FrankeNAND

How to work with modern automotive flash memory chip

NavInfo Europe
Cybersecurity team

NAVINFO

Who are we?

NavInfo Europe Cybersecurity Team

Based in Eindhoven, Netherlands

Yuriy Serdyuk
Lead Cybersecurity Researcher

Alexey Kondikov
Lead Cybersecurity Researcher

Agenda

About internet access unit (IAU) device

- **Hardware**
  - Packaging
  - Two memories in one chip
  - Dirty trick
  - Bath time
  - Destroying PCB layer by layer
  - Making the reader
- **NAND flash - JYA35**
  - Architecture of NAND memory
  - Flash dumping
  - NandTool – Integer overflow
  - OOB structure
  - Qualcomm partition table structure
  - UBIFS extraction
  - BACKUP filesystem extraction
- **NAND flash – JY990**
  - Page structure
  - File system
- **Conclusions**

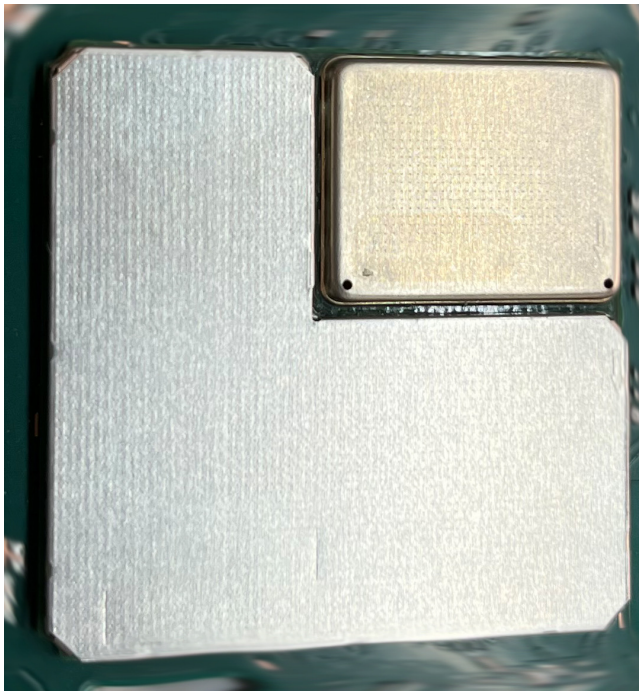
Internet access unit (IAU)

Why IAU

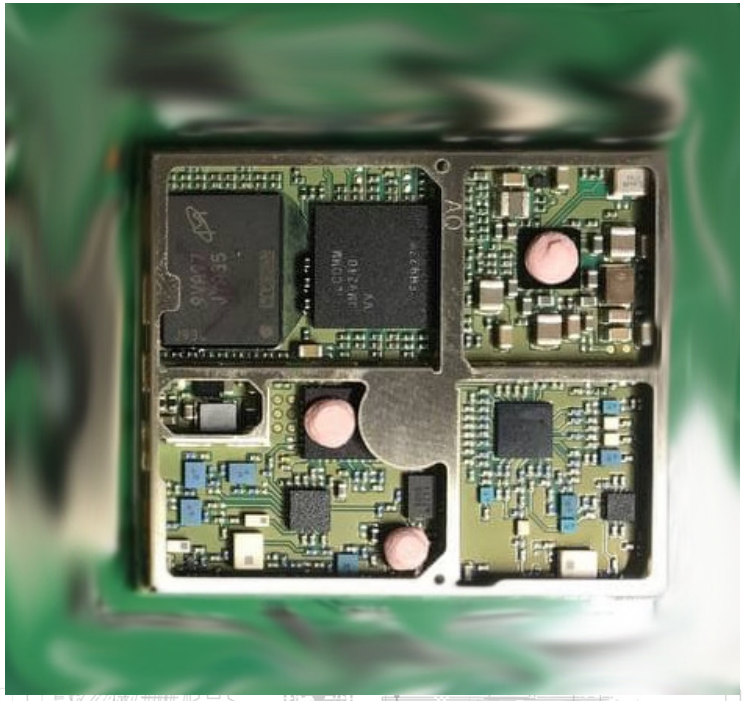
- Part of one modern electrical car
- Emergency call module
- 3G/4G connectivity (eSIM)
- Connected to company backend
- Participates in car updates via SOTA

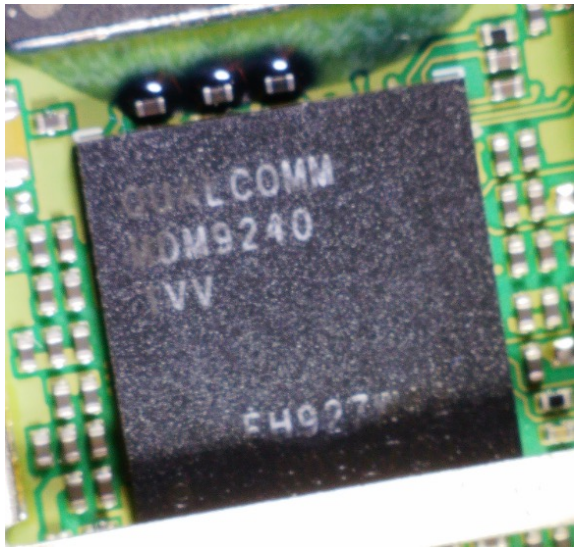
NAND flash memory MT29F4G08ABBEA

- Non-standard chip
  - NAND flash and DDR2 together
  - No sockets in the market
  - No readers
- Sometimes NAND chip is glued
- Custom format NAND pages



About Internet Access Unit (IAU) device

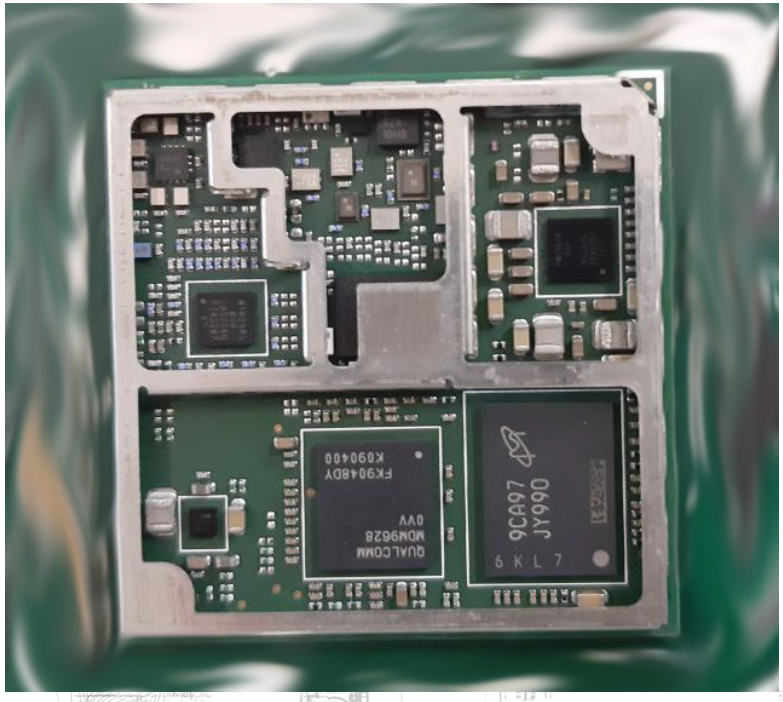




Main CPU: QUALCOMM MDM9240

Memory chip: MT29F4G08ABBEA NAND Flash with Mobile LPDDR2

About Internet Access Unit (IAU) device

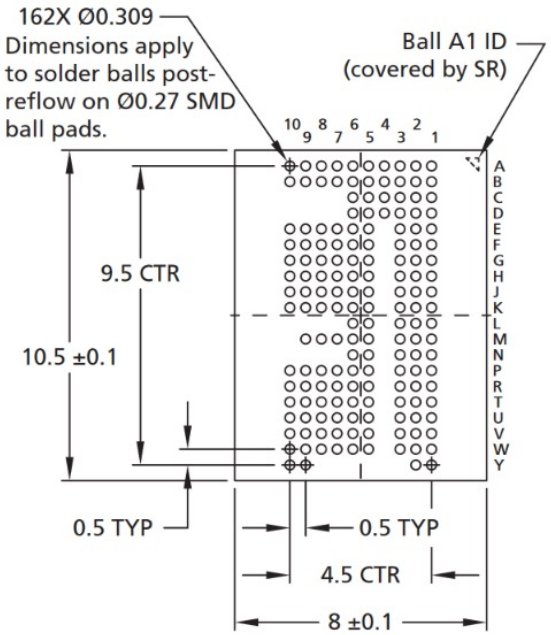


Main CPU: QUALCOMM MDM9628

Memory chip: MT29RZ4B2DZZHHW NAND Flash with Mobile LPDDR2

https://www.micron.com/support/tools-and-utilities/fbga

Packaging



Packaging of chip is BGA-162. All pins are on the bottom side of IC.

Two memories in one chip

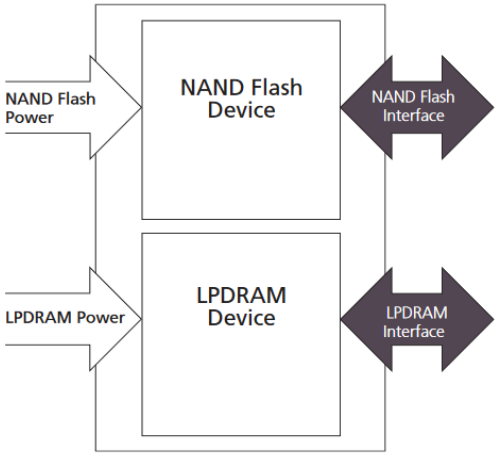
Micron

162-Ball NAND Flash with LPDDR2 MCP Features

NAND Flash with Mobile LPDDR2 162-Ball MCP

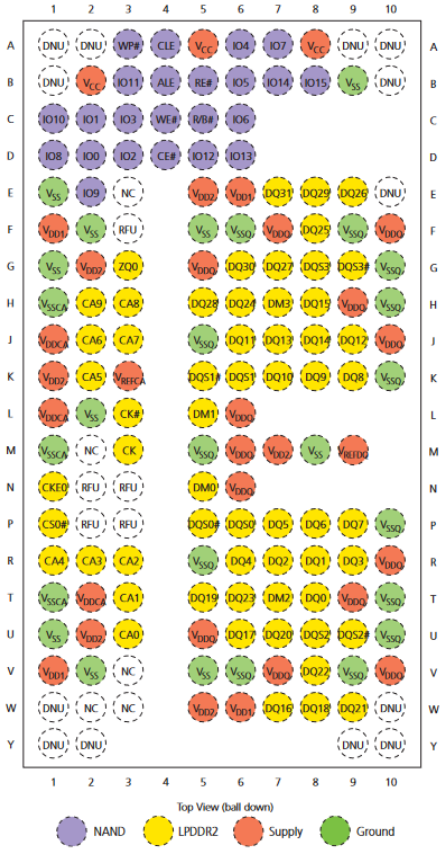
MT29RZ4B4DZZMGWD-18 I.80C

Figure 1: MCP Block Diagram

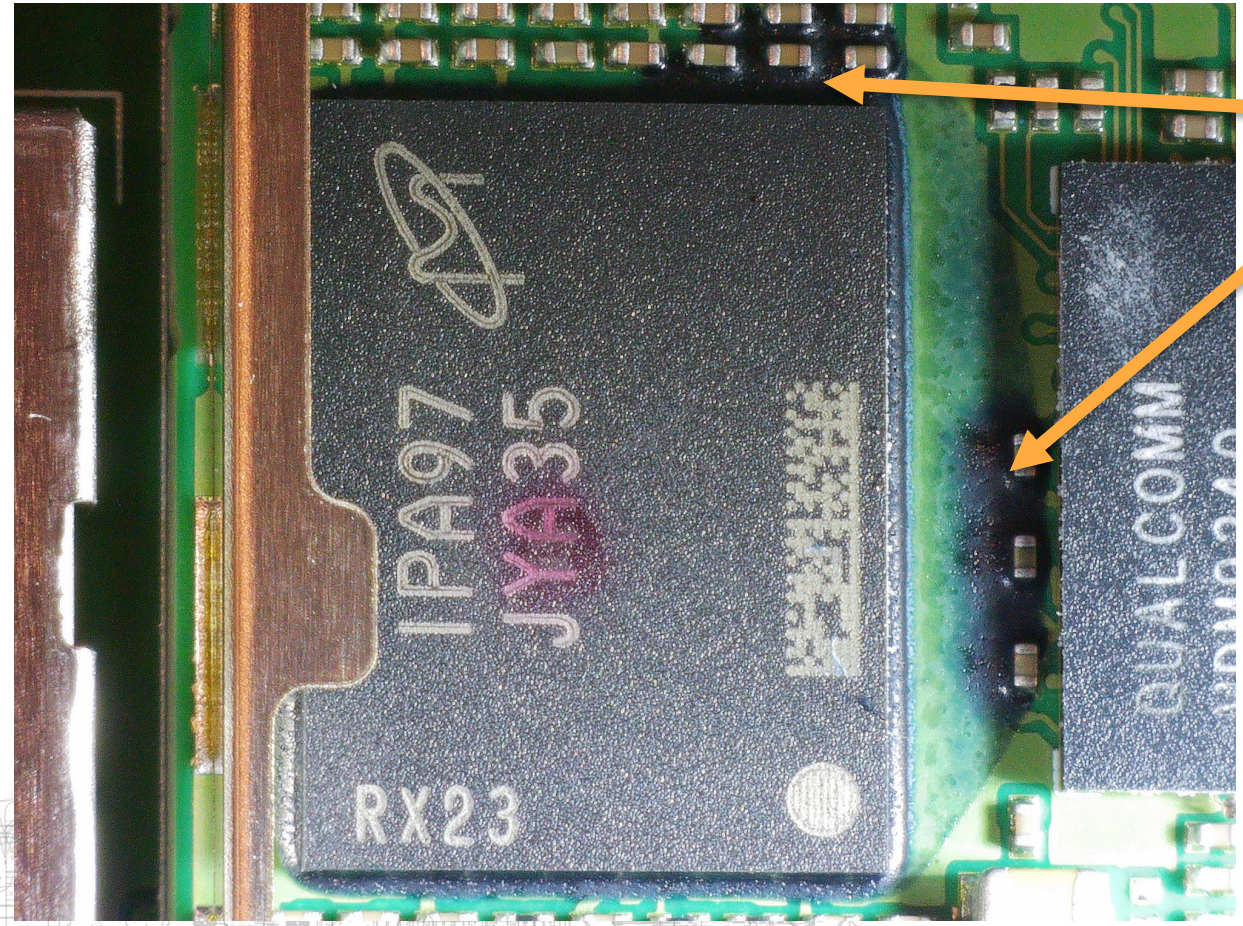


Two different types of memory embedded in one package.

Two different interfaces and two different power lines



Dirty Trick



Traces of glue

Possible solution??

Bath time


CE
ROHS
LWA
80 dB
HBM
Machines
DIGITALE ULTRASOON REINIGER
WAARSCHUWING!!!
TEMP +
TEMP-
AAN/UIT
TIJD +
TIJD-
AAN/UIT

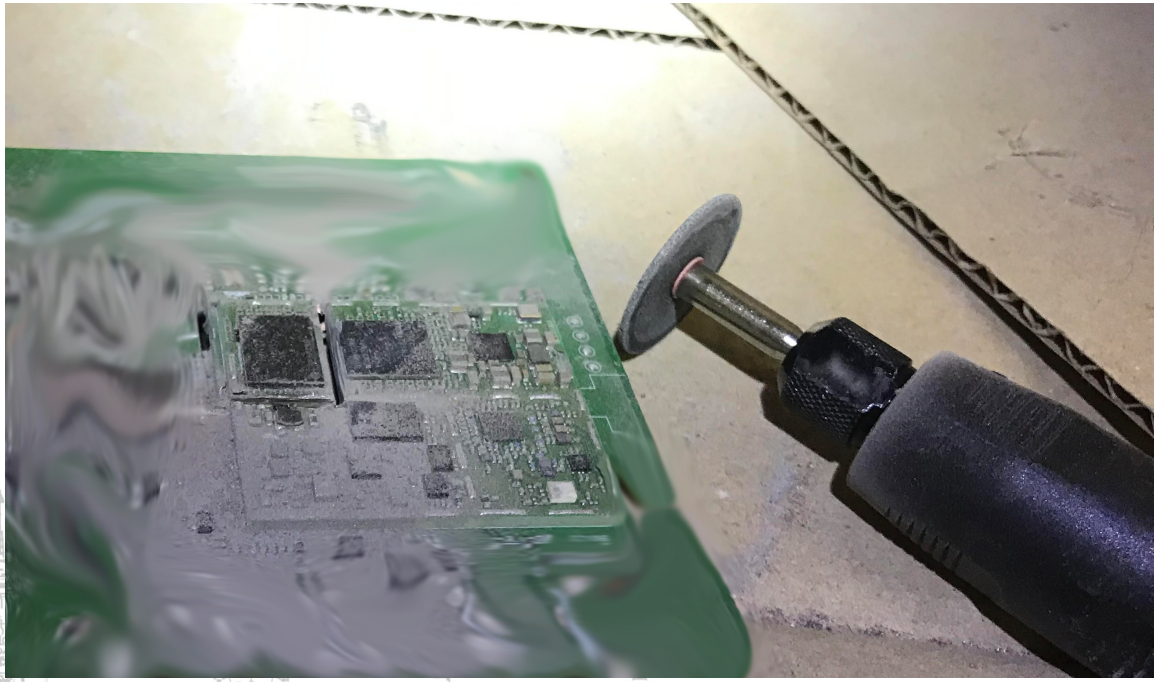
+

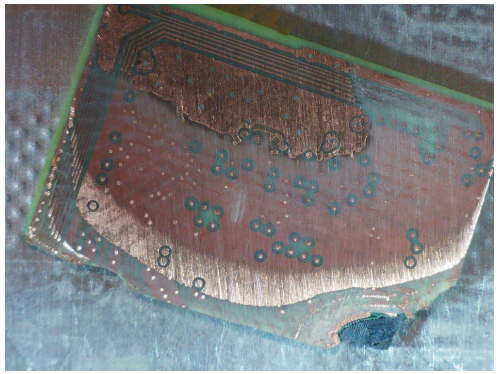

300ml


PolyesterShoppen
DC001
Aceton
1 L

Nope, that did not work...

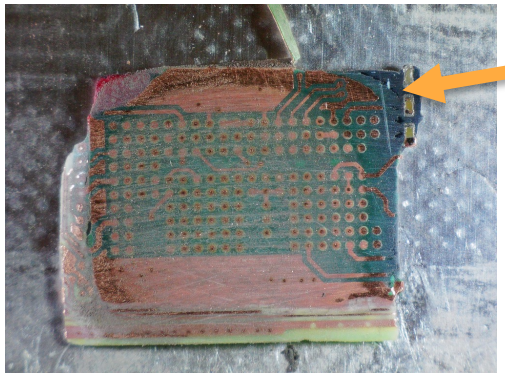
Destroying layers of PCB



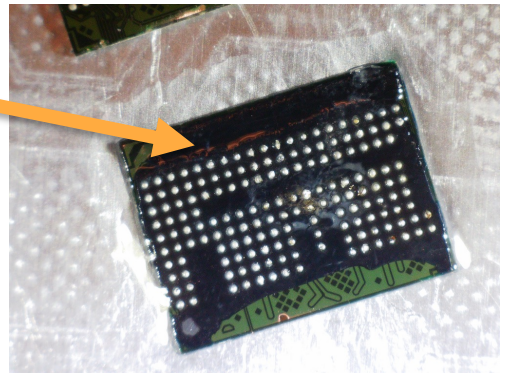


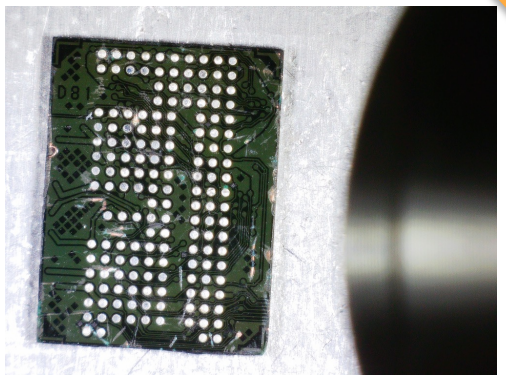
After cutting goes abrasive paper

Destroying layers of PCB

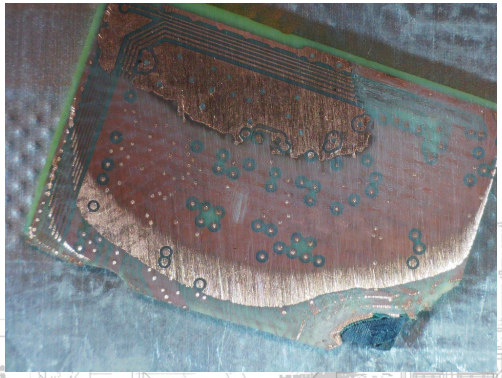


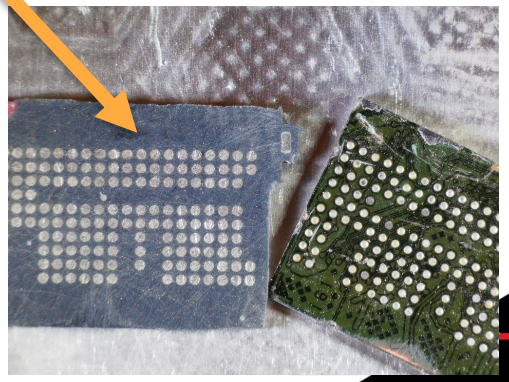
Remaining glue



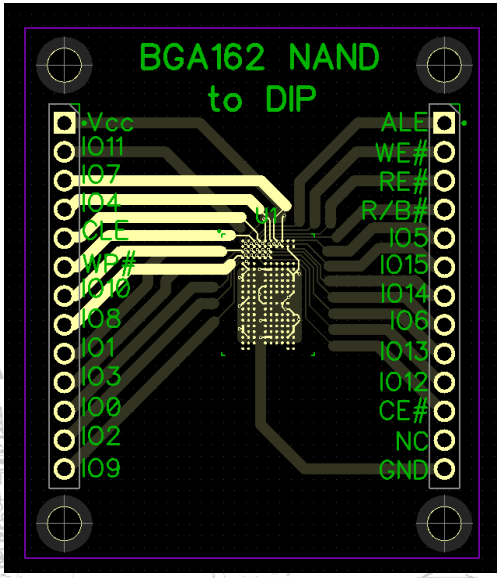


Clean memory chip

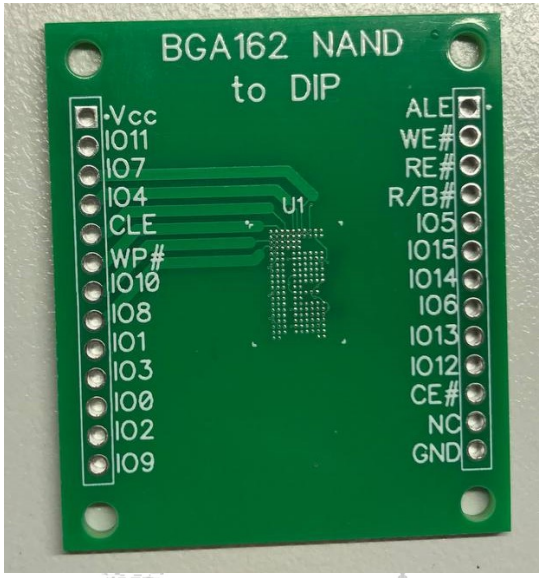




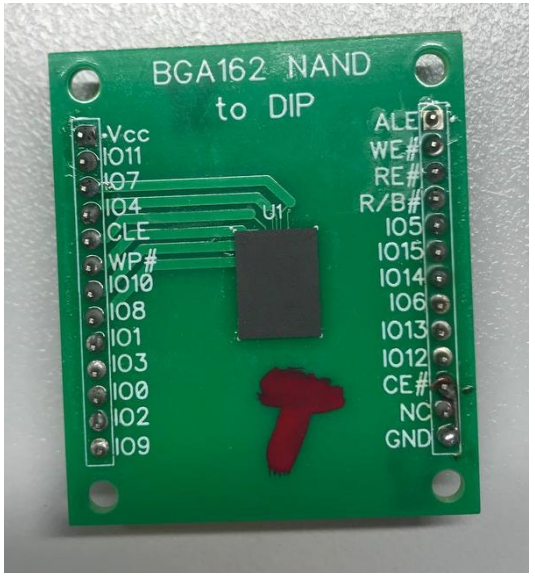
Making the reader



We have developed a connector board for chip

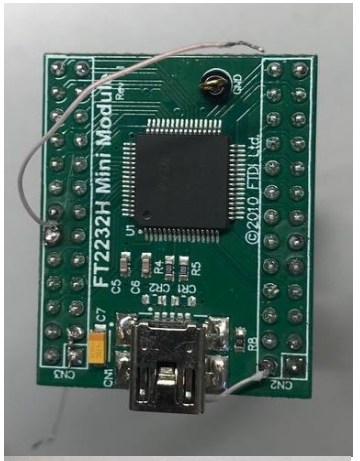


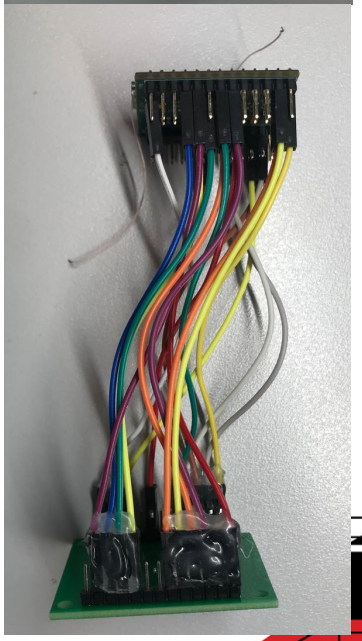
Then we made it



We reballed and soldered chip on the connector board

And we built the reader based on FT2232H board

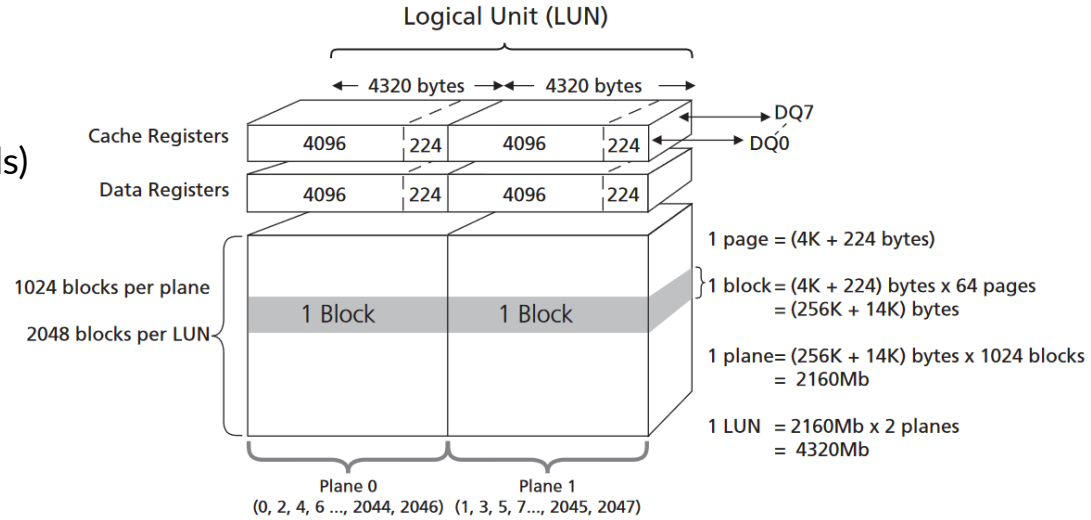




NAND flash JYA35

MT29F4G08ABBEA – Micron, 162-Ball: 4Gb (x8) NAND with 2Gb (x32) LPDDR2

- Organization
  - Page size: x8 4320 bytes (4096 + 224 bytes)
  - Page size: x16 2160 words (2084 + 112 words)
  - Block size: 64 pages (256K + 14K bytes)
  - Plane size: 2 planes x 1024 blocks per plane
  - Device size: 4Gb: 2048 blocks



https://www.micron.com/-/media/client/global/documents/products/data-sheet/mcp/162ball/162ball_nand_lpddr2_embedded_j80c.pdf

NandTool – flash dumping

Flash memory size – 0x20FE22A0 ~528 MB

NandTool output

```
FT2232H-based NAND readerUsage: [-i|-r file|-v file] [-p <start page no> <end pageno> [-t main|oob|both] [-s]
  -i       - Identify chip
  -r file - Read chip to file
  -w file - Write chip from file
  -v file - Verify chip from file data
  -p <start pageno> <end pagenon>  - Select page to operate
  -t reg   - Select region to read/write (main mem, oob ('spare') data or both, interleaved)
  -s       - clock FTDI chip at 12MHz instead of 60MHz
  -u vid:pid - use different FTDI USB vid/pid. Vid and pid are in hex.
```

https://github.com/ohjeongwook/NANDReader_FTDI.git

Binwalk - Results

Images and file systems

- Qualcomm SBL1
- EFS2 Qualcomm
- UBIFS
- Squashfs
- ELF
- Zlib
- Etc…

```
0             0x0             Qualcomm SBL1, image addr: ffffffff, image size: 4294967295, ...
2097          0x831           Qualcomm SBL1, image addr: 795775eb, image size: 4294967255, ...
4320          0x10E0          Qualcomm SBL1, image addr: ffffffff, image size: 4294967295, ...
6417          0x1911          Qualcomm SBL1, image addr: 795775eb, image size: 4294967255, ...
10737         0x29F1          ELF, 32-bit LSB processor-specific, (SYSV)
15800         0x3DB8          Certificate in DER format (x509 v3), header length: 4, sequence length: 1391
17339         0x43BB          Certificate in DER format (x509 v3), header length: 4, sequence length: 1194
18569         0x4889          Certificate in DER format (x509 v3), header length: 4, sequence length: 1151
42573         0xA64D          Ubiquiti partition header, header size: 56 bytes, name: "PARTNUM_SHFT", base ...
43105         0xA861          Ubiquiti partition header, header size: 56 bytes, name: "PARTNUM_SHFT", base ...
138240        0x21C00         Qualcomm SBL1, image addr: cb57c, image size: 83947599, code size: 3490092289, ...
...
5942400       0x2247000       EFS2 Qualcomm filesystem super block, little endian, NAND version 0x2d, 80 blocks, ...
36218880      0x228A800       EFS2 Qualcomm filesystem super block, little endian, NAND version 0x2d, 80 blocks, ...
36495360      0x22CE000       EFS2 Qualcomm filesystem super block, little endian, NAND version 0x2d, 80 blocks, ...
36771840      0x2311800       EFS2 Qualcomm filesystem super block, little endian, NAND version 0x2d, 80 blocks, ...
...
50872320      0x3084000       UBI erase count header, version: 1, EC: 0x0, VID header offset: 0x1000, data offset: 0x2000
50876640      0x30850E0       UBI volume ID header, version: 1, type: 1, volume id: 0, size: 0
50880960      0x30861C0       Squashfs filesystem, little endian, version 4.0, compression:gzip, size: 34525276 bytes...
306288042     0x124195AA      Zlib compressed data, default compression
306288129     0x12419601      Zlib compressed data, default compression
....
```

Not aligned addresses

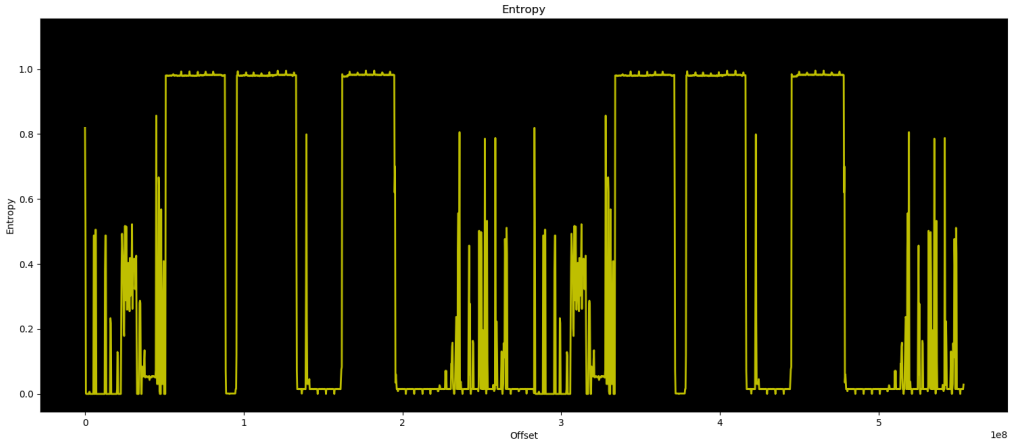
Binwalk – file system extraction

Extraction the file system failed

```
binwalk ./full_    .bin
49621549      0x2F52A2D       Unix path: /dev/icb/rpm
50872320      0x3084000       UBI erase count header, version: 1, EC: 0x0, VID header offset: 0x1000, data offset: 0x2000
50876640      0x30850E0       UBI volume ID header, version: 1, type: 1, volume id: 0, size: 0
50880960      0x30861C0       Squashfs filesystem, little endian, version 4.0, compression:gzip, size: 34525276 bytes
306288042     0x124195AA      Zlib compressed data, default compression
306288129     0x12419601      Zlib compressed data, default compression
306288216     0x12419658      Zlib compressed data, default compression


dd if=./full_    .bin of=./squashfs.bin bs=1 skip=$((0x30861C0)) count=34525276
8430+0 records in
8430+0 records out
34525276 bytes (35 MB, 33 MiB) copied, 1.06686 s, 32.4 MB/s

file squashfs.bin
squashfs.bin: Squashfs filesystem, little endian, version 1024.0, compressed, 6687860848397910016 bytes, 671088640 inodes, blocksize: 512 bytes

mount ./squashfs.bin /mnt/test/
mount: /mnt/test: wrong fs type, bad option, bad superblock on /dev/loop15, missing codepage or helper program, or other error.
```

Binwalk - entropy

Same data after the offset = 0x10E00000




```
00000000:   D1 DC 4B 84-34 10 D7 73-5A 43 0B 7D-FF FF FF FF
00000010:   FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
00000020:   FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
00000030:   FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF

10E00000:   D1 DC 4B 84-34 10 D7 73-5A 43 0B 7D-FF FF FF FF
10E00010:   FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
10E00020:   FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
10E00030:   FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
```

Firmware diff

First File - C:\pentest\ID.3\OCU\firmware\full_ .half1_0x00000000

OFFSET	00 01 02 03 04 05 06 07	08 09 0A 0B 0C 0D 0E 0F
101E21A0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21B0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21C0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21D0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21E0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21F0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2200	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2210	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2220	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2230	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2240	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2250	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2260	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2270	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2280	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2290	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E22A0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E22B0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF

Second File - C:\pentest\ID.3\OCU\firmware\full half2_0x10E00000

OFFSET	00 01 02 03 04 05 06 07	08 09 0A 0B 0C 0D 0E 0F
101E21A0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21B0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21C0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21D0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21E0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E21F0	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2200	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2210	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2220	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2230	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2240	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2250	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2260	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2270	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2280	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF
101E2290	FF FF FF FF FF FF FF FF	FF FF FF FF FF FF FF FF

NAND flash keeps full copy of data
Really?

NandTool – integer overflow

Page size – 0x10E0

Flash size – 0x20FE22A0 ~ 528MB

Pages – 0x1F483

pageno – int (4 bytes)

Address: 0x10E00000

Page number: 0x10000

Arg: 0x10000<<16 = 0x0100000000

NandDataLP.cpp

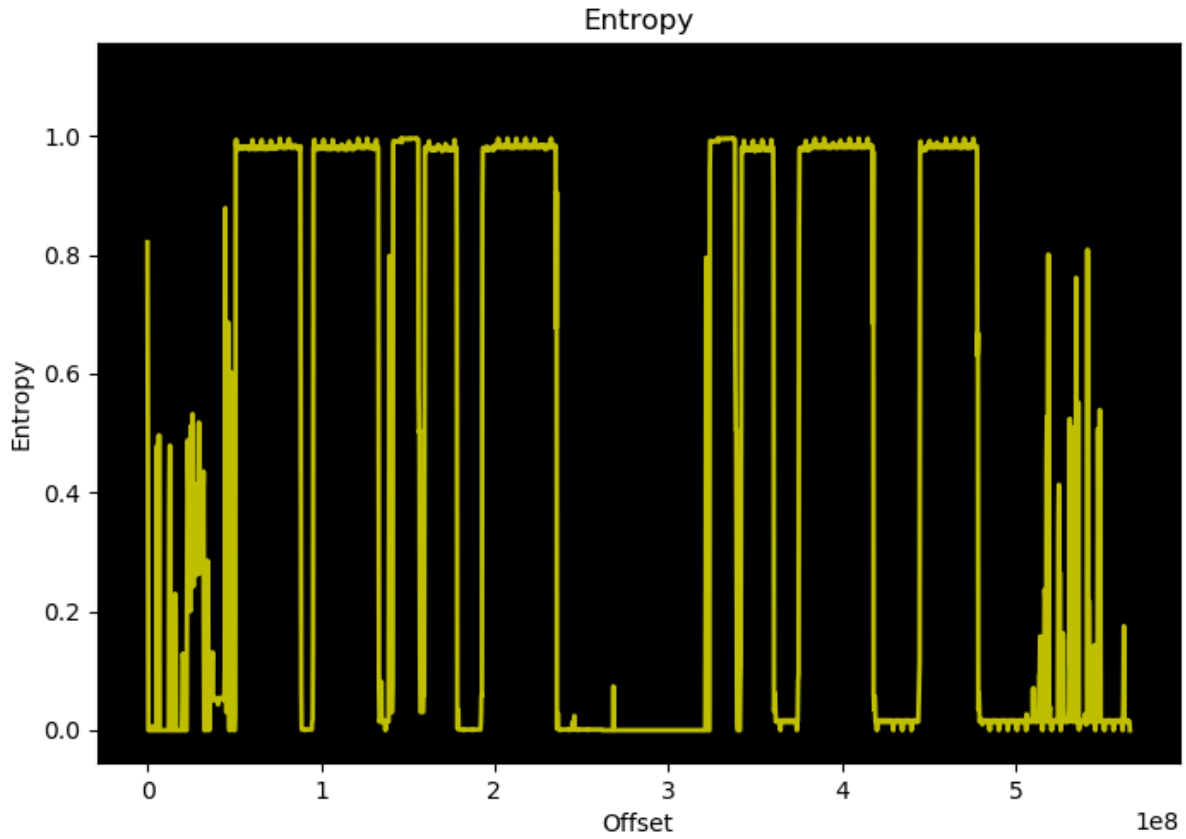
```
int NandDataLP::readPage(int pageno, char *buff, int max) {
        //Read a page
        m_ft->sendCmd(NAND_CMD_READ0);
        m_ft->sendAddr(pageno<<16L, m_id->getAddrByteCount());
        m_ft->sendCmd(NAND_CMD_READSTART);
        m_ft->waitReady();
        if (max>m_id->getPageSize()) max=m_id->getPageSize();
        if (max>0x1000)
        {
```

```
                push    rbp
                mov     ebp, esi
                xor     esi, esi        ; cmd
pageno = rbp                            ; int
                push    rbx
                mov     rbx, this
                mov     this, [this+8]  ; this
this = rbx                              ; NandDataLP *const
                shl     ebp, 16
                call    _ZN8FtdiNand7sendCmdEc ; FtdiNand::sendCmd(char)
                mov     rdi, [this+10h] ; this
                mov     r14, [this+8]
                call    _ZN6NandID16getAddrByteCountEv ; NandID::getAddrByteCount(void)
                movsxd  rsi, ebp        ; addr
                mov     rdi, r14        ; this
                mov     edx, eax        ; noBytes
                call    _ZN8FtdiNand8sendAddrExi ; FtdiNand::sendAddr(long long,int)
                mov     rdi, [this+8]   ; this
                mov     esi, 30h ; '0'  ; cmd
```

https://github.com/cheongwook/NANDReader

Binwalk – fixed memory dump

Flash memory size is 0x21C00000 ~ 540 MB



OOB structure

NAND memory - MT29F4G08ABBEA

- Dump size – 0x21C00000 ~ 540 MB
- Page size – 0x10E0 (4320)
  - Data – 0x1000 (4096)
  - OOB – 0xE0 (224)
- Block size – 0x10E0*0x40 (64)

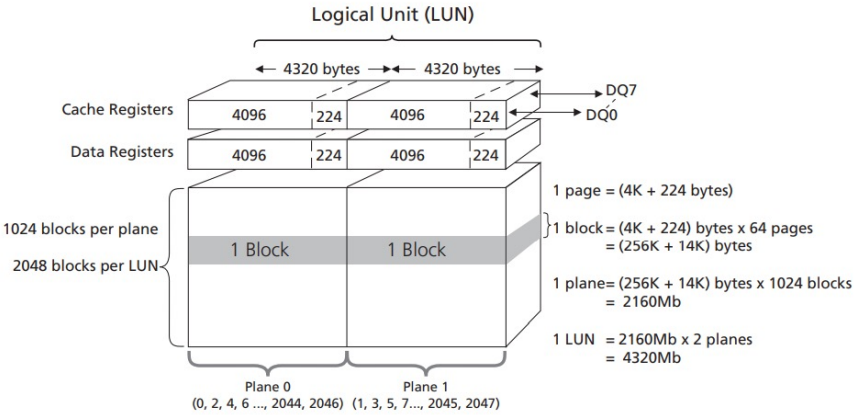
4Gb: x8, x16 NAND Flash Memory

Features

- Open NAND Flash Interface (ONFI) 1.0 compliant
- Single-level cell (SLC) technology[1]
- Organization
  - Page size: x8 4320 bytes (4096 + 224 bytes)
  - Page size: x16 2160 words (2084 + 112 words)
  - Block size: 64 pages (256K + 14K bytes)
  - Plane size: 2 planes x 1024 blocks per plane
  - Device size: 4Gb: 2048 blocks

Device and Array Organization

Figure 7: Array Organization – MT29F4G (x8)



OOB structure – strange size

Datasheet:

- OOB - 0xE0 (224)
- At the end of the page

Reality:

page 41 and 42

- At the end of the page some metadata
- Size of metadata–0x6F

page 82

- At the end of the page some metadata
- Size of metadata–0x4F

```
0002C430:  52 50 4D 20-49 6D 61 67-65 20 4C 6F-61 64 65 64  RPM Image Loaded
0002C440:  2C 20 44 65-6C 74 61 00-41 50 50 53-20 49 6D 61  , Delta APPS Ima
0002C450:  67 10 02 00-00 00 00 00-00 07 00 00-00 00 00 00  g►☻        •
0002C460:  00 04 02 00-00 14 02 00-00 00 00 00-00 01 00 00   ♦☻  ¶☻       ☺
0002C470:  00 1C 71 A2-41 D8 1E 35-6E CB 1C 21-88 E9 FF FF  ∟qóA╪▲5n╦∟!êΘ
0002C480:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C490:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C4A0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C4B0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C4C0:  65 20 4C 6F-61 64 65 64-2C 20 44 65-6C 74 61 00  e Loaded, Delta
0002C4D0:  4F 43 49 4D-45 4D 00 44-41 54 41 52-41 4D 2E 42  OCIMEM DATARAM.B

0002D510:  6F 63 6B 73-5F 66 69 6E-64 28 29 3A-20 63 61 6E  ocks_find(): can
0002D520:  27 74 20 61-6C 6C 6F 63-61 74 65 20-70 61 67 65  't allocate page
0002D530:  20 4F FC 29-1D DB F7 57-FC 04 F1 10-00 DB F7 48   Oⁿ)↔▄≈Wⁿ♦±► ▄≈H
0002D540:  FC 05 F1 08-01 DB F7 4F-FC 04 F1 14-00 DB F7 40  ⁿ♣±◘○▄≈Oⁿ♦±¶ ▄≈@
0002D550:  FC C6 20 B4-E9 9C 79 DD-F0 69 85 20-F7 A2 FF FF  ⁿ╞ ┤Θ£y▌≡ià ≈ó
0002D560:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002D570:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002D580:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002D590:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002D5A0:  62 75 66 00-64 73 5F 64-73 73 64 5F-64 61 74 61  buf ds_dssd_data
0002D5B0:  5F 62 6C 6F-63 6B 73 5F-66 69 6E 64-28 29 3A 20  _blocks_find():

00056740:  74 69 6F 6E-44 61 74 65-00 2F 2E 65-66 73 5F 70  tionDate /.efs_p
00056750:  72 78 5F 76-61 6C 69 64-5F 61 67 65-00 2F 6E 76  rx_valid_age /nv
00056760:  2F 69 74 65-6D 5F 66 69-6C 65 73 2F-77 6C 61 6E  /item_files/wlan
00056770:  2F B2 7E D4-73 2A 32 CC-CD F2 2A BD-93 85 FF FF  /▓~└s*2╠═≥*╩ôà
00056780:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
00056790:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000567A0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000567B0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000567C0:  69 76 61 74-65 00 2F 6E-76 2F 69 74-65 6D 5F 66  ivate /nv/item_f
000567D0:  69 6C 65 73-2F 6D 6F 64-65 6D 2F 6D-6D 6F 64 65  iles/modem/mmode
```

Where is our OOB?

OOB structure – ground-truth page

Page = 8 * Chunk + padding

- Data = 8 * (0x174+0x90) – 0x1020
- Meta = 8 * (1+0xF) + 0x40 – 0xC0

```
struct Chunk
{
    BYTE data0[0x174];
    BYTE meta0[1];
    BYTE data1[0x90];
    BYTE meta1[0xF];
};

struct Page
{
    Chunk chunk[8];
    BYTE padding[0x40];
};
```

Chunk – 0x214

```
00000000:  43 55 53 54-5F 46 4C 4F-57 4E 4F 54-49 44 49 53  CUST_FLOWNOTIDIS
00000010:  41 42 4C 45-00 43 55 53-54 5F 44 4D-44 49 53 41  ABLE CUST_DMDISA
00000020:  42 4C 45 00-55 53 45 52-5F 4D 4F 44-45 4D 44 49  BLE USER_MODEMDI
00000030:  53 41 42 4C-45 00 43 55-53 54 5F 42-4F 4F 54 51  SABLE CUST_BOOTQ
00000040:  55 49 45 54-44 49 53 41-42 4C 45 00-51 4F 53 5F  UIETDISABLE QOS_
00000050:  41 50 49 5F-44 49 53 41-42 4C 45 00-53 50 52 49  API_DISABLE SPRI
00000060:  4E 54 5F 44-4D 5F 44 49-53 41 42 4C-45 00 48 44  NT_DM_DISABLE HD
00000070:  52 5F 4C 4F-4E 47 5F 53-4C 45 45 50-5F 44 49 53  R_LONG_SLEEP_DIS
00000080:  41 42 4C 45-00 57 5F 44-49 53 41 42-4C 45 00 44  ABLE W_DISABLE D
00000090:  45 46 41 55-4C 54 5F 4C-54 45 5F 50-52 4F 46 49  EFAULT_LTE_PROFI
000000A0:  4C 45 00 41-56 5F 50 52-4F 46 49 4C-45 00 49 44  LE AV_PROFILE ID
000000B0:  53 5F 4F 52-49 47 5F 50-41 43 4B 41-47 45 4E 41  S_ORIG_PACKAGENA
000000C0:  4D 45 00 49-44 53 5F 50-41 43 4B 41-47 45 4E 41  ME IDS_PACKAGENA
000000D0:  4D 45 00 49-44 53 5F 50-52 45 56 5F-50 41 43 4B  ME IDS_PREV_PACK
000000E0:  41 47 45 4E-41 4D 45 00-49 44 53 5F-53 48 41 44  AGENAME IDS_SHAD
000000F0:  4F 57 5F 50-41 43 4B 41-47 45 4E 41-4D 45 00 49  OW_PACKAGENAME I
00000100:  44 53 5F 46-55 4D 4F 5F-50 4B 47 4E-41 4D 45 00  DS_FUMO_PKGNAME
00000110:  4D 56 4E 4F-5F 50 52 4F-56 49 44 45-52 4E 41 4D  MVNO_PROVIDERNAM
00000120:  45 00 55 53-42 5F 50 52-4F 44 5F 4E-41 4D 45 00  E USB_PROD_NAME
00000130:  44 41 54 41-55 53 45 5F-53 52 56 5F-4D 4F 44 45  DATAUSE_SRV_MODE
00000140:  4C 5F 4E 41-4D 45 00 49-44 53 5F 4F-53 5F 4E 41  L_NAME IDS_OS_NA
00000150:  4D 45 00 50-52 4F 44 55-43 54 5F 4E-41 4D 45 00  ME PRODUCT_NAME
00000160:  44 41 54 41-55 53 45 5F-53 52 56 5F-4E 41 4D 45  DATAUSE_SRV_NAME
00000170:  00 57 55 5F-FF 4C 41 53-54 54 49 4D-45 00 50 4F   WU_ LASTTIME PO
00000180:  57 45 52 5F-46 41 55 4C-54 5F 44 45-42 4F 55 4E  WER_FAULT_DEBOUN
00000190:  43 45 5F 54-49 4D 45 00-49 44 53 5F-46 55 4D 4F  CE_TIME IDS_FUMO
000001A0:  5F 4C 41 53-54 5F 55 50-44 41 54 45-5F 54 49 4D  _LAST_UPDATE_TIM
000001B0:  45 00 41 56-4D 53 5F 44-4D 5F 53 45-53 53 49 4F  E AVMS_DM_SESSIO
000001C0:  4E 5F 54 49-4D 45 00 49-44 53 5F 49-4E 49 54 5F  N_TIME IDS_INIT_
000001D0:  41 43 54 49-56 41 54 49-4F 4E 5F 54-49 4D 45 00  ACTIVATION_TIME
000001E0:  44 42 49 5F-44 4F 4E 45-00 56 5A 57-5F 44 45 56  DBI_DONE VZW_DEV
000001F0:  49 43 45 5F-54 59 50 45-00 41 56 4D-53 5F 44 4D  ICE_TYPE AVMS_DM
00000200:  41 43 43 5F-53 B1 E6 03-9F 6D A2 93-CD 7E 56 F4  ACC_S▒µ♥ƒmóô=~V⌠
00000210:  BB 34 FF FF-45 52 56 45-52 5F 54 59-50 45 00 41  ╗4  ERVER_TYPE A
00000220:  56 5F 44 54-4D 46 5F 44-45 54 45 43-54 5F 54 59  V_DTMF_DETECT_TY
00000230:  50 45 00 41-43 43 4F 55-4E 54 5F 54-59 50 45 00  PE ACCOUNT_TYPE
```

Metadata and padding - 0x40

```
00000E50:  45 4C 00 43-55 53 54 5F-47 50 53 53-45 4C 00 47  EL CUST_GPSSEL G
00000E60:  4E 53 53 5F-43 4F 4F 52-44 49 4E 41-54 45 5F 53  NSS_COORDINATE_S
00000E70:  45 4C 00 44-41 54 41 55-53 45 5F 44-41 06 E9 84  EL DATAUSE_DA♠Θä
00000E80:  10 DD E6 AA-E0 30 0A A4-A0 81 FF FF-54 41 5F 57  ▶▌µ¬α0◙ñáü  TA_W
00000E90:  41 52 4E 5F-4C 45 56 45-4C 00 44 41-54 41 55 53  ARN_LEVEL DATAUS
00000EA0:  45 5F 44 41-54 41 5F 43-44 4D 41 5F-52 58 5F 42  E_DATA_CDMA_RX_B
00000EB0:  49 4C 4C 00-44 41 54 41-55 53 45 5F-44 41 54 41  ILL DATAUSE_DATA
00000EC0:  5F 4C 54 45-5F 52 58 5F-42 49 4C 4C-00 44 41 54  _LTE_RX_BILL DAT
00000ED0:  41 55 53 45-5F 44 41 54-41 5F 47 57-5F 52 58 5F  AUSE_DATA_GW_RX_
00000EE0:  42 49 4C 4C-00 44 41 54-41 55 53 45-5F 44 41 54  BILL DATAUSE_DAT
00000EF0:  41 5F 43 44-4D 41 5F 54-58 5F 42 49-4C 4C 00 44  A_CDMA_TX_BILL D
00000F00:  41 54 41 55-53 45 5F 44-41 54 41 5F-4C 54 45 5F  ATAUSE_DATA_LTE_
00000F10:  54 58 5F 42-49 4C 4C 00-44 41 54 41-55 53 45 5F  TX_BILL DATAUSE_
00000F20:  44 41 54 41-5F 47 57 5F-54 58 5F 42-49 4C 4C 00  DATA_GW_TX_BILL
00000F30:  41 4E 54 5F-50 4F 4C 4C-00 41 56 5F-41 55 44 56  ANT_POLL AV_AUDV
00000F40:  4F 4C 00 57-55 5F 44 4E-4C 44 55 52-4C 00 49 44  OL WU_DNLDURL ID
00000F50:  53 5F 46 55-4D 4F 5F 50-4B 47 55 52-4C 00 57 55  S_FUMO_PKGURL WU
00000F60:  5F 53 55 4D-4D 55 52 4C-00 57 55 5F-55 50 44 54  _SUMMURL WU_UPDT
00000F70:  55 52 4C 00-49 44 53 5F-44 53 53 5F-53 45 52 56  URL IDS_DSS_SERV
00000F80:  45 52 5F 55-52 4C 00 44-41 54 41 55-53 45 5F 53  ER_URL DATAUSE_S
00000F90:  52 56 5F 55-52 4C 00 43-55 53 54 5F-4E 4F 52 4F  RV_URL CUST_NORO
00000FA0:  41 4D 00 47-4C 4F 42 41-4C 5F 44 41-54 41 5F 52  AM GLOBAL_DATA_R
00000FB0:  4F 41 4D 00-49 44 53 5F-41 55 54 4F-53 44 4D 00  OAM IDS_AUTOSDM
00000FC0:  41 4E 54 5F-4C 49 4D 00-43 55 53 54-5F 53 49 4D  ANT_LIM CUST_SIM
00000FD0:  4C 50 4D 00-4F 4D 41 44-4D 5F 56 5A-57 5F 43 4F  LPM OMADM_VZW_CO
00000FE0:  4E 46 49 47-5F 56 45 52-5F 4E 55 4D-00 43 55 53  NFIG_VER_NUM CUS
00000FF0:  54 5F 49 53-56 4F 49 43-45 4E 00 43-55 53 54 5F  T_ISVOICEN CUST_
00001000:  FF 49 50 43-48 41 4E 4E-45 4C 52 41-54 45 45 4E   IPCHANNELRATEEN
00001010:  00 43 55 53-54 5F 51 4D-49 44 45 54-41 43 48 45   CUST_QMIDETACHE
00001020:  4E 00 43 55-53 54 5F 53-54 4B 55 49-45 4E 00 49  N CUST_STKUIEN I
00001030:  44 53 5F 46-55 4D 4F 4C-45 4E 00 49-44 53 5F 46  DS_FUMOLEN IDS_F
00001040:  55 4D 4F 5F-44 4C 44 45-53 43 5F 4C-45 4E 00 43  UMO_DLDESC_LEN_C
00001050:  55 53 54 5F-47 50 53 4C-50 4D 45 4E-00 43 55 53  UST_GPSLPMEN CUS
00001060:  54 5F 46 41-53 54 45 4E-55 4D 45 4E-00 43 55 53  T_FASTENUMEN CUS
00001070:  54 4C 00 47-4E 53 53 5F-43 4F 4F 52-44 49 4E 41  TL GNSS_COORDINA
00001080:  54 45 5F 53-45 4C 00 44-41 54 41 55-53 45 5F 44  TE_SEL DATAUSE_D
00001090:  41 BB 60 53-98 C9 06 67-90 D0 F3 A4-83 14 FF FF  A╗`SÿɆ♠gÉ╨≤ñá¶
000010A0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000010B0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000010C0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000010D0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000010E0:  5F 48 41 52-44 43 4F 44-45 44 49 50-45 4E 00 43  _HARDCODEDIPEN C
000010F0:  55 53 54 5F-43 46 55 4E-50 45 52 53-49 53 54 45  UST_CFUNPERSISTE
```

OOB structure – unused buffer

Page – 0x10E0

- Data - 0x1000
- MetaData - 0xE0

```
struct Chunk1
{
    BYTE data0[0x174];
    BYTE meta0[1];
    BYTE data1[0x90];
    BYTE meta1[0xF];
};

struct Chunk2
{
    BYTE data0[0x174];
    BYTE meta0[1];
    BYTE data1[0x70];
    BYTE unused[0x20];
    BYTE meta1[0xF];
};

struct Page
{
    Chunk1 chunk[7];
    Chunk2 chunk[1];
    BYTE padding[0x40];
};
```

Weird data – 0x20

```
00056740:  74 69 6F 6E-44 61 74 65-00 2F 2E 65-66 73 5F 70  tionDate /.efs_p
00056750:  72 78 5F 76-61 6C 69 64-5F 61 67 65-00 2F 6E 76  rx_valid_age /nv
00056760:  2F 69 74 65-6D 5F 66 69-6C 65 73 2F-77 6C 61 6E  /item_files/wlan
00056770:  2F B2 7E D4-73 2A 32 CC-CD F2 2A BD-93 85 FF FF  /▒~╘s*2╠═≥*╜ôà
00056780:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
00056790:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000567A0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000567B0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
000567C0:  69 76 61 74-65 00 2F 6E-76 2F 69 74-65 6D 5F 66  ivate /nv/item_f
000567D0:  69 6C 65 73-2F 6D 6F 64-65 6D 2F 6D-6D 6F 64 65  iles/modem/mmode
```

```
0002C430:  52 50 4D 20-49 6D 61 67-65 20 4C 6F-61 64 65 64  RPM Image Loaded
0002C440:  2C 20 44 65-6C 74 61 00-41 50 50 53-20 49 6D 61  , Delta APPS Ima
0002C450:  67 10 02 00-00 00 00 00-07 00 00 00-00 00 00 00  g►☻ •
0002C460:  00 04 02 00-00 14 02 00-00 00 00 00-01 00 00 00   ♦☻ ¶☻ ☺
0002C470:  00 1C 71 A2-41 D8 1E 35-6E CB 1C 21-88 E9 FF FF  ∟qóA╪▲5n╦∟!êΘ
0002C480:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C490:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C4A0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C4B0:  FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
0002C4C0:  65 20 4C 6F-61 64 65 64-2C 20 44 65-6C 74 61 00  e Loaded, Delta
0002C4D0:  4F 43 49 4D-45 4D 00 44-41 54 41 52-41 4D 2E 42  OCIMEM DATARAM.B
```

OOB structure - ECC

- Page structure
- data0 and data1 – flash data
- meta0 – always FF
- ecc from (data0+data1)
  - Bose–Chaudhuri–Hocquenghem codes (BCH)
  - Polynomial 8219,
  - BCH BITS 8
  - Reverse
- unused – random data
- Padding – always FF FF FF …

```
struct Chunk1
{
    BYTE data0[0x174];
    BYTE meta0[1];
    BYTE data1[0x90];
    BYTE meta1[0xF];
};

struct Chunk2
{
    BYTE data0[0x174];
    BYTE meta0[1];
    BYTE data1[0x70];
    BYTE unused[0x20];
    BYTE meta1[0xF];
};

struct Page
{
    Chunk1 chunk[7];
    Chunk2 chunk[1];
    BYTE padding[0x40];
};
```

Removing OOB

Binwalk results

```
0             0x0             Qualcomm SBL1, image addr: ffffffff, image size: 4294967295, code size
2048          0x800           Qualcomm SBL1, image addr: ffffffff, image size: 4294967295, code size
4096          0x1000          Qualcomm SBL1, image addr: ffffffff, image size: 4294967295, code size
6144          0x1800          Qualcomm SBL1, image addr: ffffffff, image size: 4294967295, code size
...
131072        0x20000         Qualcomm SBL1, image addr: cb57c, image size: 83947599
148124        0x2429C         Unix path: /dev/icbcfg/boot
184084        0x2CF14         Qualcomm SBL1, image addr: 76dc419, image size: 1886057615
184092        0x2CF1C         CRC32 polynomial table, little endian
262156        0x4000C         Qualcomm SBL1, image addr: 46210400, image size: 4160112232
...
34078720      0x2080000       EFS2 Qualcomm filesystem super block, little endian, NAND version 0x2d
34340864      0x20C0000       EFS2 Qualcomm filesystem super block, little endian, NAND version 0x2d
34603008      0x2100000       EFS2 Qualcomm filesystem super block, little endian, NAND version 0x2d
...
48234496      0x2E00000       UBI erase count header, version: 1, EC: 0x0, VID header offset: 0x1000, data offset: 0x2000
48238592      0x2E01000       UBI volume ID header, version: 1, type: 1, volume id: 0, size: 0
48242688      0x2E02000       Squashfs filesystem, little endian, version 4.0, compression:gzip
290697258     0x1153B02A      Zlib compressed data, default compression
290697345     0x1153B081      Zlib compressed data, default compression
```

Addresses are aligned

Qualcomm partition table

qcomsmempart.c

```
#define SMEM_FLASH_PART_MAGIC1      0x55ee73aa
#define SMEM_FLASH_PART_MAGIC2      0xe35ebddb

struct smem_flash_ptable {
    __le32 magic1;
    __le32 magic2;
    __le32 version;
    __le32 numparts;
    struct smem_flash_pentry pentry[SMEM_FLASH_PTABLE_MAX_PARTS_V4];
} __packed __aligned(4);

struct smem_flash_pentry {
    char name[SMEM_FLASH_PTABLE_NAME_SIZE];
    __le32 offset;
    __le32 length;
    u8 attr;
} __packed __aligned(4);
```

```
00281000:  AA 73 EE 55-DB BD 5E E3-04 00 00 00-14 00 00 00  ¬sεU╦╜^π♦   ¶
00281010:  30 3A 53 42-4C 00 00 00-00 00 00 00-00 00 00 00  0:SBL
00281020:  00 00 00 00-0A 00 00 00-FF 01 00 00-30 3A 4D 49      ◙    ☺  0:MI
00281030:  42 49 42 00-00 00 00 00-00 00 00 00-0A 00 00 00  BIB         ◙
00281040:  0A 00 00 00-FF 01 FF 00-30 3A 42 41-43 4B 55 50  ◙    ☺  0:BACKUP
00281050:  00 00 00 00-00 00 00 00-14 00 00 00-36 00 00 00          ¶   6
00281060:  FF 01 FF 00-30 3A 53 53-44 41 54 41-00 00 00 00   ☺  0:SSDATA
00281070:  00 00 00 00-4A 00 00 00-08 00 00 00-FF 01 FF 00      J   ◘    ☺
00281080:  30 3A 45 46-53 32 00 00-00 00 00 00-00 00 00 00  0:EFS2
00281090:  52 00 00 00-50 00 00 00-FF 01 FF 00-30 3A 54 5A  R   P    ☺  0:TZ
002810A0:  00 00 00 00-00 00 00 00-00 00 00 00-A2 00 00 00              ó
002810B0:  0C 00 00 00-FF 01 00 00-30 3A 52 50-4D 00 00 00  ♀    ☺  0:RPM
002810C0:  00 00 00 00-00 00 00 00-AE 00 00 00-0A 00 00 00          «   ◙
002810D0:  FF 01 00 00-30 3A 6D 6F-64 65 6D 00-00 00 00 00   ☺  0:modem
002810E0:  00 00 00 00-B8 00 00 00-A0 00 00 00-FF 01 00 00      ╕   á    ☺
002810F0:  30 3A 6D 6F-64 65 6D 32-00 00 00 00-00 00 00 00  0:modem2
00281100:  58 01 00 00-A0 00 00 00-FF 01 00 00-30 3A 61 62  X☺  á    ☺  0:ab
00281110:  6F 6F 74 00-00 00 00 00-00 00 00 00-F8 01 00 00  oot         °☺
00281120:  08 00 00 00-FF 01 00 00-30 3A 62 6F-6F 74 00 00  ◘    ☺  0:boot
00281130:  00 00 00 00-00 00 00 00-00 02 00 00-40 00 00 00           ☻  @
00281140:  FF 01 00 00-30 3A 73 79-73 74 65 6D-00 00 00 00   ☺  0:system
```

https://elixir.bootlin.com/linux/latest/source/drivers/mtd/parsers/qcomsmempart.c

Qualcomm partition table

Parsed Qualcomm partition table

```
0:SBL           0x00000000-0x00280000, 0x00280000 - System bootloader
0:MIBIB         0x00280000-0x00500000, 0x00280000 - Qualcomm partition table
0:BACKUP        0x00500000-0x01280000, 0x00d80000 - GOODARCHs (zlib modem configuration files)
0:SSDATA        0x01280000-0x01480000, 0x00200000 - Dual system shared data
0:EFS2          0x01480000-0x02880000, 0x01400000 - EFS2 file system
0:TZ            0x02880000-0x02b80000, 0x00300000 - ELF - Trust Zone
0:RPM           0x02b80000-0x02e00000, 0x00280000 - ELF - Resource Power Manager
0:modem         0x02e00000-0x05600000, 0x02800000 - UBI - modem (squashfs)
0:modem2        0x05600000-0x07e00000, 0x02800000 - UBI - modem (squashfs)
0:boot          0x08000000-0x09000000, 0x01000000 - ANDROID Kernel
0:aboot         0x07e00000-0x08000000, 0x00200000 - ELF - Linux bootloader (legatoproject)
0:system        0x09000000-0x0ae00000, 0x01e00000 - UBI - rootfs (squashfs - Linux Root File System)
0:LEFWKRO       0x0ae00000-0x0ed00000, 0x03f00000 - UBI - legato (squashfs - Legato application framework)
0:CUSTOMER0     0x0ed00000-0x12300000, 0x03600000 - (Empty FFFFFFF...)
0:aboot2        0x12300000-0x12500000, 0x00200000 - ELF - Linux bootloader
0:boot2         0x12500000-0x13500000, 0x01000000 - ANDROID Kernel
0:system2       0x13500000-0x15300000, 0x01e00000 - UBI - rootfs (squashfs - Linux Root File System)
0:LEFWKRO2      0x15300000-0x19200000, 0x03f00000 - UBI - legato (squashfs - Legato application framework)
0:CUSTOMER1     0x19200000-0x1c800000, 0x03600000 - UBI - (squashfs)
0:CUSTOMER2     0x1c800000-0x20000000, 0x03800000 - UBI
```

UBI - ubireader_extract_images

Extracted images from UBI

- modem, modem2
  - modem
- System, SYSTEM2
  - rootfs - squashfs
  - rootfs_hs - verity
  - rootfs_rhs - hash
  - rootfs_srhs - digital sign
  - rootfs_cert - certificate
- LEFFWKRO, LEFFWKRO2
  - legato – squashfs
  - legato_hs
  - legato_rhs
  - legato_srhs
  - legato_cert
- CUSTOMER0 – empty
- CUSTOMER1
  - cus0 – squashfs
  - cus0_hs
  - cus0_rhs
  - cus0_srhs
  - cus0_cert
- CUSTOMER2
  - data – UBIFS
  - swv - NVR data

```
dd if=./full_OCU.bin.ecc of=./modem.bin skip=$((0x02e00)) bs=$((0x1000)) count=$((0x02800))
0240+0 records in
10240+0 records out
41943040 bytes (42 MB, 40 MiB) copied, 1.40341 s, 29.9 MB/s


file modem.bin
modem.bin: UBI image, version 1

ubireader_extract_images ./modem.bin

file  ubifs-root/modem.bin/img-19046266_vol-modem.ubifs
ubifs-root/modem.bin/img-19046266_vol-modem.ubifs: Squashfs filesystem, little endian, version 1024.0

mount ubifs-root/modem.bin/img-19046266_vol-modem.ubifs /mnt/test/

ls -lha /mnt/test/
total 4.0K
drwxr-xr-x  3 10095 10003   28 Jan 20  2021 .
drwxr-xr-x 10 root  root  4.0K May 31 03:35 ..
drwxr-xr-x  2 10095 10003  656 Jan 20  2021 image

ls -lha /mnt/test/image/
total 60M
drwxr-xr-x 2 10095 10003  656 Jan 20  2021 .
drwxr-xr-x 3 10095 10003   28 Jan 20  2021 ..
-rwxr-xr-x 1 10095 10003  60K Jan 20  2021 btfw32.tlv
-rwxr-xr-x 1 10095 10003 2.0K Jan 20  2021 btnv32.bin
-rw-r--r-- 1 10095 10003  180 Jan 20  2021 mba.b00
-rw-r--r-- 1 10095 10003 6.6K Jan 20  2021 mba.b01
-rw-r--r-- 1 10095 10003 254K Jan 20  2021 mba.b02
-rw-r--r-- 1 10095 10003 1.2K Jan 20  2021 mba.b03
-rw-r--r-- 1 10095 10003 270K Jan 20  2021 mba.mbn
....
```

https://github.com/jrspruitt/ubi_reader.git

UBI - ubireader_extract_files

UBI file system

```
ubireader_extract_images ./CUSTOMER2.bin

ls -lha  ubifs-root/CUSTOMER2.bin/
total 44M
drwxrwxrwx 1 root root    0 Jun 17 03:39 .
drwxrwxrwx 1 root root    0 Jun 17 03:39 ..
-rwxrwxrwx 1 root root  42M Jun 17 03:49 img-1178379456_vol-data.ubifs
-rwxrwxrwx 1 root root 1.7M Jun 17 03:49 img-1178379456_vol-swv.ubifs


cd ubifs-root/CUSTOMER2.bin/

ubireader_extract_files ./img-1178379456_vol-data.ubifs -w
Extracting files to: ubifs-root
read Error: Block ends at 4337377279 which is greater than file size 43933696
read Error: Block ends at 896254237 which is greater than file size 43933696
extract_files Warning: Data may be missing or corrupted, bad blocks, LEB [167,13]

ls -lha ubifs-root/
total 21K
drwxrwxrwx 1 root root 4.0K Jun 17 03:54 .
drwxrwxrwx 1 root root 4.0K Jun 17 03:54 ..
drwxrwxrwx 1 root root    0 Dec 31  1969 audioman
drwxrwxrwx 1 root root    0 Jan 20  2021 caiman
-rwxrwxrwx 1 root root    0 Jun 17 03:54 .copied_default_data
drwxrwxrwx 1 root root 4.0K Jan 20  2021 ecall_data
drwxrwxrwx 1 root root    0 Jan 20  2021 etherman
drwxrwxrwx 1 root root 4.0K Jan 20  2021 flashman
drwxrwxrwx 1 root root    0 Dec 31  1969 health
drwxrwxrwx 1 root root    0 Dec 31  1969 java
```

BACKUP partition – extract archive

Based on parsecwe.pl

Big-endian

```
struct GoodHeader
{
    DWORD crc;
    DWORD type;
    DWORD magic1; //"GOOD"
    DWORD magic2; //"ARCH"
    DWORD unk1;
    DWORD size; //size of compressed data
    DWORD unk2;
    BYTE name[82]; //name of archive
    BYTE unk3[0x20];
    BYTE compressedData[size]; //zlib compressed data
};
```

Size

Zlib data

```
00500100:  F2 69 7A 00-00 00 03-47 4F 4F 44-41 52 43 48  ≥izº   ♥GOODARCH
00500110:  00 00 00 00-00 01 76 A7-26 99 7E EC-2F 73 77 69       ☺vº&Ö~∞/swi
00500120:  62 61 63 6B-75 70 2F 4E-56 42 4B 5F-46 55 4C 4C  backup/NVBK_FULL
00500130:  5F 4C 61 74-65 73 74 2E-32 31 30 00-00 00 00 00  _Latest.210
00500140:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00500150:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00500160:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00500170:  30 31 2F 30-36 2F 38 30-00 00 00 00-01 00 00 00  01/06/80    ☺
00500180:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00500190:  78 9C EC BD-09 98 23 57-79 2E AC D6-DE EA 59 ED  x£∞╜oÿ#Wy.¼╔▌ΩYφ
005001A0:  B1 F1 86 2D-CF 78 C5 EE-E9 6D 36 3B-04 AC 56 AB  ▒±å-╧x┼εΘm6;♦¼V½
005001B0:  BB E5 D6 D2-6E A9 BB 3D-83 A1 28 49-25 75 79 A4  ╗σ╓╥n⌐╗=âí(I%uyñ
005001C0:  2A 4D 55 A9-BB 67 42 72-B1 B1 09 24-24 31 C6 26  *MU⌐╗gBr▒▒o$$1╞&
005001D0:  84 4B C2 00-21 21 09 21-36 98 6C 97-0B 5C FE 84  äK┬ !!o!6ÿlùσ\■ä
005001E0:  4B 08 F9 E1-72 03 97 87-ED 21 81 90-3C 49 7E 42  K◘·ßr♥ùçφ!üÉ<I~B
005001F0:  08 37 0B 7F-80 FF FD CE-A9 D2 D2 52-2D EA F1 9F  ◘7♂⌂Ç ²╬⌐╥╥R-Ω±ƒ
```

https://github.com/eagleusb/wwan/blob/master/scripts/parsecwe.pl

BACKUP partition – extract files

```
GoodArchHeader header;
GoodArchBlock block[0];
    GoodArchItem item;
        GoodArchItemName name;
    GoodArchItem item;
        GoodArchItemValue value;
GoodArchBlock block[1];
    GoodArchItem item;
        GoodArchItemName name;
    GoodArchItem item;
        GoodArchItemValue value;
        ...
    ...
...
```

Little-endian

```
struct GoodArchHeader
{
    WORD ver;
    WORD numberOfBlocks;
    WORD unk0;
    DWORD unk1;
};

struct GoodArchBlock
{
    DWORD size;
    WORD unk0;
    WORD unk1;
};

struct GoodArchItem
{
    WORD type; //0 or 1
    DWORD sizeOfData;
};

//type = 0
struct GoodArchItemName
{
    BYTE flag;
    BYTE fullName[sizeOfData-1];
};

//type = 1
struct GoodArchItemValue
{
    BYTE data[sizeOfData];
};
```

GoodArchHeader GoodArchItem GoodArchBlock GoodArchItemName/Value

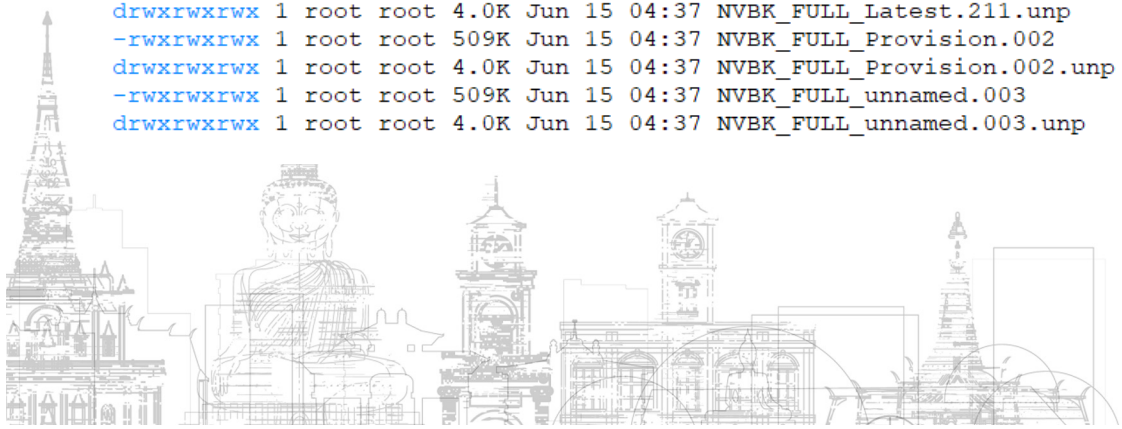
```
00000000: 00 00 94 00-00 00 00 00-00 00 30 00-00 00 02 00   ö♣      0   ☻
00000010: 01 00 01 00-13 00 00 00-00 2F 70 62-6D 5F 70 68  ☺ ☺ ‼    /pbm_ph
00000020: 6F 6E 65 5F-75 69 64 2E-64 61 74 02-00 09 00 00  one_uid.dat☻ ○
00000030: 00 00 00 00-00 00 00 00-01 38 00-00 00 03 00           ☺8   ♥
00000040: 01 00 01 00-20 00 00 00-01 2F 43 47-50 53 5F 4D  ☺ ☺     ☺/CGPS_M
00000050: 45 2F 43 47-50 53 43 65-6C 6C 44 42-43 6F 6D 6D  E/CGPSCellDBComm
00000060: 69 74 52 65-63 6F 72 64-02 00 04 00-00 00 0B 00  itRecord☻ ♦   ♂
00000070: 00 00 8C 13-00 00 02 00-01 00 01 00-18 00 00 00    î‼  ☻ ☺ ☺ ↑
00000080: 00 2F 43 47-50 53 5F 4D-45 2F 43 47-50 53 43 65   /CGPS_ME/CGPSCe
00000090: 6C 6C 44 42-46 69 6C 65-02 00 60 13-00 00 00 00  llDBFile☻ `‼
000000A0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
000000B0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
000000C0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
000000D0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
000000E0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
000000F0: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000100: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000110: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000120: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000130: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000140: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

https://github.com/eagleusb/wwan/blob/master/scripts/parsecwe.pl

BACKUP partition - files

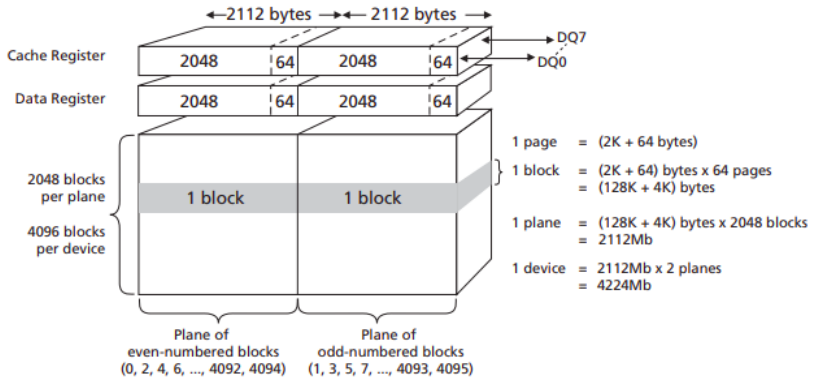
```
ls -lha swibackup/
total 3.5M
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 .
drwxrwxrwx 1 root root  48K Jun 17 03:38 ..
-rwxrwxrwx 1 root root 321K Jun 15 04:37 NVBK_FULL_Factory.001
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 NVBK_FULL_Factory.001.unp
-rwxrwxrwx 1 root root 509K Jun 15 04:37 NVBK_FULL_Latest.110
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 NVBK_FULL_Latest.110.unp
-rwxrwxrwx 1 root root 531K Jun 15 04:37 NVBK_FULL_Latest.111
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 NVBK_FULL_Latest.111.unp
-rwxrwxrwx 1 root root 509K Jun 15 04:37 NVBK_FULL_Latest.210
drwxrwxrwx 1 root root 4.0K Jun 15 04:33 NVBK_FULL_Latest.210.unp
-rwxrwxrwx 1 root root 531K Jun 15 04:37 NVBK_FULL_Latest.211
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 NVBK_FULL_Latest.211.unp
-rwxrwxrwx 1 root root 509K Jun 15 04:37 NVBK_FULL_Provision.002
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 NVBK_FULL_Provision.002.unp
-rwxrwxrwx 1 root root 509K Jun 15 04:37 NVBK_FULL_unnamed.003
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 NVBK_FULL_unnamed.003.unp
```

```
ls -lha swibackup/NVBK_FULL_Factory.001.unp/
total 25K
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 .
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 ..
drwxrwxrwx 1 root root    0 Jun 15 04:37 apn_throttle
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 CGPS_ME
drwxrwxrwx 1 root root    0 Jun 15 04:37 Data_Profiles
drwxrwxrwx 1 root root    0 Jun 15 04:37 ds
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 ecall
drwxrwxrwx 1 root root    0 Jun 15 04:37 nv
drwxrwxrwx 1 root root    0 Jun 15 04:37 nvm
-rwxrwxrwx 1 root root    9 Jun 15 04:37 pbm_phone_uid.dat
drwxrwxrwx 1 root root    0 Jun 15 04:37 rfc
drwxrwxrwx 1 root root    0 Jun 15 04:37 rpm
drwxrwxrwx 1 root root    0 Jun 15 04:37 safe
drwxrwxrwx 1 root root    0 Jun 15 04:37 scrub
drwxrwxrwx 1 root root    0 Jun 15 04:37 sd
drwxrwxrwx 1 root root    0 Jun 15 04:37 selfcal
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 sms
drwxrwxrwx 1 root root 4.0K Jun 15 04:37 swilog
drwxrwxrwx 1 root root    0 Jun 15 04:37 swinv
```



NAND flash JY990 – Page structure

MT29RZ4B2DZZHHWD-18I.84F (MT29F4G08) - Micron, 4Gb



Page structure

```
struct Chunk0
{
    BYTE data0[0x1D0];
    BYTE meta0[1];
    BYTE data1[0x34];
    BYTE ecc[0x7];
    BYTE meta1[4];
};

struct Chunk1
{
    BYTE data0[0x1D0];
    BYTE meta0[1];
    BYTE data1[0x24];
    BYTE meta1[0x10];
    BYTE ecc[0x7];
    BYTE meta2[4];
};

struct Page
{
    Chunk0 chunk[3];
    Chunk1 chunk[1];
}
```

ECC parameters

- BCH - algo
- BCH_BITS - 4
- POLY - 8219
- Reverse - False

NAND flash JY990 – File system

Offset Qualcomm partition table 0x220800

```
0:SBL        0x00000000-0x00280000, 0x00280000
0:MIBIB      0x00280000-0x00500000, 0x00280000
0:EFS2       0x00500000-0x01b80000, 0x01680000
0:SCRUB      0x01b80000-0x02a00000, 0x00e80000
0:TZ         0x02a00000-0x02b40000, 0x00140000
0:RPM        0x02b40000-0x02bc0000, 0x00080000
0:aboot      0x02bc0000-0x02d00000, 0x00140000
0:boot       0x02d00000-0x03a00000, 0x00d00000
0:recovery   0x03a00000-0x04700000, 0x00d00000
0:sec        0x04700000-0x04740000, 0x00040000
0:misc       0x04740000-0x048c0000, 0x00180000
0:UBI        0x048c0000-0x40000000, 0x3b740000
```

UBI file system

```
img-1269926124_vol-bootctl.ubifs
img-1269926124_vol-cache.ubifs
img-1269926124_vol-data.ubifs
img-1269926124_vol-dsp2.ubifs
img-1269926124_vol-system.ubifs
img-1269926124_vol-uarea.ubifs
```

```
drwxr-xr-x 23 yuriy yuriy 4,0K nov 16  2022 .
drwxr-xr-x  6 yuriy yuriy 4,0K nov 16  2022 ..
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 audio
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 ble_data
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 data_test
-rw-r--r--  1 yuriy yuriy   16 jan  1  1970 debug_state.bin
drwxr-xr-x  2 yuriy yuriy 4,0K jun 12  2020 diag_logs
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 dlt_logs
drwxr-xr-x  2 yuriy yuriy  12K jan 14  1970 ECall_persistency
drwxr-xr-x  6 yuriy yuriy 4,0K nov 16  2020 .efs
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 efs
-rw-r--r--  1 yuriy yuriy   56 jan  1  1970 efs_hal.key
-rw-r--r--  1 yuriy yuriy   56 jan  1  1970 efs.key
drwxr-xr-x  4 yuriy yuriy 4,0K jan  1  1970 flash_scrub
drwxr-xr-x  4 yuriy yuriy 4,0K jan  1  1970 log
drwxr-xr-x  3 yuriy yuriy 4,0K jan  1  1970 misc
-rw-r--r--  1 yuriy yuriy    0 dec 18  2020 mqtt_ack_decode_trace.txt
-rw-r--r--  1 yuriy yuriy   16 dec  1  2020 nadif_ecall_persistency
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 on_device_logging
drwxr-xr-x  5 yuriy yuriy 4,0K nov 16  2020 persistency
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 .privacy
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 privacy
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 provisioning
drwxr-xr-x  2 yuriy yuriy 4,0K feb  8  2021 removed_by_factory_reset
-rw-r--r--  1 yuriy yuriy    1 okt 27  2020 .scrub_direction_file
drwxr-xr-x  2 yuriy yuriy 4,0K jan 14  1970 swl
drwxr-xr-x  2 yuriy yuriy 4,0K jan  1  1970 tcam
drwxr-xr-x  2 yuriy yuriy 4,0K jun 12  2020 wav
drwxr-xr-x  3 yuriy yuriy 4,0K nov 16  2022 Wifi
```

Conclusion

- We removed the chip from PCB despite the problems with glue
- We built the reader for non-standard flash NAND memory chip
- We discovered a bug in opensource tool NandTool
- NAND flash of IAU has a custom implementation, but we were able to recover the format of pages and OOB relying only on memory dump.
- IAU uses Qualcomm partition table, we successfully parsed all information of flash memory.
- We extracted all partitions with UBI file system
- We described the format GOODARCH containers and extracted files system from BACKUP partition.

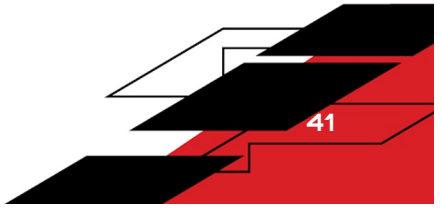
Future work

We obtain firmware and extracted file systems from IAU.
What we are planning to do next:

- Obtain SSH/ADB access to IAU
- Security assessment of device
  - Communication interfaces
  - Updating procedure
  - Company backend

Related links

- *Reverse Engineering Flash Memory for Fun and Benefit*
- *I.MX MEMORY MADNESS*
- *How To Do Firmware Analysis. Tools, Tips, and Tricks*
- *NAND Flash with Mobile LPDDR2 162-Ball MCP - MT29RZ4B4DZZMGWD*
- *https://github.com/eagleusb/wwan/blob/master/scripts/parsecwe.pl*
- *https://github.com/jrspruitt/ubi_reader*
- *https://elixir.bootlin.com/linux/latest/source/drivers/mtd/parsers/qcomsmempart.c*



THANK YOU!

NAVINFO