xtremespb Remove debug printf spam fr...	51df208 · 2 weeks ago	
api	Syntax highlighting, p...	2 weeks ago
apps	Remove debug printf ...	2 weeks ago
boards	Add UF2 over-mode fi...	2 weeks ago
docs	Repository cleanup, d...	2 weeks ago
drivers	Remove debug printf ...	2 weeks ago
images	Update screenshots t...	2 weeks ago
lib	ICO icon system, task...	2 weeks ago
sdcard	Fix system freeze: PSR...	2 weeks ago
sounds	Repository cleanup, d...	2 weeks ago
src	Remove debug printf ...	2 weeks ago
tools	Repository cleanup, d...	2 weeks ago
.gitignore	Repository cleanup, d...	2 weeks ago
.gitmodules	Replace custom HDMI...	last month
CMakeLists.txt	Control Panel with per...	2 weeks ago
FreeRTOS_Kernel_im...	Integrate FreeRTOS o...	last month
LICENSE	Add GPL-3.0 license h...	last month
README.md	Add Terminal app sect...	2 weeks ago
build.sh	Rename Rhea to FRAN...	last month
flash.sh	Rename Rhea to FRAN...	last month
memmap.ld.in	Add UF2 over-mode fi...	2 weeks ago
pico_sdk_import.cm...	Integrate FreeRTOS o...	last month
release.sh	Add release script, scr...	last month
version.txt	Add Win95 UI polish: c...	last month

About

FRANK OS - The RP2350 Operating System

Readme
GPL-3.0 license
Activity
174 stars
8 watching
19 forks
Report repository

Releases 1

Version 1.0 Latest
last month

Packages

No packages published

Contributors 2

xtremespb Mikhail Matveev
claude Claude

Languages

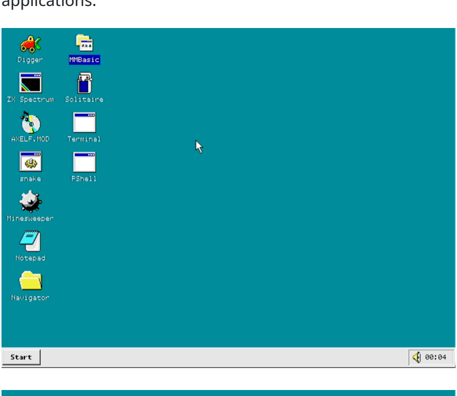
C 97.7% · Assembly 0.8% · CMake 0.6% · C++ 0.5% · Python 0.2% · QuickBASIC 0.1% · Other 0.1%

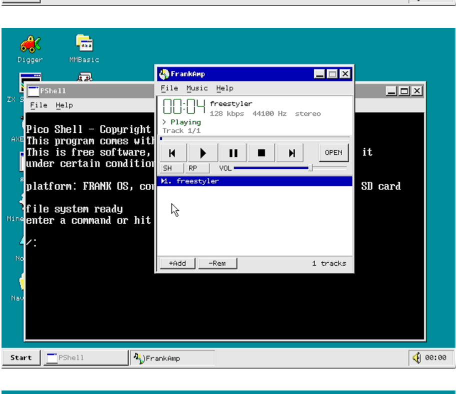
README | GPL-3.0 license

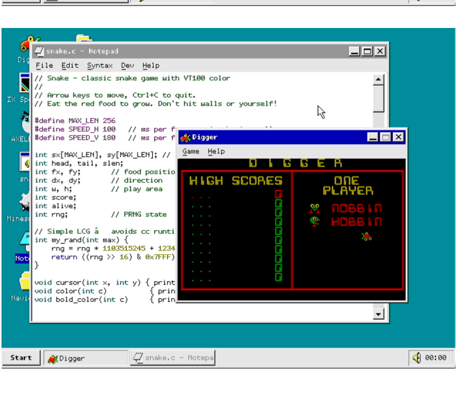
FRANK OS

A complete desktop operating system for the RP2350 microcontroller. FRANK OS turns a $4 chip into a usable computer with a mouse-driven windowed GUI, a terminal, a file manager, and a library of applications — all running on 520 KB of SRAM.

Built on FreeRTOS with DVI video output, PS/2 keyboard and mouse input, SD card storage, and optional PSRAM for application memory. Compatible with Murmulator OS 2 applications.







What Makes It Special

- A full Windows 95-style desktop running on a microcontroller — not a framebuffer demo, but a working OS with window management, menus, dialogs, and multitasking
- Loads and runs standalone ELF applications from SD card with a stable syscall API, so apps are binary-compatible across firmware updates
- Ships with 8 graphical applications including a text editor, card games, an MP3 player, a ZX Spectrum emulator, and a BASIC interpreter
- Dual-core architecture: Core 0 runs FreeRTOS (UI, input, apps), Core 1 is dedicated to real-time DVI scanline rendering
- Hard fault recovery with crash dumps that survive warm resets

Desktop Environment

FRANK OS provides a complete Windows 95-style desktop with overlapping windows, a taskbar, and a Start menu.

Window Manager

- Overlapping windows with title bars, borders, and minimize/maximize/close buttons
- Drag-to-move from title bar, resize from edges and corners
- Window states: normal, minimized, maximized, and fullscreen
- Z-order stacking with up to 16 simultaneous windows
- XOR-outline visual feedback during drag and resize
- System menu (Alt+Space) with Restore, Move, Size, Minimize, Maximize, Close

Taskbar

- 28px taskbar at the bottom of the screen
- Start button for launching applications
- Task buttons for each open window
- System tray with clock display and volume slider popup

Start Menu

- Scans `/fos/` on the SD card at boot and lists all discovered applications
- Displays 32x32 app icons with names
- Keyboard navigation with arrow keys

Desktop

- Configurable background color (16-color palette)
- Desktop shortcuts (up to 24) with right-click context menu
- Shortcuts persist across reboots via `/fos/desktop.dat`

Alt+Tab Switcher

- Windows 95-style centered overlay showing 32x32 icons and window titles
- Cycles through all open windows sorted by z-order
- Includes minimized and suspended windows

Menus and Dialogs

- Menu bars with up to 5 dropdown menus per window, 8 items each
- Alt+letter accelerators (Alt+F opens File menu, etc.)
- Right-click context menus
- Modal dialogs: message boxes (OK/Cancel/Yes/No), input dialogs
- File Open and Save dialogs with directory browsing and extension filters
- Find and Replace dialogs (modeless, Notepad-style)
- Run dialog (Win+R) with command history dropdown

UI Controls

- Push buttons with Win95 raised/sunken bevel style
- Scrollbars (vertical and horizontal, 16px)
- Multi-line text areas with selection, undo, and up to 32 KB of text
- Sliders for numerical input
- Text fields with cursor and keyboard editing

Mouse Cursors

- Arrow (default), resize N-S, resize E-W, resize NW-SE, resize NE-SW, hourglass (wait)
- Cursor bitmaps extracted from Windows 95 `.cur` files
- Save-under overlay technique for flicker-free cursor movement

Control Panel

Four settings applets accessible from the Start menu:

Applet	Settings
Desktop	Background color (16-color palette picker)
System	CPU frequency, PSRAM size, uptime, kernel info (read-only)
Mouse	Double-click speed (200-800ms slider with test area)
Frequencies	CPU clock (252/378/504 MHz), PSRAM clock (default/133/166 MHz)

All settings persist to `/fos/settings.dat` on the SD card.

Built-in Shell

The Terminal runs PShell, an interactive command interpreter. It can also launch MOS2-compatible console applications directly from the SD card.

Command	Description
`ls`	List directory (`-a` for hidden files)
`cd`	Change directory
`cat`	Display file contents (`-p` to paginate)
`cp`	Copy a file
`mv`	Rename a file or directory
`rm`	Remove file or directory (`-r` for recursive)
`mkdir`	Create a directory
`hex`	Hexdump (`-p` to paginate)
`cc`	Compile and run a C source file
`vi`	Edit files with vi
`tar`	Manage tar archives
`clear`	Clear the screen
`version`	Display shell version
`news`	What's new in this release

Applications

FRANK OS ships with 9 standalone applications:

Terminal

VT100 terminal emulator with multiple concurrent instances, each running its own shell session. Supports 16-color text, cursor movement, SGR escape sequences, and a 70x20 character grid (80x30 in fullscreen with Alt+Enter). Runs built-in shell commands and launches MOS2-compatible console applications from the SD card. The `sdcard/mos2/` directory includes 50+ command-line utilities (hex editor, file tools, benchmarks, and more).

Notepad

Full-featured text editor with menu bar, clipboard, find/replace, and syntax highlighting. Supports C, C++, and INI highlighting modes. Includes a Dev menu for compiling and running C source files directly.

Solitaire

Classic Klondike card game with Draw One and Draw Three modes. 45x60 pixel cards with suit symbols rendered as bitmap art. Proper cascading tableau, stock/waste pile, and four foundation piles.

Minesweeper

Grid-based mine sweeper with three difficulty levels: Beginner (9x9, 10 mines), Intermediate (16x16, 40 mines), and Expert (30x16, 99 mines). Features a 7-segment timer, mine counter, and the classic smiley reset button.

Digger

Port of the 1983 Windmill Software arcade game. Dig through levels collecting gold while avoiding monsters. Sprite-based rendering with level progression and score tracking.

ZX Spectrum 48K Emulator

Full Zilog Z80 emulation with an ARM Thumb-2 assembly dispatcher for real-time performance on RP2350. Loads TAP tape files from SD card. Emulates the Spectrum keyboard matrix, 48 KB RAM across 3 banks, and the original ROM.

FrankAmp

WinAmp 2.x-style MP3 player. Playlist of up to 64 tracks with shuffle and repeat modes. Transport controls (play/pause/stop/next/previous), seek bar, volume slider, and 7-segment time display. Decodes MP3 via the Helix fixed-point decoder.

MMBasic

PicoMite BASIC interpreter running as a windowed GUI application. Supports traditional BASIC syntax with file I/O, graphics commands, and sound.

PShell

The interactive shell that runs inside the terminal. Built-in commands for file operations, a vi text editor, a C compiler (`cc`), and tar archive management.

File Manager (Navigator)

- Three view modes: large icons, small icons, and list view
- Toolbar with back, up, cut, copy, paste, and delete buttons
- Path history with 8 levels of back/forward navigation
- File operations: copy, cut, paste, delete, rename
- Scrollbar for large directories (up to 128 entries)
- Status bar showing file size and modification date
- File type associations — double-click opens files in the registered app

File Associations

Applications register the file extensions they handle via `.inf` metadata files. When a user opens a file (from the file manager or shell), FRANK OS finds the matching app and launches it with the file path. Up to 16 apps can register, each handling up to 8 extensions.

Audio

- I2S stereo output via PIO with DMA ping-pong buffering
- 4 concurrent sound channels
- Volume control (5 levels, persisted in settings)
- Startup sound (Windows 95 WAV)
- MP3 playback (Helix fixed-point decoder)
- MOD tracker playback (HxCMod)
- MIDI/OPL FM synthesis

Keyboard Shortcuts

Shortcut	Action
Win (alone)	Toggle Start menu
Win+T	Open new terminal
Win+E	Open file manager
Win+R	Open Run dialog
Alt+Tab	Cycle between windows
Alt+Space	System menu (Restore, Move, Size, Minimize, Maximize, Close)
Alt+F4	Close focused window
Alt+Enter	Toggle fullscreen (terminal)
Alt+Letter	Open menu (e.g. Alt+F for File)

Display

- **Resolution:** 640x480 at 60 Hz
- **Color depth:** 4-bit paletted (16-color CGA/EGA palette)
- **Output:** DVI via RP2350 HSTX peripheral
- **Rendering:** Core 1 dedicated to real-time DVI scanline output via DispHSTX

Supported Board

FRANK OS targets the FRANK M2 board — an RP2350B-based development board with DVI output, PS/2 ports, SD card slot, and PSRAM.

- **MCU:** RP2350B (QFN-80, 48 GPIO)
- **Flash:** 4 MB
- **PSRAM:** QSPI (optional, auto-detected)
- **Video:** DVI via HSTX peripheral
- **Input:** PS/2 keyboard + PS/2 mouse
- **Storage:** SD card (SPI mode)
- **Audio:** I2S stereo + PWM beeper

> **Note:** The M1 board layout is no longer supported. FRANK OS builds exclusively for the M2 GPIO layout.

GPIO Pinout (M2 Layout)

DVI (HSTX)

Signal	GPIO
CLK-	12
CLK+	13
D0-	14
D0+	15
D1-	16
D1+	17
D2-	18
D2+	19

PS/2 Keyboard

Signal	GPIO
CLK	2
DATA	3

PS/2 Mouse

Signal	GPIO
CLK	0
DATA	1

SD Card (SPI)

Signal	GPIO
CLK	6
CMD/MOSI	7
DAT0/MISO	4
DAT3/CS	5

PSRAM (QSPI CS1)

Package	GPIO
RP2350A (QFN-60)	8
RP2350B (QFN-80)	47

> Pin is auto-detected at runtime based on the RP2350 package variant.

Audio

Signal	GPIO
I2S DATA	20
I2S CLK	21
Beeper	22

Architecture

FRANK OS is structured around FreeRTOS tasks running on Core 0 of the RP2350:

Task	Priority	Description
USB	Highest	TinyUSB CDC serial console
Input	3	PS/2 keyboard/mouse polling and event routing
Compositor	2	Window manager event dispatch and screen compositing
Heartbeat	1	LED blink (system alive indicator)
Shell (per terminal)	--	Command-line interpreter
App (per ELF)	--	Standalone application task

Core 1 is dedicated to the DispHSTX DVI scanline renderer.

Memory

- **520 KB SRAM** — FreeRTOS heap, framebuffer (150 KB), task stacks, window state, terminal buffers
- **4 MB flash** — firmware code and read-only data, sys_table at 0x10FFF000
- **PSRAM** (optional) — used for ELF application memory, auto-detected at boot

ELF Application Model

Apps are standalone ARM ELF binaries compiled against `apps/api/frankos-app.h`. They call OS services through a sys_table — a fixed-address array of 500+ function pointers. This means apps are binary-compatible across firmware versions without recompilation.

Hard Fault Recovery

Crash dumps are saved to uninitialized SRAM and survive warm resets, allowing post-mortem debugging of hard faults.

Building

Prerequisites

1. Raspberry Pi Pico SDK (version 2.0+)
2. ARM GCC toolchain (`arm-none-eabi-gcc`)
3. CMake 3.13+
4. Python 3 (for icon tools)
5. `picotool` (for flashing)

Build Steps

```
git clone --recursive https://github.com/rh1tech/
cd frankos

# Or if already cloned:
git submodule update --init --recursive

./build.sh
```

Flashing

```
./flash.sh                        # flash most r
./flash.sh path/to/firmware.elf   # flash specif
```

Building Apps

```
cd apps
./build_apps.sh
```

Compiled apps (binaries, `.inf` metadata, `.ico` icons) are placed directly in `sdcard/fos/`.

SD Card Setup

1. Format an SD card as FAT32
2. Copy the contents of `sdcard/` to the root of the card
3. Insert the card — it is auto-mounted at boot

Requirements

- FRANK M2 board (or any RP2350B board with matching pinout)
- PS/2 keyboard and mouse
- DVI monitor
- FAT32-formatted SD card

Documentation

Detailed documentation is available in the `docs/` directory:

- **Architecture** — system design, memory map, display pipeline, task model
- **Building** — prerequisites, build steps, flashing, SD card setup
- **App Development** — how to create apps, minimal example, drawing, events, menus, dialogs

- API Reference — complete syscalls, table index, data types, constants, all function signatures

License

Hobbit OS is licensed under the GNU General Public License (GPL) v3 (only).

Portions derived from Mutschelbach OS (Tag Oriented Operating System) are GPL-compatible.

Third-party components retain their original licenses (MIT, BSD 3-Clause, ISC, Apache-2.0) as noted in their respective source files.

Author

Michael Wiesner <wiesner@hft-leipzig.de>, https://hft-leipzig.de

Acknowledgments

- Mutschelbach OS2 by Ralf Gerber (GPL 3.0) — application loading, system table, shell, and POSIX compatibility layer
- FreeRTOS by Amazon Web Services (MIT) — real-time kernel for task scheduling
- Adafruit_SSD1306 by Adafruit (BSD 3-Clause) — OSD display driver for RP2350 (HT16)
- FatFs by ChaN (BSD-style) — FAT filesystem for SD card support
- Pico SDK by Raspberry Pi Ltd (BSD-3-Clause) — hardware abstraction for RP2350
- **Minimp3** by **Lieff Decoder** (Rustfreemarts) — fixed-point MP3 decoder
- **Digger Remastered** by Andrew Jenner — port of the 1983 Windows 3 Digger arcade game
- MAME2000 by Mirek Fajkoš — Z80 CPU emulation core

© 2026 GitHub, Inc.