Обзор беспроводного терминала HUAWEI FT-8090

Доброго времени суток! Хочу представить всеобщему вниманию беспроводной терминал компании HUAWEI, а именно беглый обзор его аппаратной части и более полный обзор программной. Статья рассчитана на человека, хотя бы немного знакомого с Linux.

Небольшая предыстория

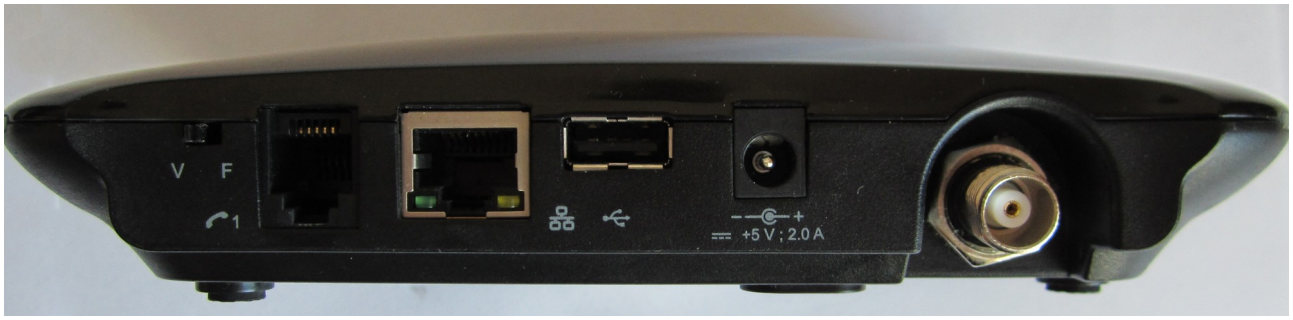
Около года назад я стал обладателем данного девайса, с помощью которого казахстанский провайдер Megaline предоставляет беспроводной доступ к Интернету, а также проводной телефонной сети посредством CDMA. Изначально услуга предоставлялась только абонентам сельской местности, но позже акция распространилась на всех.

Даже при высоком уровне сигнала от местной базовой станции соединение крайне нестабильно. Как мне объяснили, это происходит из-за перегрузки этих самых станций. Но разговор не об этом. Все началось после случайного, но успешного подключения к роутеру по протоколу telnet...

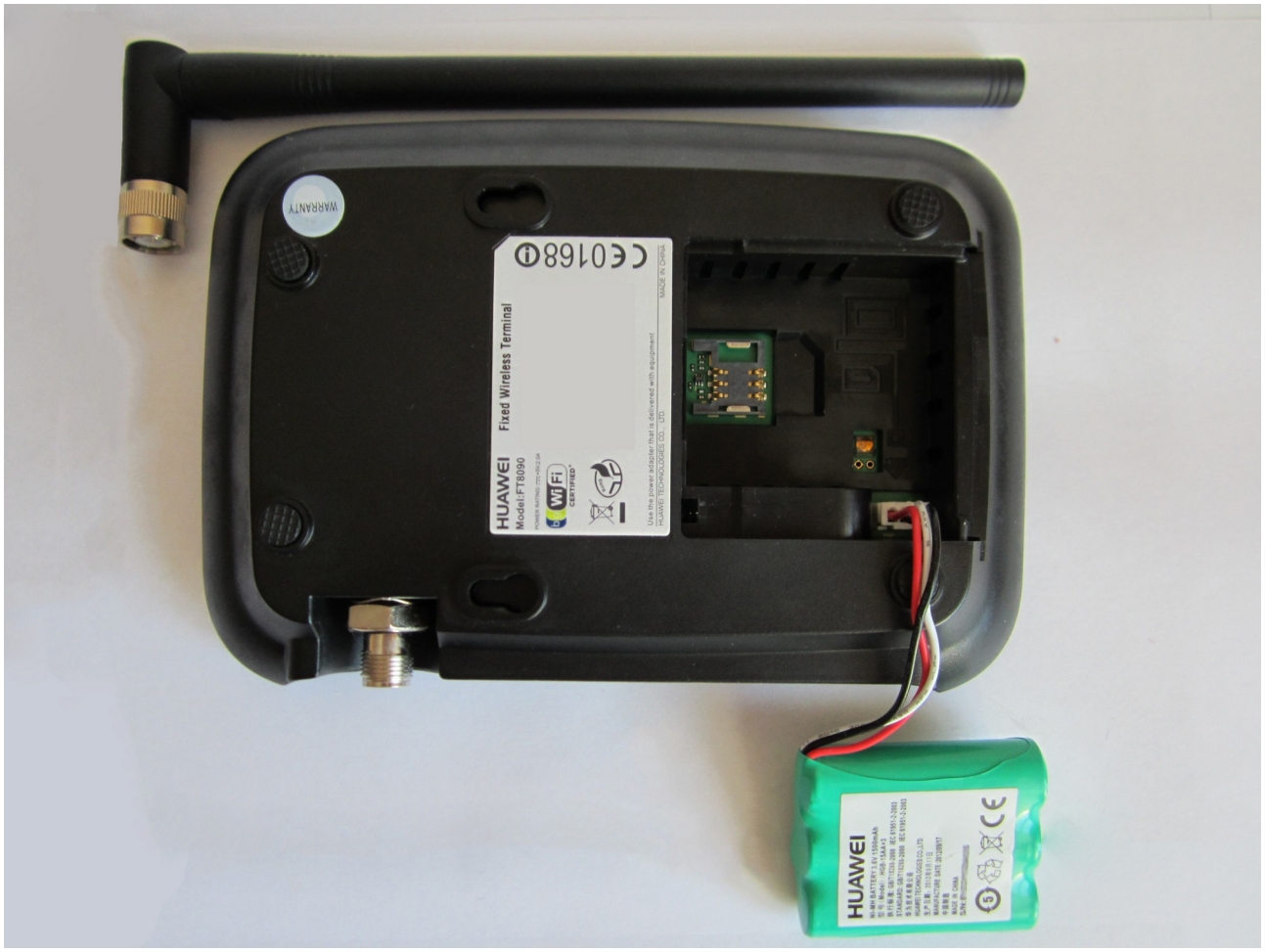
Аппаратная часть

Небольшая черная коробочка со скругленными краями и антенной. На лицевой стороне стандартные для роутеров индикаторы, за исключением индикатора батареи, и кнопка включения/отключения.
Прибор может питаться как от сети, так и от батареи, правда, во втором случае функциональность сильно урезана.

На задней стороне можно найти переключатель режима работы порта RJ-11 (телефон/факс), сам порт RJ-11, порт RJ-45, который я и буду использовать для исследования, USB, разъем питания и разъем крепления антенны.



Внизу небольшой сюрприз :) Открываем нижний отсек, извлекаем батарею и… Слот для смарт-карты, точнее для UIM-карты. Все верно. Оператор ведь должен идентифицировать абонента? Однако моя железка работает без каких-либо UIM-карт, и не только моя. Скорее всего, идентификационные данные записаны в сам терминал программно.



Начинку, к сожалению, увидеть не получится – мешает гарантия. Но самое интересное только впереди.

Поехали!

В Интернете про нашего пациента почти ничего не пишут, разве что скромная страничка на официальном сайте:

http://enterprise.huawei.com/ru/products/network/wireless/trustar/hw-264988.htm

Хотя в веб-интерфейсе и есть раздел обновления прошивки, саму прошивку мне найти не удалось. Берем на вооружение три моих любимых статьи:

http://robocraft.ru/blog/electronics/404.html

http://www.xakep.ru/post/53057/default.asp

http://www.xakep.ru/post/53486/

и начинаем со сканера портов. Встроенный роутер имеет стандартный, всем знакомый адрес 192.168.1.1. Сканируем с хоста, подключенного по LAN:

```
host# nmap 192.168.1.1

Starting Nmap 6.01 ( http://nmap.org )
Nmap scan report for 192.168.1.1
PORT     STATE SERVICE
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
5431/tcp open  park-agent
```

Как видно, TFTP нет в списке. На 80-ом порту висит web-интерфейс, на 53 – DNS, 5431 – это UPnP, а вот и telnet! Подключаемся...

```
host# telnet 192.168.1.1

Connected to 192.168.1.1.

HGW login: admin
Password: admin

BusyBox v0.60.0 (2011.10.18-03:24+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.
#
```

Стандартные admin/admin, и тут же нас встречает старый добрый шелл BusyBox; где его только не увидишь :) Начинаем осмотр:

busybox

```
BusyBox v0.60.0 (2011.10.18-03:24+0000) multi-call binary
```

```
Usage: busybox [function] [arguments]...
   or: [function] [arguments]...

        BusyBox is a multi-call binary that combines many common Unix
        utilities into a single executable.  Most people will create a
        link to busybox for each function they wish to use, and BusyBox
        will act like whatever it was invoked as.

Currently defined functions:
        busybox, cat, chmod, cp, date, dd, echo, find, free, grep,
ifconfig,
        insmod, kill, killall, klogd, ln, login, ls, lsmod, mkdir, mknod,
        more, mount, msh, mv, nc, ping, ps, pwd, reboot, rm, rmdir,
rmmod,
        route, sh, sleep, syslogd, telnetd, traceroute, umount, wget
```

Неплохой набор, что насчет файловой системы?

```
# ls -l
```

```
drwxr-xr-x    1 0        0             112 Jan  1 08:03 www
lrwxrwxrwx    1 0        0               7 Jan  1 00:00 var -> tmp/var
drwxr-xr-x    1 0        0              64 Jan  1 00:00 usr
drwxr-xr-x    1 0        0               0 Jan  1  2000 tmp
drwxr-xr-x    1 0        0             252 Jan  1 00:00 sbin
dr-xr-xr-x   38 0        0               0 Jan  1  2000 proc
drwxr-xr-x    1 0        0               0 Jan  2 15:50 mnt
drwxr-xr-x    1 0        0             148 Jan  1 00:00 lib
drwxr-xr-x    1 0        0              72 Jan  1 00:00 etc
drwxr-xr-x    1 0        0               0 Jan  1 00:00 dev
drwxr-xr-x    1 0        0             452 Jan  1 00:00 bin
lrwxrwxrwx    1 0        0               6 Jan  1 00:00 3w -> tmp/3w
```

Записывать файлы можно только в /tmp. Зато есть /proc – это, можно сказать, паспорт роутера.

```
# cat /proc/verison
```

```
Linux version 2.4.20 (root@LINUX) (gcc version 3.2.3 with Broadcom
modifications) #185 Tue Oct 18 11:25:19 CST 2011
```

Ядро не самое новое… Хотелось бы видеть 2.6. Идем дальше:

```
# cat /proc/cpuinfo
```

```
system type             : Broadcom BCM5354 chip rev 3
processor               : 0
cpu model               : BCM3302 V2.9
BogoMIPS                : 237.56
wait instruction        : no
microsecond timers      : yes
tlb_entries             : 32
extra interrupt vector  : no
hardware watchpoint     : no
VCED exceptions         : not available
VCEI exceptions         : not available
unaligned_instructions  : 3
dcache hits             : 0
dcache misses           : 0
icache hits             : 0
icache misses           : 0
instructions            : 0
```

«Broadcom BCM5354 chip rev 3» - это платформа SoC, более подробно о который можно узнать здесь: https://www.broadcom.com/products/Wireless-LAN/802.11-Wireless-LAN-Solutions/BCM5354

Сердце роутера – MIPS32 процессор с частотой 240 MHz.

Посмотрим информацию о памяти:

cat /proc/mounts

```
rootfs / rootfs rw 0 0
/dev/root / cramfs ro 0 0
none /dev devfs rw 0 0
proc /proc proc rw 0 0
ramfs /tmp ramfs rw 0 0
/dev/mtdblock/3 /tmp/var/custom cramfs ro 0 0
```

cat /proc/mtd

```
dev:    size   erasesize  name
mtd0: 00040000 00010000 "cfe"
mtd1: 00340000 00010000 "linux"
mtd2: 00268540 00010000 "rootfs"
mtd3: 00060000 00010000 "resource"
mtd4: 00010000 00010000 "factory"
mtd5: 00010000 00002000 "nvram"
```

Эта информация пригодится при извлечении файлов прошивки. Как видно, для загрузки Linux используется стандартный Broadcom'овский загрузчик CFE. Все рабочие параметры хранятся в NVRAM-памяти, управлять которой можно с помощью команды nvram. Информация хранится там в незашифрованном виде: тут и пароль админа шлюза, пароль WLAN, логин и пароль WAN, ну вы сами понимаете :)

```
# cat /proc/meminfo
```

```
        total:    used:    free:  shared: buffers:  cached:
Mem:  30982144 12759040 18223104        0  1789952  4214784
Swap:        0        0        0
MemTotal:        30256 kB
MemFree:         17796 kB
MemShared:           0 kB
Buffers:          1748 kB
Cached:           4116 kB
SwapCached:          0 kB
Active:           2964 kB
Inactive:         4596 kB
HighTotal:           0 kB
HighFree:            0 kB
LowTotal:        30256 kB
LowFree:         17796 kB
SwapTotal:           0 kB
SwapFree:            0 kB
```

Всего системе доступно 30 Мб памяти, 17 Мб свободно. Хороший запас.

Чуть не забыл, список процессов:

```
# ps
```

```
PID  Uid     Stat Command
    1 0         S    init noinitrd
    2 0         S    [keventd]
    3 0         R    [ksoftirqd_CPU0]
    4 0         S    [kswapd]
    5 0         S    [bdflush]
    6 0         S    [kupdated]
    7 0         S    [mtdblockd]
    8 0         S    [khubd]
   24 0         S    httpd
   77 0         S    [usb-storage-1]
   78 0         S    [scsi_eh_0]
   83 0         S    upnp -D -L br0 -W ppp0
  103 0         S    ats
  104 0         S    ats
  366 0         S    syslogd
  372 0         S    /usr/sbin/dnsmasq -H /tmp/hosts -n -i br0 -r
/tmp/resolv.c
  378 0         S    nas /tmp/nas.lan.conf /tmp/nas.lan.pid lan
  380 0         S    diagd
  381 0         S    udhcpd /tmp/udhcpd0.conf
  383 0         S    syswatch
  386 0         S    diagd
  393 0         S    telnetd
  398 0         S    ses -f
  399 0         S    pppd /dev/usb/tts/0 -g -g -i 0 -t 1500
  400 0         S    ses_cl -f
  407 0         S    -sh
  413 0         R    ps –a
```

Из демонов можно выделить httpd – представитель web-интерфейса, syslogd – системный лог, с telnetd все итак понятно, ну и некоторые другие. Можно заметить, что модем расположен в /dev/usb/tts/0. Больше ничего интересного.

Модули ядра:

```
# cat /proc/modules

wl                   560400   0 (unused)
et                    21920   0 (unused)
```

wl отвечает за Wi-Fi, et управляет портом LAN. Не густо. Теперь сетевые настройки (адреса изменены на иксы и нули):

```
# ifconfig

br0       Link encap:Ethernet  HWaddr XX:XX:XX:XX:XX:XX
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16778 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11709 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6952897 (6.6 Mb)  TX bytes:2264394 (2.1 Mb)

eth0      Link encap:Ethernet  HWaddr XX:XX:XX:XX:XX:XX
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16788 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12015 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:7322882 (6.9 Mb)  TX bytes:2330814 (2.2 Mb)
          Interrupt:4 Base address:0x1000

eth1      Link encap:Ethernet  HWaddr XX:XX:XX:XX:XX:XX
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:45
          TX packets:0 errors:246 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:13 Base address:0x5000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING MULTICAST  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

ppp0      Link encap:Point-Point Protocol
          inet addr:0.0.0.0  P-t-P:0.0.0.0  Mask:255.255.255.255
          UP POINTOPOINT RUNNING MULTICAST  MTU:1400  Metric:1
          RX packets:13713 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15553 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:20
          RX bytes:2160575 (2.0 Mb)  TX bytes:6814543 (6.4 Mb)

vlan0     Link encap:Ethernet  HWaddr XX:XX:XX:XX:XX:XX
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16788 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12015 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7020698 (6.6 Mb)  TX bytes:2330814 (2.2 Mb)
```

Ок, теперь прошивка... Вспомнив вывод команды «cat /proc/mtd», попробуем прочитать содержимое памяти в файл. Веб-интерфейс находится в /tmp/3w, и сервер httpd открывает доступ именно к этой директории, поэтому использовать будем ее.

```
# cd /tmp/3w
# cat /dev/mtdblock/0 > cfe.js
# cat /dev/mtdblock/1 > linux.js
# cat /dev/mtdblock/2 > rootfs.js
# cat /dev/mtdblock/3 > recource.js
# cat /dev/mtdblock/4 > factory.js
```

После этого можно смело скачивать файлы на хост, сменив расширение на “bin”. Теперь нашими инструментами будут утилиты hexdump и binwalk (http://binwalk.org). Начнем с cfe.bin:

```
host# binwalk cfe.bin

DECIMAL    HEX        DESCRIPTION
--------------------------------------------------------------------------------
6427       0x191B     88K BCS executable
36212      0x8D74     gzip compressed data, was "piggy", from Unix, last
                      modified: Tue Oct 18 09:26:52 2011, max compression
```

Это загрузчик CFE, с которым можно «пообщаться» только через последовательный порт. Следующий файл:

```
host# binwalk rootfs.bin

DECIMAL    HEX        DESCRIPTION
```

```
--------------------------------------------------------------------------------
0             0x0             CramFS filesystem, little endian size 2154496 version #2
sorted_dirs

                              CRC 0x4aa4c400, edition 0, 1525 blocks, 346 files
```

Название файла говорит само за себя – это образ корневой файловой системы. Замечу, что используется CramFS, а не SquashFS. Файл можно смело монтировать на хосте и изучать его содержимое, его даже 7z под Windows умеет открывать :) Далее:

host# binwalk linux.bin

```
DECIMAL       HEX             DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             TRX firmware header, little endian, header size: 28 bytes,
image size:

                              3039232 bytes, CRC32: 0x9EAB11AC flags/version: 0x10000
28            0x1C            gzip compressed data, was "piggy", from Unix, last
modified: Tue Oct

                              18 09:26:38 2011, max compression
883392        0xD7AC0         CramFS filesystem, little endian size 2154496 version #2
sorted_dirs

                              CRC 0x4aa4c400, edition 0, 1525 blocks, 346 files
```

Где-то мы это уже видели? Очень похоже не образ прошивки. Сначала TRX-заголовок с разметкой и контрольными суммами, затем образ CFE, ну и rootfs.

Остались еще два файла. Начну с factory.bin. Взглянув на него hexdump'ом, я увидел там серийные номера роутера и платы, MAC-адрес адаптеров, параметры начала и конца FLASH-памяти. В мануале openWRT (http://wiki.openwrt.org/toh/huawei/e970) пишут, что эти параметры использует CFE при сбросе на заводские настройки. resource.bin оказался таким же, как и rootfs.bin, образом CramFS, содержащим файлы локализации web-интерфейса и его HTML-код.

Заключение

Дело в том, что девайс не совсем стабильно себя ведет... Подключение к Интернету постоянно разрывается и не восстанавливается, хотя в соответствии с настройками должно. Приходится либо постоянно заходить в web-интерфейс и подключаться к сети вручную, либо вообще перезагружать терминал. Нужна замена кастомной прошивки.

Мое внимание привлекли два проекта: openWRT и DD-WRT. В первую очередь своими возможностями. Имея одну из двух таких прошивок, я мог бы смело поставить туда torrent-клиент, да и не только, и использовать эту железяку по максимуму, особенно, USB-порт, который по умолчанию тоже можно использовать для подключения к компу, но в этом случае встроенный роутер становится недоступным, и Wi-Fi тоже - так не интересно. На борту есть батарея, которая позволяет сделать из терминала настоящую вардрайверскую станцию. Еще было бы интересно взаимодействие прошивки и встроенного эмулятора АТС. Однако в списках поддерживаемого оборудования моей модели найдено не было.

На просторах Интернета были найдены роутеры Linksys WRT54G2 и Asus WL500gP v2, работающие на том-же чипе, что и мой. Не все потеряно :)

Вот и закончился мой первый обзор. Почему первый? Потому, что исследование продолжается. В планах вскрыть девайс, поработать с последовательным портом, научиться мигать индикаторами и многое другое. Кому интересно, вот:

Образы разделов FLASH-памяти: CFE, Linux, RootFS и Resource

http://www.fixeria.net/huawei/all.zip