

MaxDuino

Guía de uso

ÍNDICE

1.- ¿Qué son los TAPUINOS, TZXDUIINOS, CASDUIINOS y MAXDUIINOS?.....	3
2.- ¿Qué sistemas soporta el firmware MaxDuino?.....	3
3.- ¿Por qué usar estos aparatos y no una aplicación móvil?.....	4
4.- ¿Qué es MaxDuino?.....	5
5.- ¿Cómo es un dispositivo MaxDuino?.....	6
6.- ¿Cómo manejar el MaxDuino?.....	8
6.1.- Arranque del MaxDuino.....	9
6.2.- Pantalla inicial.....	10
6.3.- Cambiando la posición actual.....	10
6.4.- Desplazamientos rápidos por semi-intervalos.....	12
6.5.- Reproducir un archivo de cinta.....	13
6.6.- Menú de configuración.....	14
6.6.1.- Opción Baud Rate.....	14
6.6.2.- Opción Motor Ctrl.....	15
6.6.3.- Opción TSXCzxpUEFSW.....	16
7.- ¿Cómo actualizar el firmware MaxDuino?.....	17
8.- ¿Dónde se pueden adquirir las placas MaxDuino?.....	22
9.- ¿Desea saber más?.....	22
APÉNDICE 1. TIPOS DE BLOQUES.....	23

Agradecimientos

Quiero agradecer a Rafael Molina su gran trabajo con el proyecto MaxDuino, así como sus aportaciones para mejorar este documento.

Licencia



<https://creativecommons.org/publicdomain/zero/1.0/deed.es>

Control de versiones

Fecha	Versión	Cambios
11/04/2020	1.00	Versión inicial por desUBIKado
27/11/2022	1.01	Versión para Va de Retro µTZXdino

1.- ¿Qué son los TAPUINOS, TZXDUINOS, CASDUINOS y MAXDUINOS?

Con este nombre se denominan una serie de dispositivos que sirven para realizar cargas de programas de ordenador por audio. También sirven para denominar a los firmwares que llevan estos dispositivos, lo que puede llevar a confusión, ya que es posible en un dispositivo TZXDUINO instalar el firmware TZXduino, pero también por ejemplo el firmware ToT (Tapuino on Tzxduino), CASduino o el MAXduino.

Los ficheros con los datos para realizar la carga por audio se almacenan en una tarjeta SD o microSD, y son de diferentes tipos dependiendo del ordenador al que van dirigidos. Por ejemplo, hay ficheros .TZX que son propios del Sinclair Spectrum, y otros como el .CDT que son para Amstrad CPC. Hay algunas extensiones, por ejemplo .CAS o .TAP, que pueden ser compartidas por distintos de ordenadores, pero su codificación interna es diferente, por lo que se deben usar solo los específicos del modelo de ordenador para el se quiere hacer la carga por audio.

La principal diferencia entre firmwares es que cada uno solo es capaz de reproducir unos determinados ficheros de cinta, por lo que algunos están especializados, aunque no restringidos, a determinados modelos de ordenador, por ejemplo, el TAPuino con los ordenadores de Commodore (C64, VIC20, etc), el CASduino con los ordenadores MSX, el TZXduino con los de Sinclair (Spectrum, ZX80 o ZX81), y el MAXduino, que nació como una unificación de los firmwares TZXduino y CASduino, tiene un enfoque más generalista, y permite la carga en un gran numero de ordenadores y formatos.

Los chips Arduino Nano (Atmega328), Arduino Nano Pro y Arduino Every son los chips empleados hasta ahora por los constructores de estas placas. Recientemente ha nacido el proyecto Megduino, placa que utiliza el chip Arduino Mega 2560, más potente, en el cual la versión actual del firmware maxduino solo ocupa una pequeña parte de su capacidad, en comparación al 99% de los otros chips, por lo que es una apuesta de futuro para poder añadir nuevas funcionalidades, entre las que se encuentra el poder grabar audio, y no solo reproducirlo como hacen el resto de placas.

2.- ¿Qué sistemas soporta el firmware MaxDuino?

En su última versión los sistemas que soporta, así como el formato de archivos, son los siguientes:

- Spectrum: TAP y TZX
- Amstrad CPC: CDT
- MSX: CAS y TSX
- ZX80: O
- ZX81: P
- Acorn Atom y Electron: UEF (sin comprimir) y TSX
- Dragon 32/64 / Tandy CoCo: CAS y TSX
- SVI Spectravideo: TSX
- Oric 1 y Oric Atmos: No soporta los TAP de oric, pero sí TAP convertidos a TSX

Como veis este firmware no permite la carga de archivos de cinta TAP para Commodore VIC20, C64 o C16 / C116 o Plus/4. Sin embargo, si tenéis una placa TZXDuino, podéis instalar el firmware

ToT (Tapuino on Tzxduino) que sí lo permite, pero entonces perderéis la posibilidad de realizar cargas en otros sistemas.

Por otra parte, es posible incorporar en los ficheros TSX los datos, no ya solo de los MSX, si no de **cualquier ordenador** que usara la codificación *Kansas City Standard*. Así en el github de NataliaPC (https://github.com/nataliipc/MSX_devs/tree/master/TSXphpclass) encontramos utilidades para convertir a TSX diferentes tipos de ficheros de cinta:

- **uef2tsx.php**: Para los .UEF de los ordenadores Atom, Electron y BBC Micro de la compañía Acorn.
- **oric2tsx.php**: Para los .TAP de Oric
- **drg2tsx**: Para los ficheros .CAS de Dragon y Tandy CoCo
- **svi2tsx**: Para los .CAS de SVI Spectravideo
- **cas2tsx**: Para los .CAS de MSX

3.- ¿Por qué usar estos aparatos y no una aplicación móvil?

Hay principalmente 3 ventajas:

- Estos dispositivos leen directamente los datos a convertir en audio sin necesidad de convertirlos previamente a formato .wav, lo que lleva cierto tiempo, mientras que la reproducción desde los “duinos” es inmediata, y además consume espacio de almacenamiento. Las aplicaciones de móvil generalmente no borran estos archivos .wav de forma automática una vez usados, y ocupan mucho espacio si se carga bastante juegos, de forma que al final hay que borrarlos manualmente, lo cual es una molestia.
- Por otra parte, algunos de estos dispositivos (no todos) llevan un conector REM (remoto) que sirve para que el ordenador controle la carga, lo cual es muy útil en juegos multicarga para aquellos ordenadores que tenían la opción de controlar los “datasettes”. No obstante, aunque hay modelos que no llevan conector remote, esto no es un problema grave porque está implementado el que cuando la reproducción está en pausa, es muy sencillo retroceder o avanzar entre bloques simplemente con los botones de arriba o abajo de estos dispositivos.
- Alguno de los duinos llevan amplificación, lo cual es muy útil en ciertos modelos de ordenador que sufren un poco de sordera, y que necesitan un volumen bastante alto para que la carga por audio se realice con éxito. Esto a veces es difícil de conseguir en un móvil por las restricciones al volumen que tienen la mayoría.

Y el inconveniente, obviamente, es el precio, ya que generalmente las aplicaciones móviles especializadas en reproducir estos archivos de cinta son gratis, y un móvil lo tenemos todos hoy en día. Sin embargo, los duinos no son especialmente caros, variando su rango de precios entre los 15 y 40 euros según prestaciones, si llevan carcasa o no, etc.

4.- ¿Qué es MaxDuino?

Haciendo un poco de historia, todo parte de Sweetlilme y su proyecto TAPuino, centrado en la carga de archivos de audio .TAP para las máquinas de Commodore (PET, VIC20, C64 y C128).

Más adelante, basándose en él, Duncan Edwards y Andrew Beer crearon el CASduino para reproducir los archivos .CAS para MSX. Después nació TZXduino creado por los mismos autores, para poder reproducir ficheros .TAP y .TZX de Sinclair ZX Spectrum, al que añadieron rápidamente los ficheros .CDT de Amstrad CPC (básicamente son .TZX con distinta extensión).

Continuaron mejorándolo y añadieron el poder reproducir los ficheros .CAS de los Ordenadores Dragon data y Tandy Color Computer.

Otras mejoras fueron llegando también al TZXduino como el soporte para los ficheros .AY (melodías que se pueden reproducir en los chips de sonido AY-3-8912 que llevan por ejemplo los ordenadores Spectrum 128, Amstrad CPC o los MSX), o el soporte para poder cargar los fichero .P y .O para los ZX80 y ZX81 de Sinclair. Incluso llegaron mejoras desde otros programadores, como el soporte para archivos .TSX para los ordenadores MSX, aportación de NataliaPC (Natalia Pujol), o la corrección de un error en la rutina de impresión LCD, por Rafael Molina (rcmolina).

Fue Rafael, al que el rebobinado de directorios del TZXDuiño le parecía una manera horrible de navegar, y cuyas ideas de mejora no convencieron a Duncan Edwards, que consideraba que gastaban mucha memoria, el que se puso a optimizar el código del TZXDuiño. En su cabeza ya rondaba una idea que finalmente, el 20 de agosto de 2017, hizo pública en esta entrada (<https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=5541&start=110#p109097>), siendo el pistoletazo de salida de su proyecto de código abierto MaxDuino, en el que intentaría unificar los firmwares TZXduino y CASduino. Hay que pensar que antes, si solo tenías un duino, debías instalarle el firmware TZXduino si querías realizar cargas en un Spectrum, y si luego querías hacer cargas en un MSX debías instalar el firmware CASduino. Al unificar los firmwares esto ya no fue necesario, lo que supuso una gran ventaja. Además a partir de ese momento Rafael ha seguido optimizándolo y añadiéndole más funcionalidades. Es más, todo el desarrollo de pantalla OLED, parcheos de varios bugs, contadores de tiempo y pasos, navegación por los menús, inversión de polaridad, rebobinado de bloques, búsqueda logarítmica, desarrollos sobre BBC Micro, mejoras en Dragon y Tandy CoCo, soporte del bloque id15, soporte para Arduino Nano Every, y soporte de Tapuino sobre TZXDuiño para Commodore 64, son muchas de sus mejoras que luego han sido incorporadas al proyecto original TZXDuiño.

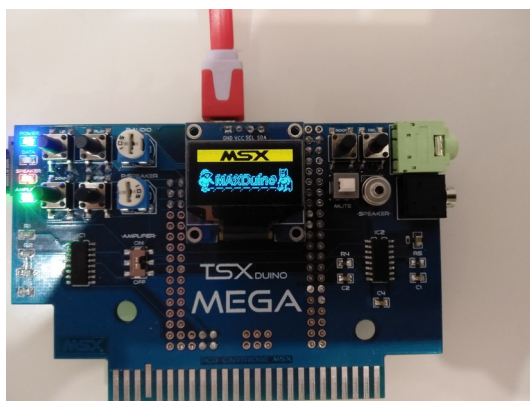
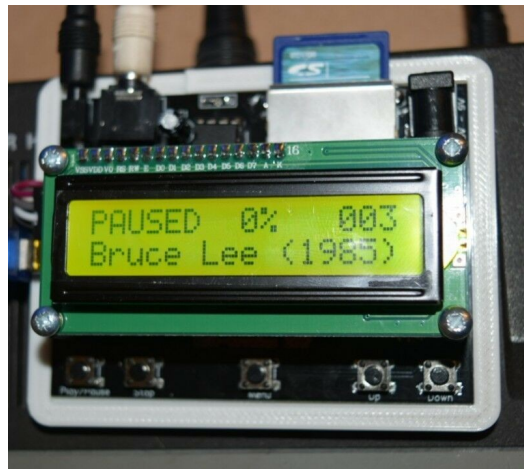
Gracias a que el proyecto MaxDuino es abierto, y que nació con carácter participativo, Rafael Molina ha recibido aportaciones de código de muchas personas, que han servido para mejorarlo. A contado con ayuda de otros programadores, como Natalia Pujol (NataliaPC), Alfredo “Retrocant” (@acf76es), Spirax, Ramón Martínez Palomares (*rampa*), sin olvidarnos de los proyectos originales de Sweetlilme -Tapuino-, y los de Duncan Edwards y Andrew Beer -CASduino y TZXduino-.

Para seguir la evolución de este proyecto se puede leer el “hilo” oficial del mismo en va-de-retro.com (<https://www.va-de-retro.com/foros/viewtopic.php?t=5541&start=9999>) y las fuentes están disponibles en su github: <https://github.com/rcmolina/MaxDuino>.

5.- ¿Cómo es un dispositivo MaxDuino?

Ya he comentado antes que el firmware se llama MaxDuino, pero los dispositivos pueden llevar diversos nombres como Tzxduino, Maxduino, Minidduino o Megadduino. Estos dispositivos son de construcción “casera”, y los que los construyen generalmente realizan tiradas limitadas, que una vez finalizadas no se producen más. Normalmente en estos casos se pone a disposición de todo el mundo la información para que cada uno se construya los suyos, básicamente los diseños de las placas y el listado de componentes. No obstante, hay otros fabricantes que tienen sus propias tiendas on-line, y venden sus propios modelos bajo su propia denominación, indicando que son compatibles con el firmware MaxDuino de Rafael Molina.

Ejemplos de distintos modelos:



Un dispositivo Maxduino se compone de:

- Un conector para proporciona la alimentación de 5v que necesita el aparato. Generalmente se utiliza un puerto MicroUSB .
- Un bahía para introducir una tarjeta SD o microSD según modelos. La tarjeta debe estar formateada en FAT32 o FAT16, funcionando sin problemas los modelos de hasta 8GB de capacidad.
- Un conector estéreo jack 3.5mm para la salida del audio. Solo usa el canal izquierdo, por lo que valen cables estéreo-estéreo o mono-mono.
- Un conector jack de 2.5 mm para la función remote (control del motor del cassette por parte del ordenador, por ejemplo en Amstrad CPC, MSX, Acorn Electron o BBC Micro). No todos los MaxDuinos lo llevan.
- Una pantalla LED u OLED para visualizar los menús y el nombre de los archivos. Los 4 tipos soportados son:
 - Pantalla LCD de 16x2 (2 líneas de 16 caracteres cada una)
 - Pantalla LCD de 20x4 (4 líneas de 20 caracteres cada una)
 - Pantalla OLED de 128x32 pixels (4 líneas de 16 caracteres)
 - Pantalla OLED de 128x64 pixels (8 líneas de 16 caracteres)

Los caracteres de las pantallas LCD suelen ser más grandes que los de las pantallas OLED, por lo que es mejor para los que ya no tienen la vista como antes, Ambos tipos de pantallas, LED y OLED, ofrecen la misma información, lo único que las OLED y la LCD de 20x4, al tener más líneas, pueden distribuir mejor los textos y dejarlo todo más vistoso.

- Cinco botones o pulsadores:
 - 1.- Play / Pause
 - 2.-Stop
 - 3.- Up
 - 4.- Down
 - 5.- Root o menú o pivot

Este último botón se llama de esas 3 formas porque ha tenido diversas funciones según ha ido evolucionando el firmware. El uso como root ya no se utiliza, y hace referencia a cuando en versiones antiguas del firmware TZXDduino (versión 1.11 o anterior), era la única forma de retroceder en el árbol de directorios, y lo que hacía era llevarte al directorio raíz de la SD. En las versiones modernas de Maxduino puede funcionar como Menú, y pulsándolo presentará el menú de configuración, o bien como pivot, que permite, mediante su pulsación conjunta con otro botón, acceder a más funciones.

Por otra parte, en las placas MegaDuino han añadido un 6º botón para poderlo usar en futuras mejoras como, por ejemplo, la posibilidad de grabación de audio.

- Un amplificador (por ejemplo LM386 monocal) para dar más potencia al sonido, útil para aquellos sistemas que sean un poco más sordos. Se puede activar/desactivar mediante jumpers en la placa. No todos los modelos llevan amplificador.

6.- ¿Cómo manejar el MaxDuino?

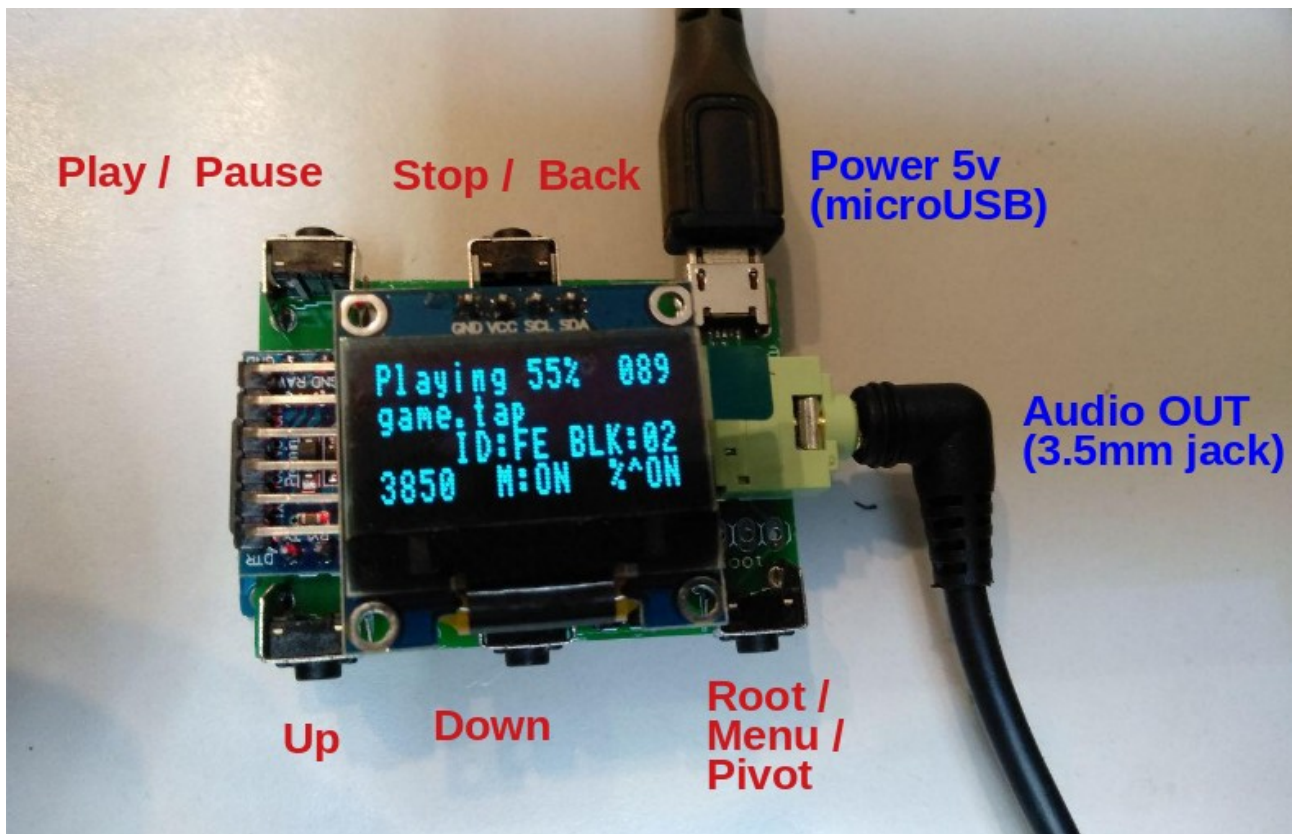
Lo primero es introducir una tarjeta SD o microSD formateada en FAT32 o FAT16 que contenga las imágenes de los archivos de cinta. Si se tienen varias máquinas físicas o alguna FPGA, una forma práctica es organizar la información en carpetas por sistemas, luego por tipo de archivos (.CAS, .TSX, etc), y finalmente separados alfabéticamente. Tenerlo todo ordenado facilita la búsqueda, pero también la incorporación posterior de nuevos ficheros.

> SD_MAXDUINO	
Nombre	Tamaño
Acorn Atom	0 elementos
Acorn BBC Micro	0 elementos
Acorn Electron	0 elementos
Amstrad CPC	1 elemento
CDT	27 elementos
#	313 elementos
a	1107 elementos
A 320 (F) (Face 1B) (1988) (1. Programmes 01 a 12) [Original] [TAPE].cdt	169,8 KiB
A 320 (F) (Face 2A) (1988) (2. Fichiers Dessins - DESS 01 a DESS 22) [Original] [TAPE].cdt	103,2 KiB
A 320 (F) (Face 2B) (1988) (3. Fichiers Dessins - DESS 23 a DESS 44) [Original] [TAPE].cdt	107,7 KiB
A 320 (F) (Face 2B) (1988) (3. Fichiers Dessins - DESS 23 a DESS 44) [Original] [TAPE](1).cdt	107,7 KiB
A New Image (UK) (1984) [Popular Computing Weekly] [Original] [TAPE].cdt	3,1 KiB

Para explicar el funcionamiento utilizaré mi Miniduino, creación de Antonio Villena con carcasa de Mejias3D.



La disposición en el Miniduino de los 5 botones es la siguiente:



Este modelo tiene una pantalla OLED de 128x64 pixels (16 caracteres en 8 líneas) de 1.3 pulgadas. Es posible que al personalizar el firmware se le diga que use las 8 líneas, pero en ese caso solo utiliza las 3 superiores y la inferior, quedando 4 líneas en negro en medio. Es mejor personalizarlo cuando se compila el firmware para ser usado como una pantalla de 128x32 (16 caracteres en 4 líneas). En ese caso solo usa las líneas pares y las impares están apagadas, por lo que los caracteres tienen una altura de 16 pixels y una anchura de 8, aumentando su legibilidad.

Como se aprecia en la imagen, este modelo no tiene el conector remote de 2.5mm, solo el conector jack de 3.5mm de salida de audio. Esto no es grave, ya que hay formatos de ficheros de cinta que tienen unos ID de bloque que indican que se debe hacer una parada antes de continuar con la reproducción del audio.

El funcionamiento que se explica a continuación es con la versión 1.54 del firmware.


6.1.- Arranque del MaxDuino

En el arranque se puede ver un logo durante segundo y medio. Al compilar el firmware se puede elegir entre una serie de ellos cuál se desea que aparezca.



6.2.- Pantalla inicial

Aparecerá el primer archivo o directorio de la carpeta raíz que se grabó en ella. Los archivos o carpetas no aparecen ordenados alfabéticamente, sino que aparecen en orden cronológico en que fueron grabados.



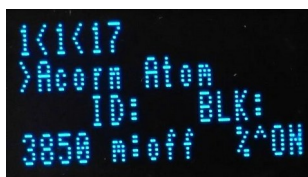
```
v1.54 MaxDuino
>Acorn Atom
ID: BLK:
3850 n:off Z^ON
```

- En la primera fila se ve la versión del firmware, en este caso es la 1.54
- En la segunda fila se ve la carpeta donde nos encontramos posicionados
- En la tercera fila se ven los campos ID y BLK sin informar. Estos campos se verán informados cuando estemos posicionados en un fichero de cinta, y sirven para indicar en que bloque nos encontramos (BLK) y el número identificativo del tipo de bloque (ID). No todos los tipo de bloques se imprimen en pantalla para no ralentizar las cargas, solo los tipos con ID 10,11, 4B y TAP.
- En la cuarta fila se ve la velocidad actual en baudios para las cargas turbo de los ficheros .CAS y .TSX de la plataforma MSX, si el control del motor (remote) está activado o no (on/off) – útil en plataformas que admitían remote como Amstrad CPC, MSX y BBC Micro - , y si está activada o no la opción TSXCzxpUEFSW que sirve para activar la carga turbo (bloques 4B) de los archivos .CAS y .TSX de MSX, el cambio de la polaridad de la señal de audio en los archivos de Spectrum y Amstrad CPC, y el control de paridad en los archivos .UEF de Acorn Electron y BBC Micro.

6.3.- Cambiando la posición actual

Se puede cambiar la posición actual pulsando los siguientes pulsadores:

PIVOT: En la línea superior desaparece la versión del firmware y aparece en que directorio o archivo estamos posicionados.

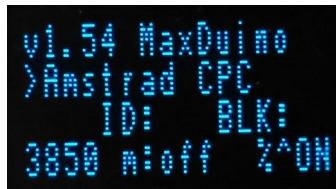


```
1<1<17
>Acorn Atom
ID: BLK:
3850 n:off Z^ON
```

De los tres valores en la línea superior, el central es la posición que ocupa el archivo o directorio en que estamos posicionados dentro de la carpeta actual, y otros dos, los valores inicial y final del intervalo actual. Cuando entramos en una carpeta el primer valor siempre es el primer archivo o directorio, y el tercero el último archivo o directorio que hay en esa carpeta.

Con $1 < 1 < 17$ al entrar en un directorio nos está diciendo que hay 17 directorios o archivos (3er valor), y que nos encontramos posicionados en el primero (2º valor).

DOWN: Para bajar dentro de la lista de archivos y directorios dentro de un carpeta. Podemos mantenerlo pulsado y se desplazará automáticamente (*autoscrolling*).



```
v1.54 MaxDuino
>Amstrad CPC
ID: BLK:
3850 m:off Z^ON
```

UP: Para subir dentro de la lista de archivos y directorios dentro de un carpeta. Podemos mantenerlo pulsado y se desplazará automáticamente (*autoscrolling*).

PLAY / PAUSE: Estando posicionado encima de un directorio entramos dentro de él.



```
Amstrad CPC
>CDT
ID: BLK:
3850 m:off Z^ON
```

Si estábamos posicionados encima del directorio Amstrad CPC, y hemos pulsado PLAY / PAUSE, entramos dentro de él. En este caso nos encontramos posicionados ahora en el directorio CDT y en la línea superior se puede ver el directorio superior desde donde hemos venido.

Aquí se podría entrar dentro de la carpeta CDT pulsando PLAY / PAUSA, y luego pulsar DOWN hasta posicionarnos en la carpeta J. Si estando encima de ella pulsamos PIVOT veremos su posición (la 11 de 27) dentro de la carpeta CDT



```
1<11<27
>j
ID: BLK:
3850 m:off Z^ON
```

STOP / BACK: Se retrocede al directorio inmediatamente superior.

6.4.- Desplazamientos rápidos por semi-intervalos

Esto es útil cuando queremos desplazarnos por una carpeta que tiene muchos archivos, y que el utilizar el botón DOWN para llegar a un determinado archivo que se encuentre bastante abajo de la lista nos llevaría muchas pulsaciones o tiempo.

La búsqueda por semi-intervalos consiste en ir acotando por mitades y quedándote siempre con el punto medio de la mitad superior o inferior.

Cuando entro en la carpeta “/Amstrad CPC / CDT / a” de mi tarjeta SD y mantengo pulsado el botón PIVOT veo la siguiente información

$1 < 1 < 1077$

Es decir, estoy posicionado en el 1er archivo (segundo valor) de una carpeta que tiene 1077 archivos.

Ahora puedo desplazarme con los botones DOWN (avanzar) y UP (retroceder) dentro de la lista de programas que empiezan por la “A”. Es posible mantener pulsados los botones DOWN o UP y nos iremos desplazando sin necesidad de realizar múltiples pulsaciones para avanzar o retroceder.

Si después de desplazarme mantengo pulsado el botón PIVOT y veo algo como

$1 < 47 < 1077$

es que ahora me encuentro posicionado en el archivo o directorio número 47 dentro de esa carpeta.

Si se que el juego que quiero jugar está en la posición 734 tengo la opción de mantener pulsado el botón DOWN hasta llegar a él, o bien usar el desplazamiento por semi-intervalos para acercarme.

Si pulso a la vez PIVOT + DOWN el intervalo pasa a ser [47, 1077] y la posición actual estará más o menos en el medio de ese intervalo.

Hecho esto, si mantengo pulsado ahora el botón PIVOT veo

$47 < 578 < 1107$

¡Acabo de avanzar más de 500 posiciones con una sola pulsación!

Siguiendo el mismo sistema voy avanzando por semi-intervalos acercándome al juego que busco.

$578 < 843 < 1177$

¡Vaya, me he pasado!

Ahora puedo también retroceder por semi-intervalos pulsando a la vez PIVOT + UP siendo el nuevo intervalo [578, 843]

$578 < 710 < 843$

Ya estoy en la posición 710, puedo seguir ahora avanzando al siguiente semi-intervalo con PIVOT + DOWN

710 < 777 < 843


Ahora retrocedo al semi-intervalo anterior con PIVOT + UP

710 < 743 < 777

Cuando ya estamos cerca es cuando tenemos que decidir si queremos seguir avanzando por semi-intervalos o bien ya usar los botones UP y DOWN para posicionarnos en el archivo o directorio que buscamos.

6.5.- Reproducir un archivo de cinta

Si estamos encima de un archivo de cinta veremos la siguiente información



```
68461 bytes
Jabato (Dinamic
ID: BLK:
3850 n:off Z^ON
```

En la primera fila el tamaño en bytes del archivo, y en la segunda el nombre del archivo, que en caso de no entrar en los caracteres de una fila, se irán desplazando para poder ver el nombre entero.

Podemos realizar las siguientes acciones pulsando los botones:

PLAY / PAUSE: La primera vez que lo pulsemos empezará a reproducir el archivo de cinta. En la 1ª línea veremos “Playing”, el % de la carga que llevamos y el número de segundos que han transcurrido. Si lo volvemos a pulsar se quedará en pausa y en la 1ª línea veremos “Paused”, de forma que si volvemos a pulsarlo por tercera vez se reanudará la reproducción desde donde estuviésemos parados.

Mientras se está reproduciendo el archivo vamos viendo en la tercera línea como vamos avanzando de número de bloque (BLK) y el tipo de bloque que se está reproduciendo en ese momento (ID).

Si tenemos activado el modo turbo (%^ON), las cargas de este tipo bloques en los archivos .CAS y .TSX de MSX se reproducirán a la velocidad que esté actualmente configurada para los bloques turbo (en la pantalla se ve que es actualmente de 3.850 baudios).

Si tenemos el cable “remote” conectado al jack 2.5mm, y la opción control motor activada (on), el ordenador tendrá el control para realizar pausas en la reproducción del archivo cuando desee, así como reanudarlo, es decir, tiene el control del botón PLAY / PAUSE.

Estando en pausa entramos en **modo bloque**, y podemos usar estas combinaciones:

UP y DOWN: Desplazarnos por los bloques del archivo. Este modo se usa básicamente para los juegos multicarga que obligaban, en caso de fallar, a repetir una fase anterior que había que volver a cargarla rebobinando el cassette. Para hacer eso, una vez que se ha cargado una fase de un juego, que generalmente es un determinado bloque, si tenemos que volver a cargarla, estando en **modo bloque** se retrocede pulsando el botón UP hasta el número de bloque en el que queremos posicionarnos, y a continuación se pulsa PLAY para cargarlo. El número de bloque en el que estamos posicionados, así como su tipo, aparecen en la fila superior.

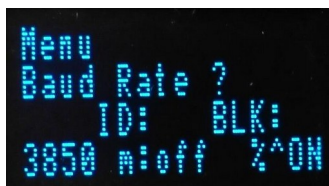
PIVOT + DOWN y PIVOT + UP: Desplazarnos por semi-intervalos, como se explicó en el apartado 6.4, pero por los bloques.

PIVOT + STOP / BACK: Activar o desactivar en ese momento la opción TSXCzxpUEFSW sin necesidad de acceder al menú de configuración.

STOP / BACK: Se para la reproducción de audio y se vuelve al principio del archivo.

6.6.- Menú de configuración

Si no estamos reproduciendo un archivo podemos acceder al menú de configuración pulsando la combinación PIVOT + STOP / BACK. Una vez dentro podemos desplazarnos entre las distintas opciones con los botones UP y DOWN.



6.6.1.- Opción Baud Rate

La primera opción que podemos configurar es la tasa de baudios. Es la velocidad de carga para los bloques turbo de los ficheros .CAS y .TSX de MSX.

Las velocidades que podemos seleccionar son: 1.200, 2.400, 3.600 y 3.850 baudios.

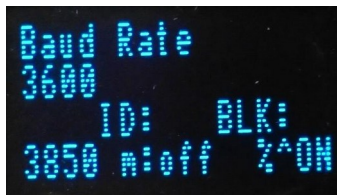
Para poder entrar en esa opción pulsamos PLAY



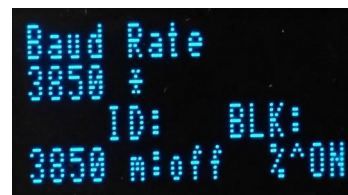
Veremos 1.200 en la segunda línea y al no tener a continuación un asterisco nos está indicando que no es la opción por defecto actual. Pulsando DOWN y UP nos movemos entre las distintas velocidades.



```
Baud Rate
2400
ID:      BLK:
3850 m:off Z^ON
```



```
Baud Rate
3600
ID:      BLK:
3850 m:off Z^ON
```



```
Baud Rate
3850*
ID:      BLK:
3850 m:off Z^ON
```


Observamos que la velocidad 3.850 lleva un asterisco porque es la velocidad actual para los bloques turbo de los ficheros .CAS y .TSX de MSX. Para cambiar a otra nos posicionamos en ella y pulsamos PLAY / PAUSA.

La selección de baudios 1200, 2400, 2700 y 3600 solo es exclusivo para el bloque 4B en los formatos .TSX y .CAS de MSX, y no influye para nada en los baudios de carga de otros tipos de bloques y tipos de ficheros de otras plataformas.

Para salir de esta opción pulsamos STOP / BACK.

6.6.2.- Opción Motor Ctrl

La segunda opción que se puede configurar es el control del motor del cassette.



```
Menu
Motor Ctrl ?
ID:      BLK:
3850 m:off Z^ON
```

Si tenemos el cable “remote” conectado al jack 2.5mm, y la opción control motor activada (on), el ordenador tendrá el control para realizar pausas en la reproducción del archivo cuando desee, así como reanudarlo, es decir, tiene el control del botón PLAY / PAUSE.



```
Motor Ctrl
off*
ID:      BLK:
3850 m:off Z^ON
```

Pulsando el botón PLAY / PAUSA entramos en esa opción y veremos el valor actual, ON* si esta activado y off * si está desactivado. Pulsando nuevamente PLAY / PAUSA activamos o desactivamos la opción.

Para salir de esta opción pulsamos STOP / BACK.

6.6.3.- Opción TSXCzxpUEFSW

Con esta tercera opción podemos habilitar 3 funcionalidades distintas, siendo realmente el nombre largo de la opción TSXCONTROLzxpolarityUEFSWITCHPARITY

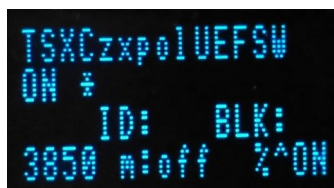


Las funcionalidades son:

- Activa la carga turbo para los archivos .CAS y .TSX de MSX
- Cambia la polaridad de la señal de audio de los archivos para Spectrum y Amstrad CPC. Necesario para la carga de ciertos juegos de Amstrad CPC en formato CDT como por ejemplo:
 - i) Sin cambio de polaridad: (off) : Tai-Pan, Forbidden Planet, Starbike
 - ii) Con polaridad invertida (on): Basil The Great Mouse Detective, Mask
- Cambia la paridad en los archivos .UEF de Acorn Electron y BBC Micro. Necesario para la carga de juegos como Starquake o The Sentinel. Se puede consultar una lista de cintas protegidas en las que hay que activar la paridad en este enlace:

[http://beebwiki.mdfs.net/List of copy protected software titles on cassette](http://beebwiki.mdfs.net/List_of_copy_protected_software_titles_on_cassette)

Pulsando el botón PLAY / PAUSA entramos en esa opción y veremos el valor actual, ON* si esta activado y off * si está desactivado. Pulsando nuevamente PLAY / PAUSA activamos o desactivamos la opción.



Para salir de esta opción pulsamos STOP / BACK.

7.- ¿Cómo actualizar el firmware MaxDuino?

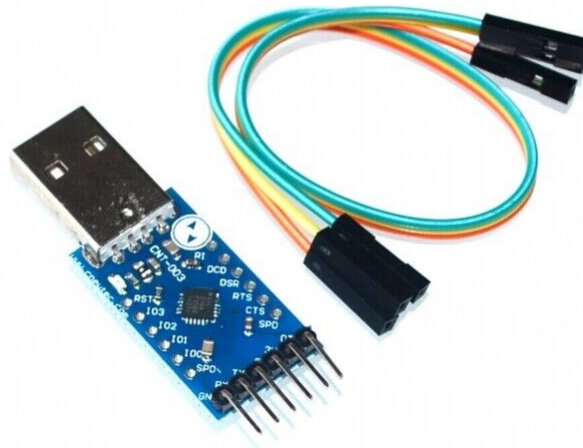
Cada cierto tiempo Rafael Molina va publicando en su github nuevas versiones del firmware para corregir algún error y/o añadir alguna nueva funcionalidad. Cada vez que se publica una nueva versión se indican los cambios nuevos, por lo que es decisión nuestra si nos merece la pena actualizar o no nuestra placa.

Voy a explicar como lo hago con mi Miniduino, pero supongo que para el resto de placas será bastante parecido.

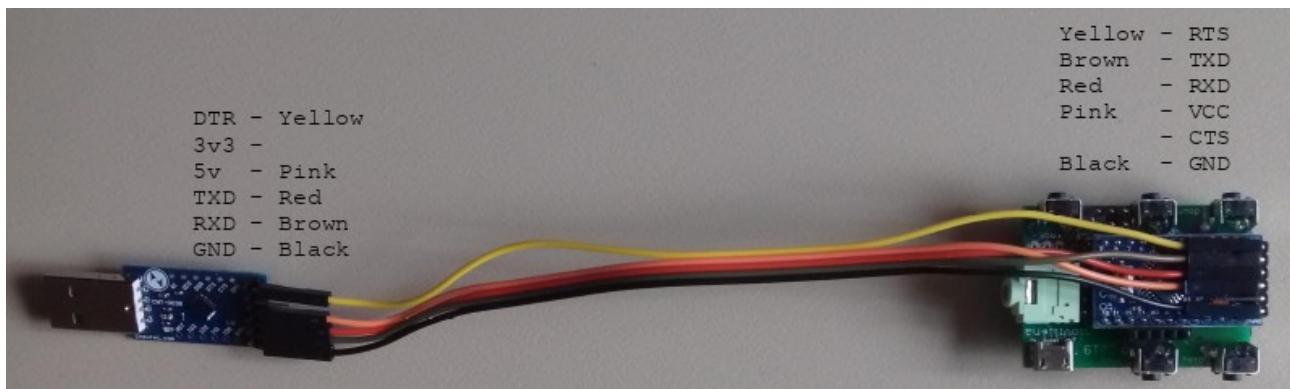
En el caso del Miniduino la actualización del firmware del chip del Arduino Nano Pro se hace a través de una serie de pines que lleva la placa. Para poder conectarlos al ordenador necesito un conversor USB a serie, en concreto el que yo uso en el siguiente:

CP2104 USB to RS232 TTL UART 6PIN Connector Module Serial Converter for Arduino

<https://www.ebay.es/itm/CP2104-USB-to-RS232-TTL-UART-6PIN-Connector-Module-Serial-Converter-for-Arduino/183426600814?ssPageName=STRK%3AMEBIDX%3AIT&trksid=p2057872.m2749.l2649>



Y la conexión entre este conversor USB a RS232 y la placa del Miniduino es la siguiente:



En caso de que no tuviésemos instalado el IDE de Arduino, accedemos a su página web (<https://www.arduino.cc/en/main/software>), descargamos la versión para nuestro sistema operativo (Windows, linux o Mac) y la instalamos. La versión que yo tengo ahora instalada es la 1.8.10 de linux. La versión que recomienda Rafael Molina es la 1.8.4, pero yo no he tenido ningún problema con la que he utilizado.

Una vez instalado accedemos al github de Rafael Molina (<https://github.com/rcmolina/MaxDuino>) y bajamos los fuentes del firmware MaxDuino al ordenador (la última versión en estos momentos es la 1.54)

Abrimos el Arduino IDE, vamos a la carpeta donde hemos dejado los fuentes y cargamos el archivo MaxDuino_v1.xx.ino (donde 1.xx es la versión).

Veremos que el código del proyecto se distribuye en una serie de pestañas:

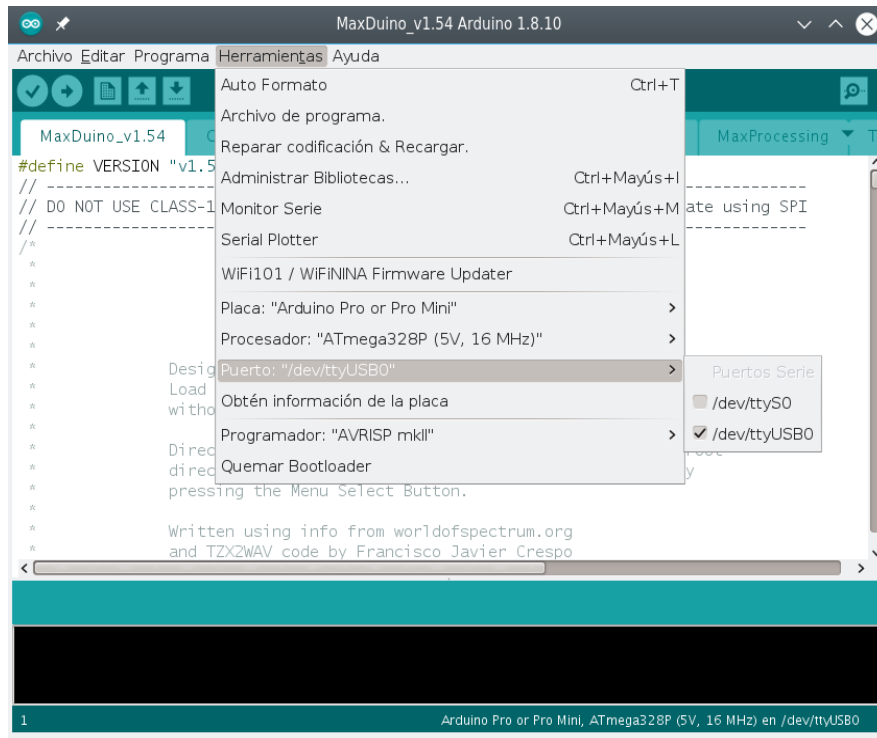


El siguiente paso es seleccionar a través del menú Herramientas la placa, el tipo de arduino y el puerto utilizado para comunicarse con el conversor USB a RS232. En mi caso las opciones que tengo que seleccionar son:

Placa: Arduino Pro o Pro mini

Procesador: ATmega328P (5v, 16Mhz)

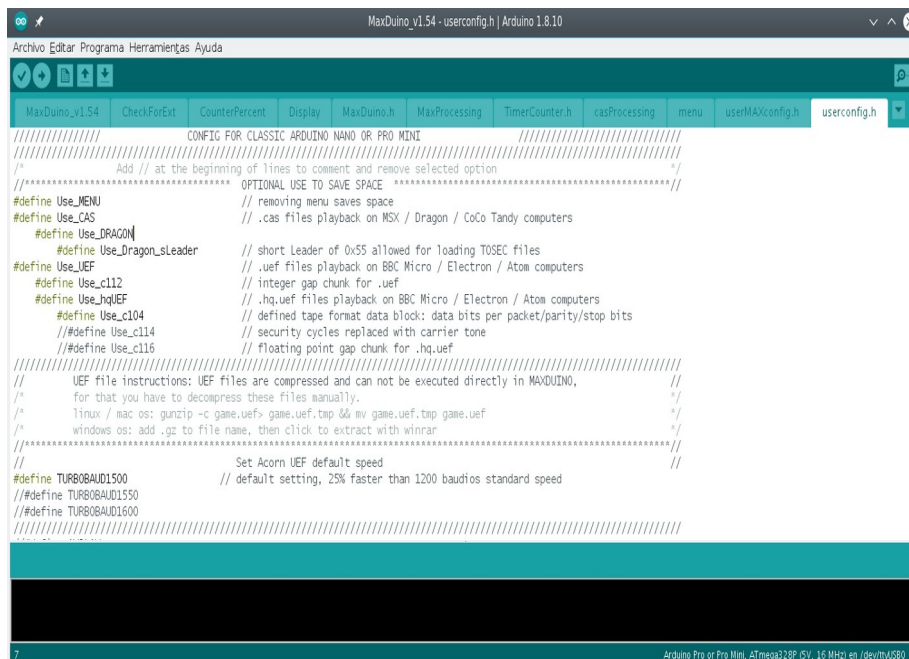
Puerto: /dev/ttyUSB0 (en Windows es más común seleccionar algo como COM4)



En el firmware hay 3 ficheros de configuración del usuario, y hay que usar el adecuado en función del procesador seleccionado en la IDE:

- userconfig.h → Para el Arduino Nano (ATmega328) y el Arduino Pro Mini
- userEVERYconfig.h → Para el Arduino Nano Every
- userMAXconfig.h → Para el Arduino Mega 2560

Como mi Minuino tiene un Arduino Pro Mini, el archivo que tengo que modificar es el **userconfig.h**



Las definiciones se descomentan (quitar la doble barra //) para que tengan efecto o bien se comentan (poner la doble barra //) para que no lo tengan. Según las funcionalidades que dejemos activas, el firmware irá pudiendo crecer en tamaño, por lo que si no entrase en la memoria del chip arduino de nuestra placa, deberíamos tener que comentar alguna para prescindir de ella, y volver a compilar.

En mi caso he dejado todas las funcionalidades que ya están activas por defecto, y solo he tenido que personalizar tres cosas para dejarlo a mi gusto:

- El tipo de pantalla que usa la placa Miniduino
- La funcionalidad de PIVOT en el botón ROOT.
- El LOGO que quiero que aparezca en el arranque

Así los cambios que he hecho son:

```
#define OLED1306 // Set if you are using OLED 1306 display
// #define OLED1306_128_64 // 128x64 resolution with 8 rows
#define OLED1106_1_3 // Use this line as well if you have a 1.3" OLED screen
```

La 2ª línea está comentada para que la definición OLED1306_128_64 no tenga efecto. La pantalla OLED del Miniduino permite las 8 líneas, pero entonces los caracteres se ven muy pequeños (tamaño 8x8) y además solo se aprovechan 4 de las 8 filas, por lo que es mejor dejarla comentada, con lo cual la información se presenta en 4 filas, y los caracteres son de 8x16 más legibles.

También dejo descomentada la siguiente definición para que el botón ROOT funcione como PIVOT

```
#define btnRoot_AS_PIVOT
```

Al final para el LOGO no he hecho ningún cambio, y con las definciones siguientes me sale el logo del MINI que creó Rafael para el Miniduino, pero podría haber comentado la última definición y haber seleccionado otro logo.

```
#ifndef OLED1306_128_64
  //#define Maxduino2Alf          // new Maxduino2 logo 128x64 by Alfredo Retrocant
  #define CosmicCruiser          // Dragon32 CosmicCruiser by rcmolina
#else
  //#define LOGOARDUITAPE          // original arduitaape logo
  //#define LOGOMAXDUINO           // new logo created by Spirax
  //#define LOGOMAXDUINO2          // new logo2 created by rcmolina
  //#define LOGOMAXDUINO3          // new logo3 created by rcmolina
  //#define LOGOMAXDUINO4          // new logo4 created by rcmolina
  //#define SUGARLESS_1            // new logo cretaed bt YoxxxoY
  //#define SUGARLESS_2            // new logo created by YoxxxoY
  #define LOGOMINIDUINO           // new logo for A.Villena's Miniduino by rcmolina
#endif
```

Ya con los cambios realizados estaríamos listos para compilar el firmware a nuestro gusto y enviarlo a la memoria del Arduino. Sin embargo, en este mismo fichero nos indican una forma de ahorrar memoria introduciendo el logo en la EEPROM.

```
////////////////////////////////////
// EEPROM LOGO. How to move to EEPROM, saving memory:
// Phase 1: Uncomment RECORD_EEPROM_LOGO define , this copies logo from memory to EEPROM.
Compile the sketch.
// Phase 2: Comment RECORD_EEPROM define, uncomment LOAD_EEPROM define. Compile the
sketch again
////////////////////////////////////
// Also it's possible to select record and load both for better testing new logo
activation, pressing MENU simulates a reset.
// And both can be deactivated also showing a black screen.
////////////////////////////////////
```

Lo que hay que hacer es compilar y enviar el firmware a la placa 2 veces:

La primera vez dejamos así estas dos líneas

```
#define RECORD_EEPROM_LOGO // Uncommenting RECORD_EEPROM deactivates #define Use_MENU
// #define LOAD_EEPROM_LOGO
```

Y le damos al botón “Subir” o CTRL+U, y si todo está bien, compilará el firmware y lo subirá a la placa. Si hay algún error este aparecerá en la ventana inferior del IDE, y habrá que solucionarlo.

La segunda vez cambiamos esas dos líneas a:

```
// #define RECORD_EEPROM_LOGO // Uncommenting RECORD_EEPROM deactivates #define Use_MENU
#define LOAD_EEPROM_LOGO
```

Y volvemos a darle al botón de subir, con lo que ya habremos terminado de actualizar el firmware.

Esta acción de compilar 2 veces solo hay que hacerlo la primera vez, o si vamos a cambiar el LOGO, para las restantes podemos dejar las definiciones como la segunda vez.

8.- ¿Dónde se pueden adquirir las placas MaxDuino?

Se pueden adquirir normalmente en las tiendas on-line de **Antonio Villena** o de **Manuél Fernández Higuera**s

- <https://www.antoniovillena.es/store/>
- <https://manuferhi.com/>

Mirando en el foro de va-de-retro por si se hace algún tipo de tirada nueva como esta de **merlinkv**

- **Tirada MegaDuino:** <https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=8496>

Buscando en eBay **tzxduino** o **maxduino**

Y si se es mañoso con la electrónica, puede hacérselo uno mismo, por ejemplo, con los diseños compartidos de:

- Va de Retro **pTZXduino**: <https://www.va-de-retro.com/foros/viewtopic.php?p=177736#p177736>
- Edu Arana de su **TZDuino Reloaded**: <https://github.com/arananet/TzxDuino-Reloaded>

9.- ¿Desea saber más?

El contenido de este manual es básicamente una puesta ordenada de la información dispersa que se encuentra en estos hilos de **va-de-retro.com**:

- Probando el TZXDuino: <https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=5541>
- Miniduino (TZXDuino low cost): <https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=7448>
- Probando el TSXDuino MEGA: <https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=8488>
- Tirada Megaduino: <https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=8496>

y del README del proyecto:

https://github.com/rcmolina/MaxDuino_v1.54/blob/master/README.md

Para saber más sobre como están codificados estos archivos de cinta se puede leer el documento

Tutorial para la generación de archivos TSX de Natalia Pujol (NataliaPC)
https://github.com/nataliapc/makeTSX/blob/master/docs/Tutorial_TSX_es.pdf

En YouTube podemos ver muchos vídeos muy interesantes del MaxDuino en acción en el canal de Rafa Molina: <https://www.youtube.com/channel/UC0L03I8mb4CMytW7MuhF--g/videos>

También hay muchos vídeos en el facebook de Alfredo “Retrocant”:
<https://www.facebook.com/retrocant>

APÉNDICE 1. TIPOS DE BLOQUES

Formato: **TZX**

ID	Valor	Descripción
ID10	0x10	Standard speed data block
ID11	0x11	Turbo speed data block
ID12	0x12	Pure tone
ID13	0x13	Sequence of pulses of various lengths
ID14	0x14	Pure data block
ID15	0x15	Direct recording block -- TBD - curious to load OTLA files using direct recording (22KHz)
ID18	0x18	CSW recording block (No soportado)
ID19	0x19	Generalized data block (No soportado)
ID20	0x20	Pause (silence) or 'Stop the tape' command
ID21	0x21	Group start
ID22	0x22	Group end
ID23	0x23	Jump to block
ID24	0x24	Loop start
ID25	0x25	Loop end
ID26	0x26	Call sequence
ID27	0x27	Return from sequence
ID28	0x28	Select block
ID2A	0x2A	Stop the tape if in 48K mode
ID2A	0x2B	Set signal level
ID30	0x30	Text description
ID31	0x31	Message block
ID32	0x32	Archive info
ID33	0x33	Hardware type
ID35	0x35	Custom info block
ID4B	0x4B	Kansas City block (MSX/BBC/Acorn/...)
IDPAUSE	0x59	Custom Pause processing
ID5A	0x5A	Glue block (90 dec, ASCII Letter 'Z')
AYO	0xFB	AY file
ZXO	0xFC	ZX80 O file
ZXP	0xFD	ZX81 P File
TAP	0xFE	Tap File Mode
IDEOF	0xFF	End of file

Formato: UEF

Chunks	Valor	Descripción
ID0000	0x0000	origin information chunk
ID0100	0x0100	implicit start/stop bit tape data block
ID0104	0x0104	defined tape format data block: data bits per packet/parity/stop bits
ID0110	0x0110	carrier tone (previously high tone)
ID0111	0x0111	carrier tone (previously high tone) with dummy byte at byte
ID0112	0x0112	Integer gap: cycles = (this.baud/1000)*2*n
ID0114	0x0114	Security Cycles replaced with carrier tone
ID0116	0x0116	floating point gap: cycles = floatGap * this.baud
IDCHUNKEOF	0xffff	