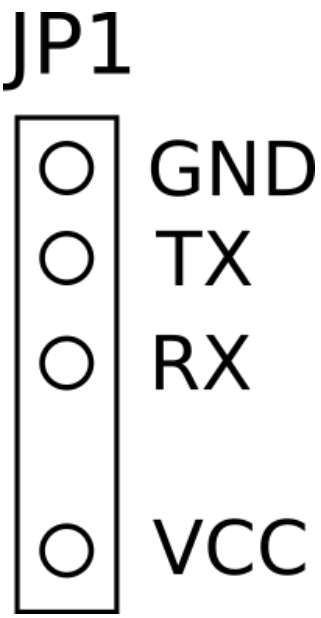
Ralf's Blog

COMMUNICATIONS, ENGLISH

HACKING THE GENEXIS FIBERTWIST-P2410

2016-03-29 | RALF BERGS | 16 COMMENTS

In my previous article I described the key components the Genexis FiberTwist-P2410 is comprised of. One of these components is the serial console connector, and its presence was so tempting that I simply *had* to play with it…



Layout of Serial Console Connector

So I connected a UART-to-USB converter and watched the console output while the device boots… Communications parameters were easy to guess: 115,200 bps, 8N1, no handshake (neither HW, nor SW)…

```
ROM VER: 1.0.0
CFG 06
NAND

ROM VER: 1.0.0
CFG 06
NAND
```

```
bootstrap-polar-2.1.0-R (Dec  1 2015 - 15:47:13)

DDR autotuning Rev 1.0
DDR size from 0xa0000000 - 0xa3ffffff
DQS GATE ECHO DLL Delay Slice0:00000014
DQS GATE ECHO DLL Delay Slice1:00000016
Read DQS Delay Slice0:00000026
Read DQS Delay Slice1:00000026
Write DQS Delay Slice0:00000025
Write DQS Delay Slice1:00000025


bootloader-polar-2.1.0-R (Dec 01 2015 - 15:46:40)

CLOCK CPU 600M RAM 300M
16 Bit RAM
DRAM:  128 MiB
NAND:  NAND device: Manufacturer ID: 0xc8, Chip ID: 0xd1 (Unknown NAND
128MiB 3,3V 8-bit)
128 MiB
Bad block table found at page 65472, version 0x01
Bad block table found at page 65408, version 0x01
*** Warning - bad CRC or NAND, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   internal phy using 25Mhz clock
Internal phy firmware version: 0x8434
ar10 Switch

Type "run flash_nfs" to mount root filesystem over NFS

Hit any key to stop autoboot:  1  0
Creating 1 MTD partitions on "nand0":
0x000000280000-0x000007f80000 : "mtd=4"
UBI: attaching mtd1 to ubi0
UBI: physical eraseblock size:   131072 bytes (128 KiB)
UBI: logical eraseblock size:    129024 bytes
UBI: smallest flash I/O unit:    2048
UBI: sub-page size:              512
UBI: VID header offset:          512 (aligned 512)
```

```
UBI: data offset:               2048
UBI: attached mtd1 to ubi0
UBI: MTD device name:           "mtd=4"
UBI: MTD device size:           125 MiB
UBI: number of good PEBs:       1000
UBI: number of bad PEBs:        0
UBI: max. allowed volumes:      128
UBI: wear-leveling threshold:   4096
UBI: number of internal volumes: 1
UBI: number of user volumes:    2
UBI: available PEBs:            0
UBI: total number of reserved PEBs: 1000
UBI: number of PEBs reserved for bad PEB handling: 20
UBI: max/mean erase counter: 2/0
UBIFS: mounted UBI device 0, volume 1, name "data"
UBIFS: mounted read-only
UBIFS: file system size:   120250368 bytes (117432 KiB, 114 MiB, 932
LEBs)
UBIFS: journal size:       9033728 bytes (8822 KiB, 8 MiB, 71 LEBs)
UBIFS: media format:       w4/r0 (latest is w4/r0)
UBIFS: default compressor: LZO
UBIFS: reserved for root:  0 bytes (0 KiB)
feed 'geneos-polar-2.1.0-R.img', ino 82, new f_pos 0x9b0a8e3find file
geneos-polar-2.1.0-R.img on position 0Loading file 'fw/0/geneos-polar-
2.1.0-R.img' to address 0x83000000 (size 0)
Loading file 'fw/0/geneos-polar-2.1.0-R.img' to addr 0x83000000 with
size 6546932 (0x0063e5f4)...
Done
## Booting kernel from FIT Image at 83000000 ...
   Using 'conf@1' configuration
   Trying 'kernel@1' kernel subimage
     Description:  Generic initramfs
     Type:         Kernel Image
     Compression:  lzma compressed
     Data Start:   0x83000118
     Data Size:    6535067 Bytes = 6.2 MiB
     Architecture: MIPS
     OS:           Linux
     Load Address: 0x80002000
     Entry Point:  0x80002000
     Hash algo:    sha1
     Hash value:   42bd16e172686233005096bde4abefe44bcf566b
```

```
   Verifying Hash Integrity ... sha1+ OK
## Flattened Device Tree from FIT Image at 83000000
   Using 'conf@1' configuration
   Trying 'fdt@1' FDT blob subimage
     Description:  Genexis Polar FDT blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x8363b9a8
     Data Size:    10482 Bytes = 10.2 KiB
     Architecture: MIPS
     Hash algo:    sha1
     Hash value:   5050dde93e7d83b3c5339da2b8e9cdf227f44658
   Verifying Hash Integrity ... sha1+ OK
   Booting using the fdt blob at 0x8363b9a8
data_blob [0x8363b9a8], gpio [15]
Flash system LED
   Uncompressing Kernel Image ... OK

Starting kernel ...

[    0.000000] Linux version 3.10.12 (jenkins@jenkins) (gcc version
4.8.3 (OpenWrt/Linaro GCC 4.8-2014.04 unknown) ) #2 Tue Dec 1 15:53:05
CET 2015
[    0.000000] SoC: xRX330 rev 1.1
[    0.000000] bootconsole [early0] enabled
[    0.000000] CPU0 revision is: 00019556 (MIPS 34Kc)
[    0.000000] adding memory size:133169152 from DT
[    0.000000] Determined physical RAM map:
[    0.000000]  memory: 07f00000 @ 00000000 (usable)
[    0.000000] Initrd not found or empty - disabling initrd
[    0.000000] Zone ranges:
[    0.000000]   Normal   [mem 0x00000000-0x07efffff]
[    0.000000] Movable zone start for each node
[    0.000000] Early memory node ranges
[    0.000000]   node   0: [mem 0x00000000-0x07efffff]
[    0.000000] Primary instruction cache 32kB, 4-way, VIPT, linesize 32
bytes.
[    0.000000] Primary data cache 32kB, 4-way, VIPT, cache aliases,
linesize 32 bytes
[    0.000000] Built 1 zonelists in Zone order, mobility grouping on.
Total pages: 32258
[    0.000000] Kernel command line: ubi.mtd=system_sw
```

```
console=ttyLTQ0,115200 init=/etc/preinit bootstrap_ver="bootstrap-polar-
2.1.0-R" bootloader_ver="bootloader-polar-2.1.0-R" fw_number=0
[    0.000000] PID hash table entries: 512 (order: -1, 2048 bytes)
[    0.000000] Dentry cache hash table entries: 16384 (order: 4, 65536
bytes)
[    0.000000] Inode-cache hash table entries: 8192 (order: 3, 32768
bytes)
[    0.000000] Writing ErrCtl register=00004100
[    0.000000] Readback ErrCtl register=00004100
[    0.000000] Memory: 118548k/130048k available (4128k kernel code,
11500k reserved, 1070k data, 4776k init, 0k highmem)
[    0.000000] NR_IRQS:256
[    0.000000] Setting up vectored interrupts
[    0.000000] CPU Clock: 600MHz
[    0.000000] Calibrating delay loop... 397.82 BogoMIPS (lpj=795648)
[    0.032000] pid_max: default: 32768 minimum: 301
[    0.036000] Mount-cache hash table entries: 512
[    0.044000] pinctrl core: initialized pinctrl subsystem
[    0.048000] NET: Registered protocol family 16
[    0.060000] dma-xway 1e104100.dma: Init done - hw rev: 8, ports: 5,
channels: 24
[    0.068000] pinctrl-xway 1e100b10.pinmux: Init done
[    0.072000] Init done
[    0.072000] gpio-stp-xway 1e100bb0.stp: Reserved = 0x00000000
[    0.076000] gpio-stp-xway 1e100bb0.stp: edge = 67108864, groups = 3,
dsl = 0
[    0.080000] gpio-stp-xway 1e100bb0.stp: phy1 = 0, phy2 = 0, phy3 = 0,
phy4 = 0
[    0.084000] gpio-stp-xway 1e100bb0.stp: Init done
[    0.088000] gpio-stp-xway 1e100bb0.stp: AR = 0x00000000
[    0.092000] gpio-stp-xway 1e100bb0.stp: CPU0 = 0x000000ff
[    0.096000] gpio-stp-xway 1e100bb0.stp: CPU1 = 0x00000000
[    0.100000] gpio-stp-xway 1e100bb0.stp: CON0 = 0x84008000
[    0.104000] gpio-stp-xway 1e100bb0.stp: C0N1 = 0x81000003
[    0.108000] !!!!!!! WAVE400 system registeration on AHB
[    0.112000] MTLK_MEM_BAR1_START is 1a000000
[    0.116000] MTLK_MEM_BAR1_END is 1a7fffff
[    0.120000] MTLK_WIRELESS_IRQ_IN_INDEX is 26
[    0.124000] dcdc-xrx200 1f106a00.dcdc: Core Voltage : 0 mV
[    0.736000] pcie_wait_phy_link_up port 1 timeout
[    1.248000] pcie_wait_phy_link_up port 1 timeout
[    1.760000] pcie_wait_phy_link_up port 1 timeout
```

```
[    1.764000] pcie_rc_initialize port 1 link up failed!!!!!
[    1.768000] Lantiq PCIe Root Complex Driver - 2.0.3
[    1.772000] Copyright(c) 2009 - 2013 LANTIQ DEUTSCHLAND GMBH
[    1.796000] bio: create slab <bio-0> at 0
[    1.800000] SCSI subsystem initialized
[    1.804000] usbcore: registered new interface driver usbfs
[    1.808000] usbcore: registered new interface driver hub
[    1.812000] usbcore: registered new device driver usb
[    1.816000] NET: Registered protocol family 8
[    1.820000] NET: Registered protocol family 20
[    1.824000] Switching to clocksource MIPS
[    1.828000] NET: Registered protocol family 2
[    1.836000] TCP established hash table entries: 1024 (order: 1, 8192
bytes)
[    1.840000] TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
[    1.848000] TCP: Hash tables configured (established 1024 bind 1024)
[    1.856000] TCP: reno registered
[    1.856000] UDP hash table entries: 256 (order: 0, 4096 bytes)
[    1.864000] UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
[    1.872000] NET: Registered protocol family 1
[    6.348000] gptu: totally 6 16-bit timers/counters
[    6.352000] gptu: misc_register on minor 63
[    6.356000] gptu: succeeded to request irq 126
[    6.360000] gptu: succeeded to request irq 127
[    6.364000] gptu: succeeded to request irq 128
[    6.368000] gptu: succeeded to request irq 129
[    6.372000] gptu: succeeded to request irq 130
[    6.376000] gptu: succeeded to request irq 131
[    6.384000] vpe1_mem = 0
[    6.388000] Wired TLB entries for Linux read_c0_wired() = 0
[    6.396000] squashfs: version 4.0 (2009/01/31) Phillip Lougher
[    6.400000] jffs2: version 2.2 (NAND) (SUMMARY) (LZMA) (RTIME)
(CMODE_PRIORITY) (c) 2001-2006 Red Hat, Inc.
[    6.412000] msgmni has been set to 231
[    6.416000] io scheduler noop registered
[    6.420000] io scheduler deadline registered (default)
[    6.428000] lantiq,asc 1e100c00.serial: pins are not configured from
the driver
[    6.436000] 1e100c00.serial: ttyLTQ0 at MMIO 0x1e100c00 (irq = 112)
is a lantiq,asc
[    6.452000] console [ttyLTQ0] enabled, bootconsole disabled
[    6.452000] console [ttyLTQ0] enabled, bootconsole disabled
```

```
[    6.464000] loop: module loaded
[    6.472000] NAND device: Manufacturer ID: 0xc8, Chip ID: 0xd1
(Unknown NAND 128MiB 3,3V 8-bit), 128MiB, page size: 2048, OOB size: 64
[    6.480000] Scanning device for bad blocks
[    6.524000] 5 ofpart partitions found on MTD device 14000000.nand-
parts
[    6.528000] Creating 5 MTD partitions on "14000000.nand-parts":
[    6.536000] 0x000000000000-0x000000080000 : "bootstrap"
[    6.544000] 0x000000080000-0x000000180000 : "bootloader"
[    6.548000] 0x000000180000-0x000000200000 : "reserved_1"
[    6.552000] 0x000000200000-0x000000280000 : "reserved_2"
[    6.560000] 0x000000280000-0x000007f80000 : "system_sw"
[    6.568000] IMQ driver loaded successfully. (numdevs = 3, numqueues =
1)
[    6.576000]     Hooking IMQ after NAT on PREROUTING.
[    6.580000]     Hooking IMQ after NAT on POSTROUTING.
[    6.588000] Lantiq VRX318 Version 2.0.0
[    6.588000] LTQ ETH SWITCH API, Version 2.0.1.
[    6.592000] SWAPI: Registered char device [switch_api] with major no
[81]
[    6.600000] Switch API: PCE MicroCode loaded !!
[    6.604000] gphy_driver_init: fw_mode:11G-FW, no of phys:4, mode:0
[    6.612000] PPP generic driver version 2.4.2
[    6.616000] PPP MPPE Compression module registered
[    6.620000] NET: Registered protocol family 24
[    6.624000] res = 87908f00
[    6.628000] wdt 1f8803f0.watchdog: Init done
[    6.632000] leds-gpio gpio-leds.13: pins are not configured from the
driver
[    6.644000] Lantiq DEU driver version 2.0.0
[    6.648000] LTQ DEU DES initialized.
[    6.652000] LTQ DEU AES initialized.
[    6.652000] LTQ DEU ARC4 initialized
[    6.656000] LTQ DEU SHA1 initialized
[    6.660000] LTQ DEU MD5 initialized
[    6.664000] LTQ DEU SHA1_HMAC initialized
[    6.668000] LTQ DEU MD5_HMAC initialized
[    6.672000] DEU driver initialization complete!
[    6.676000] u32 classifier
[    6.680000]     input device check on
[    6.684000]     Actions configured
[    6.684000] nf_conntrack version 0.5.0 (1852 buckets, 7408 max)
```

```
[    6.692000] xt_time: kernel timezone is -0000
[    6.696000] ipip: IPv4 over IPv4 tunneling driver
[    6.704000] ip_tables: (C) 2000-2006 Netfilter Core Team
[    6.708000] TCP: cubic registered
[    6.708000] Initializing XFRM netlink socket
[    6.712000] NET: Registered protocol family 10
[    6.720000] NET: Registered protocol family 17
[    6.724000] NET: Registered protocol family 15
[    6.728000] Bridge firewalling registered
[    6.732000] Ebtables v2.0 registered
[    6.736000] lec:lane_module_init: lec.c: initialized
[    6.740000] mpoa:atm_mpoa_init: mpc.c: initialized
[    6.744000] KOAM is loaded successfully.
[    6.748000] 8021q: 802.1Q VLAN Support v1.8
[    6.756000] UBI: attaching mtd4 to ubi0#
[    6.968000] UBI: scanning is finished
[    6.984000] UBI: attached mtd4 (name "system_sw", size 125 MiB) to
ubi0
[    6.992000] UBI: PEB size: 131072 bytes (128 KiB), LEB size: 129024
bytes
[    6.996000] UBI: min./max. I/O unit sizes: 2048/2048, sub-page size
512
[    7.004000] UBI: VID header offset: 512 (aligned 512), data offset:
2048
[    7.012000] UBI: good PEBs: 1000, bad PEBs: 0, corrupted PEBs: 0
[    7.016000] UBI: user volume: 2, internal volumes: 1, max. volumes
count: 128
[    7.024000] UBI: max/mean erase counter: 2/0, WL threshold: 4096,
image sequence number: 768042961
[    7.032000] UBI: available PEBs: 0, total reserved PEBs: 1000, PEBs
reserved for bad PEB handling: 20
[    7.044000] UBI: background thread "ubi_bgt0d" started, PID 326
[    7.076000] Freeing unused kernel memory: 4776K (80516000 - 809c0000)
[    7.104000] input: gpio-keys-polled.10 as /devices/gpio-keys-
polled.10/input/input0
[    7.116000] usbcore: registered new interface driver usb-storage
[   11.408000] IFXOS, Version 1.5.92 (c) Copyright 2009, Lantiq
Deutschland GmbH
[   11.416000] ip6_tables: (C) 2000-2006 Netfilter Core Team
[   11.432000] i2c /dev entries driver
[   11.436000] i2c-gpio i2c.14: using pins 211 (SDA) and 209 (SCL)
[   11.444000] switch_module_init(): Module initializing...
```

```
[   11.448000] read_physical_2_logical_lan_ports_mapping(): lan-ports-
mapping <2, 4, 1, 3>
[   11.456000] switch_module_init(): External PHY present!
[   11.480000] phy_get_identifier(): PHY identifier: 321
[   11.484000] External PHY present id = 321
[   11.508000] switch_module_init(): Module initialized...
[   11.516000] hidraw: raw HID events driver (C) Jiri Kosina
[   13.176000] Loading D5 (MII0/1) driver ......
[   13.180000]
[   13.180000] Cannot find wlanm
[   13.204000] CHIPID: 1, chipid address: 0xbf107344
[   13.208000] Succeeded!
[   13.212000] PPE datapath driver info:
[   13.212000]   Version ID: 128.3.3.1.0.0.3
[   13.212000]   Family    : AR10
[   13.212000]   DR Type   : Normal Data Path | Indirect-Fast Path
[   13.212000]   Interface : MII0 | MII1
[   13.212000]   Mode      : Routing
[   13.212000]   Release   : 0.0.3
[   13.236000] PPE firmware info:
[   13.236000]   Version ID: 10.5.2.16.1
[   13.236000]   Family    : GRX390
[   13.236000]   FW Package: D5
[   13.236000]   Release   : 2.16.1
[   13.236000] PPE firmware feature:
[   13.236000]   Packet Acceleration            Support
[   13.236000]   IPv4                           Support
[   13.236000]   IPv6                           Support
[   13.236000]   6RD                            Support
[   13.236000]   DS-Lite                        Support
[   13.364000] PPA API --- init successfully
[   14.956000] UBIFS: mounted UBI device 0, volume 1, name "data", R/O
mode
[   14.960000] UBIFS: LEB size: 129024 bytes (126 KiB), min./max. I/O
unit sizes: 2048 bytes/2048 bytes
[   14.968000] UBIFS: FS size: 120250368 bytes (114 MiB, 932 LEBs),
journal size 9033728 bytes (8 MiB, 71 LEBs)
[   14.980000] UBIFS: reserved for root: 0 bytes (0 KiB)
[   14.984000] UBIFS: media format: w4/r0 (latest is w4/r0), UUID
648017B7-983B-4FE2-9327-15032C0F2A06, small LPT model
[   15.676000] gphy-fw gphy-fw.8: proc_write_phy_fw:   Found:VR9 V1.2
GPHY GE  FW
```

```
[   15.688000] gphy-fw gphy-fw.8: booting GPHY0 firmware at 5CE0000 for
GRX390
[   15.692000] gphy-fw gphy-fw.8: booting GPHY1 firmware at 5CE0000 for
GRX390
[   15.700000] gphy-fw gphy-fw.8: booting GPHY2 firmware at 5CE0000 for
GRX390
[   15.708000] gphy-fw gphy-fw.8: booting GPHY3 firmware at 5CE0000 for
GRX390
[   15.712000] ltq_gphy_firmware_config: fw_mode:11G-FW, no of
phys:4,data_ptr:5CE0000
[   19.564000] device eth1 entered promiscuous mode
```

From this boot loader/kernel boot log we can gather many more details about the ONT's hardware:

- CPU: MIPS 34Kc @ 600 MHz, 397.82 BogoMIPS

There is the following MTD partitions:

```
Creating 5 MTD partitions on "14000000.nand-parts":
0x000000000000-0x000000080000 : "bootstrap"
0x000000080000-0x000000180000 : "bootloader"
0x000000180000-0x000000200000 : "reserved_1"
0x000000200000-0x000000280000 : "reserved_2"
0x000000280000-0x000007f80000 : "system_sw"
```

One of these partitions is attached as a UBI partition:

```
UBI: attached mtd4 (name "system_sw", size 125 MiB) to ubi0
```

Then later one volume is mounted:

```
UBIFS: mounted UBI device 0, volume 1, name "data", R/O mode
```

There was also the following interesting console output:

```
Press the [f] key and hit [enter] to enter failsafe mode
Press the [1], [2], [3] or [4] key and hit [enter] to select the debug
```

```
level
[...]
geneos login:
```

So you can boot into a “failsafe” mode (can we exploit that?!), and you can set the debug level.

At the end there is a login prompt… But how to get in???

Ok, I tried the “failsafe” mode, and look what I got:

```
f
- failsafe -
/etc/preinit: line 1: telnetd: not found


BusyBox v1.22.1 (2015-12-01 15:47:20 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.

ash: can't access tty; job control turned off
  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 BARRIER BREAKER (14.07, unknown)
 -----------------------------------------------------
  * 1/2 oz Galliano         Pour all ingredients into
  * 4 oz cold Coffee        an irish coffee mug filled
  * 1 1/2 oz Dark Rum       with crushed ice. Stir.
  * 2 tsp. Creme de Cacao
 -----------------------------------------------------
root@(none):/#
```

This looks very familiar… 😀

So the firmware is very obviously based on OpenWrt 14.07, codenamed “Barrier Breaker” (with a device target of “lantiq/generic”)… I think I need to write the Genexis guys a nice email, asking for the source code… 😀

Anyway, let’s continue:

```
root@(none):/# cat /etc/passwd
root:x:0:0:root:/root:/bin/ash
operator:x:0:0:Operator:/root:/usr/bin/oxsh
admin:x:25197:25197:End User:/var:/bin/false
daemon:x:1:1:daemon:/var:/bin/false
ftp:x:55:55:ftp:/home/ftp:/bin/false
network:x:101:101:network:/var:/bin/false
nobody:x:65534:65534:nobody:/var:/bin/false
root@(none):/# cat /etc/shadow
root:!:0:0:99999:7:::
operator:$6$FardvCZyI71$Uxu5a/76M8LMeaubaNqdGb
/3/oMn7Dmmj2THQrV6bWaysO2tKACck3kRkEJgeTI8rkMn4xUHDxXAoXC2E7L580:0:0:999
99:7:::
admin:!:0:0:99999:7:::
daemon:*:0:0:99999:7:::
ftp:*:0:0:99999:7:::
network:*:0:0:99999:7:::
nobody:*:0:0:99999:7:::
```

Ok, so I have to log in as “operator”… What if I change the password for that user, and try to boot into multi-user mode?

Duh, that didn’t work out… Could change the password, but not continue to boot into multi-user mode… When I rebooted the router the password I chose didn’t work…

But wait, often it’s “admin” as the login, and “admin” as the password… Now the login is “operator”, so why not try “operator” as the password:

```
geneos login: operator
Password:
Genexis Operating System (GeneOS)
Copyright (c) 2014-2015 Genexis B.V. All rights reserved.
GeneOS version: geneos-polar-2.1.0-R
geneos#
```

Oh joy, I did it!!! 😀

Naive as I am I expected to have a full-blown OpenWrt… But not so… Not even “help” worked… But pressing “?” *did* work… 🙂

```
geneos#
  configure   Enter configuration mode
  copy        Copy from one file to another
  ping        Send ICMP echo requests
  quit        Exit shell
  reload      Reload system
  show        Show running system information
  write       Write running configuration
```

Wait… This somehow looks familiar… Like Cisco's IOS?!

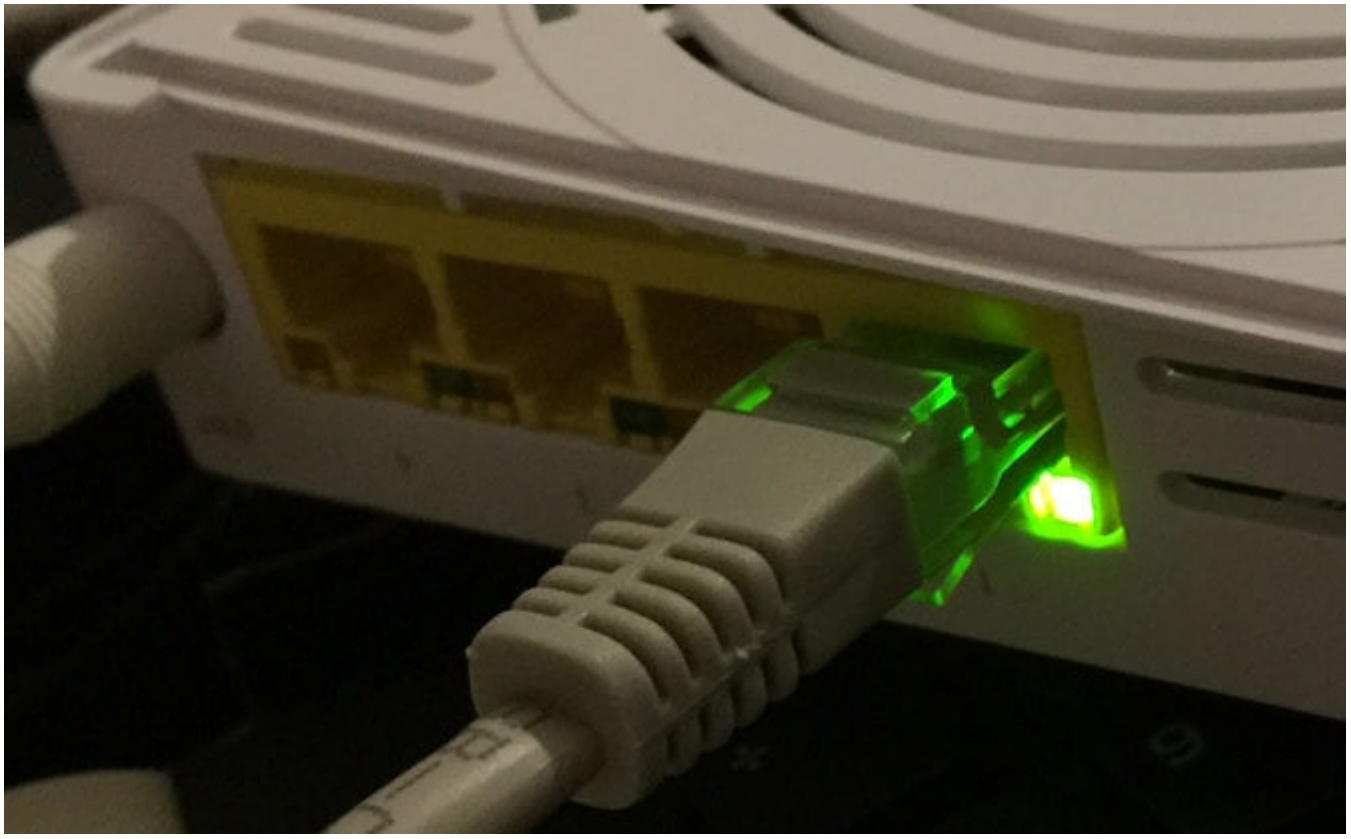
```
geneos# show <press "?">
  clock           Show system clock
  cwmp            Show CWMP information
  dhcp            DHCP information
  history         Show command line history
  interface       Interface information
  logging         Log messages
  running-config  Show running configuration
  tech-support    Show information for Technical Support
  version         Show version info
```

Yes, that's right. So it should be fairly easy to fiddle with this thing… 🙂

Let's first try to bring the Ethernet interfaces up, which are administratively down by default:

```
geneos# conf term
 geneos(config)#
 geneos(config)# interface lan/ethernet1
 geneos(config-if-lan-eth)# no shutdown
 geneos(config-if-lan-eth)# exit
 geneos(config)# interface lan/ethernet2
 geneos(config-if-lan-eth)# no shutdown
 geneos(config-if-lan-eth)# exit
 geneos(config)# interface lan/ethernet3
 geneos(config-if-lan-eth)# no shutdown
 geneos(config-if-lan-eth)# exit
 geneos(config)# interface lan/ethernet4
 geneos(config-if-lan-eth)# no shutdown
 geneos(config-if-lan-eth)# exit
```

Ok, now let's connect a LAN cable that's connected to my laptop… Ok, that look's great:



Port is now up, "link" LED is lit.

From the default config options I can figure out that the WAN interface acts as a DHCP client. So my guess that they do port-based security seems to be true… As I have a dedicated fiber into the PoP this is not a security risk… Someone would have to physically connect their fiber to my fiber that comes from the PoP in order to impersonate me… Doesn't sound very easy…

Let's continue...

```
geneos# show tech-support
 --------------- show logging level debugging ---------------
 [...]
 Dec  1 14:50:32 geneos local1.info mgmt-agent[871]:
usp.product.prodname = 'FiberTwist-P2410', length = 16
 [...]
 Dec  1 14:50:32 geneos local1.info mgmt-agent[871]: [truncated]
usp.dropbear.rsakey = 'AAAAB3NzaC1yc2EAAAA...
```

Duh... *This* I don't like... So they can remotely log into my router?! Boooo!!!

Ok, maybe modifying the boot image brings us any further... But how to extract it??? Let's boot into failsafe mode again... After a while of playing around I figured out how to do it:

```
## Sends firmware to external VR9 PHY
# unlzma -c /etc/gphy/gphy_firmware.img.lzma > /proc/driver/ltq_gphy
/phyfirmware
# cd /lib/modules/3.10.12/
# modprobe ltqmips_ppe_drv.ko
# ifconfig eth0 192.168.2.43
## Mount volume from flash ROM
# mount -t ubifs /dev/ubi0_1 /mnt/
# cd /mnt/fw/0
# md5sum -b geneos-polar-2.1.0-R.img
## On remote side (your PC?): netcat -l 9999 >geneos-polar-2.1.0-R.img
# cat geneos-polar-2.1.0-R.img | nc 192.168.2.10 9999
## On remote side check that FW image is not corrupted
# md5sum -b geneos-polar-2.1.0-R.img
```

Ok, now we have a copy of the "Polar" (which is the platform name) boot image. From the boot log we can tell this is a "FIT Image." But what is that? It's the "Flattened Image Tree" for the U-Boot boot loader. There's a `dumpimage` tool available to unpack these images, so let's play with it...

```
$ mkimage -l geneos-polar-2.1.0-R.img
FIT description: Image tree for Polar platform products.
Created: Tue Dec 1 15:53:16 2015
 Image 0 (kernel@1)
 Description: Generic initramfs
```

```
 Created: Tue Dec 1 15:53:16 2015
 Type: Kernel Image
 Compression: lzma compressed
 Data Size: 6535067 Bytes = 6381.90 kB = 6.23 MB
 Architecture: MIPS
 OS: Linux
 Load Address: 0x80002000
 Entry Point: 0x80002000
 Hash algo: sha1
 Hash value: 42bd16e172686233005096bde4abefe44bcf566b
 Image 1 (fdt@1)
 Description: Genexis Polar FDT blob
 Created: Tue Dec 1 15:53:16 2015
 Type: Flat Device Tree
 Compression: uncompressed
 Data Size: 10482 Bytes = 10.24 kB = 0.01 MB
 Architecture: MIPS
 Hash algo: sha1
 Hash value: 5050dde93e7d83b3c5339da2b8e9cdf227f44658
 Default Configuration: 'conf@1'
 Configuration 0 (conf@1)
 Description: Configuration for all Polar variants
 Kernel: kernel@1
 FDT: fdt@1
```

So there's two images in it, plus a common config. Let's extract the images… To do so you must build the U-boot tools:

```
git clone git://git.denx.de/u-boot.git
cd u-boot
make O=sandbox sandbox_config
make O=sandbox
```

Then you have the tool we require in `sandbox/tools/dumpimage`.

To extract the two images from the FIT image do the following:

```
dumpimage -i geneos-polar-2.1.0-R.img -T flat_dt -p 0
kernel.initramfs.lzma
dumpimage -i geneos-polar-2.1.0-R.img -T flat_dt -p 1 fdt.img
```

At this point I'm currently stuck. I can un-lzma the initramfs file, but I cannot find out how to un-pack the resulting file. Any idea?

Similarly to the FIT image I transferred the `config.db` file from `/mnt/config/`. This is a SQLite3 file that can easily be viewed and edited.

Now back to the boot loader. The boot loader seems to be U-Boot. If you're quick and press any key within a second you can interrupt auto boot, and you will be in the boot loader's command line:

```
GRX330 #
GRX330 # help
?       - alias for 'help'
base    - print or set address offset
bootm   - boot application image from memory
bootp   - boot image via network using BOOTP/TFTP protocol
chpart  - change active partition
cmp     - memory compare
cp      - memory copy
crc32   - checksum calculation
echo    - echo args to console
fdt     - flattened device tree utility commands
go      - start application at address 'addr'
help    - print command description/usage
loadb   - load binary file over serial line (kermit mode)
loady   - load binary file over serial line (ymodem mode)
loop    - infinite loop on address range
md      - memory display
mm      - memory modify (auto-incrementing address)
mtdparts- define flash/nand partitions
mtest   - simple RAM read/write test
mw      - memory write (fill)
nand    - NAND sub-system
nboot   - boot from NAND device
nm      - memory modify (constant address)
ping    - send ICMP ECHO_REQUEST to network host
printenv- print environment variables
rarpboot- boot image via network using RARP/TFTP protocol
reset   - Perform RESET of the CPU
run     - run commands in an environment variable
setenv  - set environment variables
tftpboot- boot image via network using TFTP protocol
```

```
ubi     - ubi commands
ubifs_genload- load file from an UBIFS filesystem
ubifsload- load file from an UBIFS filesystem
ubifsls - list files in a directory
ubifsmount- mount UBIFS volume
upgrade - upgrade - forward/backward copy memory to pre-defined flash
location
version - print monitor version
GRX330 #
```

I think it would be nice to create an image of the current system on an NFS server, modify it, and boot from there… This way I can't brick the device, and still play with it… 🙂

For reference purposes here's the output of `printenv`:

```
GRX330 # printenv
bootcmd=run flash_flash
bootdelay=1
baudrate=115200
preboot=echo;echo Type \"run flash_nfs\" to mount root filesystem over
NFS;echo
bootfile="uImage"
mem=118M
phym=128M
wlanm=119M
ipaddr=192.168.1.1
serverip=192.168.1.2
ethaddr=00:E0:92:XX:XX:XX
netdev=eth0
console=ttyLTQ0
tftppath=
loadaddr=0x83000000
rootpath=/mnt/full_fs
rootfsmtd=/dev/mtdblock3
nfsargs= setenv bootargs ubi.mtd=system_sw root=/dev/nfs rw
nfsroot=$(serverip):$(rootpath)
ramargs=setenv bootargs root=/dev/ram rw
addip=setenv bootargs $(bootargs)
ip=$(ipaddr):$(serverip):$(gatewayip):$(netmask):$(hostname):$(netdev):o
n
flash_nfs=run nfsargs addip addmisc;bootm $(kernel_addr)
```

```
net_nfs=tftp $(loadaddr) $(tftppath)$(bootfile);run nfsargs addip
addmisc;bootm
net_flash=tftp $(loadaddr) $(tftppath)$(bootfile); run flashargs addip
addmisc; bootm
net_ram=tftp $(loadaddr) $(tftppath)$(bootfile); run ramargs addip
addmisc; bootm
u-boot=u-boot.ltq
rootfs=rootfs.img
firmware=firmware.img
fullimage=fullimage.img
totalimage=totalimage.img
load=tftp $(loadaddr) $(u-boot)
update=protect off 1:0-2;era 1:0-2;cp.b $(loadaddr) B0000000 $(filesize)
flashargs=setenv bootargs ubi.mtd=system_sw
flash_flash=run flashargs addmisc;ubi part system_sw;ubifsmount
data;setenv bootargs $(bootargs) fw_number=0;ubifs_genload $(loadaddr)
fw/0/;bootm $(loadaddr);setenv bootargs $(bootargs) fw_number=1;
ubifs_genload $(loadaddr) fw/1/;bootm $(loadaddr)
update_nandboot=tftp $(loadaddr) $(tftppath)u-boot-nand.bin;nand erase 0
17FFFF;nand erase 1C0000 31FFFFF;nand write.partial $(loadaddr) 0
$(filesize)
ubi_init=setenv kernelA_id 0;setenv rootfsA_id 1;setenv firmwareA_id
2;setenv kernelB_id 3;setenv rootfsB_id 4;setenv firmwareB_id 5;setenv
setbank check_image$(update_chk);run $(setbank);ubi part system_sw
update_chk=0
switchbankA=setenv active_bank A;setenv kernel_id $(kernelA_id);setenv
rootfs_id $(rootfsA_id);setenv f_kernel_size f_kernel_sizeA;setenv
kernel_vol kernelA;setenv rootfs_vol rootfsA;setenv firmware_vol
firmwareA;setenv rootfsname rootfsA
switchbankB=setenv active_bank B;setenv kernel_id $(kernelB_id);setenv
rootfs_id $(rootfsB_id);setenv f_kernel_size f_kernel_sizeB;setenv
kernel_vol kernelB;setenv rootfs_vol rootfsB;setenv firmware_vol
firmwareB;setenv rootfsname rootfsB
check_image0=run switchbankA
check_image1=run switchbankB;setenv update_chk 0;save
check_image2=run switchbankB
check_image3=run switchbankA;setenv update_chk 2;save
update_uboot=tftp $(loadaddr) $(tftppath)$(u-boot); nand write.partial
$(loadaddr) 0x4000 $(filesize);reset
update_kernel=run ubi_init;tftpboot $(loadaddr)
$(tftppath)$(bootfile);run switchbankB;upgrade $(loadaddr)
$(filesize);run switchbankA;set update_chk 0;upgrade $(loadaddr)
```

```
$(filesize)
update_bootloader=update_uboot
update_rootfs=run ubi_init;tftpboot $(loadaddr) $(tftppath)$(rootfs);run
switchbankB;upgrade $(loadaddr) $(filesize);run switchbankA;set
update_chk 0;upgrade $(loadaddr) $(filesize)
update_firmware=run ubi_init;tftpboot $(loadaddr)
$(tftppath)$(firmware);run switchbankB;upgrade $(loadaddr)
$(filesize);run switchbankA;set update_chk 0;upgrade $(loadaddr)
$(filesize)
update_fullimage=run ubi_init;tftpboot $(loadaddr)
$(tftppath)$(fullimage);run switchbankB;upgrade $(loadaddr)
$(filesize);run switchbankA;set update_chk 0;upgrade $(loadaddr)
$(filesize)
update_totalimage=run ubi_init;tftpboot $(loadaddr)
$(tftppath)$(totalimage);upgrade $(loadaddr) $(filesize)
reset_uboot_config=nand erase $(f_ubootconfig_addr)
$(f_ubootconfig_range)
reset_ddr_config=nand write.partial 80400000 $(f_ddrconfig_addr)
$(f_ddrconfig_size)
reset_sysconfig=run ubi_init;ubi remove sysconfig;ubi remove
sysconfigA;ubi remove sysconfigB
mtdparts=mtdparts=ifx_nand:512k(bootstrap),1m(bootloader),512k(reserved_
1),512k(reserved_2),125m(system_sw),-(bbt)
part0_begin=0x00000000
part1_begin=0x00040000
part2_begin=0x000C0000
part3_begin=0x002C0000
part4_begin=0x07000000
part5_begin=0x07040000
part6_begin=0x07080000
total_part=7
flash_end=0x07FFFFFF
data_block0=uboot
data_block1=firmware
data_block2=kernel
data_block3=rootfs
data_block4=sysconfig
data_block5=ubootconfig
data_block6=dectconfig
total_db=7
f_uboot_addr=0x00000000
f_uboot_size=0
```

```
f_ubootconfig_addr=0x100000
f_ubootconfig_size=0x4000
f_ubootconfig_end=0x07040FFF
f_ubootconfig_range=0x80000
f_gphy_firmware_addr=IFX_CFG_FLASH_GPHY_FIRMWARE_IMAGE_START_ADDR
f_gphy_firmware_size=IFX_CFG_FLASH_GPHY_FIRMWARE_IMAGE_SIZE
f_gphy_firmware_end=IFX_CFG_FLASH_GPHY_FIRMWARE_IMAGE_END_ADDR
f_kernel_addr=0x000C0000
f_kernel_size=0
f_kernel_end=IFX_CFG_FLASH_KERNEL_IMAGE_END_ADDR
f_rootfs_addr=0x002C0000
f_rootfs_size=0
f_rootfs_end=IFX_CFG_FLASH_ROOTFS_IMAGE_END_ADDR
f_firmware_addr=0x00040000
f_firmware_size=0
f_fwdiag_addr=IFX_CFG_FLASH_FIRMWARE_DIAG_START_ADDR
f_fwdiag_size=IFX_CFG_FLASH_FIRMWARE_DIAG_SIZE
f_sysconfig_addr=0x07000000
f_sysconfig_size=0x10000
f_dectconfig_addr=0x07080000
f_dectconfig_size=0x400
f_wlanconfig_addr= IFX_CFG_FLASH_WLAN_CFG_START_ADDR
f_wlanconfig_size=IFX_CFG_FLASH_WLAN_CFG_SIZE
f_ddrconfig_addr=0x00003fe0
f_ddrconfig_size=32
f_ddrconfig_end=0x00003fff
stdin=serial
stdout=serial
stderr=serial
ver=U-Boot-2010.06-LANTIQ-v-2.3.08
ethact=ar10 Switch
addmisc=setenv bootargs $(bootargs) console=$(console),$(baudrate)
init=/etc/preinit  bootstrap_ver="bootstrap-polar-2.1.0-R"
bootloader_ver="bootloader-polar-2.1.0-R"
mtdids=nand0=ifx_nand
partition=nand0,0
mtddevnum=0
mtddevname=bootstrap

Environment size: 5407/16380 bytes
```

BTW, there's also JP2 with 10 pads which look like it could be two USB ports (4 each plus a spare

each?). This guess is backed by the fact that the onboard Linux has USB support... 🙂

FIBER GLASFASER HACKING ONT

16 THOUGHTS ON “HACKING THE GENEXIS FIBERTWIST-P2410”

Gero

2016-03-30 AT 18:22

Respekt! Sehr interessant Ralf.

★ Ralf Bergs

2016-04-06 AT 09:45

Danke. Macht Spaß so eine Kiste aufzumachen, passiert selten genug heutzutage dass ich Zeit und Gelegenheit dazu habe...

Wenne

2016-06-01 AT 19:52

Kannst du auch einen Genexis Titanium in den Brigdmodus bringen ?

Ralf Bergs

2016-07-05 AT 07:56

Weiß ich nicht... Ich bin aber auch kein Auftragshacker... 🙂

Ich mache sowas nur, weil es mir Spaß macht und spannend ist... Außerdem ist das relativ zeitaufwändig, deshalb habe ich auch schon seit einer ganzen Weile keine Zeit mehr gehabt mich dieser Sache zu widmen, und es ist auch nicht absehbar wann ich wieder Zeit haben werde...

Heiko Rintelen

2016-09-28 AT 23:12

Hallo! Würde gern an so einen Genexis 2410 von der Deutschen Glasfaser (Helinet) einen ganz normalen Router mit VOIP anschließen, einen Speedling 5510 von Zyxel. Den bekomme ich nur nicht so richtig konfiguriert bislang, beim Profil “Hinter einem Router” holt er sich wenigstens schon eine IP.. Hat da jemand Erfahrungen? Normalerweise verwenden die die FritzBox, die sich irgendwie selbst konfiguriert,

aber der Support kann mir da nichts genaues sagen..

Ralf Bergs

2016-10-01 AT 11:04

Was genau ist Dein Problem?

Ich glaube verstanden zu haben, dass Du wenigstens schon surfen kannst mit Deinem Zyxel-Router, richtig? Also brauchst Du dann nur noch den VoIP/SIP-Account im Zyxel-Router zu konfigurieren. Hast Du denn die Zugangsdaten Deines SIP-Accounts? Die musst Du ansonsten bei Deinem Provider abfragen. Dann sollte es kein Problem sein, die im Router zu konfigurieren…

Viele Grüße,

Ralf

Wolfgang Poppelreuter

2016-11-26 AT 10:44

Hallo,

ich habe hier hinter einem DG FiberTwist einen Openwrt Router laufen. Der DG Sip Account läuft jetzt, wenn auch nach einigem probieren, erfolgreich auf Asterisk Anlage.

Grüße,

Wolfgang

Ralf Bergs

2016-12-01 AT 14:56

Hallo Wolfgang.

Warum sollte das auch nicht funktionieren? Das FiberTwist ist ja genau für den Fall da. Sicherlich, es kann auch selbst als Router fungieren, aber die DG sieht ja i. d. R. einen dedizierten Router vor (entweder den eigenen DG-Router, oder einen Kundenrouter). Daher wird der ONT quasi nur als Bridge/Medienkonverter verwendet…

Bei uns ist immer noch kein Licht im PoP… Hoffentlich bald…

Viele Grüße,

Ralf

Huy

2017-03-07 AT 15:16

Hi Ralf,

grandiose Arbeit! Bei uns in Rommerskirchen (Süd) wurde jetzt erst mit den Tiefbauarbeiten begonnen. Sobald ich den FiberTwist in den Händen habe werde ich mir den Serial-Port auch mal näher anschauen. Du hast da ja schon einiges an Vorarbeit geleistet 🙂

VG aus Rommerskirchen.

Huy

★ Ralf Bergs

2017-03-13 AT 13:05

Hallo Huy,

Danke. 🙂

Leider habe ich im Moment keine Zeit mehr, mich diesem Thema zu widmen… Unsere Leitung ist ja schon fast drei Monate in Betrieb… Wäre schon cool da noch ein paar Statistiken etc. “rauszukitzeln”… 😉

Viele Grüße nach Roki,

Ralf

Peer

2021-10-02 AT 13:07

Nice work! We just got a Fibertwist connection in our home.
The provider has P2420 as a standard setup, but I adked for the F2120 twistpad to directly connect to the sfp port of my mikrotik router.
(was not easy to get this)
Which settings is the P2420 using? I understood that it is symply acting as DHCP client?

---

administrator

2021-10-02 AT 18:28

Hi Peer.

Nice equipment you got there, I'm a bit envious… 🙂

How did you get the ONT, directly from your ISP? Or did you buy it on the open market?

Anyway, the FiberTwist ONT I have functions as a DHCP server to my OpenWrt router "behind", correct. PPPoE is not used.

Does that help?

Kind regards,

Ralf

Peer

2021-10-11 AT 11:56

Hallo Ralf,

als Kunde hat man ja ein Recht auf einen passiven Netzabschluss, was defakto nur mit der F2110 oder F2120 (wenn man CATV nicht nutzt) gegeben ist. Das F2120 hat mir mein Anbieter vorbeigebracht.
Seit 1.10. jetzt bin ich mit der Hotline zugange, das die mir technische Details zum Anschluss geben.
PPPoE User/Passwort SIP account habe ich.
Leider ist auf dem SFP bidi Modul kein PPPoE Server zu finden. Zeitweilig war dort wenigstens ein DHCP Swrver am laufen, der aber nur ne IP ausspuckt, wenn man dem SFP Modul die MAC Adresse des alten P2420 verpasst hat.

administrator

2021-10-19 AT 16:52

Bei was für einem Provider bist Du denn eigentlich?

Ich fürchte, ich kann Dir da leider nicht weiterhelfen. PPPoE wird bei der Deutsche Glasfaser gar nicht eingesetzt.

Hat Dir denn Dein Provider tatsächlich bestätigt, dass da wirklich PPPoE "gesprochen" wird? Ist das "glaubhaft"? Vielleicht war es schlicht ein "überforderter" Hotline-Mitarbeiter, der gar nicht genau Bescheid weiß… Die überwiegende Mehrzahl der Kunden wird sicherlich das Standard-Setup nutzen, so dass man bei abweichenden

Konfigurationen immer wieder Falschauskünfte bekommt (war hier in unserem Neubaugebiet nicht anders…).

Das mit der “ge-fake-ten” MAC-Adresse ist übrigens gar nicht so selten. Das habe ich vor vielen, vielen Jahren schon bei Verwandten in Florida erlebt, die hatten damals noch einen Kabelanschluss… War ein Drama, das ans Laufen zu bekommen…

Viel Erfolg noch!

Sebastian

2022-06-09 AT 09:59

Hallo zusammen,

weiß jemand ob es möglich ist, den Fibertwist so zu konfigurieren, sodass er als Bridge funktioniert. D.h. der Fibertwist der DCHP Service soll ausgeschalten werden und die öffentliche IP soll direkt an den Router weitergereicht werden.

★ Ralf Bergs

2022-06-13 AT 12:16

Hallo Sebastian.

Das musst Du bei der Deutsche Glasfaser beauftragen. Ich habe das genau so am laufen bei mir.

Mein eigener OpenWrt-Router ist direkt an den ONT (Fibertwist) angeschlossen. Das WAN-Interface des Routers bekommt per DHCP vom ONT die (private) IPv4-Adresse (aus dem 100.64.0.0/10-Block für CNG) sowie den öffentlichen Prefix für IPv6.

Viel Erfolg und viele Grüße,

Ralf