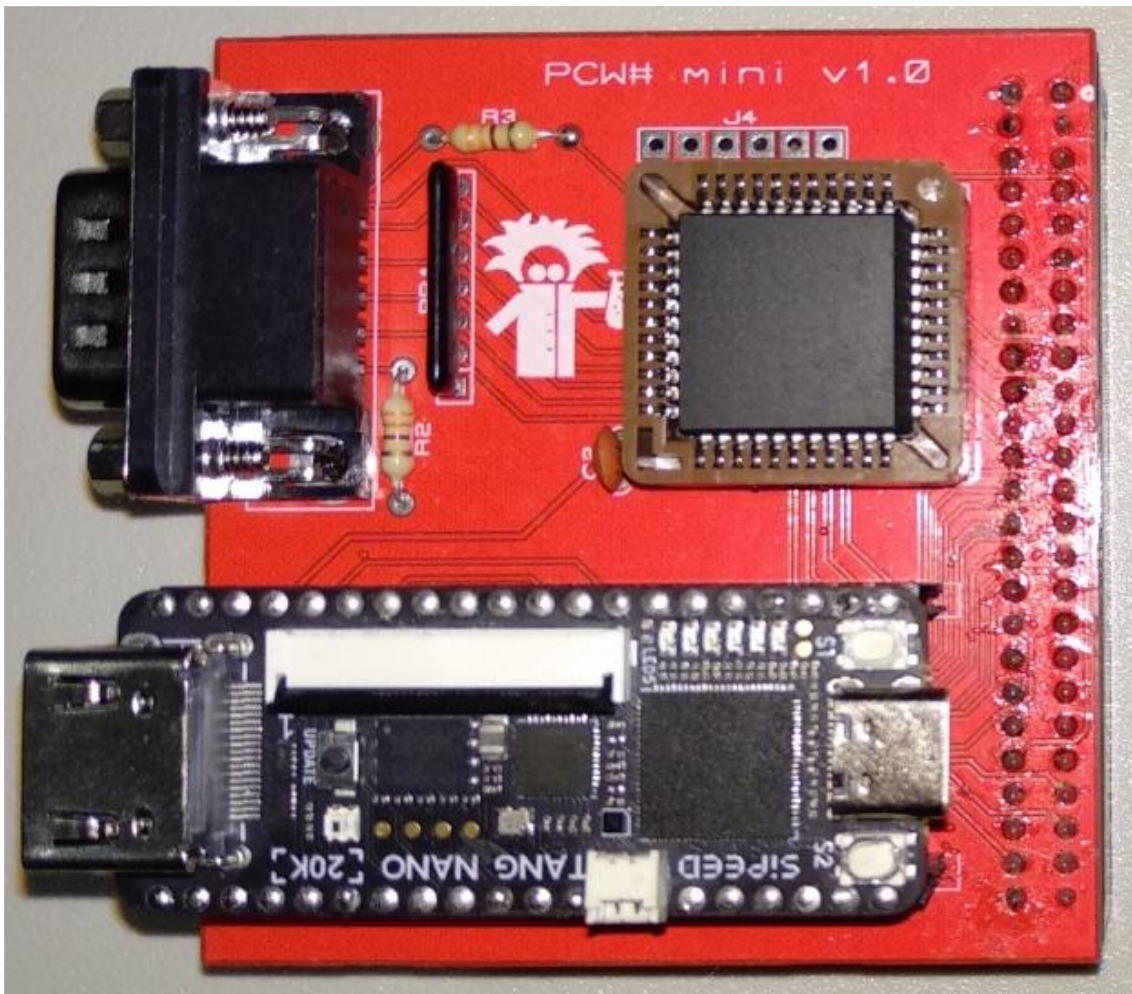


PCW# mini



Manual de usuario

Contenido

Introducción	3
Características.....	6
Definiciones.....	6
Hardware	7
Software	8
ColorIn	8
Dk'TurboSound.....	9
Software incluido	11
EdiPal.....	11
ParBios.....	11
Actualización	12
Temas legales.....	14

Introducción

PCW# mini es una versión simplificada de PCW#, proyecto nunca publicado por ser demasiado caro y complejo; y reconozco que también por vaguería.

PCW# surge como experimento; muchos juegos, al ser portados a Amstrad PCW desde otras plataformas, lo hicieron con los gráficos tal cual si no se veían mal. La mayoría de los juegos de Opera, por ejemplo, utilizan los mismos gráficos que sus versiones PC CGA; la situación en este caso es casi idéntica a la relación CGA / Hercules del mundo del PC.

Otras compañías fueron más lejos; Level 9 es famosa por haber publicado discos de 3" que funcionan tanto en ZX Spectrum +3, Amstrad CPC y PCW. En ellos, los gráficos de PCW son los mismos de CPC modo 0 (16 colores).

En cualquier caso, la salida de vídeo en el PCW se produce a niveles TTL. Podemos interpretarla como bits y agruparlos de 2 en 2 o de 4 en 4 para obtener modos gráficos similares a los que usaba la versión original, reinterpretando la señal.



Ejemplos de reinterpretación de vídeo, 4 y 16 colores.

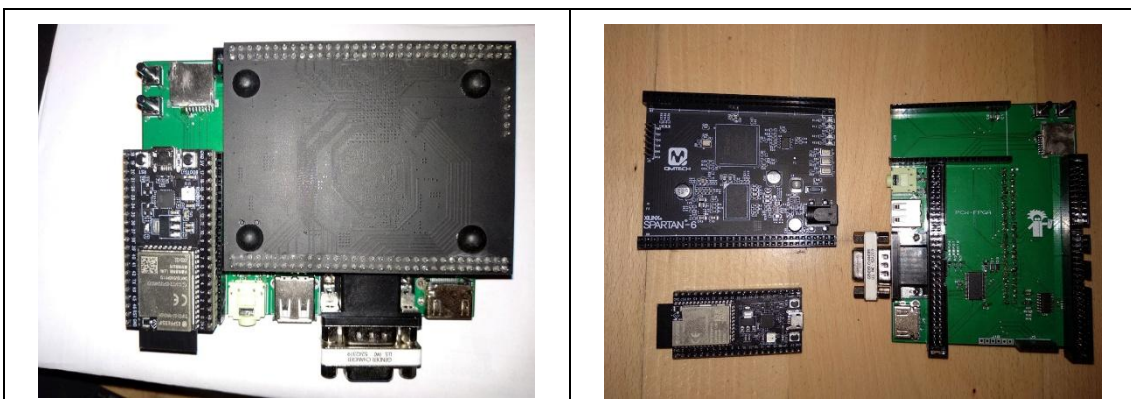
Al ser un experimento, se le añadió también un modo de atributos, que no sólo facilita el porteo de software de Spectrum, MSX, Thomson, ... sino que dobla potencialmente la velocidad del volcado a video.

Inicialmente empecé haciendo pruebas con una Cyclone II y un simple DAC de 6 bits (64 colores EGA). Eventualmente migré a una Spartan 6 con SDRAM (con soporte LVDS

que nos permite codificar TMDS, y por tanto HDMI) y finalmente otra más grande con DDR. Por el camino les fui añadiendo acceso completo al bus del PCW, WiFi, SD, USB, ...



Presentación “en sociedad” del primer prototipo. Seguro que alguien reconoce la ubicación.

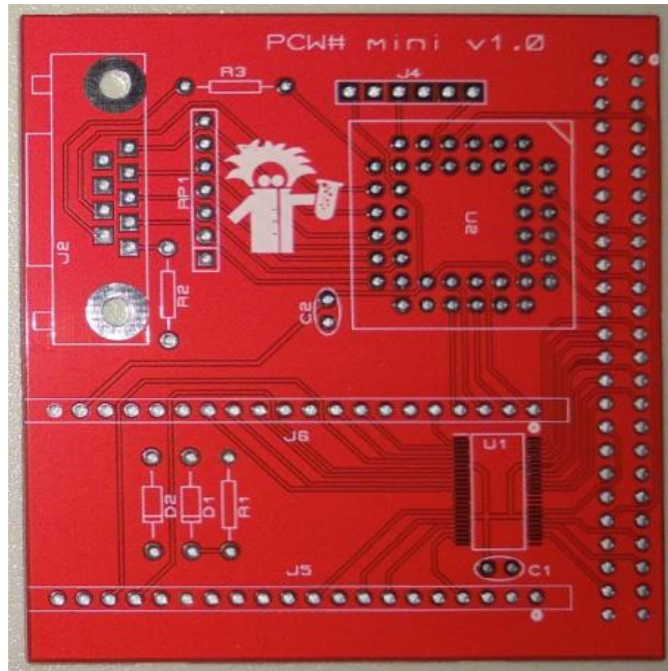


Último prototipo, montado y desmontado.

Viendo la buena acogida original, y aprovechando que el emulador tenía añadidos los modos e interfaz de la PCW# (incluyendo características no documentadas), decidí publicarlo.

No es, sin embargo, hasta años más tarde cuando hablando con Enrique G. del tema me decidí finalmente a sacar una versión simplificada utilizando una Tang Nano que tenía olvidada en un cajón; la idea era simplificar y crear algo barato y fácil de montar, sin sacrificar lo principal: ofrecer modos de color en un PCW, y añadir luego tantos extras como fuese posible.

Y así es como nace la PCW# mini; está basada en una Tang nano 20K, placa que trae una FPGA mediana y viene con conector HDMI de serie. Aunque tiene pocas patas expuestas aprovechables, la situación se alivia usando un pequeño CPLD.



Al poder inyectarse audio, se le añadió una doble tarjeta Dk'tronics (mi variante estéreo con DAC), así como modos de scanlines / efecto CRT (similares al emulador), joystick normal Atari, etc.

Quizás no pueda ampliar la memoria, interceptar el arranque, proporcionar WiFi, ... Pero tampoco es algo que se necesite para disfrutar de unas partidas con audio y video mejorados, lo cual es su objetivo.

Características

PCW# mini ofrece:

- Enchufar y listo; no hay que abrir un PCW ni modificar nada.
- Salida HDMI, estándar digital actual.
- Incluye vídeo y audio.
- Control completo por software.
- Se pueden conmutar modos y scanlines de forma manual.
- Incluye los modos: 0-4 de ColorIn.
- Incluye paletas por defecto y programables de 24b y 8b simplificada.
- Incluye varios modos de scanlines y filtro CRT, de forma similar al emulador.
- Incluye DOBLE tarjeta de sonido Dk'tronics: dos chips de sonido, dos DACs y soporte para dos joysticks. Todo en estéreo.
- Conector de joystick norma Atari, con soporte de dos botones (CPC y MSX), así como para un segundo joystick con un splitter de CPC.
- Actualizable (incluyendo nuevas funcionalidades) con un simple cable USB.

Limitaciones:

- La interfaz espera video PAL, pues no existieron comercialmente PCWs NTSC. Quizás se añada soporte en el futuro. Se descartó la idea de forzarlo porque no es algo que corresponda a esta placa.

Definiciones

- Bit: Unidad básica de información, con dos estados lógicos: "0" y "1". Por convenio, se escribirán en orden de más a menos significativo si hay más de uno.
- Nibble: 4 bits, numerados por tanto de 3 a 0.
- Byte: 8 bits, numerados de 7 a 0 (o bien dos nibbles). Cualquier secuencia se especificará en orden creciente de memoria o acceso a puerto.
- PSG: Programmable sound generator, el chip de sonido utilizado. En el caso de la PCW#mini se trata del Yamaha YM2149.

Hardware

El núcleo de la interfaz consiste en una Tang Nano 20K, placa con HDMI de serie (entre otras cosas), basada en una FPGA GW2AR-18. Debido a la escasez de patas aprovechables, se usa un CPLD XC9536XL como apoyo (decodificación de direcciones y serializado de los joysticks). El resto es simplemente soporte.

El conector de Joystick admite norma Atari, con dos botones (tipo CPC o MSX), con común en pin 8 o 9 (selección tipo CPC), y alimentación en 5. Los pines 5 y 9 están protegidos con resistencias por si acaso se conectan joysticks que los usen para otras cosas. A la vez, se pueden usar para comunicaciones, incluyendo por bucle de corriente (p.ej. MIDI).

La interfaz se alimenta directamente del PCW, usando algo menos de 130 mA. Se usa la línea de 5V para alimentar a la Tang, y los 3.3V generados por ésta para el CPLD. Existe un diodo protector por si se conecta un cable USB a un ordenador con el PCW encendido (o un dispositivo HDMI que entregue alimentación).

Se utilizan los botones S1 y S2 para cambiar manualmente de modo gráfico (con cambio de paleta) y de scanlines / filtro, respectivamente.

El botón UPDATE así como los 6 leds normales están disponibles para el usuario. El led RGB indica estado al arrancar durante 2 segundos (rojo = no detecta PCW, ámbar = no detecta HDMI, verde = todo bien), y después también se hace disponible para el usuario.

Se anulan por incompatibilidad la interfaz LCD (apagando el inversor para ahorrar) y el DAC + amplificador. Siguen disponibles la tarjeta SD (modo SPI), los tres relojes adicionales, acceso a la Flash y características internas de la FPGA (SDRAM, oscilador, un PLL, posibilidad de multiboot, etc.).

De los dos PLL que ofrece la GW2 se emplea uno para elevar la frecuencia para el HDMI. Se usa el reloj del PCW de 32 Mhz como reloj global. El diseño es completamente síncrono y sin latches.

De momento no se usa la posibilidad de ofrecer USB OTG con el BL616 incluido en la Tang. Se está experimentando con la posibilidad de utilizar mandos USB de PC.

Precauciones (aparte de las de siempre, como: no conectar o desconectar la PCW# mini con el PCW encendido, no conectarla al revés, no mojarla, etc.):

- No se recomienda programar el CPLD o la Tang con la PCW# mini conectada a un PCW encendido (es posible, pero es preferible evitar inestabilidades).
- No conectar la Tang al revés (el HDMI va en la misma dirección que el joystick) o desplazada.

Software

Desde el punto de vista de un programador, la interfaz integra los siguientes dispositivos:

ColorIn

Este dispositivo permite reinterpretar la salida monocroma del PCW como modos de color. Utiliza dos puertos, ambos de lectura y escritura:

\$80 – Contiene el índice al registro al cual se accede por el puerto \$81.

\$81 – Contiene el registro a ser accedido.

Los registros se agrupan por funcionalidad en el nibble superior del índice; el acceso, ya sea lectura o escritura, incrementa el índice internamente para poder enviar fácilmente datos desde el Z80 excepto en los registros específicos.

No debe asumirse ningún valor en registros o bits no definidos. En escritura, un bit no definido debe ser 0 por compatibilidad futura.

Mapa de registros

Grupo	Índice	Función
Modo vídeo	\$00	Modo de vídeo usado
Paleta 24b	\$10-\$1F	Valores de la paleta 24b
Paleta 8b	\$20-\$2F	Valores de la paleta 8b
Borde	\$30	Valores del borde 24b
	\$31	Valor del borde 8b
	\$32	Control del borde
Específico PCW# mini	\$FE	LED RGB
	\$FF	Depurado

\$00 – Tanto en lectura como escritura, los bits 6-0 especifican el modo de vídeo utilizado / a utilizar. En escritura, el bit 7 indica si no queremos cambio de la paleta a la predeterminada (0 = cambio, 1 = no cambio). En lectura, el bit 7 no está definido.

\$10 - \$1F – Acceso a la paleta 24b del color correspondiente en el nibble bajo. Se necesitan 3 accesos por cada entrada, en orden B-G-R.

\$20-\$2F – Acceso a la paleta simplificada, un solo byte por entrada. El formato es $R_2R_1R_0G_2G_1G_0B_1B_0$. Internamente se convertirá a 24 bits, tanto en lectura como escritura, truncando / doblando los bits necesarios ($R_2R_1R_0G_2G_1G_0B_1B_0 \Leftrightarrow R_2R_1R_0R_2R_1R_0R_2R_1G_2G_1G_0G_2G_1G_0G_2G_1B_1B_0B_1B_0B_1B_0B_1B_0$).

\$30 – Funciona igual que \$10-\$1F, pero para el borde.

\$31 – Funciona igual que \$20-\$2F, pero para el borde.

\$32 – El bit 0 controla si tenemos activo el borde o no (1 = activo, 0 = no activo). El color de la zona de overscan quedará determinado por el color del borde o de la entrada 0 de la paleta, respectivamente.

\$FE – Funciona de forma similar a \$30, pero para el led RGB de la Tang.

\$FF – Registro de depurado; los 6 bits inferiores se corresponden a los LEDs de la Tang (1 = apagado, 0 = encendido), y el superior en lectura al botón nombrado como “UPDATE”. Ambas cosas pueden ser útiles para depurar sobre hardware real.

Los modos de vídeo son:

Número	Modo (ancho X alto X colores)
0	720x256x2, modo original del PCW
1	360x256x4
2	180x256x16
3	360x256x16 atributos
4	90x256x256 (no oficial)

Todos los modos de vídeo reinterpretan la señal del PCW. Es decir, los datos están en el orden en el que se visualizan, no en el orden en el que estén en memoria.

Modo 0 – Igual que el PCW original. Cada bit representa un píxel, y se pintan del 7 al 0.

Modo 1 – Agrupa los bits de 2 en 2 para obtener 4 colores. 7-6 definen el índice en la paleta para el primero, 5-4 el segundo, etc. La paleta por defecto es la CGA BMCW con brillo.

Modo 2 – Agrupa los bits de 4 en 4 para obtener 16 colores. 7-4 definen el índice del primer píxel y 3-0 el del segundo. La paleta por defecto es la gama completa CGA RGBI, sin cambiar el amarillo oscuro a marrón como hace el monitor de IBM.

Modo 3 – Éste es un modo de atributos. Se leen dos bytes, el primero de atributos y el segundo de mapa de bits. El byte de atributos contiene los índices para fondo (bits 7-4) y tinta (bit 3-0). Se pintan los bits del mapa de bits como fondo (0) o tinta (1).

Modo 4 – En este modo experimental cada byte es un índice sobre la paleta de 8 bits, logrando así 256 colores.

Dk'TurboSound

Este dispositivo implementa dos interfaces “Dk'tronics Sound and Joystick” ampliadas con DAC y estéreo, usando como base el integrado YM2149 (con respecto al AY-3-8910: dobles niveles de envolventes, bits de registros completos).

Para la programación de los chips de sonido se recomienda leer previamente el datasheet correspondiente.

El funcionamiento es similar al de la expansión original, pero añade el que con los valores \$FE y \$FF del latch se selecciona el PSG a utilizar.

La interfaz utiliza 4 puertos:

\$A8 – Lectura del latch del PSG actual (exclusivo PCW# mini).

\$A9 – Lectura del valor del registro del PSG actual apuntado por el latch.

\$AA – Escritura del latch del PSG actual.

\$AB – Escritura del valor del registro del PSG actual apuntado por el latch.

Con el valor de latch \$FF se selecciona el PSG0, y con \$FE el PSG1. Cada PSG tiene un joystick en el puerto A (lectura) y un DAC de 8b en el puerto B (escritura).

En arranque o reinicio estará activo el PSG0 en modo normal (estéreo ABC + DAC centrado). Cuando se use el latch para seleccionar PSG se asumirá TurboSound y cambiará el modo estéreo a PSG0+DAC0 en el canal izquierdo y PSG1+DAC1 en el derecho.

En lectura, cada joystick tiene el formato:

b7	b6	b5	b4	b3	b2	b1	b0
Disparo 2	Disparo 1	Arriba	Abajo	Derecha	Izquierda	1	1

Éste se lee a través del puerto A (registro \$E) del PSG correspondiente. El puerto B (registro \$F) corresponde a su DAC.

Software incluido

EdiPal

Programa para definir y establecer la paleta y modo gráfico de la PCW# mini. Admite parámetros en línea de comandos y modo interactivo.

Su sintaxis es la siguiente:

EdiPal [Fichero] [Modo] [Acción]

- [Fichero] es el fichero con la paleta a cargar.
- [Modo] es el modo gráfico (0-4) a establecer.
- [Acción] es la acción a llevar a cabo: “Y” para volver a CP/M y “Z” para reiniciar (igual que las teclas en el modo interactivo).

Si no se especifican los tres, se arrancará en modo interactivo. Al iniciar éste, se leerán el modo y paleta actual desde el hardware y, en caso de estar especificados en línea de comandos, estos últimos tendrán preferencia.



La interfaz permite definir los colores con su índice en hexadecimal (0-9, A-F) en formato RGB hexadecimal, cambiar modo, poner la paleta por defecto en el modo (con dos paletas en el caso del modo 0), guardar la paleta (siempre con el nombre “pal.bin”, puede renombrarse desde CP/M a posteriori), volver al CP/M con la paleta y modo elegidos y simular un reinicio con la paleta y modo elegidos.

ParBios

Parche simple para la BIOS, permite capturar las llamadas a la misma para establecer el color de la paleta y del borde y efectuarlas a través de la PCW# mini. Simplemente ejecutar y se parcheará en memoria, para no alterar el fichero EMS.

Actualización

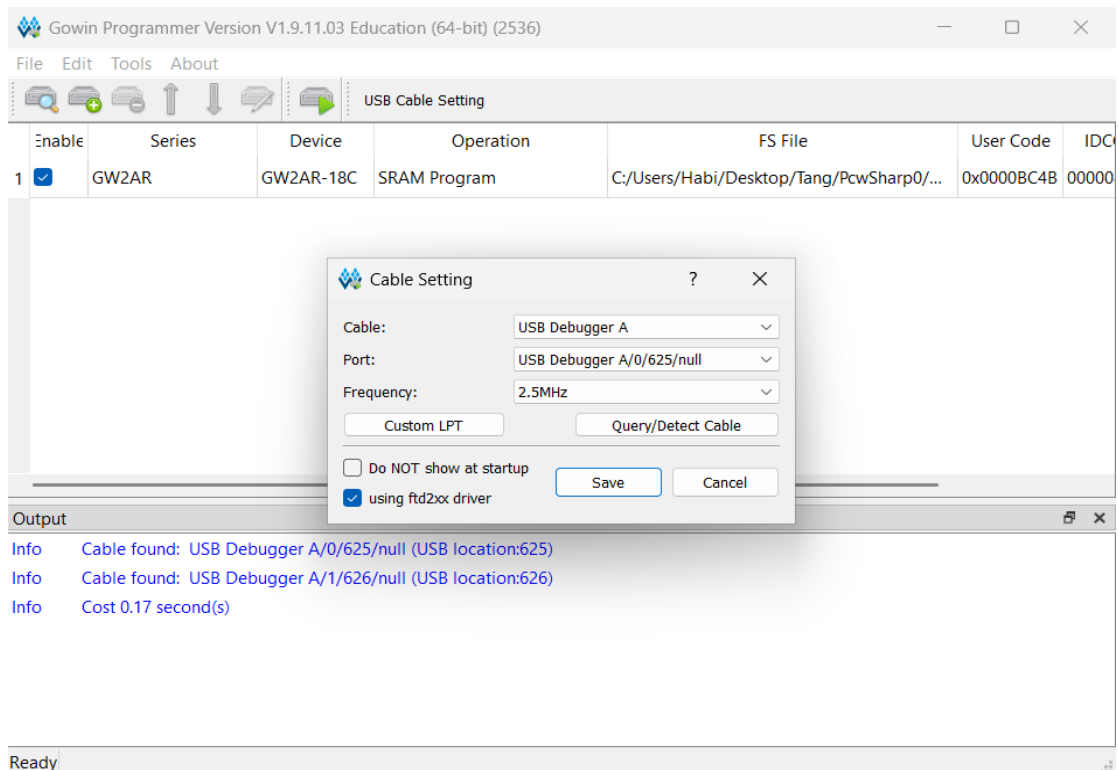
PCW# mini contiene tres componentes que son actualizables: el CPLD, el BL616 y la FPGA (estos dos últimos en la propia Tang). Salvo cambio radical, sólo se actualizará la FPGA; es un proceso sencillo y al alcance de cualquier usuario, el cual se describe a continuación.

Para actualizar se necesitan:

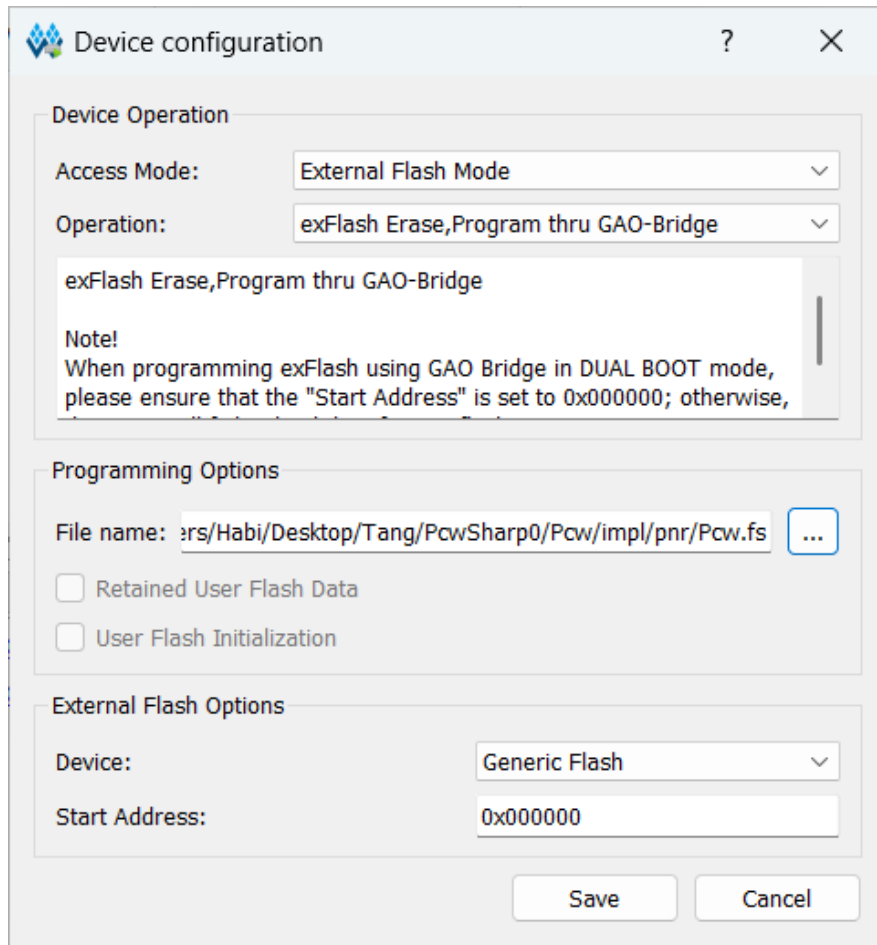
- 1) La placa Tang, preferiblemente desconectada de la PCW# mini.
- 2) Un cable USB (p.ej. el incluido con la Tang).
- 3) El programa “[GOWIN Programmer](#)” instalado en un PC, para su sistema operativo correspondiente.

Los pasos para actualizarla son los siguientes:

- 1) Bajar el fichero correspondiente a la versión que queramos instalar.
- 2) Conectar la Tang con el cable USB al PC.
- 3) Abrir el programa; debiera detectar la FPGA:



- 4) Escoger la opción de grabado “External flash” en el programa como la siguiente captura, seleccionar el fichero y darle a grabar:



5) Listo. Ya se puede desconectar el cable USB, cerrar el programa, etc.

Temas legales

Todas las marcas o derechos de cualquier tipo, nombres registrados, logos o insignias, son propiedad de sus respectivos dueños.