

cowlark.com

Fuzix on the Raspberry Pi Pico

6-8 minutes

2021 February 16

So, my new [Raspberry Pi Picos](#) (two of them) arrived just as I was finishing up the [ESP8266 Fuzix port](#), and naturally I had to port Fuzix to this too. I didn't video this one, which is a shame as it would probably have been a more interesting watch.

The Pico is an interesting device: two Cortex M0+ cores running at approximately 130MHz, overclockable up to lots; 2MB of NAND flash on chip; 269kB of RAM; and a large collection of interesting hardware, including two high-speed IO coprocessors which allow you to do some really interesting things. The [Fuzix](#) port only uses one core but it runs really nicely on it, with RAM to spare. Compared to the ESP8266 it seems a little slower, but I haven't touched the overclocking settings yet. Performance is still completely adequate for an interactive Unix.

Working with the Pico was an experience: the documentation is excellent, as is the C SDK. The SDK provides a set of libraries which are thin wrappers around the underlying hardware, making most features utter simplicity to use. Unlike the ESP8266's libraries, the Pico SDK is unopinionated and doesn't require you to use any of its features: if you want to talk directly to the hardware, you can (and in fact there's library support for doing just this).

There are some high-level features like a heap, stdio emulation, multicore primitives, etc which I'm not using, but if you don't use them you don't pay for them. For my embedded-systems brain it's the ideal ratio of functionality to complexity. I wish more platforms provided libraries this good. Maybe we'll get compatible ports of the Pico SDK to other systems; I can only hope.

There are some niggles, such as the almost non-negotiable requirement to use cmake as your build system. cmake is pretty ghastly but to be honest build systems are always terrible, and cmake's not the worst choice. Once I nerved myself to trust it, it was a breeze to use and the Pico SDK's cmake addons make building binaries and flashable images trivial.

Development was done using the Pico port of OpenOCD. I don't have a JTAG debugger which will work on the Pico, but that's fine, because there's a turnkey image which will let you use a Pico as a JTAG debugger for a Pico! Which is why I bought two. And the debugger [was written with the SDK](#), too...

So, the Fuzix port provides:

- user binaries using up to 64kB of code and data each (this could be expanded, as there's plenty of spare RAM)
- up to 15 processes
- a proper Unix filesystem
- SD card support, used for both the filesystem and swap space
- serial console on UART0
- the full set of Fuzix core binaries work — fsck, Bourne shell, the standard Unix tools, a vi clone, etc, plus some simple games

What you don't get:

- NAND flash support: the code's done, but I've discovered that a bad filesystem will crash the dhara FTL library. As the Pico's flash is too small to use for swap, you have to have an SD card anyway, so I haven't bothered fixing this.
- Multitasking support: currently only the most recent process runs. Most things will work fine, with the big exception of pipes. This *should* work but doesn't due to a bug.
- an absence of bugs

Screenshots are kinda meaningless, but here's a serial terminal dump. if this looks similar to the ESP8266 one, it's because I'm doing all the same things.

```
FUZX version 0.4pre1
```

```
Copyright (c) 1988-2002 by H.F.Bower, D.Braun,  
S.Nitschke, H.Peraza
```

```
Copyright (c) 1997-2001 by Arcady Schekochikhin,  
Adriano C. R. da Cunha
```

```
Copyright (c) 2013-2015 Will Sowerbutts  
<will@sowerbutts.com>
```

```
Copyright (c) 2014-2020 Alan Cox  
<alan@etchedpixels.co.uk>
```

```
Devboot
```

```
64kB total RAM, 64kB available to processes (15  
processes max)
```

```
Enabling interrupts ... ok.
```

```
SD drive 0: hda: hda1 hda2
```

```
Mounting root fs (root_dev=2, ro): warning:  
mounting dirty file system, forcing r/o.
```

OK

Starting /init

init version 0.9.0ac#1

Cannot open file

Current date is Mon 2021-02-15

Enter new date:

Current time is 22:20:37

Enter new time:

/etc/rc: /var/run/utmp: cannot create

^ ^

n n Fuzix 0.3.1

>@<

Welcome to Fuzix

m m

login: root

Welcome to FUZIX.

stty erase '^?'

cd /

ls

bin

dev

etc

init

mnt

root

tmp

usr

```
var
# prtroot
/dev/hda2 / fuzix rw 0 0
# df
Filesystem            Blocks    Used    Free   %Used
Mounted on
/dev/hda2              65279    3752   61527     5% /
# cd /usr/games
# ls fortune*
fortune
fortune-gen
fortune.dat
# ls -l fortune*
-rwxr-xr-x    1 root      0                3720 Feb 15
22:20 fortune
-rwxr-xr-x    1 root      0                4876 Feb 15
22:20 fortune-gen
-rw-r--r--    1 root      0               241378 Feb 15
22:20 fortune.dat
# ./fortune
The sendmail configuration file is one of those
files that looks like someone
beat their head on the keyboard.  After working
with it... I can see why!
                -- Harry Skelton
# ./fortune
The only thing worse than X Windows: (X Windows)
- X
# ./fortune
A certain monk had a habit of pestering the Grand
```

Tortue (the only one who
had ever reached the Enlightenment 'Yond
Enlightenment), by asking whether
various objects had Buddha-nature or not. To
such a question Tortue
invariably sat silent. The monk had already
asked about a bean, a lake,
and a moonlit night. One day he brought to
Tortue a piece of string, and
asked the same question. In reply, the Grand
Tortue grasped the loop
between his feet and, with a few simple
manipulations, created a complex
string which he proffered wordlessly to the monk.
At that moment, the monk
was enlightened.

From then on, the monk did not bother Tortue.
Instead, he made string after
string by Tortue's method; and he passed the
method on to his own disciples,
who passed it on to theirs.

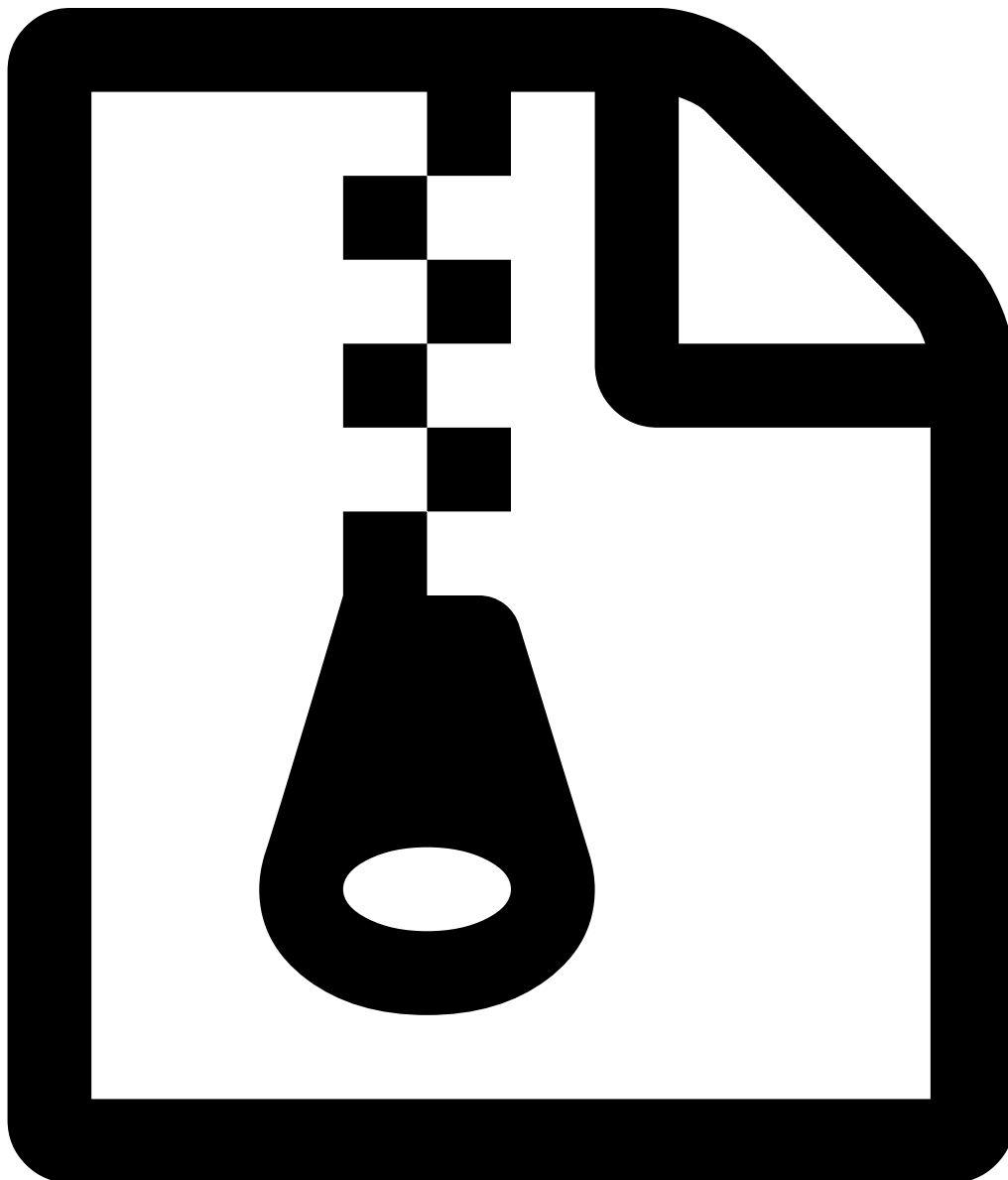
./startrek

```
*
*
*      * * Super Star Trek * *
*
*
```

Do you need instructions (y/N):

If you're looking for the source code, I'm currently upstreaming it piece by piece to [the main FUZIX repository](#). Until that's done, look in [my own fork](#).

If you just want a binary to flash and try for yourself, here's one:



[Raspberry Pi Pico Fuzix binaries](#) 501 kB

Poorly put-together, bugridden and unsupported Fuzix binaries for the Raspberry Pi Pico. Instructions are enclosed, more or less.