

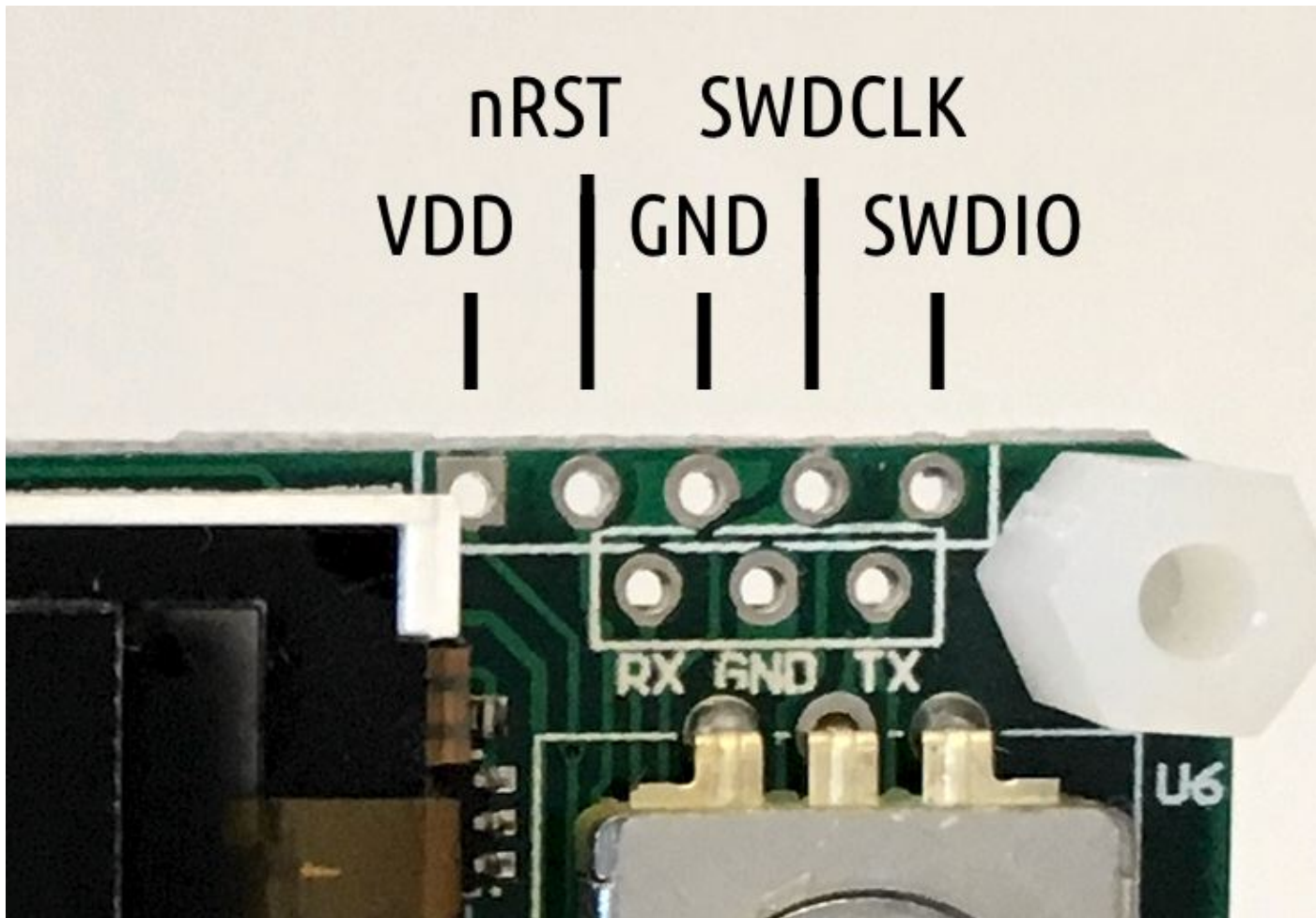
Contents: [Dobrica PavlinuÅiÄ 's random unstructured stuff]

- Dobrica PavlinuÅiÄ 's random unstructured stuff (OpenDPS)
- Dobrica PavlinuÅiÄ 's random unstructured stuff (pinout)
- Dobrica PavlinuÅiÄ 's random unstructured stuff (upgrade to opensource firmware)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (st-link)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (openocd)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (backup few states)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (erase)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (flash)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (client software)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (serial upgrade)
  - ◆ Dobrica PavlinuÅiÄ 's random unstructured stuff (esp8266)
    - ◇ Dobrica PavlinuÅiÄ 's random unstructured stuff (wifi config)
    - ◇ Dobrica PavlinuÅiÄ 's random unstructured stuff (build)
    - ◇ Dobrica PavlinuÅiÄ 's random unstructured stuff (flash)
    - ◇ Dobrica PavlinuÅiÄ 's random unstructured stuff (test)
    - ◇ Dobrica PavlinuÅiÄ 's random unstructured stuff (powering esp8266)

## OpenDPS

- <https://johan.kanflo.com/hacking-the-dps5005/>
- <https://johan.kanflo.com/opendps-design/>
- <https://johan.kanflo.com/upgrading-your-dps5005/>

## pinout



## upgrade to opensource firmware

<https://github.com/kanflo/opensdps.git>

```
dpavlin@nuc:/nuc$ git clone --recursive https://github.com/kanflo/opensdps.git
dpavlin@nuc:/nuc$ cd opensdps/
dpavlin@nuc:/nuc/opensdps$ make -C libopenm3
dpavlin@nuc:/nuc/opensdps$ make -C opensdps
```

## st-link

connecting rst pin to unpowered st-link stops dps5005 from booting

<https://johan.kanflo.com/upgrading-your-dps5005/>

## openocd

```
pi@pihmi:~$ sudo openocd -f interface/stlink-v2.cfg -f target/stm32f1x.cfg
Open On-Chip Debugger 0.10.0+dev-01489-g06c7a53f1-dirty (2020-11-14-15:21)
Licensed under GNU GPL v2
```

```
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
WARNING: interface/stlink-v2.cfg is deprecated, please switch to interface/stlink.cfg
Info : auto-selecting first available session transport "hla_swd". To override use 'transport sel
Info : The selected transport took over low-level target control. The results might differ compar
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : clock speed 1000 kHz
Info : STLINK V2J28S7 (API v2) VID:PID 0483:3748
Info : Target voltage: 3.273018
Info : stm32f1x.cpu: hardware has 6 breakpoints, 4 watchpoints
Info : starting gdb server for stm32f1x.cpu on 3333
Info : Listening on port 3333 for gdb connections
```

## backup few states

from another terminal

```
pi@pihdmi:/nuc/opendps $ ./ocd-client.py all | tee 5V-off.txt

# exit openocd, turn output on

pi@pihdmi:/nuc/opendps $ ./ocd-client.py all | tee 5V-on.txt
```

## erase

```
pi@pihdmi:/nuc/opendps $ telnet localhost 4444
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Open On-Chip Debugger
> reset halt
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x080010a4 msp: 0x20001ff0
> flash erase_address unlock 0x08000000 0x10000
device id = 0x10016420
flash size = 64kbytes
erased address 0x08000000 (length 65536) in 0.145284s (440.517 KiB/s)
```

## flash

```
pi@pihdmi:/nuc/opendps/opendps $ make flash
FLASH opendps_DPS5005.srec
(echo "halt; program /nuc/opendps/opendps/opendps_DPS5005.srec verify reset" | nc -4 localhost 44
    openocd -f interface/stlink-v2.cfg \
    -f target/stm32f1x.cfg \
    -c "program opendps_DPS5005.srec verify reset exit" \
    2>/dev/null
iç¾iç¾iç¾iç¾iç¾iç¾iç¾iç¾iç¾iç¾Open On-Chip Debugger
> halt; program /nuc/opendps/opendps/opendps_DPS5005.srec verify reset
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x08000bb8 msp: 0x20001fb8
target halted due to debug-request, current mode: Thread
```



```
V_in      : 10.78 V
V_out     : 1.73 V
I_out     : 0.001 A
```

```
pi@pihdmi:/nuc/opendps/dpsctl $ python3 ./dpsctl.py --device /dev/ttyUSB2 --query
Func      : cv (off)
  voltage : 5000
  current : 1000
V_in      : 10.78 V
V_out     : 0.01 V
I_out     : 0.000 A
```

## serial upgrade

```
pi@pihdmi:/nuc/opendps/dpsctl $ python3 dpsctl.py --device /dev/ttyUSB2 -U ../opendps/opendps_DPS
Download progress: 2% Error: timeout talking to device /dev/ttyUSB2.
```

### SEL on power-on to stay in bootloader

```
# it bricked my board first time I tried it
```

## esp8266

I had trouble finding toolchain which works with it, I ended up using <https://github.com/pfalcon/esp-open-sdk>

```
dpavlin@nuc:/nuc/esp8266/esp-open-sdk$ git remote -v
origin  https://github.com/pfalcon/esp-open-sdk.git (fetch)
origin  https://github.com/pfalcon/esp-open-sdk.git (push)
```

I also needed to replace esptool, since this machine requires python3 version of it

```
ln -s /nuc/esp32/esptool/esptool.py /nuc/esp8266/esp-open-sdk/xtensa-lx106-elf/bin/esptool.py
```

## wifi config

```
dpavlin@nuc:/nuc/opendps$ ls -al esp8266-proxy/esp-open-rtos/include/private_ssid_config.h
-rw-r--r-- 1 dpavlin dpavlin 60 Aug  4 10:21 esp8266-proxy/esp-open-rtos/include/private_ssid_con
dpavlin@nuc:/nuc/opendps$ vi esp8266-proxy/esp-open-rtos/include/private_ssid_config.h
```

## build

```
dpavlin@nuc:/nuc/opendps$ cd esp8266-proxy/
dpavlin@nuc:/nuc/opendps/esp8266-proxy$ cat env.sh
```

```
export EOR_ROOT=`pwd`/esp-open-rtos
export PATH=/nuc/esp8266/esp-open-sdk/xtensa-lx106-elf/bin:$PATH
dpavlin@nuc:/nuc/opendps/esp8266-proxy$ . env.sh
```

```
dpavlin@nuc:/nuc/opendps/esp8266-proxy$ make
```

```
Merged 1 ELF section
```

## flash

```
dpavlin@nuc:/nuc/opendps/esp8266-proxy$ make flash
esptool.py -p /dev/ttyUSB0 --baud 115200 write_flash -fs 16m -fm qio -ff 40m \
    0x0 esp-open-rtos/bootloader/firmware_prebuilt/rboot.bin 0x1000 esp-open-rtos/bootloader/
WARNING: Flash size arguments in megabits like '16m' are deprecated.
Please use the equivalent size '2MB'.
Megabit arguments may be removed in a future release.
esptool.py v3.2-dev
Serial port /dev/ttyUSB0
```

```
pi@pihdm1:/nuc/opendps/esp8266-proxy $ /nuc/esp32/esptool/esptool.py --port /dev/ttyUSB3 write_fl
esptool.py v3.2-dev
Serial port /dev/ttyUSB3
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 5c:cf:7f:c2:6b:19
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x00000fff...
Flash will be erased from 0x00001000 to 0x00001fff...
Flash will be erased from 0x00002000 to 0x000043fff...
Compressed 3104 bytes to 2169...
Wrote 3104 bytes (2169 compressed) at 0x00000000 in 0.3 seconds (effective 79.1 kbit/s)...
Hash of data verified.
Compressed 2048 bytes to 23...
Wrote 2048 bytes (23 compressed) at 0x00001000 in 0.1 seconds (effective 132.3 kbit/s)...
Hash of data verified.
Compressed 268660 bytes to 195361...
Wrote 268660 bytes (195361 compressed) at 0x00002000 in 18.0 seconds (effective 119.6 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

## test

reset board and verify that it's connecting to wifi

```
pi@rpi4:/nuc/opendps/dpsctl $ microcom -p /dev/ttyUSB0 -s 9600
connected to /dev/ttyUSB0
Escape character: Ctrl-\
Type the escape character to get to the prompt.
2dnÜ%î¿%bdbl¿î¿%$î¿%`dlî¿%î¿%x$2î¿%+`b: `:rî¿%pdî¿%zt {î¿%[î¿%8)lJî¿%î¿%î¿%î¿%î¿%î¿%: sta(4c:11:ae:0
```

```
add if0
scandone
add 0
aid 16
cnt

connected with dreamhouse, channel 10
dhcp client start...
ip:192.168.3.111,mask:255.255.255.0,gw:192.168.3.1
```

## now test client

```
pi@rpi4:~ $ /nuc/opendps/dpsctl/dpsctl.py -S
192.168.3.111
^C

pi@rpi4:~ $ export DPSIF=192.168.3.111

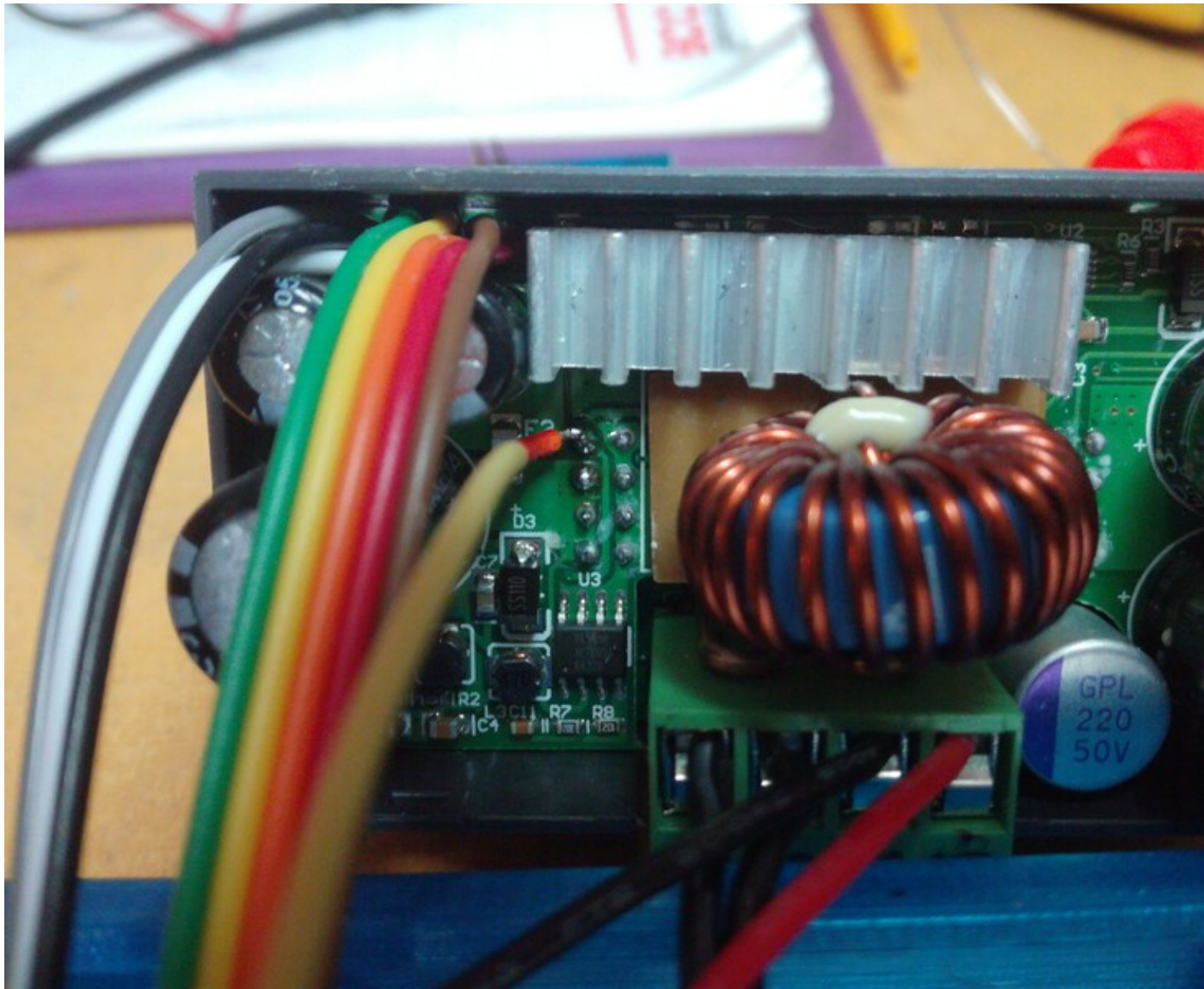
pi@rpi4:~ $ /nuc/opendps/dpsctl/dpsctl.py -q
Func      : cv (off)
  voltage : 5000
  current  : 1000
V_in      : 9.71 V
V_out     : 0.00 V
I_out     : 0.000 A
```

## powering esp8266

3.3v voltage is available at jtag pins, but it can't supply enough current to run esp8266

There is 5v pin from buck converter on top-left pin of 2x4 pins on left side of board (when looking from back side of module)

```
5v gnd
o o
o o
o o
o o
```



power drain from input voltage is roughly doubled when powering esp8266 from 5v rail

input voltage	no esp8266	with esp8266	
7.68 v	55 mA	110 mA	[0, 2]
9v	46 mA	97 mA	[1, 2]
			[2, 2]