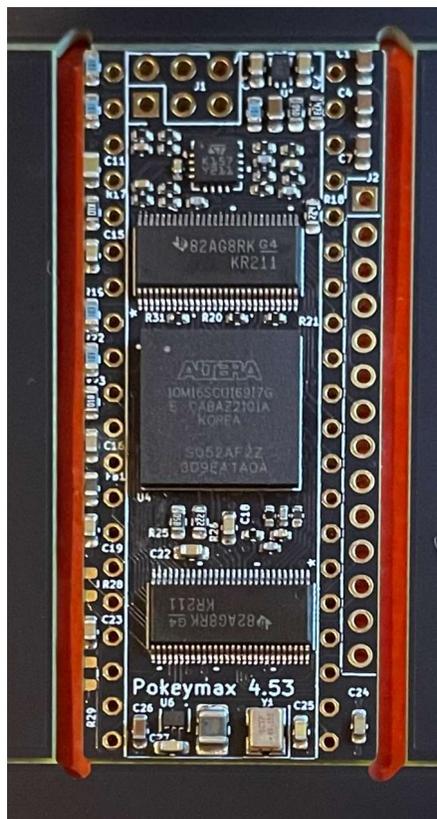


PokeyMAX

Device Overview & Configuration Guide

For PokeyMAX v4.5 • pokeycfg v1.6



PokeyMAX v4.53 — Altera MAX 10 FPGA in a 40-pin DIP package

www.64kib.com

Introduction

PokeyMAX is an FPGA-based replacement for the Atari 8-bit POKEY sound chip. Rather than being a straight one-for-one clone, it is a platform: depending on the build variant fitted, it can simultaneously implement up to four POKEY cores alongside SID, PSG (AY/YM), a DMA sample player, and COVOX-style digital audio. All of these share the original 40-pin DIP footprint, making it a drop-in fit for any Atari 400/800/XL/XE or compatible machine.

The v4.5 'full fat' build described in this guide includes all of the above. Configuration is performed using the `pokeycfg` utility, which runs on the Atari itself and writes settings to flash memory inside the FPGA so that preferences survive power cycles.

NOTE

This guide covers what the features mean and how to configure them. For hardware installation, address-line wiring, and audio output connections please refer to the separate Pokeymax Installation Guide.

Document scope

This guide is organised in two main parts:

- Part 1 — Background: what each implemented chip or technology is, why it exists, and what the configuration choices relate to.
- Part 2 — Using `pokeycfg`: how to navigate the tool, what each setting does, and the critical distinction between applying and saving.

Part 1 — Architecture & Implemented Chips

1.1 How PokeyMAX is structured

The block diagram below shows the internal signal flow. Understanding it makes the configuration options much more intuitive.

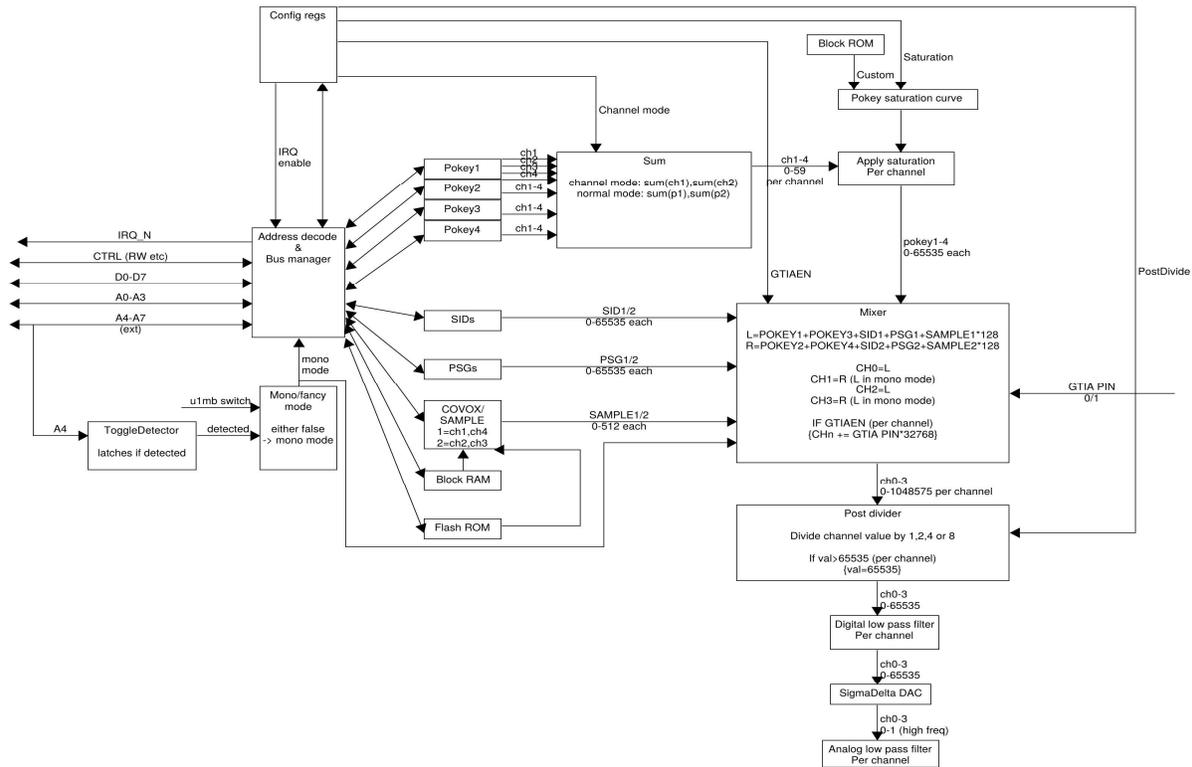


Figure 1: PokeyMAX internal signal flow

Working left to right through the diagram:

- The bus interface decodes addresses \$D200–\$D2FF and routes reads and writes to the correct internal block.
- Up to four POKEY cores, two SID cores, two PSG cores, and the COVOX/SAMPLE engine each produce 16-bit audio values per channel.
- A mixer combines them:
 - Left = POKEY1+POKEY3+SID1+PSG1+SAMPLE1
 - Right = POKEY2+POKEY4+SID2+PSG2+SAMPLE2.
- A post-divider scales each output channel independently (divide by 1, 2, 4, or 8) to control level.
- The signal is then fed to a Sigma-Delta DAC, and finally an analogue low-pass filter before reaching the physical output pins. For S/PDIF output it is digitally low pass filtered first then goes to a S/PDIF encoder.
- GTIA audio can be injected into any output channel, replacing the motherboard mixing path.
- Configuration registers control all of this behaviour and can be stored in on-chip flash memory.

TIP

The A4–A7 address lines (pins 6–9 on connector J2) must be wired to the CPU for the full memory map to work. Without them, PokeyMAX operates in mono POKEY-only mode. See the Installation Guide for wiring details.

1.2 Capability register — what your build includes

PokeyMAX reads a CAPABILITY register at \$D211 on power-up. The pokeycfg utility displays a summary of detected capabilities on its status line. The possible capabilities are:

Capability	Description
1× POKEY	Single POKEY — mono only
2× POKEY	Stereo POKEY pair
4× POKEY (Quad)	Full quad POKEY — broadest software compatibility
2× SID	Two SID cores (one per stereo channel)
2× PSG	Two AY/YM PSG cores
COVOX	Four 8-bit manual volume registers for CPU-driven samples
SAMPLE	DMA sample player with 42 KiB block RAM
FLASH	On-chip UFM/CFM flash — required for saving configuration

The v4.5 full-fat build includes all capabilities listed above.

1.3 POKEY

POKEY (POtentiometer KEYboard) is the original Atari custom chip combining audio generation, keyboard scanning, serial I/O, timers, and a hardware random-number generator. For PokeyMAX users the audio side is what matters.

POKEY sound uses frequency dividers and polynomial-counter noise sources, producing up to four audio channels per chip. A standard Atari has one POKEY; dual-POKEY stereo machines gained popularity through hardware expansions and the Atari XE Game System. Software written for dual POKEY expects two chips at \$D200 and \$D210. Quad POKEY adds a third and fourth at \$D220 and \$D230.

Saturation curve

Real POKEY hardware does not sum channels linearly — it uses a non-linear saturation curve that compresses loud signals slightly, giving a characteristic warmth at high volumes. PokeyMAX can replicate this curve (Pokey mode) or use a mathematically linear sum (Linear mode). For most software the Pokey curve is preferred.

Channel mode

In normal mode each POKEY chip's four channels are summed to a single output value per chip. Channel mode instead routes individual voices to separate output pins. On v4.5 there is only one physical output pin in this path, giving three usable channels in total rather than four, so this option has limited practical use and can generally be left off.

1.4 SID

The SID (Sound Interface Device) is the audio chip of the Commodore 64, famous for its filters, three-voice synthesis, and characterful sound. PokeyMAX implements two SID cores, placing SID1 on the left channel (\$D240) and SID2 on the right (\$D260).

6581 vs 8580

There are two major SID generations that composers and playback engines care about:

Model	Character
6581 (1982–85)	Earlier revision. Rougher, dirtier sound. Filter behaviour varies considerably between real chips. Associated with older C64 music and the warm 'classic' SID tone.
8580 (1987+)	Later revision. Cleaner, more stable, different filter response. Preferred for later productions and emulation-friendly playback.

When a SID music file or tracker specifies a target model, choose accordingly. When in doubt, try 6581 first for older material and 8580 for newer.

Filter distortion (6581 mode)

Real 6581 chips were electrically imperfect: their filter stages introduced non-linear distortion that many composers relied on for texture. The 6581 mode in PokeyMAX models this behaviour. Disabling it gives a cleaner emulation that may sound more like 8580 even when the type is set to 6581.

Digifix

Early C64 demo coders exploited DAC offsets and analogue biases in real 8580 hardware to play digitised samples. Digifix compensates for the absence of these imperfections in a digital implementation by injecting a small bias into the fake ADC input, improving the fidelity of those sample-playback tricks. It is only relevant in 8580 mode — leave at 0 when using 6581.

SID clock

SID is clocked at approximately 1 MHz, derived from PHI2: 5/9 PHI2 in PAL mode, 4/7 PHI2 in NTSC mode. This matches original C64 timing closely enough for correct pitch playback.

1.5 PSG (AY-3-8910 / YM2149)

PSG stands for Programmable Sound Generator, specifically the General Instrument AY-3-8910 and its Yamaha-compatible relative the YM2149. These are three-voice square-wave chips with a shared noise source and hardware envelope generator, used in ZX Spectrum, MSX, Amstrad CPC, Atari ST, and many arcade boards.

PokeyMAX implements two PSG cores at \$D2A0 and \$D2B0. Because registers are directly memory-mapped, writes to \$D2A0–\$D2AF control PSG1 directly without the indirect register-address bus used on the original chips.

Frequency / clock

The PSG clock can be set to 1 MHz, 2 MHz, or PHI2 directly. The correct choice depends on the playback software: most Spectrum and MSX ports assume 1.77 MHz and work best with the 2 MHz setting as the closest available option. Some Atari ST ports are tuned for 2 MHz exactly.

Stereo modes

The AY/YM chips have three tone channels (A, B, C). Different national and regional conventions for which channel goes to which speaker became established over time:

Mode	Channel routing
Mono	All channels (A, B, C) on both left and right
Polish	A+B → Left, B+C → Right (default)
Czech	A+C → Left, B+C → Right
Max / L-R	PSG1 entirely to Left, PSG2 entirely to Right

The Polish mode is the most widely supported by Atari 8-bit AY player software. If a track sounds panned or unbalanced, try a different mode.

Envelope steps

The AY-3-8910 uses a 16-step envelope generator; the YM2149 uses 32 steps for smoother volume sweeps. The default of 32 steps matches YM2149 behaviour and sounds smoother. Switch to 16 steps only when targeting strict AY-3-8910 compatibility for software that relies on the coarser envelope shape.

Volume curve

The AY-3-8910 uses a logarithmic volume table; the YM2149 is slightly different. PokeyMAX offers a log curve matching the AY, or a linear curve for software that assumes linearity. Most material sounds better with the log curve.

1.6 COVOX and the Sample Player

COVOX

COVOX is a simple resistor-ladder DAC concept that maps a byte value directly to an analogue voltage level, allowing software to play digital audio by rapidly writing sample values to a register. On PokeyMAX the COVOX registers (\$D280–\$D283) provide four 8-bit volume channels for CPU-driven digital audio. The CPU writes sample bytes in a tight loop at the desired playback rate.

DMA Sample Player

The DMA sample player extends the COVOX concept dramatically. It stores up to 42 KiB of audio data in dedicated block RAM inside the FPGA and plays it back without CPU intervention once started. The CPU only needs to set up address, length, period, and volume for each of the four channels, then respond to the end-of-sample IRQ to queue the next buffer.

Three data formats are supported:

- 4-bit signed PCM — same storage footprint as ADPCM but uncompressed. Easier to seek and restart mid-sample since there is no encoder state to track.
- 8-bit signed PCM — standard quality, double the storage of 4-bit
- 4-bit IMA ADPCM — compressed with approximately 13-bit perceptual quality. Requires encoding in advance: `sox input.wav -t ima-adpcm output.ima`. Harder to seek mid-stream but gives the best quality-per-byte.

Sample rates up to 48 kHz are supported, though the available 42 KiB of storage limits duration at high rates. For context, 8-bit mono at 22 kHz gives roughly 1.9 seconds per fill of RAM.

NOTE

The sample player is heavily inspired by the Amiga's Paula chip, which pioneered DMA-driven four-channel sample playback in home computers. You may see Paula mentioned in related discussions — it refers to this conceptual lineage, not a literal compatibility mode.

1.7 Output path and level control

All audio engines feed into a common mixer. The mixed signal then passes through the post-divider, a digital low-pass filter, the Sigma-Delta DAC, and finally an analogue low-pass filter. Understanding this chain explains why level and routing settings exist.

The mixer produces four output channels. All four sum the same set of chip audio engines (POKEY, SID, PSG, SAMPLE), but AUDIO_0/1 and AUDIO_2/3 differ in what else they include:

- AUDIO_0 (Left) → DIP pin 37: feeds to internal audio outputs. No GTIA or ADC mixed in — the Atari motherboard already handles those on the internal audio path.
- AUDIO_1 (Right) → Not present on v4.5
- AUDIO_2 (Left) → J2 line out + S/PDIF source: chip audio plus GTIA and SIO/ADC audio mixed in. These are needed on the external path because the motherboard mixing is bypassed.
- AUDIO_3 (Right) → J2 line out + S/PDIF source: same as AUDIO_2 but for the right channel.

Post-divide

PokeyMAX natively outputs 0–5 V per channel. A single audio engine at maximum volume produces 5 V; with all four POKEY cores plus SID and PSG playing simultaneously the internal sum can reach much higher before clipping at 5 V. The post-divider lets you divide each physical output channel (CH0–CH3) by 1, 2, 4, or 8.

For line-level output (~1 V peak) from a single device, /4 is a reasonable starting point and is the default for the external outputs (CH3/CH4). The internal pin-37 output (CH1) defaults to /1 since the motherboard amplifier stage handles level. If you have multiple engines playing simultaneously, /8 may be needed to prevent distortion. The install guide has more on connecting audio outputs to specific hardware.

GTIA mixing

The Atari motherboard normally mixes GTIA's switch output into the audio amplifier alongside POKEY. Most notably this provides the keyboard click buzzer, but also software can toggle the GTIA audio line directly. Some clever programmers even exploited this path to play back digitised samples by rapidly toggling the output. When using PokeyMAX's external outputs (line-level or S/PDIF), this motherboard mixing is bypassed, so GTIA audio would otherwise be absent. PokeyMAX can inject the GTIA signal digitally into any of the four output channels to preserve these sounds.

GTIA mixing is configurable independently per channel. Channel 1 is pin 37 (the internal path) and defaults to off, since the motherboard already handles GTIA there. Channels 3 and 4 are the external left and right line outputs and default to on. Channel 2 is not populated on v4.5 hardware. The GTIA mixing setting in `pokeycfg` shows all four channels; set channel 1 to 1 only if you want GTIA injected on the internal path as well (unusual).

Mono detection

When enabled, PokeyMAX monitors the right output channel and, if it is silent, automatically mirrors the left channel to both outputs. This prevents silent right channels when playing back mono software through stereo equipment.

Part 2 — Using pokeycfg

2.1 Overview

pokeycfg (pokeycfg.xex) is the official configuration utility for PokeyMAX. It runs on the Atari itself — no PC, no JTAG connection required for normal use. The program reads the current live register values and displays them in a text-mode interface. You can change settings and either apply them immediately to hear the effect, or save them to flash so they take effect automatically from the next power-on.

pm_update.xex is an unofficial tool developed by MADRAFi. That is not described here. It has a very nice menu based interface.

IMPORTANT

pokeycfg will only work if the A4–A7 address lines are connected. Without them the device cannot distinguish which register bank is being accessed, and the tool will report that PokeyMAX was not found.

2.2 Starting the tool

Load pokeycfg.xex from disk or cassette as you would any Atari program. On launch it detects the PokeyMAX by reading the ID register at \$D20C. If detection succeeds it displays the firmware version string and capability summary, then shows the first configuration section (Core).

```
PokeyMAX config v1.5   Core: v4.5Q
Sid:2 pokey:4 psg:2 covox:1 sample:1

▶ Mono support   : Left only
Post divide     : 1=1 2=1 3=4 4=4
GTIA mixing     : 1=0 2=0 3=1 4=1
Restrict        : quad sid psg covox
Output          : 1=1 2=1 3=1 4=1 5=1
PHI2->1MHz      : PAL (5/9)
ADC volume      : 1x
SIO DATA vol   : 1x

Cursor keys + Enter to change   SPACE to change section
(A)pply config (S)toze config (U)pdate core (Q)uit
```

Figure 2: pokeycfg — Core section (reconstructed screen)

2.3 Navigation

The Atari keyboard layout means that arrow-key functions are mapped as follows:

The Atari cursor keys have arrow symbols printed on them. Their key-code mappings in pokeycfg are:

Key	Action
↑ Up cursor (prints -)	Move selection up
↓ Down cursor (prints =)	Move selection down
← Left cursor (prints +)	Change selected value left / decrement
→ Right cursor (prints *)	Change selected value right / increment
SPACE	Cycle to the next configuration section
RETURN	Confirm / toggle the highlighted item
A	Apply current settings to live registers immediately
S	Store (save) current settings to flash
U	Update FPGA core from core.bin on D1-4
V	Verify FPGA core against core.bin on D1-4
Q	Quit pokeycfg and return to DOS

There are four configuration sections, visited in order by pressing SPACE:

- Core — system-wide settings: level, GTIA, restrictions, clock
- POKEY — saturation curve, channel mode, IRQ routing
- PSG — clock frequency, stereo mode, envelope, volume curve
- SID — chip model, filter type, digifix

TIP

Press SPACE repeatedly to cycle through all four sections. The section name is visible at the top of the screen. There are no direct jump keys.

2.4 Apply vs Save — the most important distinction

This is the single most important concept in pokeycfg. The two actions have completely different effects:

Key	Effect	Persists after reboot?
A (Apply)	Writes current settings directly to PokeyMAX live registers. You hear the effect immediately, without any reboot.	NO. Lost on power-off.
S (Save)	Writes settings to flash memory inside the FPGA. Does NOT change the currently running configuration. Takes effect from the next power-on.	YES. Loaded automatically at boot.

The recommended workflow when tuning a setting is:

- Navigate to the setting and change it with the left/right cursor keys.
- Press A to apply. The change takes effect immediately — listen to the result.
- Adjust further if needed, applying each time.
- Once satisfied, press S to store. This becomes the new power-on default.

2.5 Core section settings

The Core section contains system-wide settings that affect all sound engines and output channels.

Setting	What it does
Mono support	When set to 'Play on both channels', if the right output is silent the left channel signal is copied to it automatically. Useful when playing mono software through stereo speakers. Default: Left only.
Post divide (CH1–4)	Divides each physical output channel's level by 1, 2, 4, or 8. Default is CH1/CH2 = ÷1 (internal path at full level) and CH3/CH4 = ÷4 (external line outputs attenuated to approximately line level). Increase division further if output is distorting with multiple engines active.
GTIA mixing (CH1–4)	Enables digital injection of the GTIA audio signal per output channel. CH1 = pin 37 (internal), defaults off — the motherboard already mixes GTIA here. CH2 = not populated on v4.5. CH3/CH4 = external left/right line outputs, default on — enable these to preserve GTIA sounds on the line-level and S/PDIF outputs.
Restrict	Enables or disables individual sound engines. Disabled engines have their address range shadowed by POKEY 1, maintaining compatibility with software that scans the bus. Options: mono/dual/quad POKEY, SID on/off, PSG on/off, COVOX on/off.
Output (CH1–5)	Five individual on/off toggles, one per output channel, plus the S/PDIF output. Set a channel to 0 to disable it entirely. Disabling S/PDIF (channel 5) when not in use reduces digital noise on the ADC input. All channels are enabled by default.
PHI2→1MHz	Sets the clock divider for the 1 MHz derived clock used by SID and PSG. Choose PAL (5/9 PHI2) for PAL machines, NTSC (4/7 PHI2) for NTSC. Incorrect setting causes pitch errors.
ADC volume	Amplification factor applied to the digitised SIO/PBI audio input (0x = silent, 1x–4x). Set to 0 if you have not connected the SIO audio pin.
SIO DATA vol	Mixes SIO serial data interference into the audio digitally, replicating the analogue bleed present on real hardware. Set to 0 for clean audio, or a low value for authentic feel.

2.6 POKEY section settings

```
PokeyMAX config v1.5   Core: v4.5Q   [ POKEY ]
▶ Mixing               : Non-linear
Channel mode          : Off
IRQ                   : Pokey 1
```

Figure 3: POKEY section

Setting	What it does
Mixing (Saturation)	Non-linear: applies the POKEY hardware saturation curve, compressing loud signals and giving the characteristic warm Atari sound. Linear: mathematically sums channels without compression. Non-linear is correct for the vast majority of Atari software.
Channel mode	Off: each POKEY chip's four channels are mixed to a single output (normal). On: each individual voice is routed to a separate audio output pin. On v4.5 there is only one physical output pin in this path, so only three channels are available rather than four. Used for multi-channel recording rigs, not normal listening.
IRQ	Pokey 1: only POKEY 1 generates IRQ signals to the CPU (standard behaviour). All: enables IRQ from all POKEY chips. Only relevant for software that uses multiple POKEYs for timer/serial functions, not just audio.

2.7 PSG section settings

```
PokeyMAX config v1.5   Core: v4.5Q   [ PSG ]
▶ PSG frequency       : 2MHz
PSG stereo            : polish (L:AB R:BC)
PSG envelope          : 32 steps
PSG volume            : Log 0
```

Figure 4: PSG section

Setting	What it does
PSG frequency	Clock speed for both PSG chips. 2MHz suits most Spectrum and MSX ports. 1MHz matches some original AY hardware clocking. PHI2 feeds the bus clock directly. Incorrect clocking causes pitch errors.

PSG stereo	Channel routing across left and right outputs. Polish (A+B / B+C) is the Atari 8-bit community default. Czech (A+C / B+C) matches some other platforms. L/R sends the two PSG chips to separate channels entirely.
PSG envelope	32-step envelopes give smoother volume sweeps. 16-step matches some original hardware behaviour more closely. Most modern players assume 32 steps.
PSG volume	Log 0: AY-3-8910 logarithmic volume table (recommended). Linear: equal-step volume increments. Log curves closer to the original; linear can sound harsher at high volumes.

2.8 SID section settings

```
PokeyMAX config v1.5   Core: v4.5Q   [ SID ]
▶ SID detect          : Auto
SID version           : 1:6581      2:6581
SID ext                : 1:Digifix   2:Digifix
```

Figure 5: SID section

Setting	What it does
SID detect	Auto: PokeyMAX detects whether SID-capable software is running by monitoring bus activity, and enables the SID address range only when needed. Manual: SID addresses are always (or never) active. Auto is recommended for most users.
SID version (1 and 2)	Sets the emulated SID model for each chip independently. 6581: older, rougher, warmer, with non-linear filter character. 8580: later, cleaner, more stable. Match the model to what the music was written for.
SID ext	Controls what is fed into the SID chip's external audio input pin. GND: the input is held at ground (silent, cleanest signal). Digifix: a small bias signal is injected to compensate for the absence of the analogue ADC imperfections that real 8580 hardware relied on for digi-sample playback tricks. Only relevant in 8580 mode — leave at GND when using 6581.

2.9 Updating the FPGA core

pokeycfg can update the FPGA bitstream directly from the Atari. Place a file called core.bin in D1-D4 then:

- Press U to start a core update. Confirm with Y when prompted.

- The process erases flash, writes the new core, then verifies it. Do not turn off the computer during this operation.
- Press V to verify an existing core against core.bin in D1-D4 without overwriting.
- If the core version string does not match, the tool warns and asks for confirmation. You can force with F if you are certain the file is correct for your hardware — a mismatched core leaves the device unresponsive until re-flashed via JTAG.

IMPORTANT	The core.bin file must match your specific board variant. Always download the correct core for your hardware from www.64kib.com .
------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.10 Companion documents

This guide covers what the features mean and how to configure them. Two other documents complete the set:

Document	Content
Pokeymax Installation Guide	Physical fitting, pin-1 orientation, address-line wiring, audio output connections (analogue and S/PDIF), GTIA and SIO audio wiring, PS/2 keyboard support.
PokeyMAX Developers Guide	Full register map, bit-level descriptions of all configuration and extended registers, memory map variants, SID/PSG/SAMPLE register details. Essential for software authors.

Both are available at www.64kib.com alongside pokeycfg.xex, core.bin files, and example projects.

Quick Reference: Recommended Starting Points

Standard Atari use

- POKEY mixing: Non-linear
- Mono support: Left only (unless using stereo speakers and mono software)
- PHI2→1MHz: match your machine (PAL or NTSC)
- Post divide: defaults are CH1/CH2 = ÷1, CH3/CH4 = ÷4. Increase if output clips with multiple engines active
- ADC volume: 0x if SIO audio not connected
- Save after testing — do not save experimental settings before testing them with Apply

SID playback

- Try 6581 first for C64 music from the 1980s
- Try 8580 for later productions or if a tune sounds over-distorted on 6581
- Enable Digifix on 8580 if digi-samples sound wrong
- Use Auto SID detect unless you need it always-on

PSG / AY playback

- Frequency: 2MHz for most Spectrum and MSX software
- Stereo: Polish for Atari 8-bit AY players; try Czech or L/R if tracks sound unbalanced
- Volume: Log 0 (default AY curve)
- Envelope: 32 steps (default; switch to 16 only if a specific title sounds wrong)

Sample playback

- Keep post-divide levels conservative — samples can be louder than chip audio
- Test with Apply before saving
- Use IMA ADPCM format for longer samples within the 42 KiB block RAM limit

Appendix: Memory Map (v4.5 Full)

The full memory map for the v4.5 build is shown below. Configuration registers are accessed by writing 0x3F to \$D20C to bank them into the \$D210–\$D21F range.

Address range	When config mapped	Function
\$D200–\$D20F	—	POKEY 1
\$D210–\$D21F	Config registers	POKEY 2 / Configuration bank
\$D220–\$D22F	—	POKEY 3
\$D230–\$D23F	—	POKEY 4
\$D240–\$D25F	—	SID 1
\$D260–\$D27F	—	SID 2
\$D280–\$D283	—	COVOX (4 × 8-bit volume)
\$D284–\$D29F	—	DMA Sample player
\$D2A0–\$D2AF	—	PSG 1
\$D2B0–\$D2BF	—	PSG 2
\$D2C0–\$D2FF	—	Reserved

Access the configuration registers by writing 0x3F to \$D20C (the SKCTL register in normal operation). Once banked in, \$D210 becomes MODE, \$D211 CAPABILITY, \$D212 POSTDIVIDE, etc. Writing any value other than 0x3F to \$D20C restores \$D210–\$D21F to POKEY 2.

Further information and downloads: www.64kib.com