Fuzix for the Pico Computer

Unix and BBC BASIC on the Pico Computer 2 and 3

July 2026 — kernel branch pc3

Contents

1 Introduction

Fuzix is Alan Cox's compact V7-style Unix. This port makes it a third first-class environment for the Pico Computer, beside MMBasic and MicroPython: a genuine multi-tasking Unix booting from SD card, with a full set of classic utilities — and R. T. Russell's BBC BASIC as its flagship application, complete with BBC graphics modes on the HDMI display, the four-channel BBC sound system on the audio DAC, joystick and analogue inputs, and a spare serial port.

One kernel image serves both machines. At boot it probes GP27 for the DS3231's 32 kHz clock (the same test MMBasic and MicroPython use): present means Pico Computer 3, absent means Pico Computer 2. The practical difference is the SD card wiring, handled automatically; the boot banner names the detected board.

Headline specification as configured here:

- CPU at 375 MHz (the XGA-capable clock, as MMBasic uses)
- 320 KB of program RAM managed as swap-backed 4 KB pages, with the 8 MB PSRAM as the swap device — up to 30 concurrent processes, each up to 256 KB
- Root filesystem on SD card; the on-board NAND flash holds a recovery system
- 80×40 colour ANSI console on HDMI, mirrored to the USB-C serial port, with USB keyboard support (six layouts)
- Pre-emptive multitasking: a runaway program can always be stopped from the keyboard

2 Installing Fuzix

2.1 Flashing the kernel

The kernel is a standard UF2 file, `fuzix.uf2`, flashed exactly like any other Pico Computer firmware:

1. Set the USB HUB switch to **DISABLE** and connect the **Prog** port to your PC.
2. Hold **BOOT**, click **RESET**, release BOOT. A USB drive appears.
3. Drag `fuzix.uf2` onto the drive. The board reboots when the copy completes.
4. Set the HUB switch back to **ENABLE**.

Flashing does not touch the SD card.

2.2 Writing the SD card

Fuzix boots from a ready-made card image, `pc3-sd.img` (about 101 MB). Write it to a card of 128 MB or larger with any raw-image tool — Raspberry Pi Imager ("use custom image"), Win32DiskImager, balenaEtcher, or `dd` on Linux/macOS. **The whole card is overwritten.**

The image lays the card out as three partitions:

Partition	Size	Type	Purpose
1	64 MB	FAT	File interchange with Windows/macOS/Linux
2	32 MB	Fuzix root	The Unix filesystem (boot device `hdb2`)
3	4 MB	0x7F	Reserved for future on-card swap

Partition 1 ships unformatted: format it as **FAT or FAT32 (not exFAT)** in Windows before first use — see section 6. Partition 2 is Fuzix's own filesystem format; no desktop OS can mount it, which is why the FAT partition and the `fat` command exist.

2.3 First boot

Connect a monitor to the HDMI port and/or a terminal program (for example TeraTerm at 115200) to the USB-C console port. Switch on: the kernel banner, board identification and device probe appear on both, then:

```
login: root
```

There is no password. The system arrives at a Bourne shell; the console is an 80×40 colour terminal (termcap type `pc3`) and the USB keyboard and serial input feed the same session interchangeably. Set the keyboard layout in `/etc/rc` (`picoctl keymap uk` by default).

To power off, type `shutdown` (or at minimum `sync`, then wait a moment). After an unclean power-off the next boot repairs the filesystem automatically (`fsck -a -y`).

2.4 Setting the clock

The board's DS3231 battery-backed clock supplies the date and time: at every boot `/etc/rc` runs `setdate`, which reads the chip and sets system time silently. If the chip cannot supply a valid time (fresh battery, or the battery was removed) the same command falls back to prompting on the console:

```
Current date is Mon 2026-07-27
Enter new date:
```

Press Enter to keep a value, or type a new one (`2026-07-28`, then `16:30:00`). To set the clock at any other time run `setdate -u`, which always prompts.

Setting the system time does not touch the chip. To make the time permanent, write it back:

```
# setdate -w
writing
```

That also restarts a stopped oscillator, so the sequence for a new battery is: `setdate -u` (enter the time), then `setdate -w`. The kernel message `oscillator was stopped, time needs setting` at boot means exactly that sequence is wanted.

While running, system time is kept by the crystal-driven timer tick and gently re-synchronised from the DS3231 about once an hour, the same policy as MMBasic. `date` prints the current time.

If a transaction to the chip is ever interrupted at the wrong moment (a reset mid-read), the battery-backed DS3231 can be left jamming the I2C bus - even across power cycles. The kernel detects this at boot (`ds3231: SDA held low, clocking bus free`) and clears it automatically.

2.5 Escape routes

- **Esc** — inside BBC BASIC, stops the running program.
- **Ctrl-** — kernel emergency break: kills the foreground program even if it has the terminal in raw mode, and restores a sane terminal. Use it instead of the RESET button when something wedges.
- `kill` from the shell works on anything; Fuzix pre-emption guarantees a spinning process can't lock you out.

3 The consoles

Everything is mirrored: kernel and program output appear on the HDMI display and the serial console simultaneously, and input is merged from the USB keyboard and the serial line. TeraTerm is resized to 80×40 automatically at boot.

While BBC BASIC has a graphics MODE on screen, the HDMI display belongs to the graphics; text output continues to the serial side as a plain stream, so a transcript survives even a full-screen game.

4 BBC BASIC

Start it by name; leave with `QUIT`:

```
# bbcbasic
BBC BASIC for Fuzix Console v0.50
(C) Copyright R. T. Russell, 2025
>
```

This is R. T. Russell's BBC BASIC (the BBCSDL console edition), so the language is the full modern dialect: 64-bit integers, IEEE double floats, long variable names, `WHILE`/`ENDWHILE`, multi-line `IF`/`ELSE`/`ENDIF`, structures, and the inline assembler. Keywords must be typed in **capitals** (`PRINT`, not `print`); `*LOWERCASE ON` relaxes this. Star commands are case-insensitive.

4.1 Loading programs, including plain text

`LOAD` and `CHAIN` accept **both** program formats:

- the tokenised internal format written by `SAVE` (fast, exact), and
- **plain ASCII listings**, detected automatically. An unnumbered modern-style listing is given line numbers 10, 20, 30…; a listing with its own numbers keeps them. Windows line endings are fine, and typographic characters that desktop editors introduce (curly quotes, non-breaking spaces, tabs) are silently converted to their ASCII equivalents — an invisible "smart" character can otherwise produce a baffling `Syntax error` on a line that looks perfect.

The comfortable round trip: edit `myprog.bas` on your PC, copy it to the FAT partition, then:

```
# fat get myprog.bas
# bbcbasic
>LOAD "myprog.bas"
>RUN
>SAVE "MYPROG"          save tokenised for fast loading next time
```

4.2 Graphics

`MODE 0`–`5` switch the display to 1024×768 and provide the classic screens; `MODE` with any higher number (e.g. `MODE 6`) returns to the text console. Errors leave you at the prompt *in* the current mode, exactly like the original machine.

MODE	Resolution	Colours	Presentation on the monitor
0, 3	640×256	2	Full width (5:8 smooth upscale), full height
1, 4	320×256	4	960×768 — every pixel a crisp 3×3 block
2, 5	160×256	16	960×768 — every pixel a crisp 6×3 block

Coordinates are the authentic 1280×1024 graphics units, origin bottom-left, movable with `VDU 29`. Implemented: `MOVE`, `DRAW`, `PLOT` (move/line families 0–15, points 64–71, filled triangles 80–87 — so `PLOT 85` works), `CLG`, `GCOL`, `COLOUR`, `POINT(x,y)`, `VDU 19` palette changes (instant, no redraw), `TAB(x,y)`, and text printed with the built-in 8×8 font (40 or 80 columns × 32 rows, with scrolling). `MODE 1/4` store 4 bits per pixel, so all 16 logical colours are available there via `VDU 19` even though the default palette is the authentic four.

4.3 Sound

The full BBC sound system plays through the PCM5102 DAC and the 3.5 mm jack:

```
SOUND 1,-15,89,20                              A4 (440 Hz) for one second
ENVELOPE 1,1,4,-4,4,10,10,10,126,-2,0,-10,120,100
SOUND 1,1,100,30                               envelope-shaped note
SOUND &101,-15,53,20:SOUND &102,-15,69,20:SOUND &103,-15,81,20
                                               a synchronised C-E-G chord
```

Channels 1–3 are square-wave tones, channel 0 is noise. Each channel queues up to 8 notes; `&1x` in the channel number flushes the queue, `&1xx`–`&3xx` synchronise channels. `ENVELOPE` implements the three-section pitch envelope and ADSR amplitude, stepped at the authentic 100 Hz. Pitch follows the BBC scale: 4 units per semitone, 89 = A4 = 440 Hz. Durations are in 20ths of a second; 255 means "until further notice". `SOUND` blocks BBC-style when a queue is full (Esc still works).

4.4 ADVAL: joystick, analogue inputs, sound queues

Call	Returns
`ADVAL(0)`	Joystick switches on GP34–37 as a mask: bit 0 = GP34, bit 1 = GP35, bit 2 = GP36, bit 3 = GP37; pressed (grounded) = 1
`ADVAL(1)`–`ADVAL(4)`	Analogue readings of GP41–GP44, 0–65520 (12-bit ADC ×16)
`ADVAL(-1)`	Characters waiting in the keyboard buffer
`ADVAL(-5)`–`ADVAL(-8)`	Free queue slots on sound channels 0–3

The joystick inputs have pull-ups and Schmitt-trigger inputs enabled; wire switches directly to ground. The analogue inputs are 3.3 V full-scale.

4.5 Serial port

A second serial port lives on the I/O header: **TX = GP0, RX = GP1** (3.3 V logic, no flow control), visible as `/dev/tty2`. BBC BASIC reaches it as a *port channel* — raw and unbuffered:

```
*stty 9600 </dev/tty2
ch% = OPENUP("/dev/tty2")
BPUT#ch%,65                send a byte
X% = BGET#ch%              receive a byte (waits for one)
CLOSE#ch%
```

Any `stty` setting applies (baud rate, parity, size). The same `OPENIN`/`OPENUP`/`BGET#`/`BPUT#` mechanism works on any `/dev` device; ordinary file names get the buffered 256-byte file channels instead, exactly as on the original machine.

4.6 Star commands and the shell

The built-in set includes `*DIR`/`*.`, `*CD`, `*TYPE`, `*COPY`, `*DELETE`/`*ERA`, `*MKDIR`, `*RMDIR`, `*KEY`, `*SPOOL`, `*EXEC`, `*LOWERCASE`, `*TEMPO`, `*QUIT`. Anything unrecognised is passed to the Unix shell, so `*ls -l`, `*stty 9600 </dev/tty2` and even `*fat get demo.bas` work from inside BASIC.

4.7 The inline assembler

The assembler between `[` and `]` targets the machine it runs on: **ARM Thumb (v6-M subset)**, not 6502. `CALL` and `USR` work as documented for BBCSDL's ARM editions. Machine code written for a BBC Micro will not run; this is the same situation as every modern BBC BASIC.

4.8 Differences from the original BBC Micro

Better than the original:

- ~110 KB of program workspace (about 80 KB when graphics are in use) against the Model B's 32 KB shared with the screen
- 64-bit integers, double-precision floats, the modern language
- Long file names on a hierarchical filesystem
- The machine multitasks underneath — BASIC is just a process

Not (yet) present, compared with an original machine or full BBCSDL:

- **MODE 7** teletext (MODE 6 and 7 return to the console)
- `VDU 5` text-at-the-graphics-cursor; user-defined characters (`VDU 23`) are accepted but not rendered
- PLOT families beyond points/lines/triangles: no circles, arcs, ellipses, flood fill or rectangle/parallelogram fills yet
- `GCOL` action modes 1–4 (OR/AND/EOR/invert) plot as mode 0 (set)
- Flashing colours (8–15 map to their steady equivalents)
- `*FX` calls, `INKEY` with negative arguments (keyboard scanning), and the 6502 `CALL` interface
- `TIME` advances in 0.1 s steps (the kernel clock granularity)
- Cassette/Econet/ROM filing systems, for obvious reasons

5 The FAT partition and the `fat` command

Partition 1 of the SD card is ordinary FAT, readable and writable by any PC — this is how files travel to Fuzix, because nothing else can mount the Unix partition. Format it once in Windows (FAT or FAT32, **not exFAT**), then simply copy files onto it.

On the Fuzix side, the `fat` command reads it (long filenames and subdirectories included):

```
# fat info
/dev/hdb1: FAT16, 32695 clusters of 2048 bytes (62 MB)
# fat ls
    4523  My Program.bas
   <dir>  BASIC
# fat ls basic
# fat get "my program.bas" /root/prog.bas
# fat get demo.bin                    (lowercased name in current dir)
```

`fat` is read-only by design — the desktop does the writing. To move a file *out* of Fuzix, use the serial port, or write it with `doswrite` (the venerable Minix FAT12/16 tool, also included) if the partition is

FAT16.

6 The filesystem and included software

6.1 Layout

```
/bin          core utilities (always available)
/usr/bin      larger tools, BBC BASIC, fat, picoctl
/usr/games    the games collection
/usr/lib/bbc  BBC BASIC library directory (@lib$)
/usr/man      manual pages (man <name>)
/dev          devices - see below
/etc          rc (boot script), inittab, passwd, termcap
/tmp          scratch space
/root         root's home directory
```

Key devices:

Device	What it is
`/dev/tty1`	The console (HDMI + USB keyboard + serial mirror)
`/dev/tty2`	The GP0/GP1 serial port
`/dev/hda`	On-board NAND flash filesystem
`/dev/hdb1`–`hdb3`	SD card partitions (root is `hdb2`)
`/dev/hdc`	The 8 MB PSRAM (used as swap; see `free`)
`/dev/rtc`	The DS3231 clock (`setdate` reads and sets it)
`/dev/sys`	Platform control (graphics, sound, ADVAL ioctls)

`/etc/rc` runs at boot: filesystem check, clock from the DS3231, swap activation, keyboard layout. Edit it with any of the editors below.

6.2 Applications

Editors: `vi` (levee), `ue` (a WordStar-diamond micro-Emacs: Ctrl-W write, Ctrl-Q quit), `ed`, `fleamacs`.

Shell & core: Bourne `sh` (plus `fsh`), and the classic set — `ls ll cp mv ln rm mkdir rmdir cat more less head tail grep fgrep sed tr cut sort uniq wc find xargs diff diff3 cmp comm join split rev tar dd df du free ps kill killall uptime date cal banner echo sleep tee touch which who su passwd stty mount umount sync fsck mkfs fdisk chmod chown chgrp od hd factor seq units dc expr m4 make cron at mail write wall` ``` (and more - seels /bin /usr/bin‘).

Machine tools: `picoctl` (keyboard layout, reboot to the flasher), `picogpio`/`gpiotool`, `gfxtest` (display test card), `setdate` (DS3231), `flashrom`, `setboot`, `dosread`/`doswrite` (FAT12/16 floppy-era transfers), `fat`.

Languages: `bbcbasic`, `fforth` (a complete ANS Forth), the `as09`/`ld09` assembler pair, and `dc`.

Games (`/usr/games`): the original Colossal Cave `advent`, the complete Scott Adams `adv01`–`adv14` and Mysterious Adventures `myst01`–`myst11` collections, Infocom Z-machine interpreters `z1`–`z8` and `l9x` (Level 9), `startrek`, `hamurabi`, `backgammon`, `invaders`, `2048`, `moo`, `ttt`, `fish`, `arithmetic`, `fortune`, `cowsay`, `wump`.

7 Appendix A: pin usage

GPIO	Function
GP0 / GP1	Serial port `/dev/tty2` (TX / RX)
GP8 / GP9	System console UART (USB-C CH340)
GP10/11/22	Audio I2S: BCLK / LRCLK / DATA (PCM5102 DAC)
GP12–GP19	HDMI (HSTX)
GP20 / GP21	I2C0: DS3231 RTC and QWIIC connector (SDA / SCL)
GP27	DS3231 32 kHz — board identification (PC3 only)
GP28/30/31/33	SD card, Pico Computer 3 (MISO/SCK/MOSI/CS, SPI1)
GP29/30/31/32	SD card, Pico Computer 2 (CS/SCK/MOSI/MISO, bit-banged)
GP32	DS3231 alarm interrupt (PC3; unused by Fuzix)
GP34–GP37	Joystick switches — `ADVAL(0)` bits 0–3
GP40	Analogue input (reserved; not read by ADVAL)
GP41–GP44	Analogue inputs — `ADVAL(1)`–`ADVAL(4)`
GP45/GP46	Analogue-capable, free
GP23/24/25/29	CYW43 wireless (PC3; unused by Fuzix)
GP47	PSRAM chip select

All spare header pins remain ordinary 3.3 V GPIO. Do not apply 5 V to any GPIO.

8 Appendix B: boot options

The kernel command line (held by the bootloader) accepts:

- `hda` / `hdb2` … — root device override
- `kbd=us|uk|de|fr|es|be` — early keyboard layout override (the layout in `/etc/rc` then applies for the session)

The build, source and development notes live in the `pc3` branch of `github.com/UKTailwind/FUZIX` under `Kernel/platform/platform-rpipico/` — see `PC3-DEVNOTES.md` for the engineering history and `PC3-GFX-DESIGN.md` for the display design.