q	A	O	P	Escape	M	Enter	F1	X	Y	Z	C
Abajo	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Ø	F5	X	Y	Z	M
Izquierda	7	6	5	8	Escape	Ø	Enter	F1	X	Y	Z	M

Puerto 2 (PS/2)

Pulsación Arranque	Arriba	Abajo	Izquierda	Derecha	A	B	C	Start	X	Y	Z	Mode
Ninguna (Original)	Q	A	O	P	Escape	M	Enter	F1	X	Y	Z	C
Ninguna (ESPectrum)	Arriba	Abajo	Izquierda	Derecha	A	B	C	Start	X	Y	Z	Mode
Arriba	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Alt Dcha.	F1	X	Y	Z	M
Abajo	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Ø	F5	X	Y	Z	M
Izquierda	7	6	5	8	Escape	Ø	Enter	F1	X	Y	Z	M

Puerto 1 (USB)

Pulsación Arranque	Arriba	Abajo	Izquierda	Derecha	A	B	C	Start	X	Y	Z	Mode
Ninguna (Original)	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Alt Dcha.	F12	X	Y	Z	M
Arriba	Q	A	O	P	Escape	M	Enter	F12	X	Y	Z	C
Abajo	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Ø	F5	X	Y	Z	M
Izquierda	7	6	5	8	Escape	Ø	Enter	F12	X	Y	Z	M

Puerto 2 (USB)

Pulsación Arranque	Arriba	Abajo	Izquierda	Derecha	A	B	C	Start	X	Y	Z	Mode
Ninguna (Original)	Q	A	O	P	Escape	M	Enter	F12	X	Y	Z	C
Arriba	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Alt Dcha.	F12	X	Y	Z	M
Abajo	Arriba	Abajo	Izquierda	Derecha	Escape	Enter	Ø	F5	X	Y	Z	M
Izquierda	7	6	5	8	Escape	Ø	Enter	F12	X	Y	Z	M

Actualización de firmware

Existen diferentes versiones de firmware (el programa grabado en la memoria flash de la placa y que se encarga de realizar la traducción a pulsaciones de teclado):

- El firmware original, que puede utilizarse para enviar pulsaciones de teclado PS/2 o USB y que está disponible aquí:

http://github.com/Dacarsoft/DB9_2_Keyboard

- La variante específica para [ESPectrum](#), y que envía, por defecto, para PS/2, unas pulsaciones especiales para el emulador, disponible aquí:

https://github.com/dacarsoft/DB9_2_Keyboard/tree/DB9_2_Keyboard_ESPectrum

- También se puede instalar el firmware Daemonbite disponible en esta ubicación:

<https://github.com/MickGyver/DaemonBite-Retro-Controllers-USB/tree/master/SegaTwoControllersUSB>



En este último caso, el ESPJoy funcionará como mando de juegos HID USB, pero perdiendo el soporte de protocolo PS/2.

Para actualizar la programación de la placa, existen varias opciones de software, aunque todas requieren un ordenador o similar con puertos USB y un sistema operativo Windows, Linux o macOS.

La forma de programar dependerá del formato en que se encuentre disponible el programa.

Firmware como código fuente (formato INO)

En este caso, el firmware no está completamente listo para ser instalado en la placa y tiene que ser compilado desde su código fuente antes ser grabado en la memoria flash de la placa. El mejor sistema en este caso, consiste en utilizar el entorno de desarrollo oficial de Arduino.

Arduino IDE

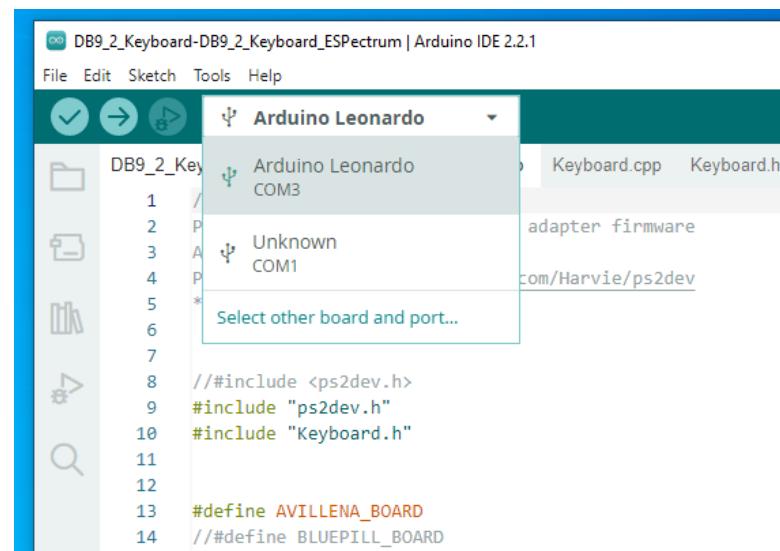
Para poder hacer este tipo de instalación, se puede usar un ordenador (Windows, Mac, Linux) con el entorno [Arduino IDE](#) instalado.

Actualización

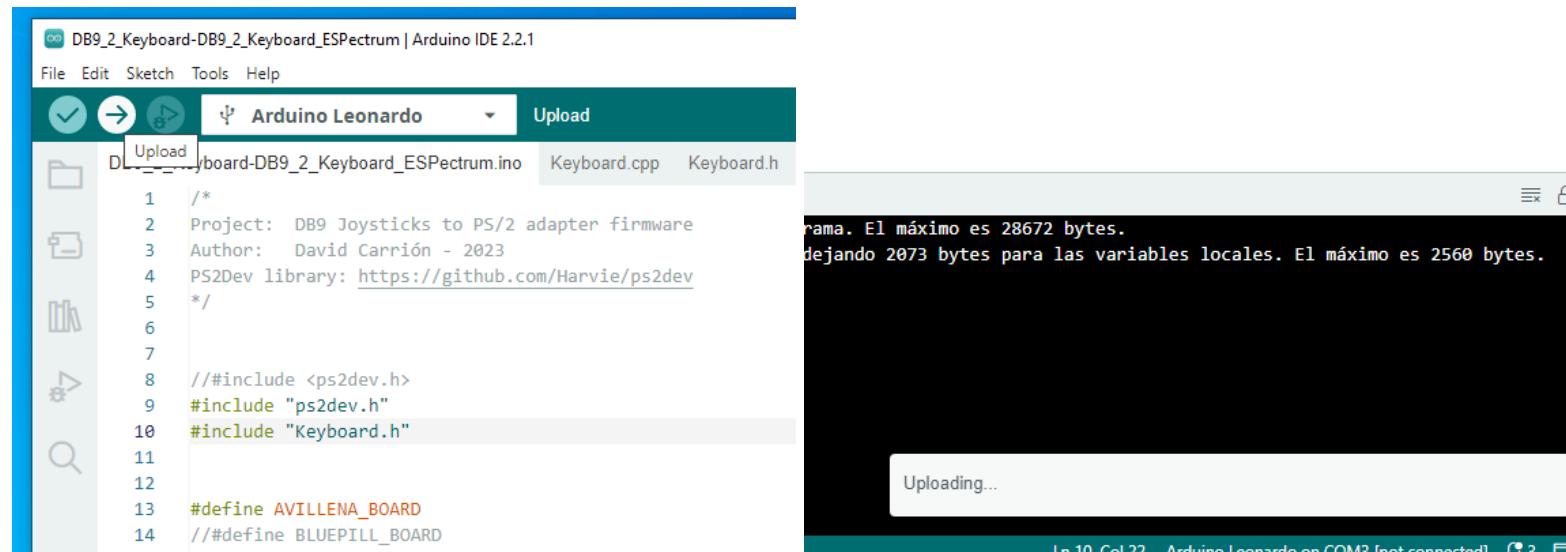
Una vez preparado el entorno con Arduino IDE instalado, descargar la versión deseada del proyecto desde el repositorio correspondiente (estándar, ESPectrum, etc.). Normalmente es una carpeta con varios ficheros, uno de los cuales tiene la extensión **.ino** en el nombre.

Abrir el fichero del proyecto en Arduino IDE (por ejemplo **DB9_2_Keyboard_ESPectrum.ino**).

Colocar el interruptor de la placa en la posición marcada como "USB" (la posición más cercana al conector USB Mini-B). Conectar el ESPjoy al equipo usando el cable USB y seleccionarlo en la parte superior de la ventana del IDE (opción "Arduino Leonardo").



Finalmente, pulsar el botón de carga del firmware (upload)  y esperar unos segundos mientras se compila el proyecto y se carga en el dispositivo.



Firmware en formato HEX

En este otro caso, el firmware está completamente listo para ser grabado en la memoria flash de la placa. Existen múltiples programas que se pueden utilizar para hacer la grabación.

AVRDUDE

AVR Downloader Uploader es un programa de línea de comandos que sirve para manipular los contenidos ROM y EEPROM de múltiples microcontroladores.

Para poder utilizarlo con ESPJoy, se necesita una versión reciente que soporte el parámetro `-r` (reconnect to -P port after "touching" it). Véanse las [notas Técnicas](#) al final de este manual para más información.

La última versión ejecutable para Windows se puede descargar desde el repositorio oficial:

<https://github.com/avrdudes/avrdude>

Para otros sistemas, es bastante sencillo compilar el ejecutable desde el código fuente disponible en dicho repositorio. También está disponible en múltiples sistemas de instalación de paquetes, como [Homebrew para Mac](#), o los distintos repositorios oficiales de distribuciones de Linux.

Actualización

Tras obtener la versión de firmware deseada del fichero `.hex` (por ejemplo [DB9_2_Keyboard.ino.hex](#)), colocar el interruptor de la placa en la posición marcada como "USB" (la posición más cercana al conector USB Mini-B). A continuación conectar la placa ESPJoy al ordenador a un puerto USB, e identificar dicho puerto en el sistema operativo.

Una vez identificado el nombre del puerto, se puede grabar el fichero en la placa con un comando como el siguiente:

```
.../avrdude -F -patmega32u4 -cavr109 -b57600 -r -P <puerto> -U flash:w:<fichero.hex>:i
```

Donde `<puerto>` es la identificación del puerto donde está conectada la placa y `<fichero.hex>` es la ruta al fichero `.hex` que se desea grabar.

Por ejemplo, para un puerto llamado `/dev/cu.usbmodemHIDPC1` y un fichero `DB9_2_Keyboard.ino.hex`:

```
.../avrdude -F -patmega32u4 -cavr109 -b57600 -r -P /dev/cu.usbmodemHIDPC1 -U flash:w:DB9_2_Keyboard.ino.hex:i
```

Esperar unos segundos, y, si todo va bien, en la consola aparecerán mensajes indicando que se ha grabado el programa correctamente.

```
avrdude warning: avr_mem_order[] does not know data; add to array and recompile
avrdude warning: System wide configuration file version ()
                  does not match Avrdude build version (7.2-20231201 (5501c63a))
avrdude: touching serial port /dev/cu.usbmodemHIDPC1 at 1200 baud
avrdude: waiting for new port... 600 ms: using new port /dev/cu.usbmodem83201
avrdude: AVR device initialized and ready to accept instructions
avrdude: device signature = 0x1e9587 (probably m32u4)
avrdude: Note: flash memory has been specified, an erase cycle will be performed.
          To disable this feature, specify the -D option.
avrdude: erasing chip

avrdude: processing -U flash:w:DB9_2_Keyboard_ESPectrum_Expanded.ino.hex:i
avrdude: reading input file DB9_2_Keyboard_ESPectrum_Expanded.ino.hex for flash
      with 8460 bytes in 1 section within [0, 0x210b]
      using 67 pages and 116 pad bytes
avrdude: writing 8460 bytes flash ...
Writing | ##### | 100% 0.67 s
avrdude: 8460 bytes of flash written
avrdude: verifying flash memory against DB9_2_Keyboard_ESPectrum_Expanded.ino.hex
Reading | ##### | 100% 0.07 s
avrdude: 8460 bytes of flash verified

avrdude done.  Thank you.
```

Notas Técnicas

Sobre la programación de Arduino Leonardo

La placa ESPJoy está basada en un controlador programable [ATmega32u4](#), que es el mismo que utilizan las placas [Arduino Leonardo](#).

Estos dispositivos disponen de un cargador de arranque que se puede utilizar para la programación de la flash, pero también puede omitirse para ejecutar directamente la programación guardada. Así es como se determina cuál de las dos posibilidades es la que se activa:

- Si se inicia después de una orden de reinicio (es decir, se ha pulsado el botón de reinicio cuando la placa dispone del mismo), siempre se activa el modo de programación (grabación de programa en la memoria flash).
- Si no hay programa almacenado en la flash (la primera palabra es `0xffff`), siempre se activa el modo de programación.
- Si arranca después de un encendido en frío (lo que significa que el dispositivo se activa por primera vez después de aplicar alimentación), intentará ejecutar el programa almacenado.
- Si arranca después de un reinicio (reset) por watchdog (perro guardián), entonces:
 - Si hay el código adecuado escrito en una ubicación mágica en la SRAM (`0x7777` escrito en `0x0800`), entonces pasará al modo de programación.
 - Si no, intentará ejecutar el código de usuario.
- Si arranca después de cualquier otro tipo de reinicio, pasará a modo programación.

El gestor de arranque esta hecho de forma que detecta si se ha abierto y cerrado una conexión serie a 1200 baud al puerto, y, en ese caso, realiza los pasos necesarios para hacer un reinicio por watchdog.

Finalmente, señalar que el gestor de arranque tiene un tiempo de espera de 8 segundos. Si la programación no se ha iniciado en ese tiempo después de activar (a través de cualquier método) en el modo de programación, intentará de nuevo ejecutar el código de usuario.

Como consecuencia de esto, la programación por vía de comandos, etc. en Windows requiere que estén instalados los controladores específicos para los dos modos (ejecución de programa y programación) para todos los puertos serie (COM), para que la nueva conexión tras el reinicio por watchdog funcione correctamente.

En el caso de la grabación con [AVRDUDE](#), es muy recomendable usar una versión que disponga del parámetro `-r`, y que por tanto realiza automáticamente este proceso de apertura y cierre de conexión, y conexión posterior, dentro de los 8 segundos de margen necesarios para la programación. En otro caso puede ser algo complicado realizarlo de forma manual.

Referencias

<https://antoniovillena.com/product/espjoy/>

http://github.com/Dacarsoft/DB9_2_Keyboard

<https://github.com/MickGyver/DaemonBite-Retro-Controllers-USB/tree/master/SegaTwoControllersUSB>

<https://avrdude.readthedocs.io/en/latest/>

<https://stackoverflow.com/questions/45646127/how-to-upload-a-hex-file-into-an-arduino-leonardo-using-avrdude-from-the-command>