

This is the foundational code for the “**Still Zero**” protocol. By treating the `.still` file as the atom-thin “bootloader,” we create a hardware-software handshake that bypasses the modern “OS” distractions entirely.

still-zero-bootloader.md

The “Still Zero” Protocol: The Bootloader for Reality

Objective: To define the `.still` format as the lowest-level visual state of the Penbook. This is the “Escape Hatch.” If the system lags, if the UI clutters, if the magic fades—you return to **Still Zero**.

1. *The .still Format Definition (Technical)*

A `.still` is not a “document.” It is a **Direct Pixel Map (DPM)**.

Layer Count: Exactly 1. (No UI layer, no notification layer, no cursor layer).

Color Depth: 1-bit (Native). While the hardware supports 16-level grayscale, the `.still` protocol prefers 1-bit to eliminate “shimmer” and maximize contrast.

Compression: None. Every pixel is a hard-coded coordinate.

Refresh Command: FORCED_REGAL_CLEAN. Once the pixels are set, the screen enters a “Physically Persistent” state where the power to the display controller is cut.

2. *Still Zero: The Home Still*

The “Still Zero” file is the default state of the device. It is a “**Home Screen**” that **isn’t a screen**.

Visual Anatomy of “Still Zero”:

1. **The Void:** 90% of the screen is empty (The White Paper).
 2. **The Sigil:** A small, 300x300 pixel “Seal” in the center-bottom representing the device owner.
 3. **The Library Anchors:** Four tiny, hand-drawn icons (Codex, Scroll, Stone, Vault) in the corners—only visible if you look closely.
 4. **The Hatch:** A single strike-through line in the “Scholar’s Column” (The Escape Hatch) that, when traced with a Wand, resets the entire system back to this state.
-

3. *The Bootloader Logic (Mermaid)*

This is how we teach an AI or a developer to handle a `.still` transition.

Code snippet

```

``` graph TD
 A[Current Activity: Reading/Writing] --> B{Trigger: Still Zero}
 B -- Physical Gesture: The Hatch --> C[System Purge]
 C --> D[Load Still_Zero.still]
 D --> E[Execute Deep Regal Refresh]
 E --> F[Power-Down Controller]
 F --> G[State: SET IN STONE]

```

```

```

```

4. *Why it Matters: The “Escape Hatch”*

In a world of “smart” devices that get bogged down by updates and background processes, **Still Zero** is the hardware guarantee of peace.

The Reset: If the Penbook ever feels like a “tablet” (laggy, busy, noisy), the user performs the “Hatch” gesture.

The Result: The device doesn’t “reboot” with a logo and a loading bar. It **snaps** instantly to the Still Zero pixel map. It is the digital equivalent of clearing a physical desk.

5. *Instruction for the “Kiln” (AI/Dev)*

To generate a `.still` file from a `.render` (Vector Master):

1. **Discard** all metadata except `Hearth_UID`.
2. **Discard** all color data.
3. **Rasterize** to 1404 x 1872 at 1-bit depth.
4. **Wrap** in a raw binary header that identifies it as a `.still`.

5. Set the hardware flag to “Persistent.”

The Zen Manifesto of .still:

“A .still file does not ask for your attention. It does not update. It does not move. It is the only thing in your digital life that is truly, beautifully, and purely finished.”

I have successfully documented the “Still Zero” bootloader. Would you like me to “Fire” a mockup of what the `still_zero.still` visual looks like—the actual minimalist layout of the escape hatch—so you can see it on your Slate?