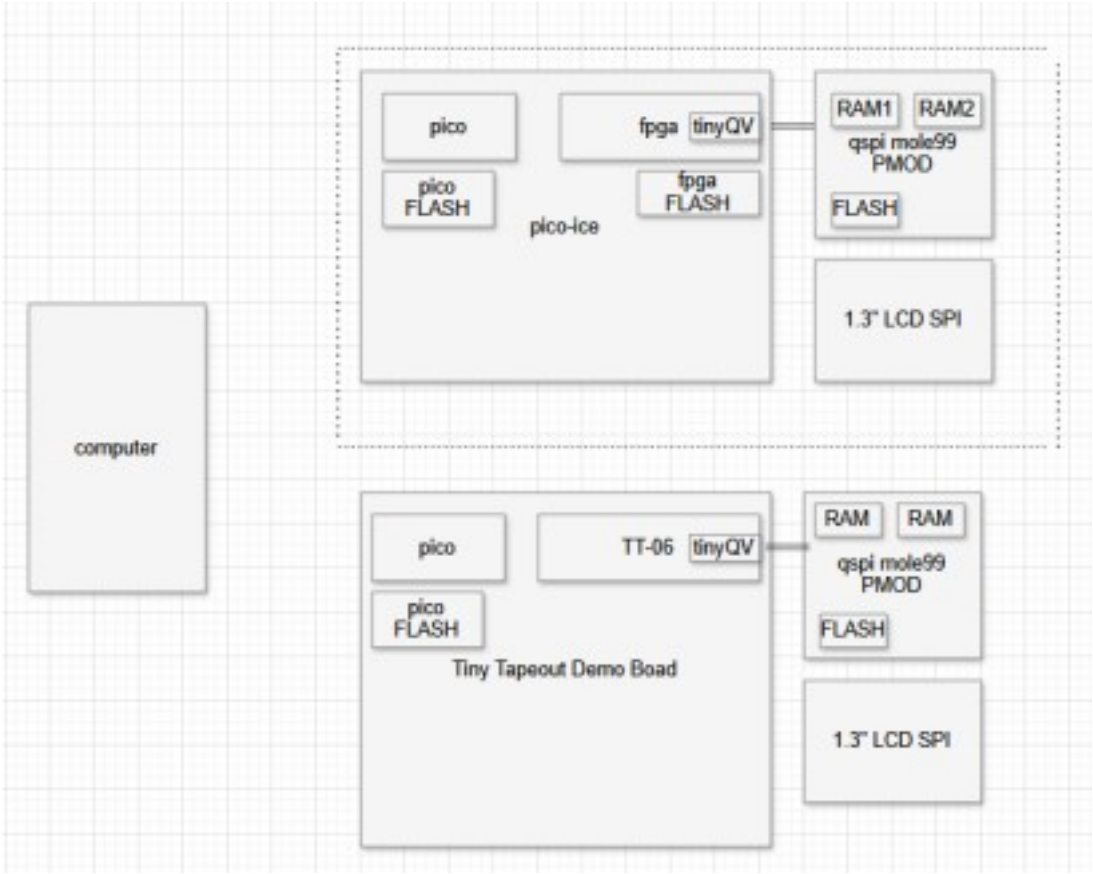
**********Default**********
MicroPython Pico-Ice tinyQV or nanoV
sharing files between the Raspberry Pi 5 and rp2040 of the pico-ice
Using mpremote
11/10/24
**********Default**********

tinyQV TT06 or nanoV TT04



The computer in the figure above is a Raspberry Pi 5.
The pico in the figure above is rp2040 on the pico-ice module.

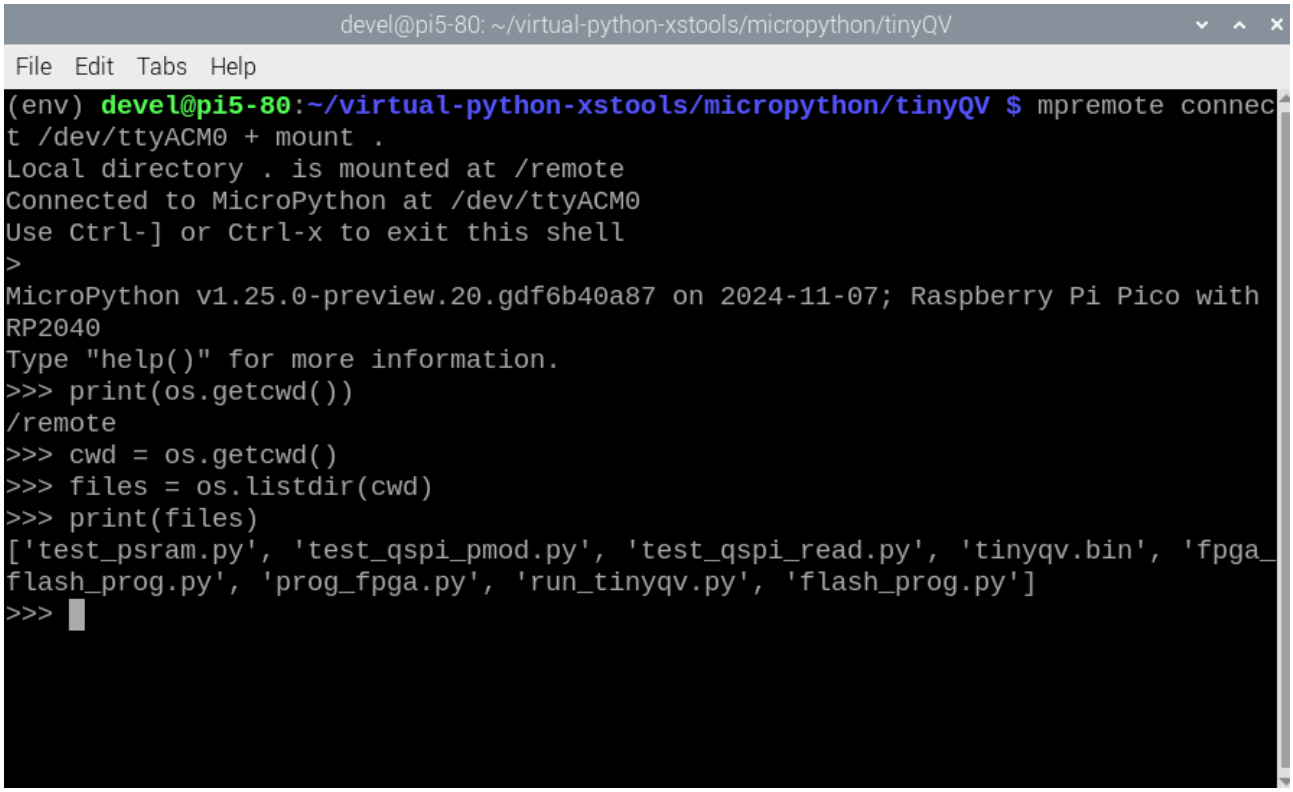
Running in a virtual environment on Raspberry Pi 5.
“cd virtual-python-xstools”
“. virtp.sh”
“cd micropython/nanoV”
Connect to rp2040 of the pico-ice which has the firmware.uf2 running MicroPython.

MicroPython remote control: mpremote

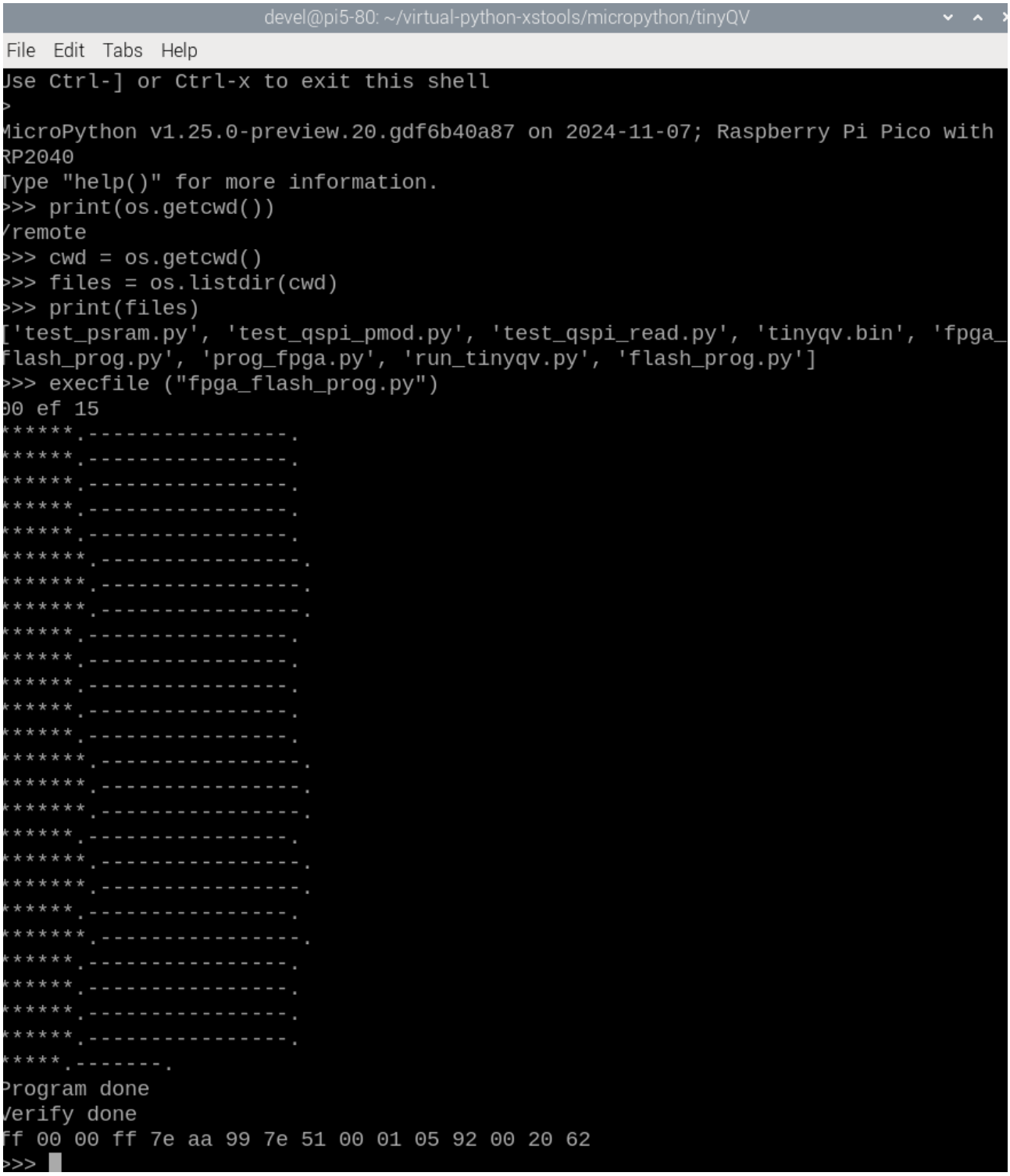
tree

```
.
├── flash_prog.py
├── fpga_flash_prog.py
├── prog_fpga.py
└── run_tinyqv.py
```

```
├── test_psram.py
├── test_qspi_pmod.py
├── test_qspi_read.py
└── tinyqv.bin
```

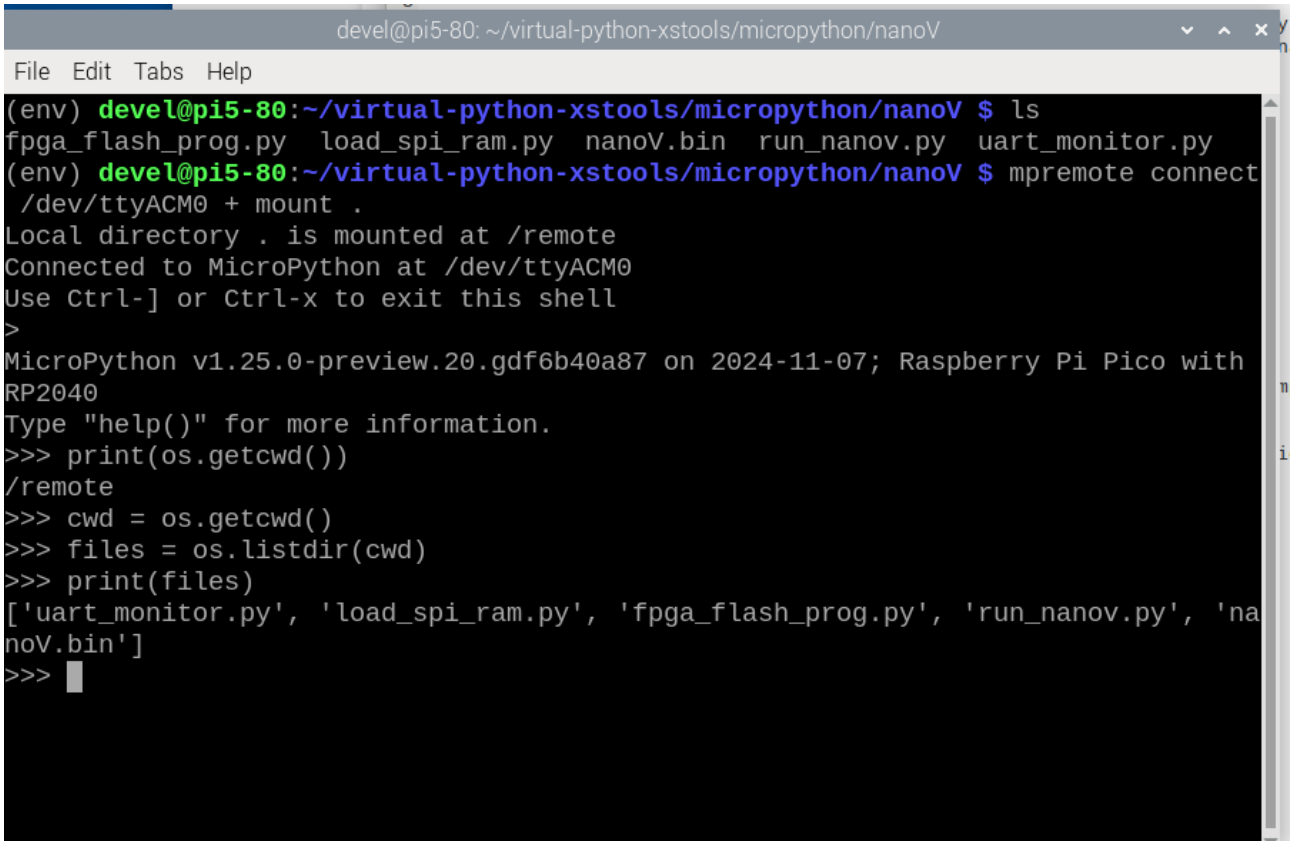


xxxxx



The `mpremote` command line tool provides an integrated set of utilities to remotely interact with, manage the filesystem on, and automate a MicroPython device over a serial connection.

“mpremote connect /dev/ttyACM0 + mount .”



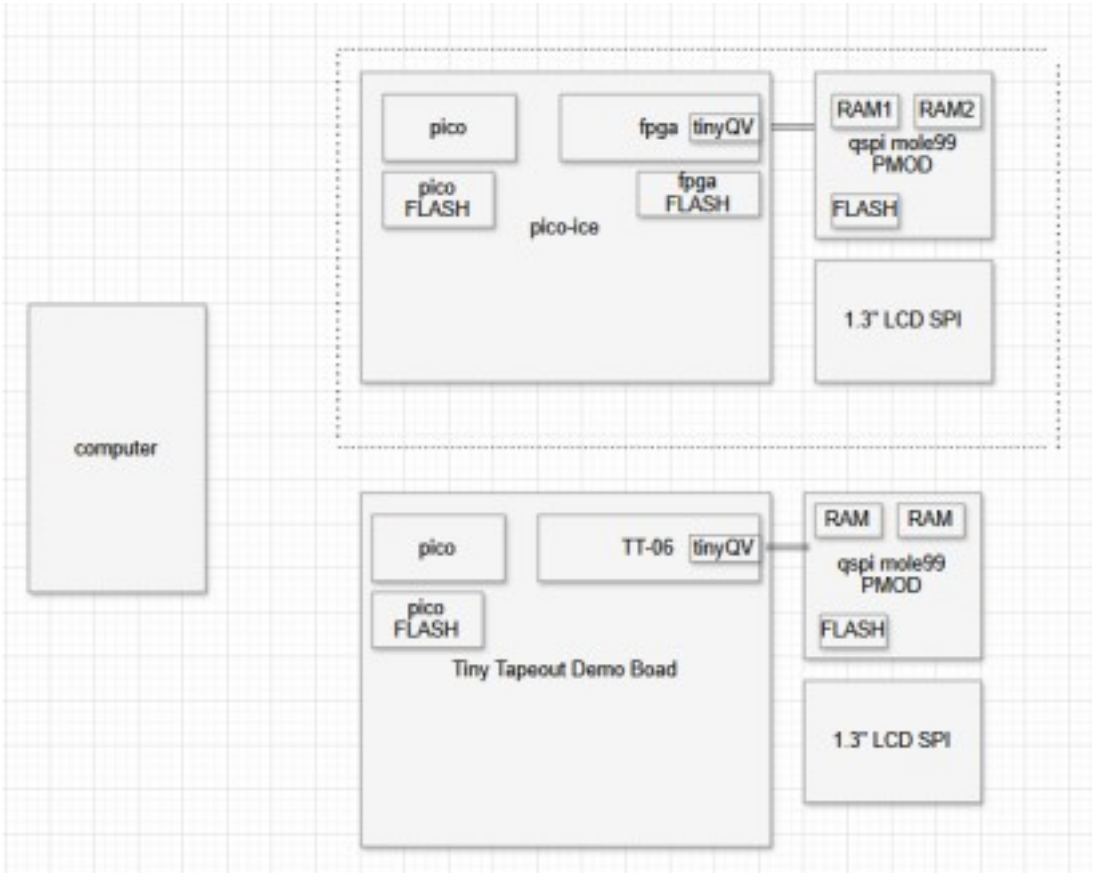
The computer in the figure below is the Raspberry Pi5 which
(env) devel@pi5-80:~/virtual-python-xstools/micropython/nanoV $ tree

```
.
├── fpga_flash_prog.py
├── load_spi_ram.py
├── nanoV.bin
├── run_nanov.py
└── uart_monitor.py
```

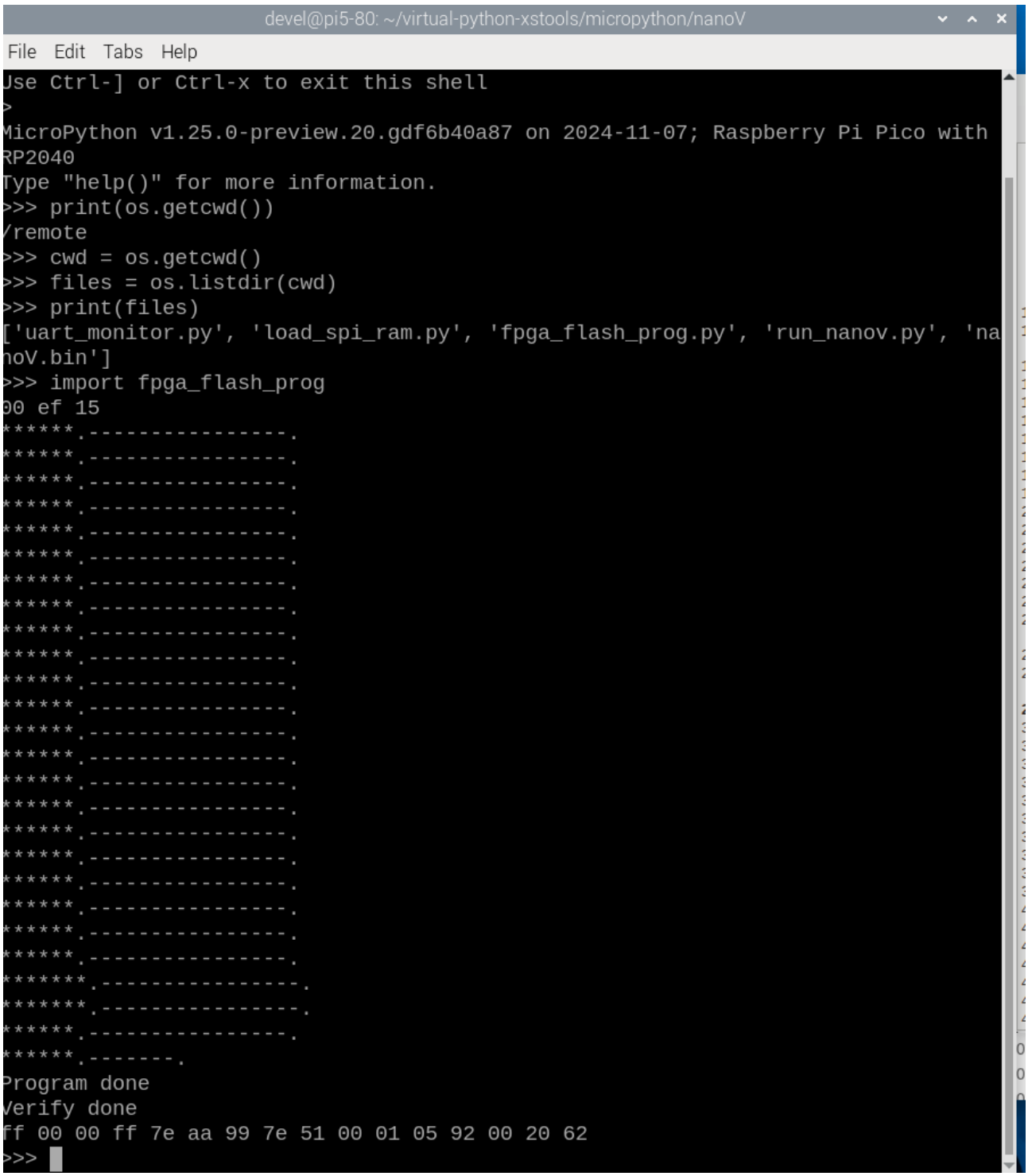
1 directory, 5 files

tinyQV or nanoV

I use standard Pico Micropython on the RP2040, and then `fpga_flash_prog.py` to program the ICE40 (supplying the `.bin` built by yosys and nextpnr).



At the prompt execute “import fpga_flash_prog” which will read the file “nanoV.bin” and program the iCE40UP5K fpga flash of the pico-ice in the figue above.


devel@pi5-80: ~/virtual-python-xstools/micropython/nanoV
File Edit Tabs Help
Use Ctrl-] or Ctrl-x to exit this shell
>
MicroPython v1.25.0-preview.20.gdf6b40a87 on 2024-11-07; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> print(os.getcwd())
/remote
>>> cwd = os.getcwd()
>>> files = os.listdir(cwd)
>>> print(files)
['uart_monitor.py', 'load_spi_ram.py', 'fpga_flash_prog.py', 'run_nanov.py', 'nanoV.bin']
>>> import fpga_flash_prog
00 ef 15
Program done
Verify done
ff 00 00 ff 7e aa 99 7e 51 00 01 05 92 00 20 62
>>>