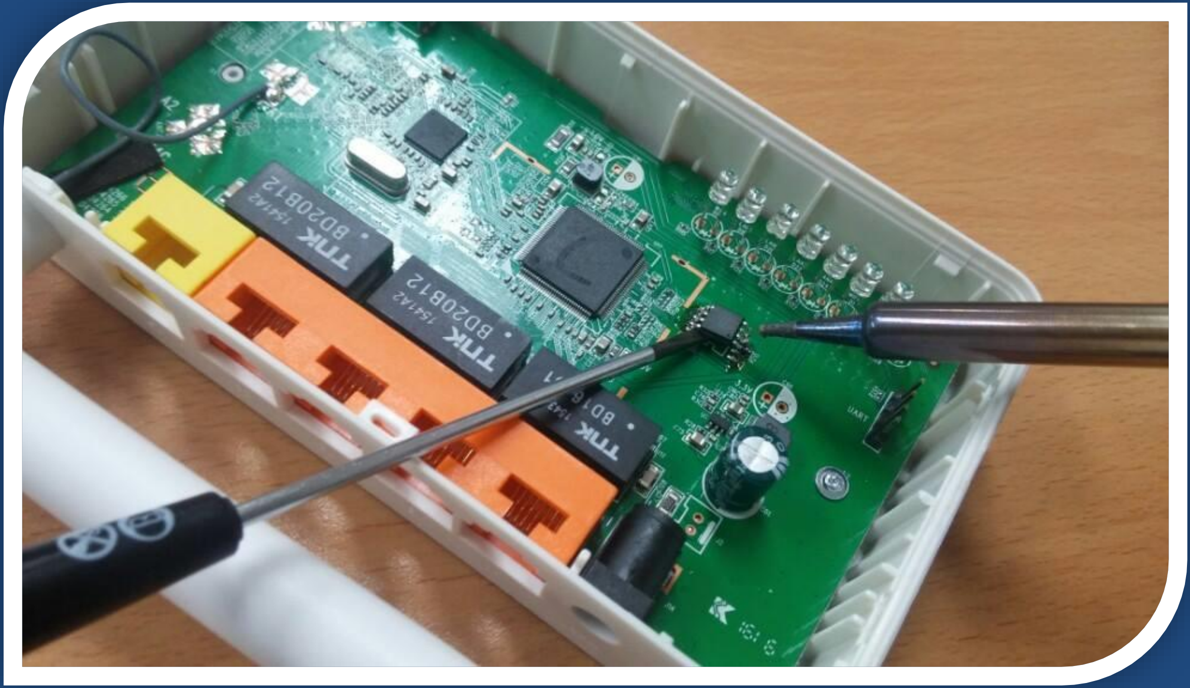


Flash Memory Dump 실습

mongii@grayhash

개요

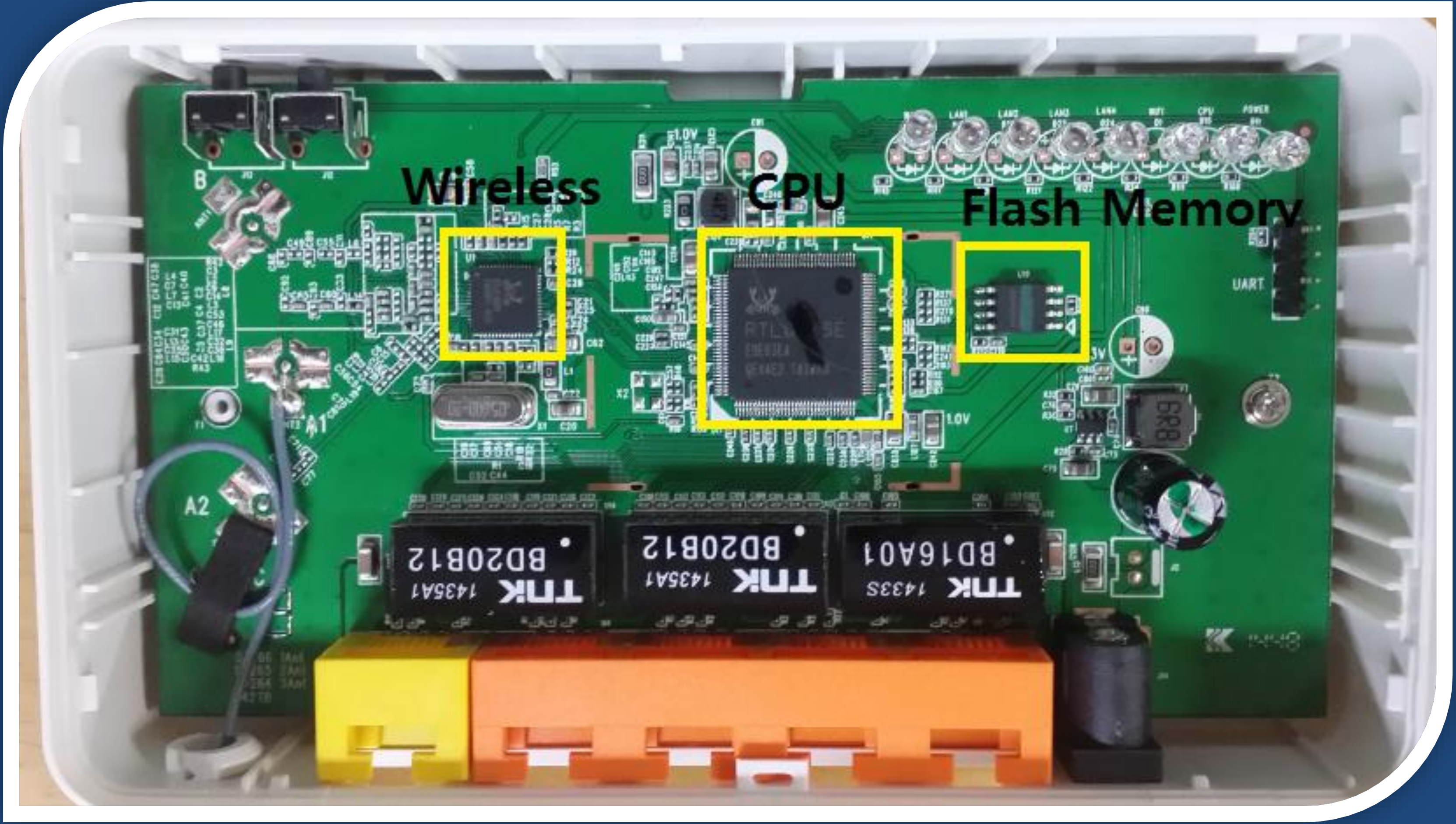
- 임베디드 장비가 사용하는 운영체제 및 어플리케이션 코드들은 대부분 Flash Memory에 저장된다.

- Flash Memory Dumper 구현을 통해 작동 방식을 이해하고, CPU 주변의 IC들을 자유롭게 제어할 수 있는 개발 능력을 향상시킨다.

- IC의 Datasheet를 이해하고 활용하는 방법을 습득한다.

목차

IPTIME N104p의 HW 구조



IPTIME N104p의 HW 구조

- CPU
  - Realtek RTL8196E (MIPS 400MHz)
- Flash Memory
  - Winbond S02157
    - 2 Mbytes
- 무선 네트워크 칩
  - RealTek RTL8188ER
    - Wireless LAN (WLAN) network interface

Flash Memory란?

- 전기적으로 데이터를 지우거나 기록할 수 있는 비휘발성 기억 장치
- Flash(일 순간에 번쩍)하게 삭제(혹은 읽기/쓰기) 가능한 저장장치

- 1984년 도시바의 마스오카 후지오 박사가 발명
- 1988년 인텔에서 최초의 상용 제품 출시

- 소형 저장장치의 발전 단계
  - ROM (Read Only Memory)
  - PROM (Programmable ROM)
  - EPROM (Erasable PROM)
  - EEPROM (Electrically Erasable Programmable Read-Only Memory)
  - Flash Memory (Flash EEPROM)

Flash Memory 속도

- DRAM, SRAM에 비해 매우 느림
- 일반 자기 하드디스크에 비해서도 느림

- SSD가 빠른 이유는?
  - 다 수의 Flash Memory를 병렬로 연결
  - 이론상 N개를 연결할 수록 속도는 N배가 된다

Flash memory에 데이터 쓰고 읽기

- ROM Writer
  - Firmware Writing 전용 장비
- JTAG
  - Hardware Debugging 장비

Flash Memory Dump 절차

1. Flash Memory Chip Desoldering
2. Datasheet 획득 및 학습
3. Flash Memory Dumper 제작
4. Flash Memory Data 추출
5. Data(Hex) -> Binary 변환
6. Firmware 코드 분석

Flash Memory Desoldering

방법1

- Hot air gun 사용

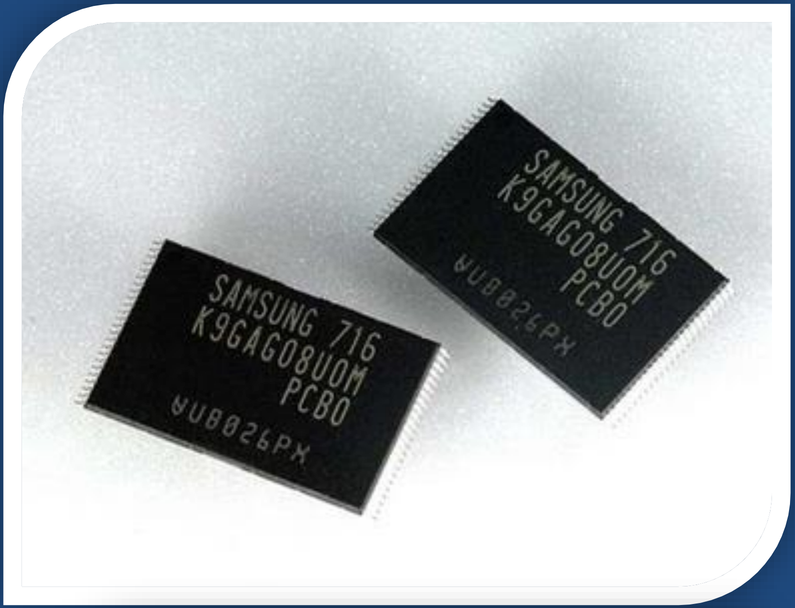
Hot Air Gun 사용

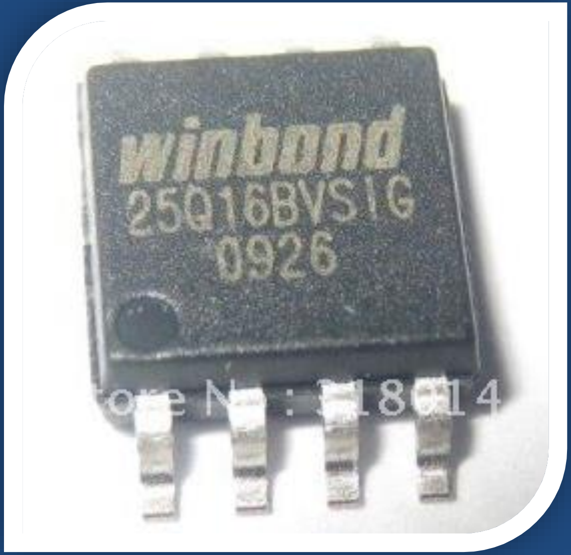
- https://www.youtube.com/watch?v=dBzQwnlp_yk

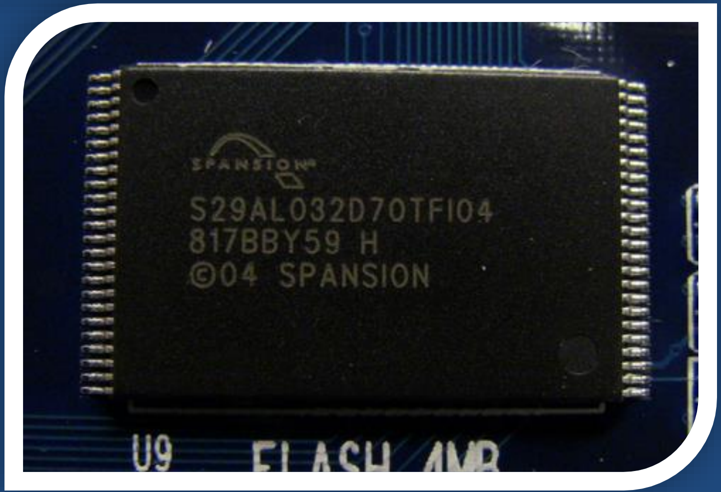
방법2

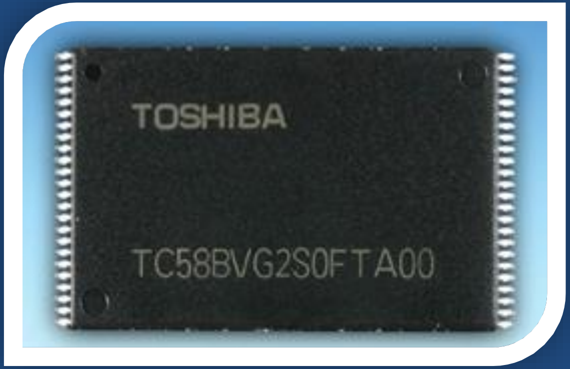
- 인두기 사용
- https://www.youtube.com/watch?v=ogd3KFfE0Cg

플래시 모델명 알아내기

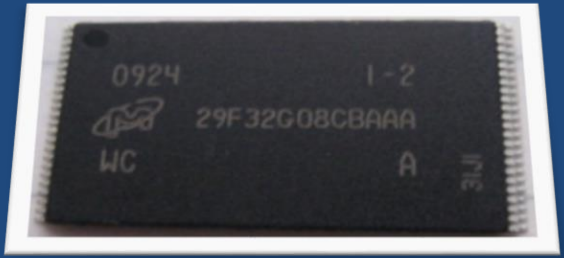

SAMSUNG 716
K9GAG08U0M
PCB0
AU8026PX
SAMSUNG 716
K9GAG08U0M
PCB0
AU8026PX


winbond
25Q16BVSIG
0926


SPANSION
S29AL032D70TFI04
817BBY59 H
©04 SPANSION
U9
FLASH 4MB


TOSHIBA
TC58BVG2S0FTA00

Logo로 알아보는 제조사



SPANSION

AMD

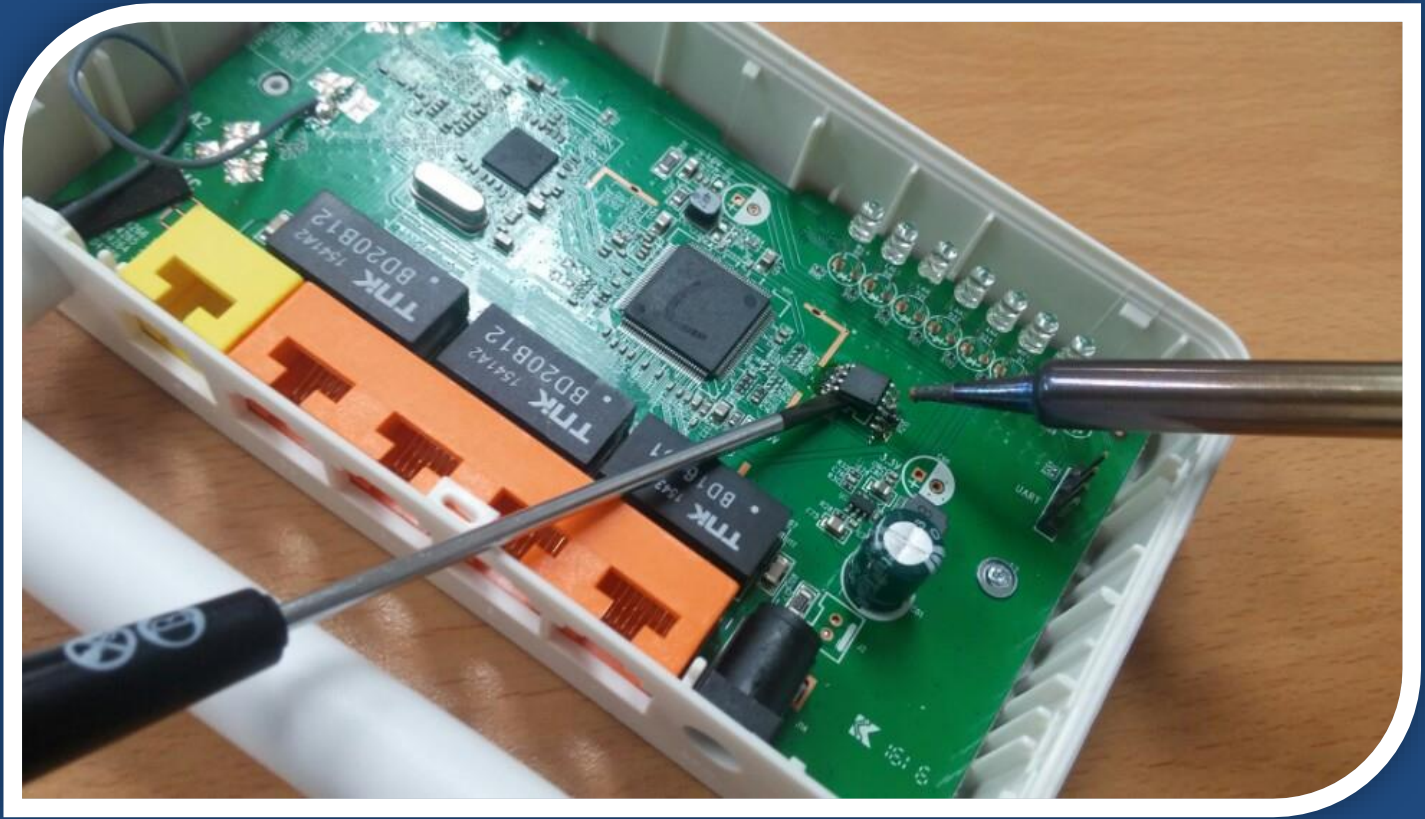
Micron

SAMSUNG

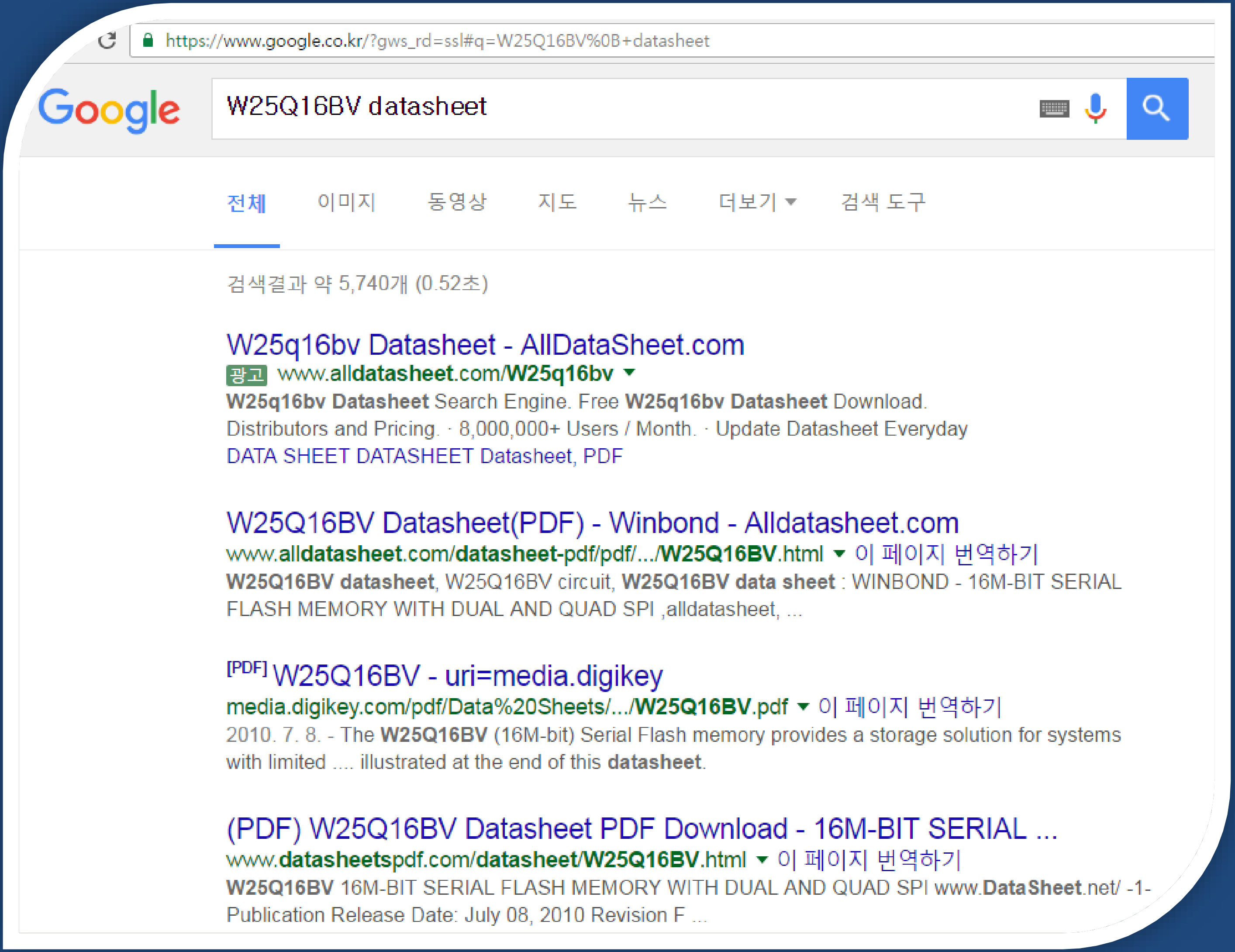
MARVELL

Winbond
Electronics Corp.

Desoldering 실습



데이터 시트 찾기


https://www.google.co.kr/?gws_rd=ssl#q=W25Q16BV%0B+datasheet

W25Q16BV datasheet

검색결과 약 5,740개 (0.52초)

W25q16bv Datasheet - AllDataSheet.com
광고 www.alldatasheet.com/W25q16bv
W25q16bv Datasheet Search Engine. Free **W25q16bv Datasheet** Download. Distributors and Pricing. · 8,000,000+ Users / Month. · Update Datasheet Everyday
DATA SHEET DATASHEET Datasheet, PDF

W25Q16BV Datasheet(PDF) - Winbond - Alldatasheet.com
www.alldatasheet.com/datasheet-pdf/pdf/.../W25Q16BV.html ▾ 이 페이지 번역하기
W25Q16BV datasheet, W25Q16BV circuit, **W25Q16BV data sheet** : WINBOND - 16M-BIT SERIAL FLASH MEMORY WITH DUAL AND QUAD SPI ,alldatasheet, ...

[PDF] W25Q16BV - uri=media.digikey
media.digikey.com/pdf/Data%20Sheets/.../W25Q16BV.pdf ▾ 이 페이지 번역하기
2010. 7. 8. - The **W25Q16BV** (16M-bit) Serial Flash memory provides a storage solution for systems with limited .... illustrated at the end of this **datasheet**.

(PDF) W25Q16BV Datasheet PDF Download - 16M-BIT SERIAL ...
www.datasheetspdf.com/datasheet/W25Q16BV.html ▾ 이 페이지 번역하기
W25Q16BV 16M-BIT SERIAL FLASH MEMORY WITH DUAL AND QUAD SPI www.**DataSheet**.net/ -1- Publication Release Date: July 08, 2010 Revision F ...

데이터 시트 찾기

W25Q16BV

winbond

spiflash®

16M-BIT
SERIAL FLASH MEMORY WITH
DUAL AND QUAD SPI

Winbond W25Q16BV Serial Flash Memory

Spec 요약

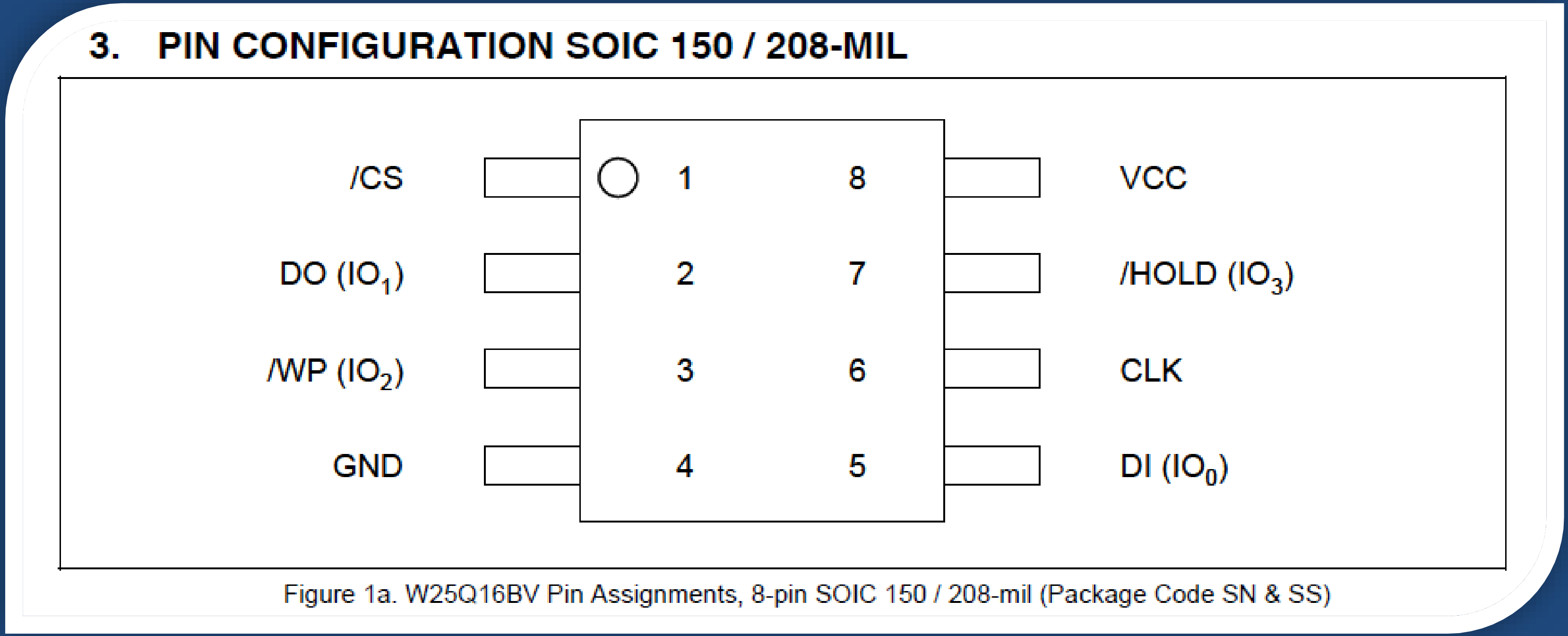
- 통신 방식 : SPI
  - Serial Peripheral Interface
    - Clock에 맞추어 데이터 송수신
- 용량 : 16Mbit (2MB)
- 패키지 방식 : SOIC
- 작동 전압 : 3.3v
- Cell 저장 방식 : NAND

NAND vs NOR

- NAND
  - 각 메모리 Cell이 직렬 형태로 이루어짐
  - Read 속도는 느리고, Write/Erase는 속도는 빠름
  - 데이터 영역으로 적합 (ex. Data 파티션, 이동식 장치)
- NOR
  - 각 메모리 Cell이 병렬 형태로 이루어짐
  - Read 속도는 빠르고, Write/Erase 속도는 느림
  - 코드 영역으로 적합 (ex. OS 파티션)
  - NAND보다 비쌈

PIN 목록

- 총 8개의 핀 사용



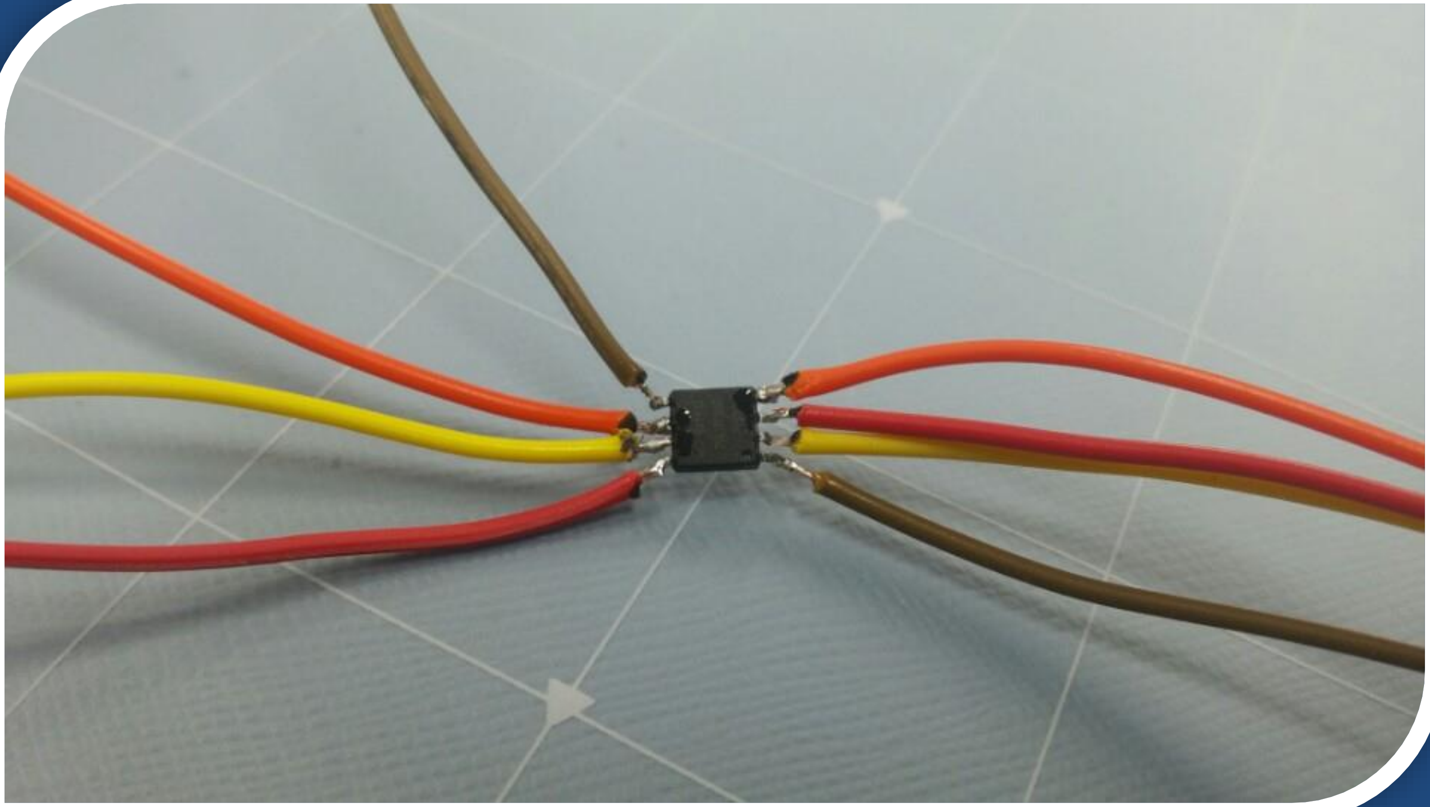
3. **PIN CONFIGURATION SOIC 150 / 208-MIL**

Figure 1a. W25Q16BV Pin Assignments, 8-pin SOIC 150 / 208-mil (Package Code SN & SS)

PIN 목록

- VCC, GND : 전원 제공
- /CS : 대상 Flash Memory 선택
- DO : Data Out (칩으로부터 데이터 출력)
- DI : Data In (칩으로 데이터 입력)
- CLK : Clock 제공
- WP# : 쓰기 방지 기능
- HOLD# : Pause 기능

각 PIN에 케이블 연결



Flash ←→ 아두이노 핀 연결 구성

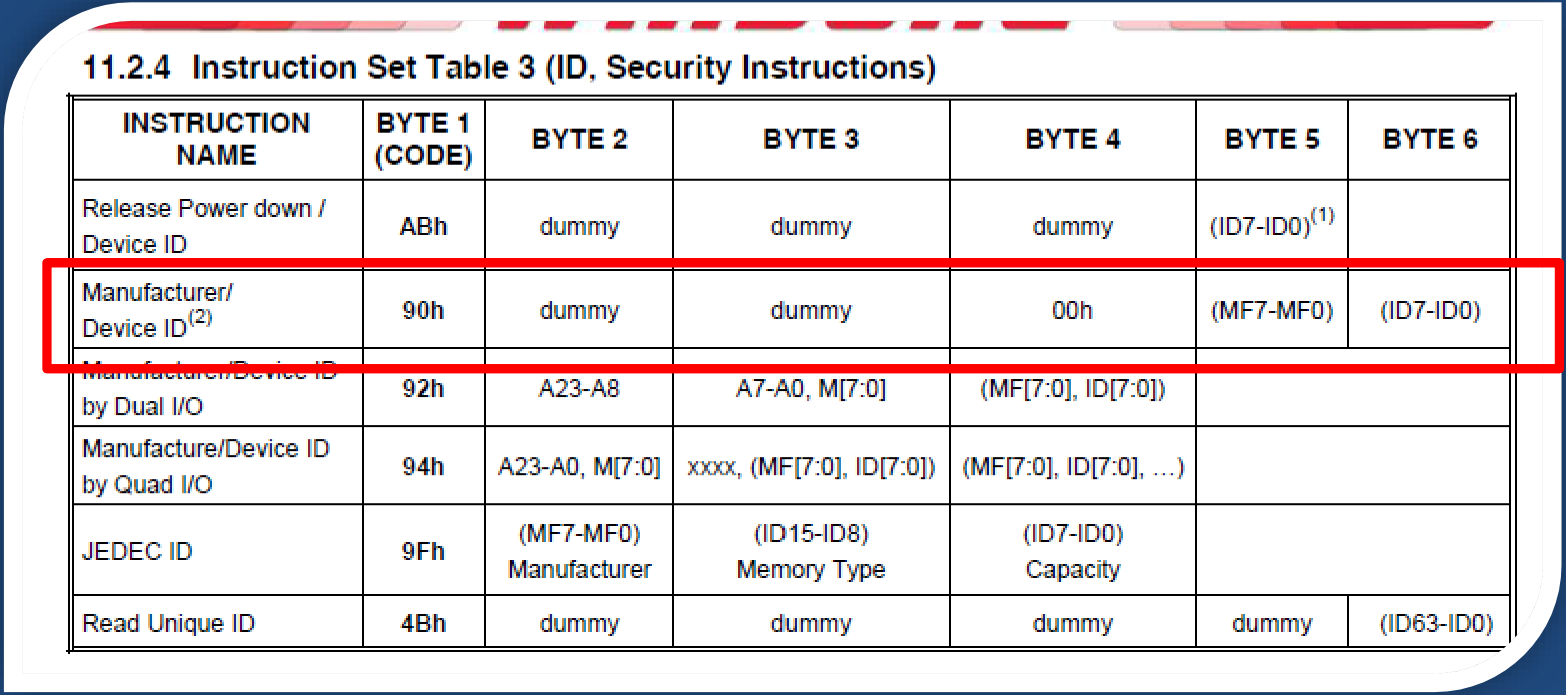
- 1번 핀(/CS) ←→ 2번 : HIGH->LOW 변경
- 2번 핀(DO) ←→ 3번 : Data Out
- 3번 핀(/WP) ←→ 4번 : 항상 HIGH
- 4번 핀(GND) ←→ 5번 : 항상 LOW
- 5번 핀(DI) ←→ 6번 : Data In
- 6번 핀(CLK) ←→ 7번 : CLOCK
- 7번 핀(/HOLD) ←→ 8번 : 항상 HIGH
- 8번 핀(VCC) ←→ 3.3v

W25Q16BV 제어 기본 규칙

COMMAND + ADDRESS + DATA

ID 코드 읽기

- 제조사의 ID를 읽을 수 있다.
- 이 값이 잘 읽힌다면 회로에 문제가 없는 것이다.

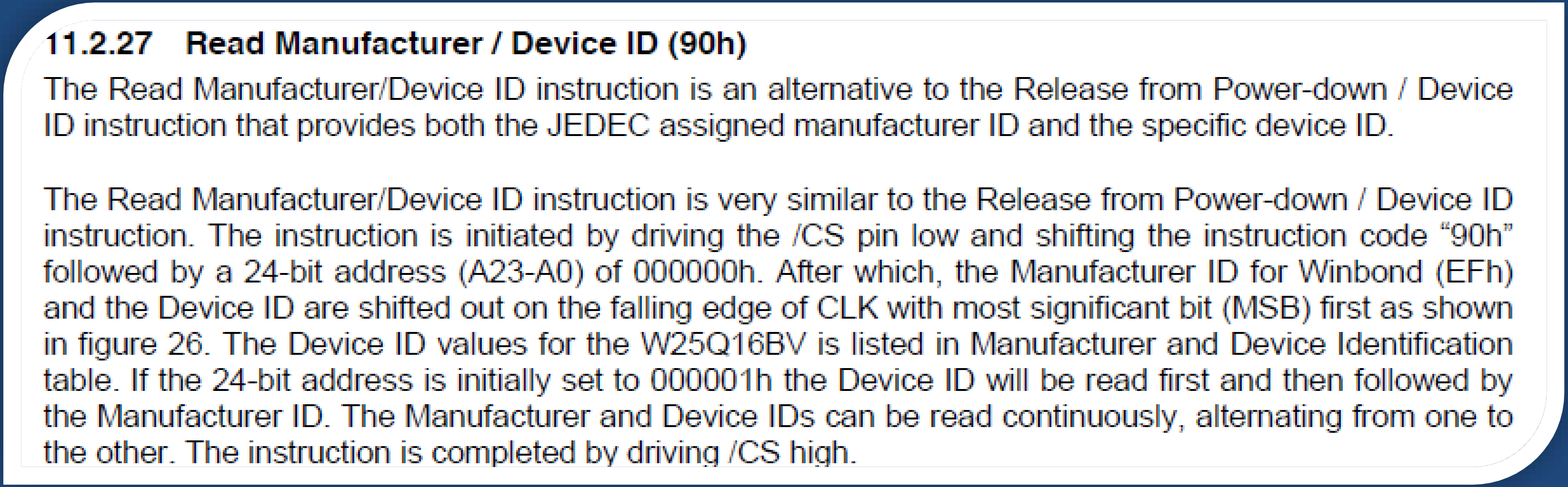


11.2.4 Instruction Set Table 3 (ID, Security Instructions)

INSTRUCTION NAME	BYTE 1 (CODE)	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
Release Power down / Device ID	ABh	dummy	dummy	dummy	(ID7-ID0)(1)	
Manufacturer/ Device ID(2)	90h	dummy	dummy	00h	(MF7-MF0)	(ID7-ID0)
Manufacturer/Device ID by Dual I/O	92h	A23-A8	A7-A0, M[7:0]	(MF[7:0], ID[7:0])		
Manufacture/Device ID by Quad I/O	94h	A23-A0, M[7:0]	xxxx, (MF[7:0], ID[7:0])	(MF[7:0], ID[7:0], …)		
JEDEC ID	9Fh	(MF7-MF0) Manufacturer	(ID15-ID8) Memory Type	(ID7-ID0) Capacity		
Read Unique ID	4Bh	dummy	dummy	dummy	dummy	(ID63-ID0)

ID 코드 읽기

- 0x90 instruction == ID 코드 읽기
- 24bit의 0이 다음으로 이어진다.
- 다음으로 DO 핀에 ID가 shift된다.



11.2.27 Read Manufacturer / Device ID (90h)

The Read Manufacturer/Device ID instruction is an alternative to the Release from Power-down / Device ID instruction that provides both the JEDEC assigned manufacturer ID and the specific device ID.

The Read Manufacturer/Device ID instruction is very similar to the Release from Power-down / Device ID instruction. The instruction is initiated by driving the /CS pin low and shifting the instruction code "90h" followed by a 24-bit address (A23-A0) of 000000h. After which, the Manufacturer ID for Winbond (EFh) and the Device ID are shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in figure 26. The Device ID values for the W25Q16BV is listed in Manufacturer and Device Identification table. If the 24-bit address is initially set to 000001h the Device ID will be read first and then followed by the Manufacturer ID. The Manufacturer and Device IDs can be read continuously, alternating from one to the other. The instruction is completed by driving /CS high.

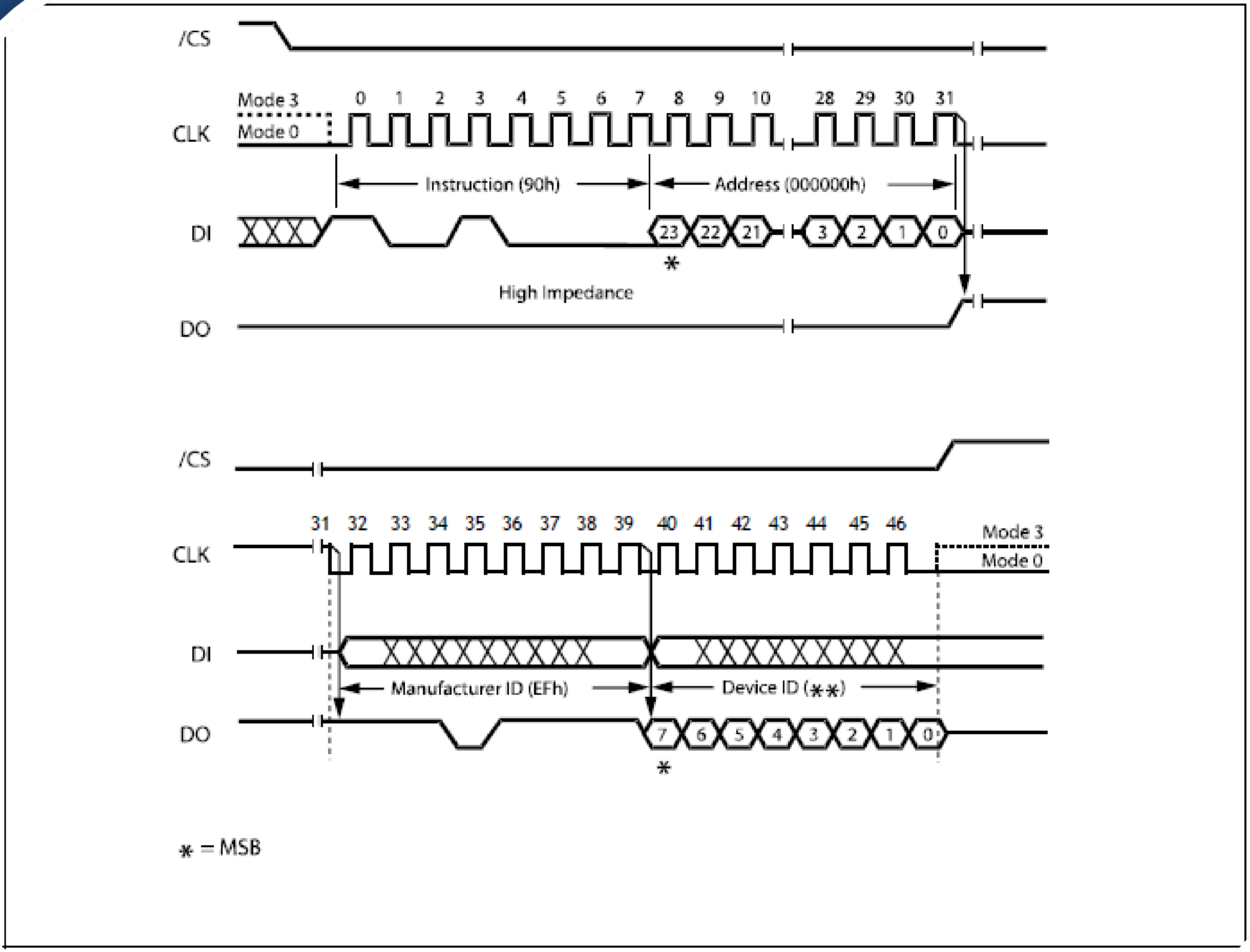
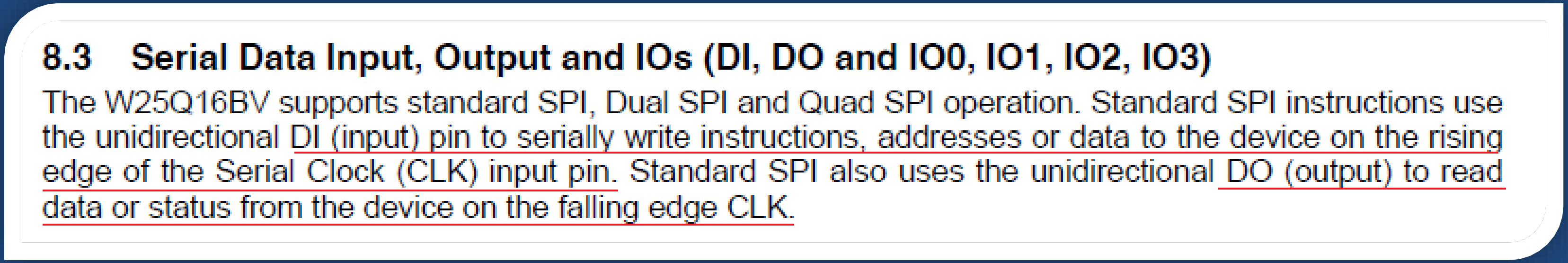


Figure 26. Read Manufacturer / Device ID Instruction Sequence Diagram

Rising/Falling Edge

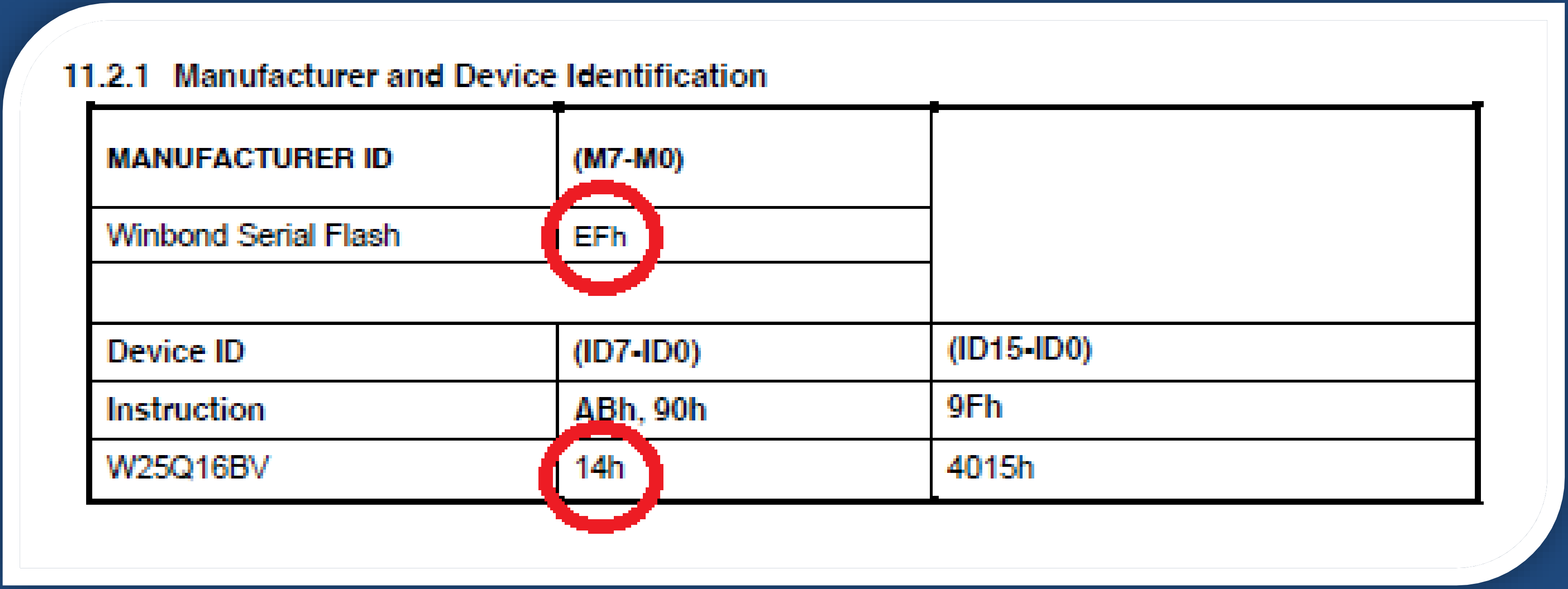
- MCU가 DATA를 보낼 때(DI) : Rising Edge
- MCU가 DATA를 받을 때(DO) : Falling Edge


8.3 Serial Data Input, Output and IOs (DI, DO and IO0, IO1, IO2, IO3)

The W25Q16BV supports standard SPI, Dual SPI and Quad SPI operation. Standard SPI instructions use the unidirectional DI (input) pin to serially write instructions, addresses or data to the device on the rising edge of the Serial Clock (CLK) input pin. Standard SPI also uses the unidirectional DO (output) to read data or status from the device on the falling edge CLK.

실습

- ID 코드를 읽어서 UART로 출력해 보세요.
  - 제조사 ID, 디바이스 ID
  - 11101111(HHHLHHHH), 00010100(LLHLHLL)



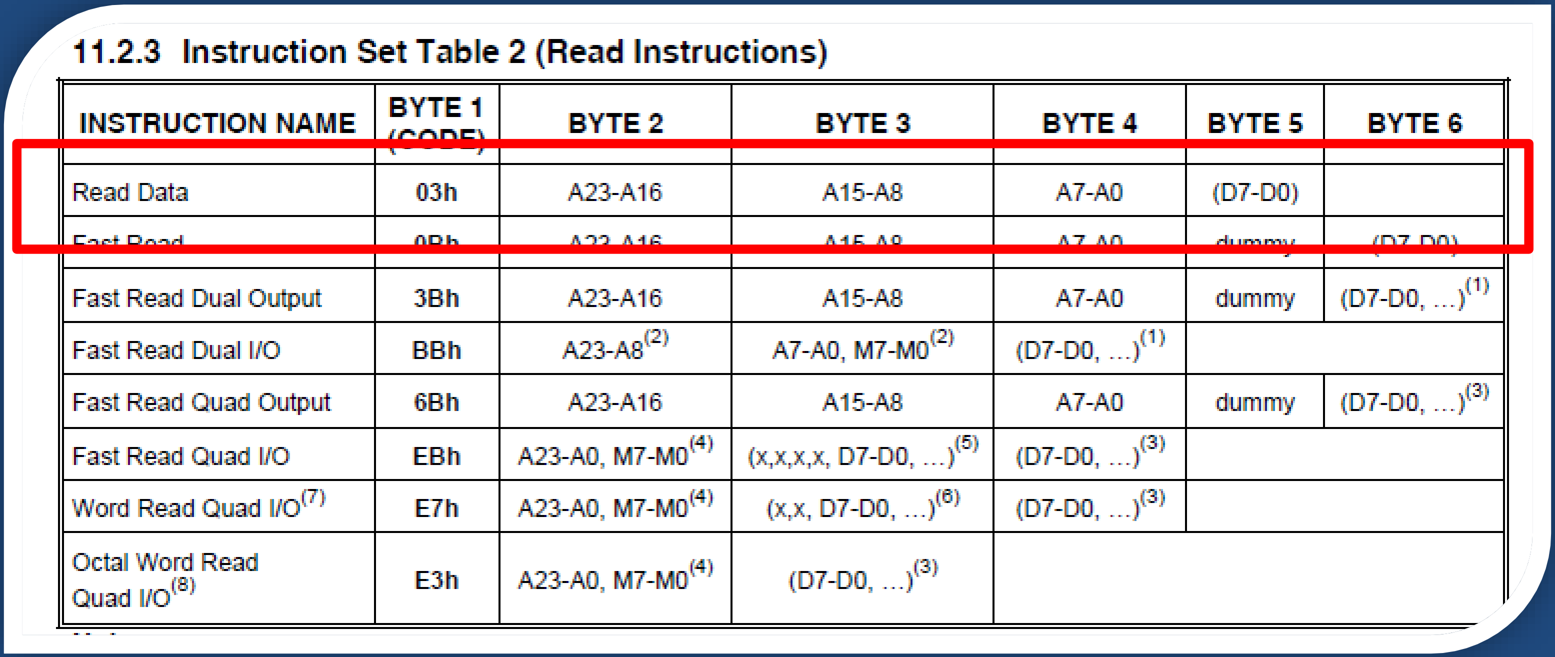
11.2.1 Manufacturer and Device Identification

MANUFACTURER ID	**(M7-M0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7-ID0)**	**(ID15-ID0)**
Instruction	ABh, 90h	9Fh
W25Q16BV	14h	4015h

Flash Memory Data 읽기

읽기 커맨드

- 기본적인 방식인 Read Data로 실습 진행
- 최대 주소 : 3바이트==0xFFFFFF==16,777,215



11.2.3 Instruction Set Table 2 (Read Instructions)

INSTRUCTION NAME	BYTE 1 (CODE)	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	
Fast Read	0Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)
Fast Read Dual Output	3Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0, …)(1)
Fast Read Dual I/O	BBh	A23-A8(2)	A7-A0, M7-M0(2)	(D7-D0, …)(1)		
Fast Read Quad Output	6Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0, …)(3)
Fast Read Quad I/O	EBh	A23-A0, M7-M0(4)	(x,x,x,x, D7-D0, …)(5)	(D7-D0, …)(3)		
Word Read Quad I/O(7)	E7h	A23-A0, M7-M0(4)	(x,x, D7-D0, …)(6)	(D7-D0, …)(3)		
Octal Word Read Quad I/O(8)	E3h	A23-A0, M7-M0(4)	(D7-D0, …)(3)			

읽기 커맨드 타이밍 차트

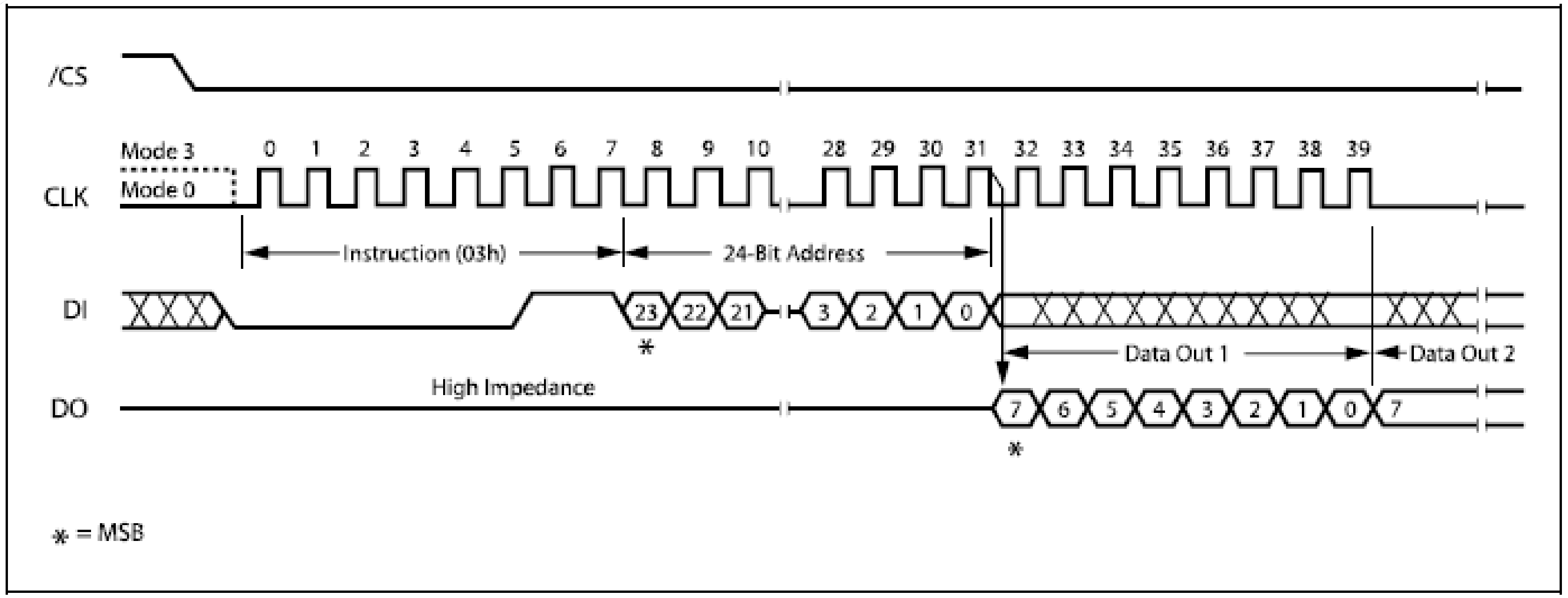
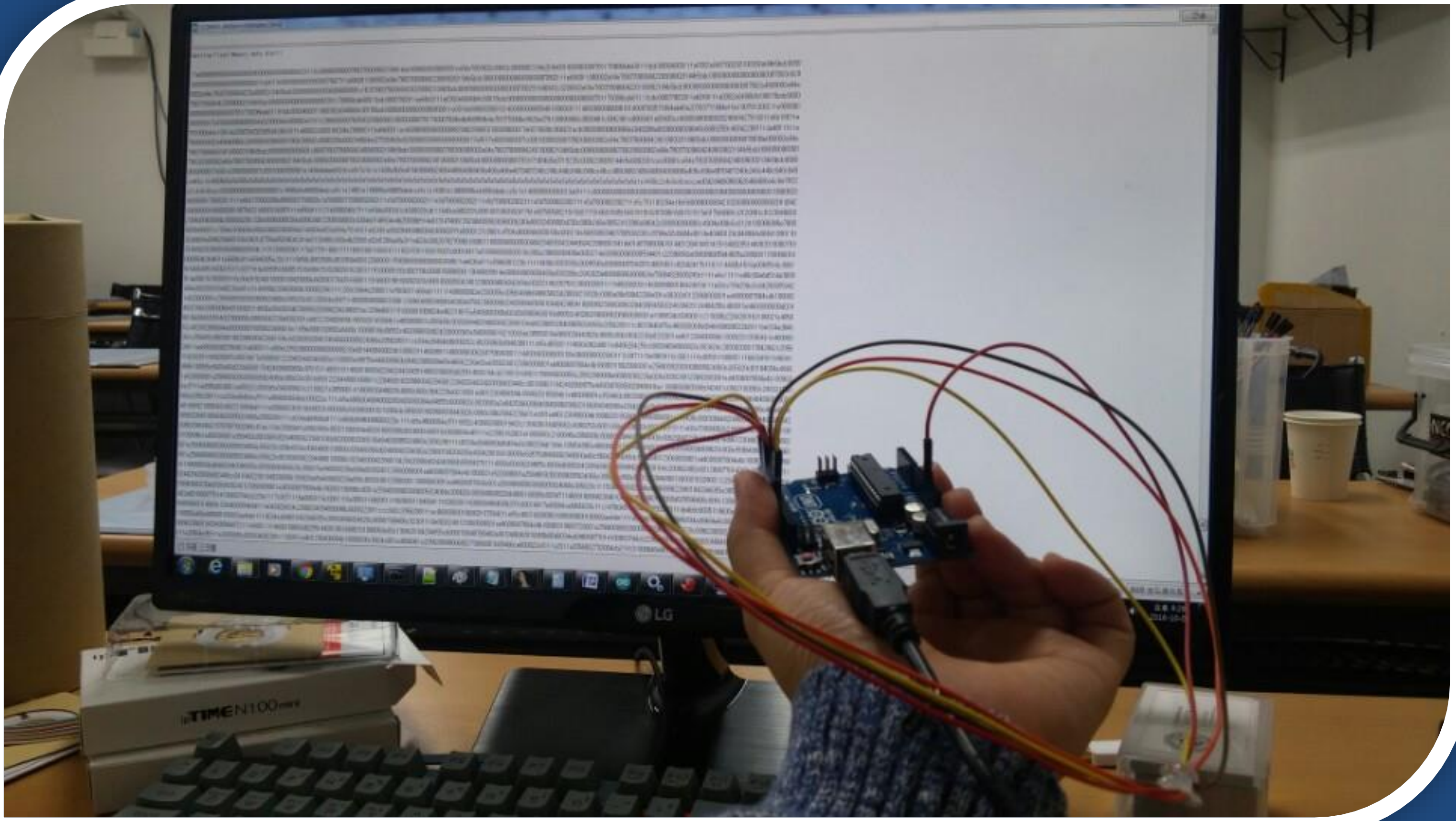


Figure 8. Read Data Instruction Sequence Diagram

실습

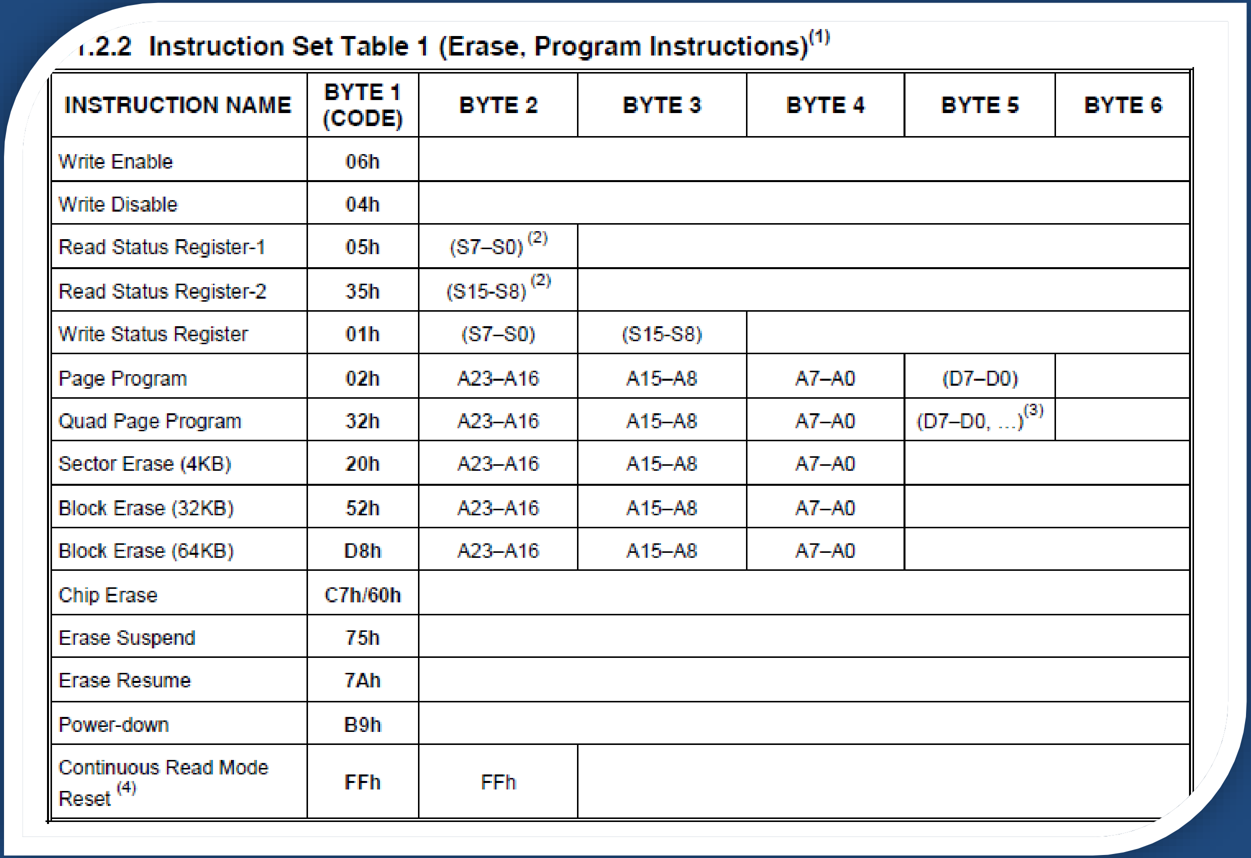
- IPTIME의 펌웨어를 획득해 보세요!

Flash Memory 덤프 화면



쓰기, 삭제 커맨드

- Write 기능 활성화 + 쓰기 순서로 진행된다.
- Write Enable + Page Program

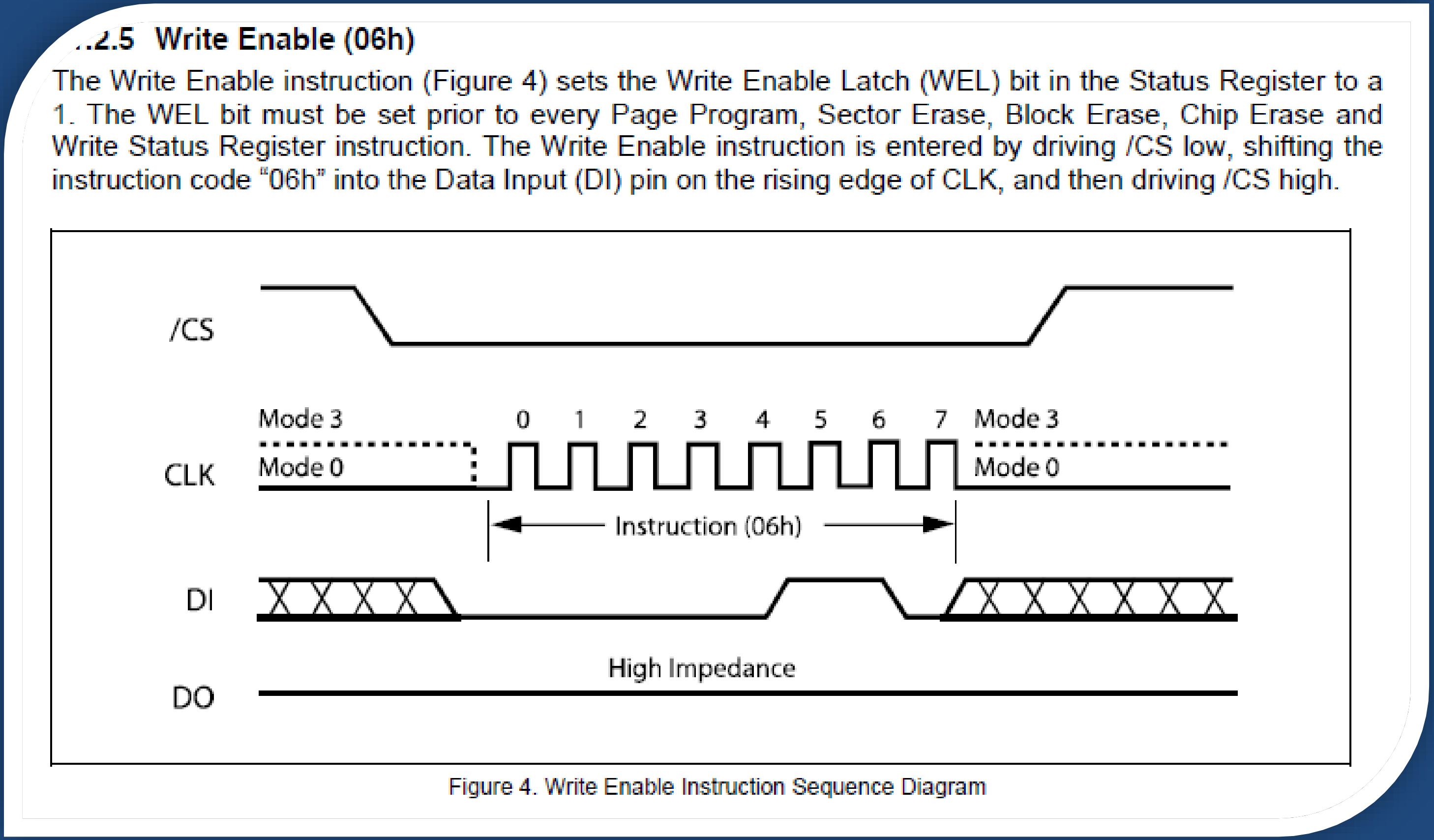


.2.2 Instruction Set Table 1 (Erase, Program Instructions)[1]

INSTRUCTION NAME	BYTE 1 (CODE)	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
Write Enable	06h					
Write Disable	04h					
Read Status Register-1	05h	(S7–S0) [2]				
Read Status Register-2	35h	(S15-S8) [2]				
Write Status Register	01h	(S7–S0)	(S15-S8)			
Page Program	02h	A23–A16	A15–A8	A7–A0	(D7–D0)	
Quad Page Program	32h	A23–A16	A15–A8	A7–A0	(D7–D0, …)[3]	
Sector Erase (4KB)	20h	A23–A16	A15–A8	A7–A0		
Block Erase (32KB)	52h	A23–A16	A15–A8	A7–A0		
Block Erase (64KB)	D8h	A23–A16	A15–A8	A7–A0		
Chip Erase	C7h/60h					
Erase Suspend	75h					
Erase Resume	7Ah					
Power-down	B9h					
Continuous Read Mode Reset [4]	FFh	FFh				

Write Enable의 타이밍 차트

- 모든 쓰기/삭제 명령 전에 Write Enable이 필요하다.



..2.5 Write Enable (06h)

The Write Enable instruction (Figure 4) sets the Write Enable Latch (WEL) bit in the Status Register to a 1. The WEL bit must be set prior to every Page Program, Sector Erase, Block Erase, Chip Erase and Write Status Register instruction. The Write Enable instruction is entered by driving /CS low, shifting the instruction code "06h" into the Data Input (DI) pin on the rising edge of CLK, and then driving /CS high.

Figure 4. Write Enable Instruction Sequence Diagram

Page Program 타이밍 차트

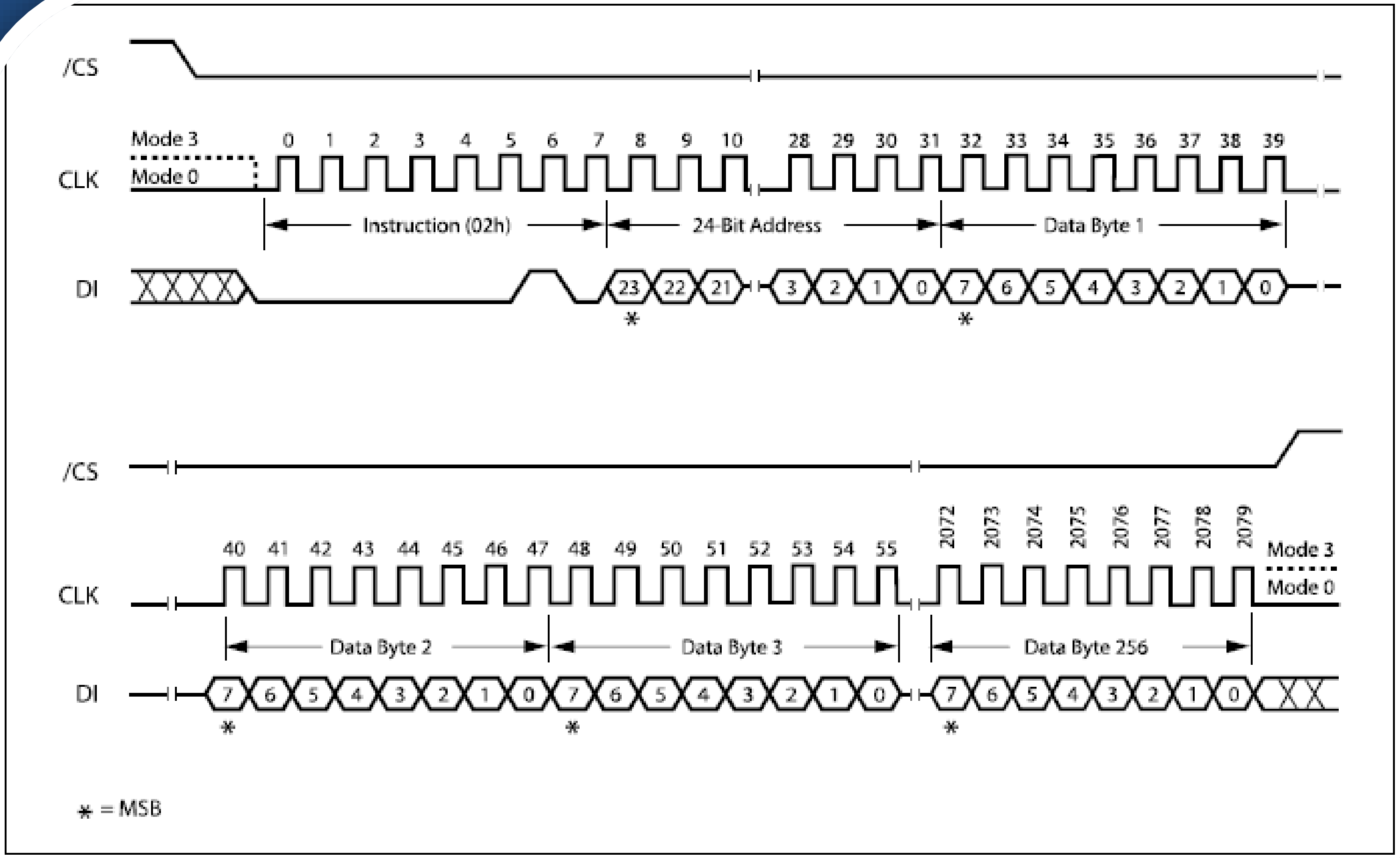


Figure 16. Page Program Instruction Sequence Diagram

전체 삭제

- Flash Memory의 데이터를 전체 삭제한다.

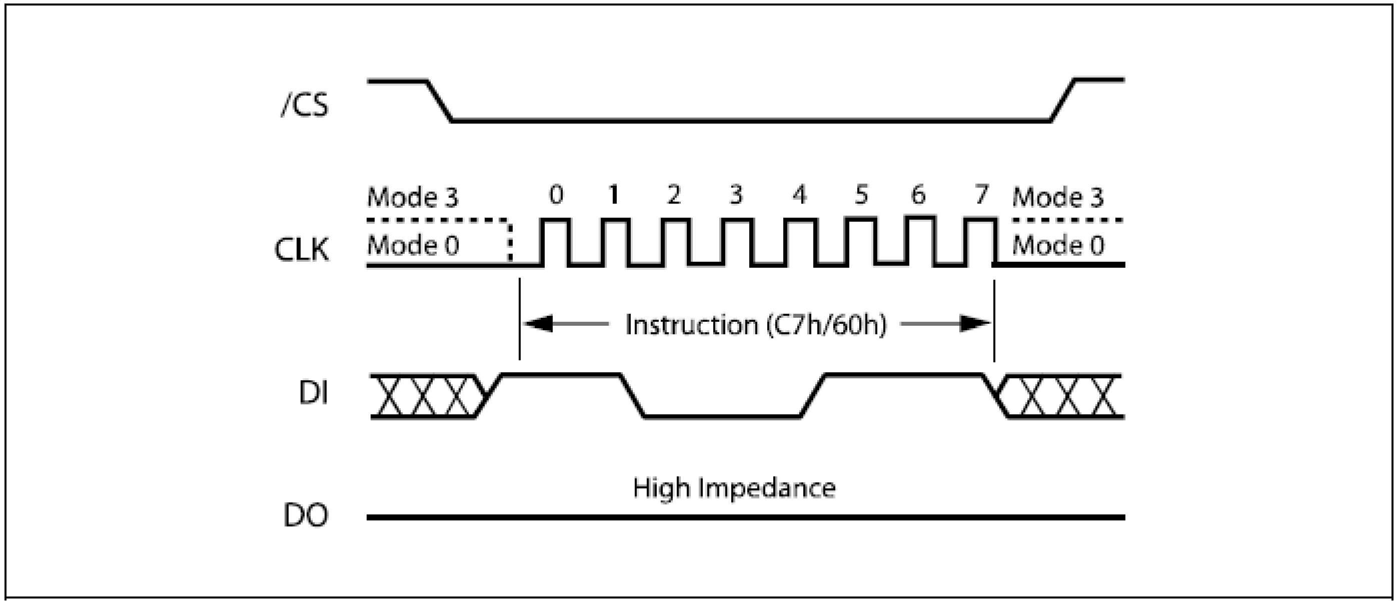
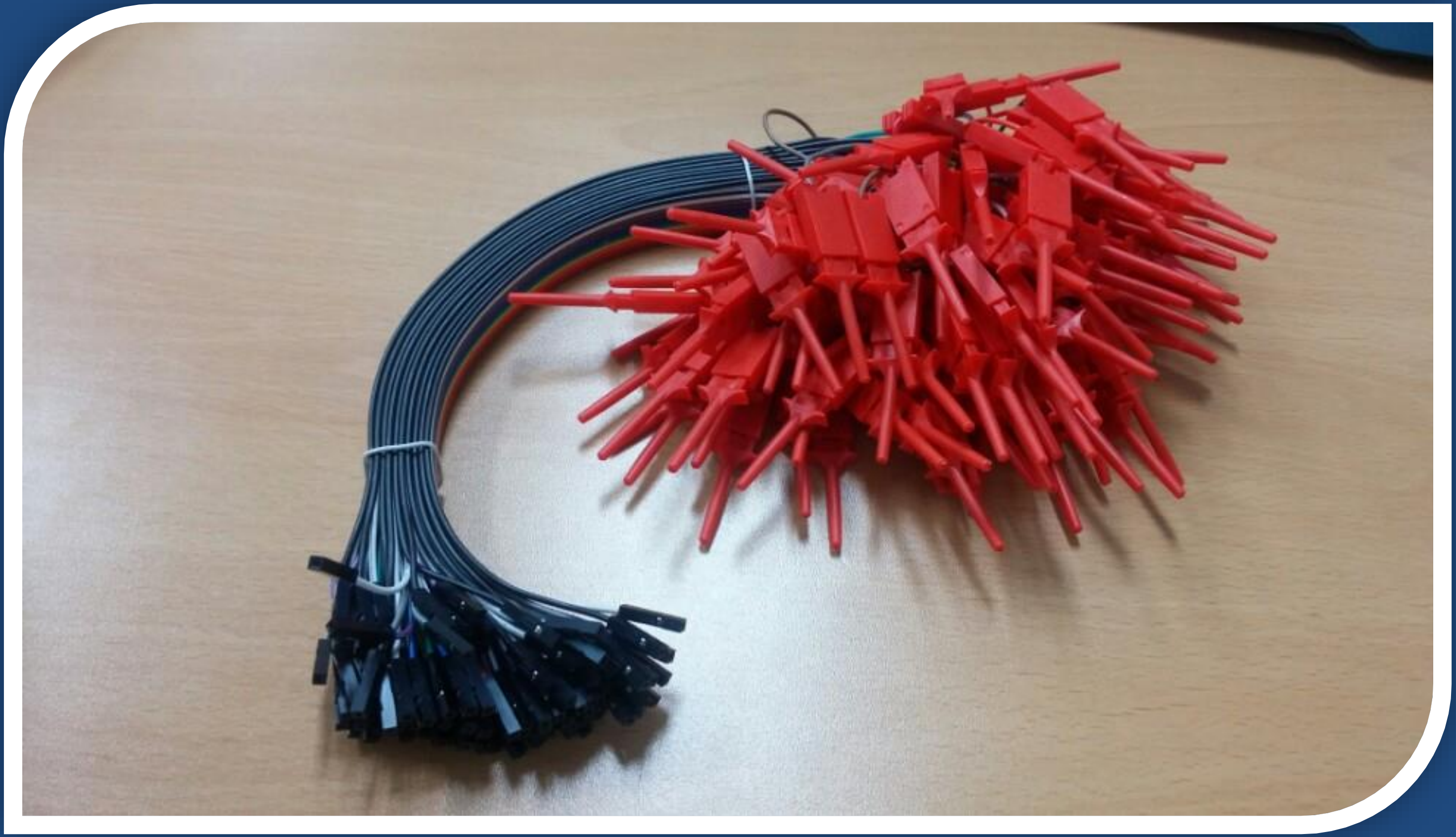


Figure 21. Chip Erase Instruction Sequence Diagram

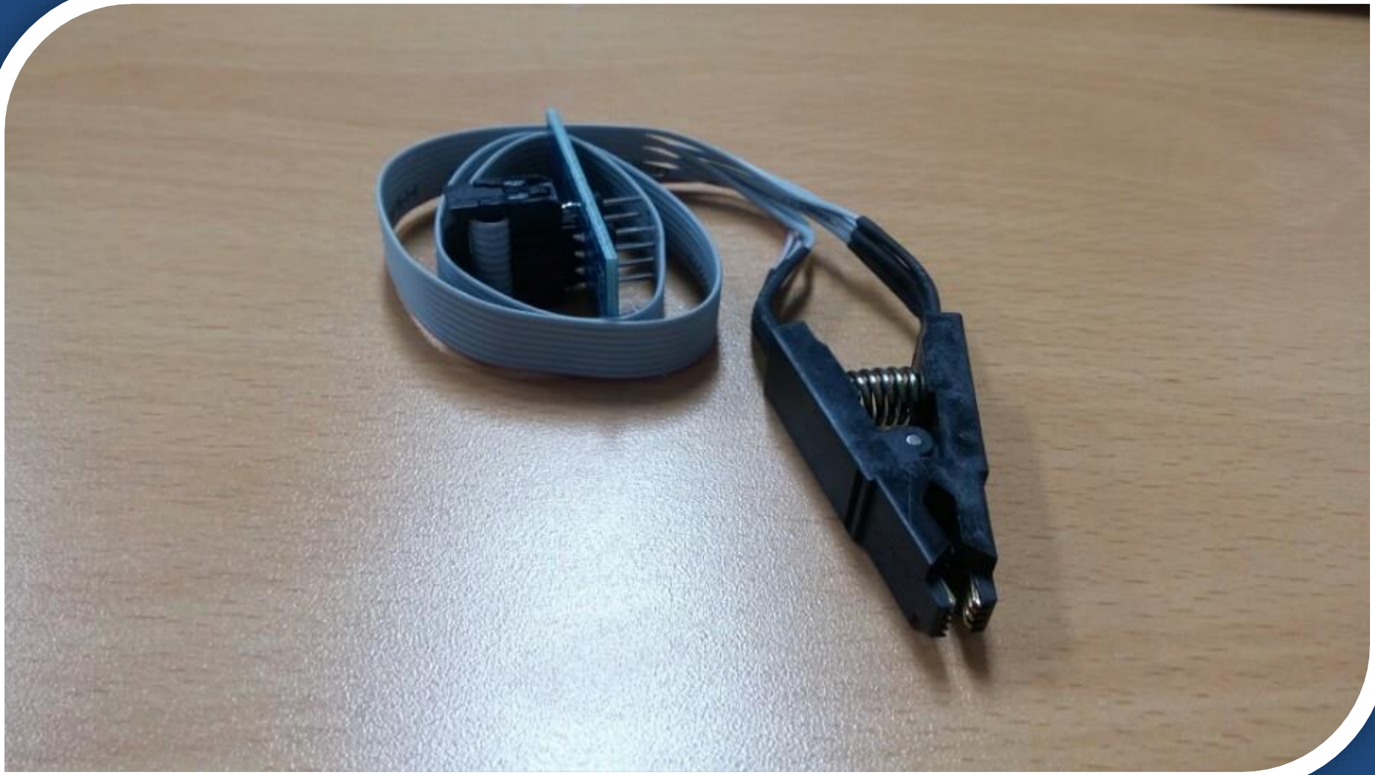
Flash Memory의 초기 데이터

- 초기 구매 플래시엔 1로 가득 차 있음
- 데이터 프로그래밍은 AND 연산으로 이루어짐
  - 1 & 1 = 1
  - 1 & 0 = 0
- 그렇기 때문에 새로운 값을 쓰기 위해서는 우선 해당 Block을 1로 초기화(Erase) 해야 함
  - 데이터는 최소 Block 단위로만 삭제 가능
  - 삭제 횟수에 제한 있음 (ex. 10만회)

ITEM1 : Hook Cable (test hook clip)



ITEM2 : SOIC Test-Clip

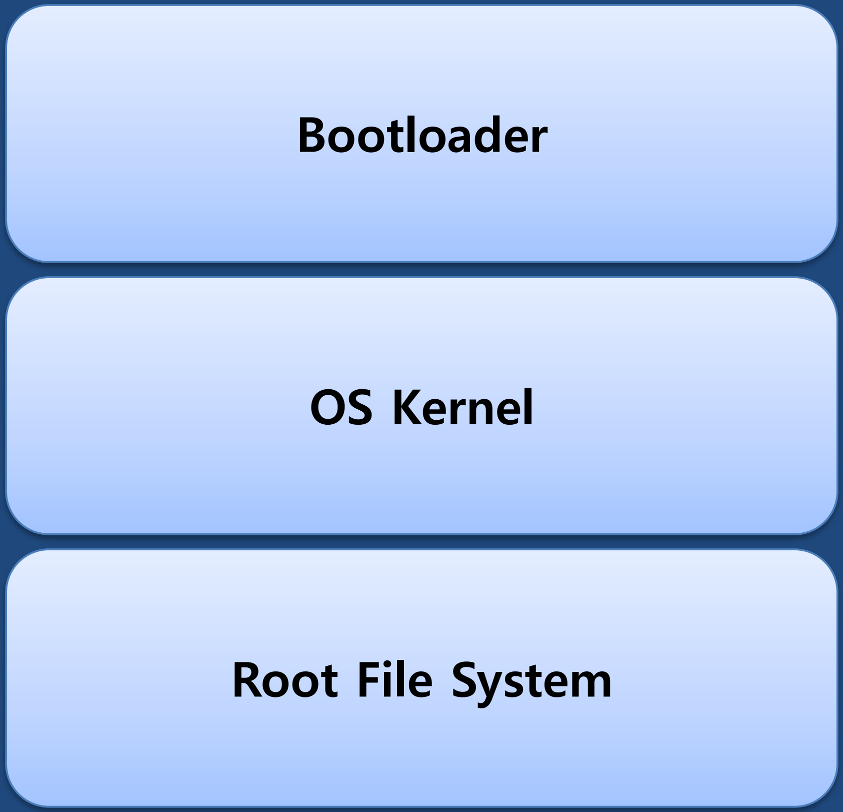


Hex -> Binary 변환

```
# hex2bin.py

import os

fp = open("firm_dump.log")
fp2 = open("firmware.bin", "wb")

while 1:
	ch = fp.read(2).decode('hex')
	if ch == "":
		break
	fp2.write(ch)

fp.close()
fp2.close()
```

공유기 Firmware 분석하기

Embedded Linux의 구조



Firmware 자동 분석 툴

- Binwalk (Firmware Analysis Tool)
  - 펌웨어 파일의 구성 분석
    - 펌웨어 분석의 원리
      - Signature 탐색
      - Ex> squashfs == “hsqs”
  - http://binwalk.org/
  - apt-get install binwalk

- FMK (Firmware Mod Kit)
  - 펌웨어 파일 내에서 각종 파일 추출
  - 혹은 수정된 파일을 기반으로 새 펌웨어 빌드
  - https://code.google.com/p/firmware-mod-kit/

binwalk

```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
--------------------------------------------------------------------
--------------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from
Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian,
version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes,
created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

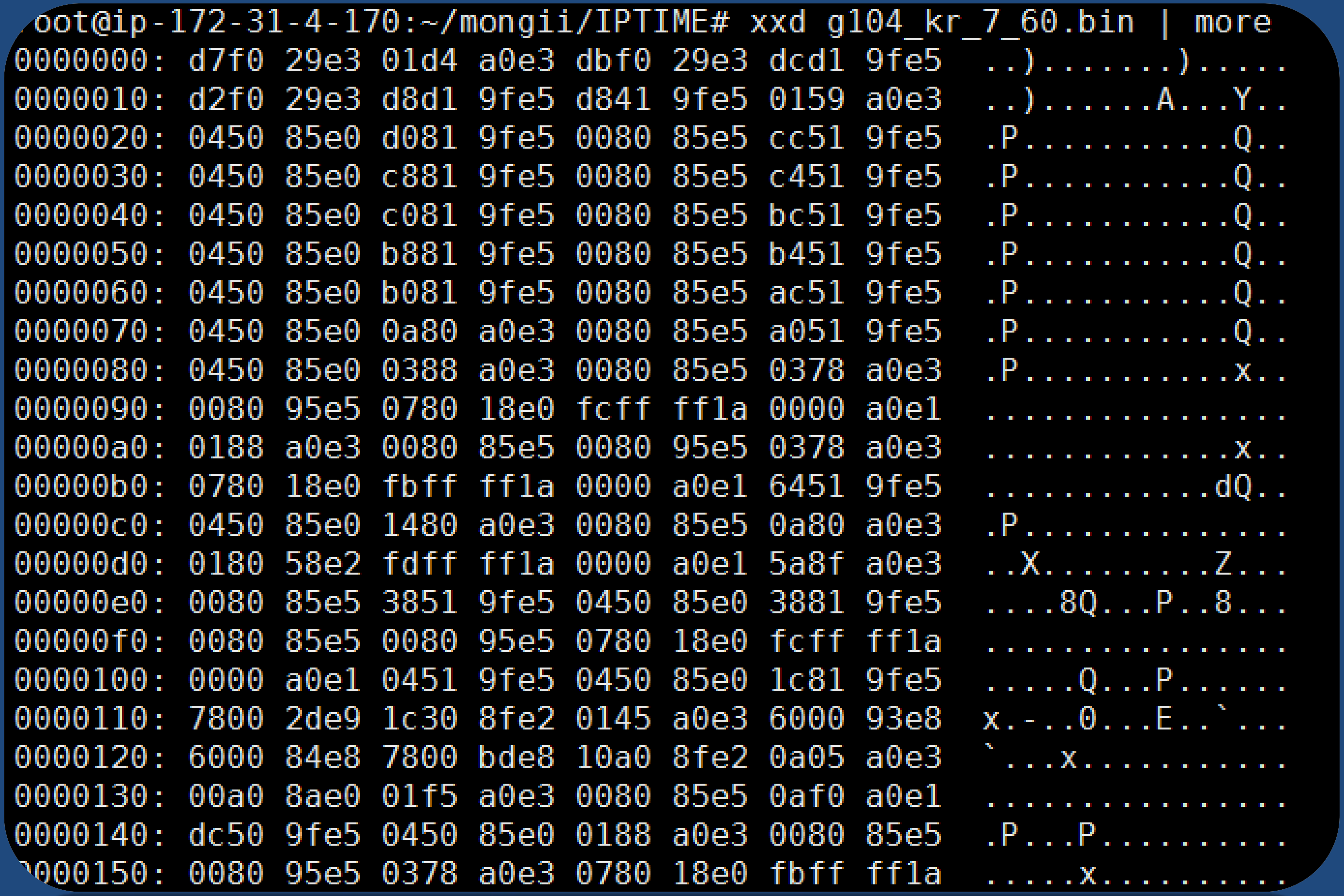
Bootloader 분석

Binwalk 결과 재확인

```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
--------------------------------------------------------------------
---------------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from
Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian,
version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes,
created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

* Offset이 65592라는 말은 그 앞에 무언가가 더 있다라는 것을 의미함

펌웨어의 시작 부분



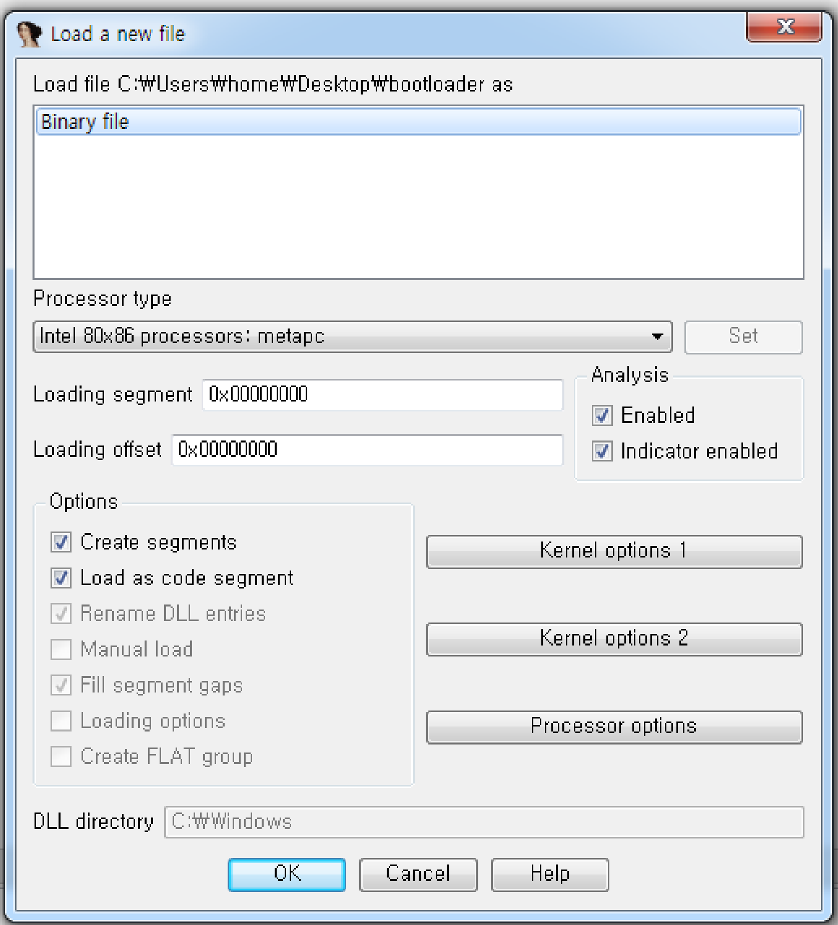
```
oot@ip-172-31-4-170:~/mongii/IPTIME# xxd g104_kr_7_60.bin | more
0000000: d7f0 29e3 01d4 a0e3 dbf0 29e3 dcd1 9fe5  ..).......).....
0000010: d2f0 29e3 d8d1 9fe5 d841 9fe5 0159 a0e3  ..).......A...Y..
0000020: 0450 85e0 d081 9fe5 0080 85e5 cc51 9fe5  .P...........Q..
0000030: 0450 85e0 c881 9fe5 0080 85e5 c451 9fe5  .P...........Q..
0000040: 0450 85e0 c081 9fe5 0080 85e5 bc51 9fe5  .P...........Q..
0000050: 0450 85e0 b881 9fe5 0080 85e5 b451 9fe5  .P...........Q..
0000060: 0450 85e0 b081 9fe5 0080 85e5 ac51 9fe5  .P...........Q..
0000070: 0450 85e0 0a80 a0e3 0080 85e5 a051 9fe5  .P...........Q..
0000080: 0450 85e0 0388 a0e3 0080 85e5 0378 a0e3  .P...........x..
0000090: 0080 95e5 0780 18e0 fcff ff1a 0000 a0e1  ................
00000a0: 0188 a0e3 0080 85e5 0080 95e5 0378 a0e3  .............x..
00000b0: 0780 18e0 fbff ff1a 0000 a0e1 6451 9fe5  ............dQ..
00000c0: 0450 85e0 1480 a0e3 0080 85e5 0a80 a0e3  .P..............
00000d0: 0180 58e2 fdff ff1a 0000 a0e1 5a8f a0e3  ..X.........Z...
00000e0: 0080 85e5 3851 9fe5 0450 85e0 3881 9fe5  ....8Q...P..8...
00000f0: 0080 85e5 0080 95e5 0780 18e0 fcff ff1a  ................
0000100: 0000 a0e1 0451 9fe5 0450 85e0 1c81 9fe5  .....Q...P......
0000110: 7800 2de9 1c30 8fe2 0145 a0e3 6000 93e8  x.-..0...E..`...
0000120: 6000 84e8 7800 bde8 10a0 8fe2 0a05 a0e3  `...x...........
0000130: 00a0 8ae0 01f5 a0e3 0080 85e5 0af0 a0e1  ................
0000140: dc50 9fe5 0450 85e0 0188 a0e3 0080 85e5  .P...P..........
000150: 0080 95e5 0378 a0e3 0780 18e0 fbff ff1a  .....x..........
```

Bootloader 분석

```
root@ip-172-31-4-170:~/mongii/IPTIME# dd if=./g104_kr_7_60.bin of=./bootloader count=65592 bs=1
65592+0 records in
65592+0 records out
65592 bytes (66 kB) copied, 0.07132 s, 920 kB/s
root@ip-172-31-4-170:~/mongii/IPTIME#
```

```
root@ip-172-31-4-170:~/mongii/IPTIME# xxd bootloader
0000000: d7f0 29e3 01d4 a0e3 dbf0 29e3 dcd1 9fe5  ..).......).....
0000010: d2f0 29e3 d8d1 9fe5 d841 9fe5 0159 a0e3  ..)......A...Y..
0000020: 0450 85e0 d081 9fe5 0080 85e5 cc51 9fe5  .P...........Q..
0000030: 0450 85e0 c881 9fe5 0080 85e5 c451 9fe5  .P...........Q..
...
000fff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0010000: 6731 3034 0000 0000 372e 3630 0000 0000  g104....7.60....
0010010: 5475 6520 4170 7220 3132 2031 363a 3535  Tue Apr 12 16:55
0010020: 3a33 3120 3230 3131 0a00 0000 0000 0b00  :31 2011........
0010030: c85f 1c00 b1f0 860e                      ._......
root@ip-172-31-4-170:~/mongii/IPTIME#
```

Bootloader 분석



Load a new file

Load file C:₩Users₩home₩Desktop₩bootloader as

Binary file

Processor type

Intel 80x86 processors: metapc

Set

Loading segment 0x00000000

Loading offset 0x00000000

Analysis

- [✓] Enabled
- [✓] Indicator enabled

Options

- [✓] Create segments
- [✓] Load as code segment
- [✓] Rename DLL entries
- [ ] Manual load
- [✓] Fill segment gaps
- [ ] Loading options
- [ ] Create FLAT group

Kernel options 1

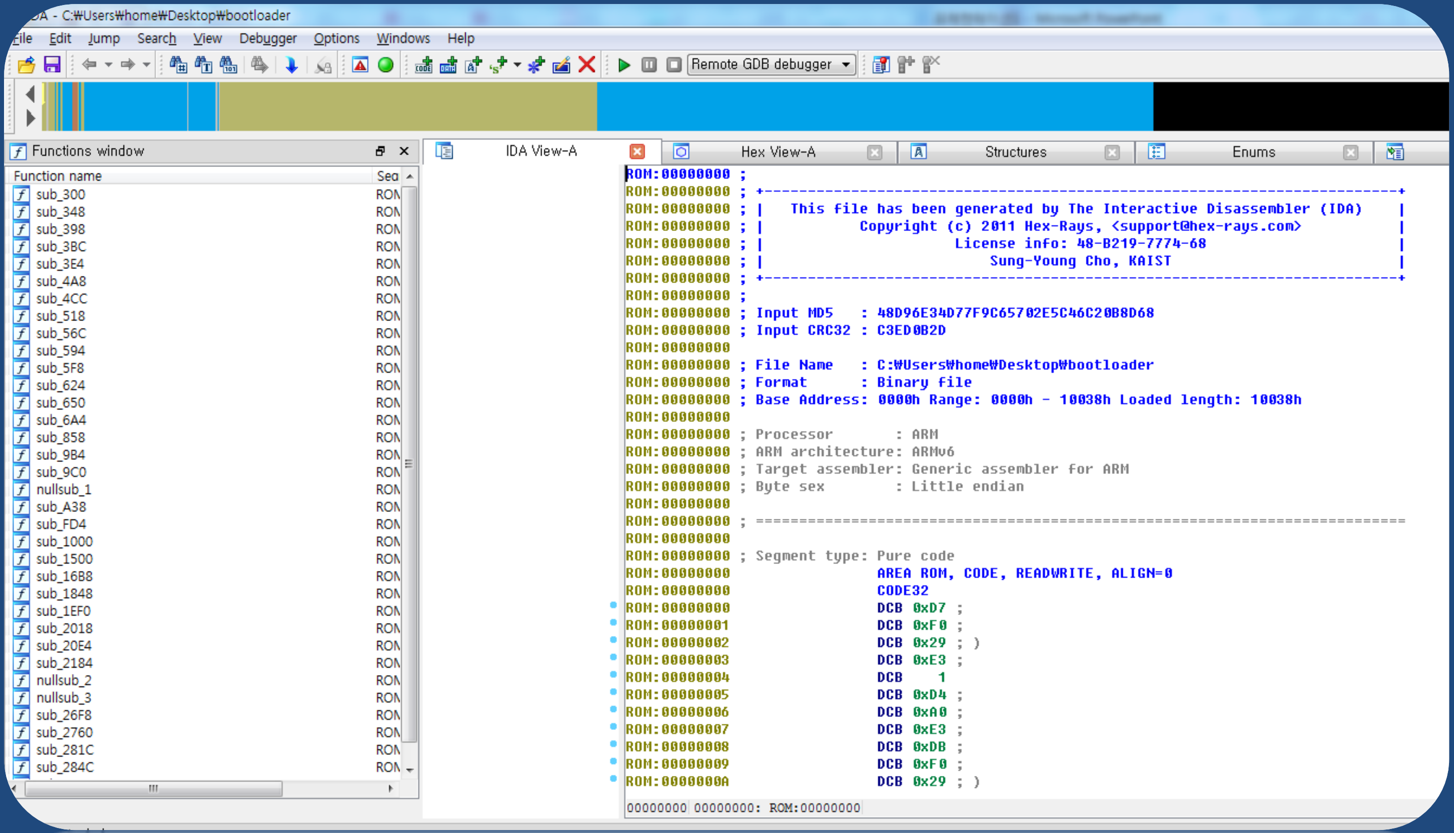
Kernel options 2

Processor options

DLL directory C:₩Windows

OK Cancel Help

IDA로 Bootloader 확인

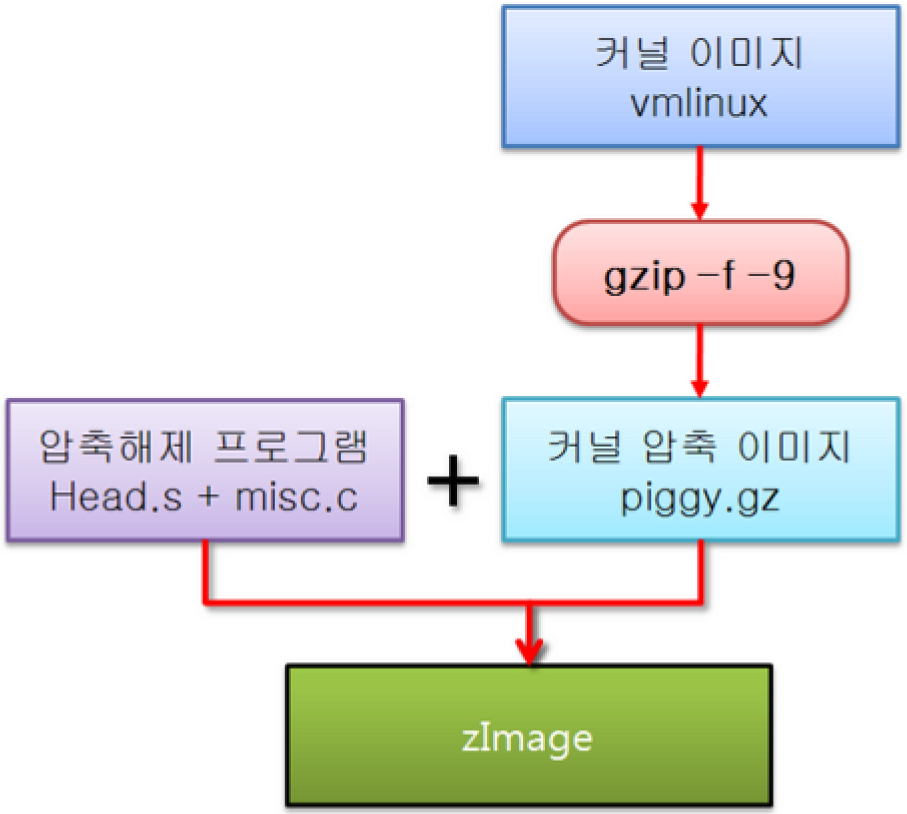

DA - C:₩Users₩home₩Desktop₩bootloader
File Edit Jump Search View Debugger Options Windows Help
Remote GDB debugger
Functions window
IDA View-A
Hex View-A
Structures
Enums
Function name
Seg

Function name	Seg
sub_300	ROM
sub_348	ROM
sub_398	ROM
sub_3BC	ROM
sub_3E4	ROM
sub_4A8	ROM
sub_4CC	ROM
sub_518	ROM
sub_56C	ROM
sub_594	ROM
sub_5F8	ROM
sub_624	ROM
sub_650	ROM
sub_6A4	ROM
sub_858	ROM
sub_9B4	ROM
sub_9C0	ROM
nullsub_1	ROM
sub_A38	ROM
sub_FD4	ROM
sub_1000	ROM
sub_1500	ROM
sub_16B8	ROM
sub_1848	ROM
sub_1EF0	ROM
sub_2018	ROM
sub_20E4	ROM
sub_2184	ROM
nullsub_2	ROM
nullsub_3	ROM
sub_26F8	ROM
sub_2760	ROM
sub_281C	ROM
sub_284C	ROM

```
ROM:00000000 ;
ROM:00000000 ; +-------------------------------------------------------------------------+
ROM:00000000 ; |    This file has been generated by The Interactive Disassembler (IDA)    |
ROM:00000000 ; |           Copyright (c) 2011 Hex-Rays, <support@hex-rays.com>            |
ROM:00000000 ; |                      License info: 48-B219-7774-68                       |
ROM:00000000 ; |                          Sung-Young Cho, KAIST                           |
ROM:00000000 ; +-------------------------------------------------------------------------+
ROM:00000000 ;
ROM:00000000 ; Input MD5    : 48D96E34D77F9C65702E5C46C20B8D68
ROM:00000000 ; Input CRC32  : C3ED0B2D
ROM:00000000
ROM:00000000 ; File Name    : C:₩Users₩home₩Desktop₩bootloader
ROM:00000000 ; Format       : Binary file
ROM:00000000 ; Base Address: 0000h Range: 0000h - 10038h Loaded length: 10038h
ROM:00000000
ROM:00000000 ; Processor       : ARM
ROM:00000000 ; ARM architecture: ARMv6
ROM:00000000 ; Target assembler: Generic assembler for ARM
ROM:00000000 ; Byte sex        : Little endian
ROM:00000000
ROM:00000000 ; ===========================================================================
ROM:00000000
ROM:00000000 ; Segment type: Pure code
ROM:00000000                 AREA ROM, CODE, READWRITE, ALIGN=0
ROM:00000000                 CODE32
ROM:00000000                 DCB 0xD7 ;
ROM:00000001                 DCB 0xF0 ;
ROM:00000002                 DCB 0x29 ; )
ROM:00000003                 DCB 0xE3 ;
ROM:00000004                 DCB    1
ROM:00000005                 DCB 0xD4 ;
ROM:00000006                 DCB 0xA0 ;
ROM:00000007                 DCB 0xE3 ;
ROM:00000008                 DCB 0xDB ;
ROM:00000009                 DCB 0xF0 ;
ROM:0000000A                 DCB 0x29 ; )
00000000 00000000: ROM:00000000
```

Kernel 분석

Kernel의 구조



출처 : http://bmfrog.tistory.com/m/post/view/id/101

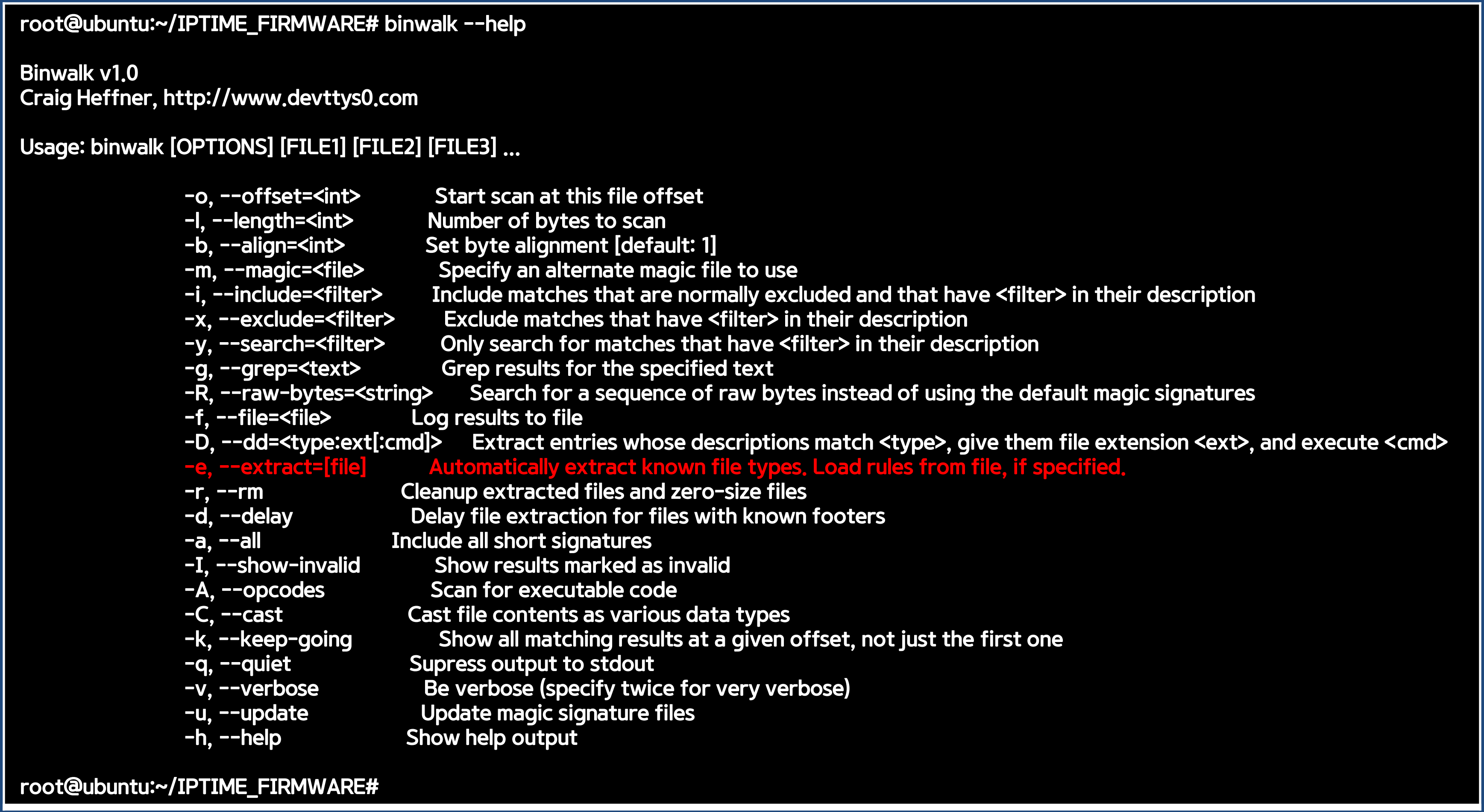
Binwalk 결과 재확인

```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
--------------------------------------------------------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from
Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian,
version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes,
created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

Extraction

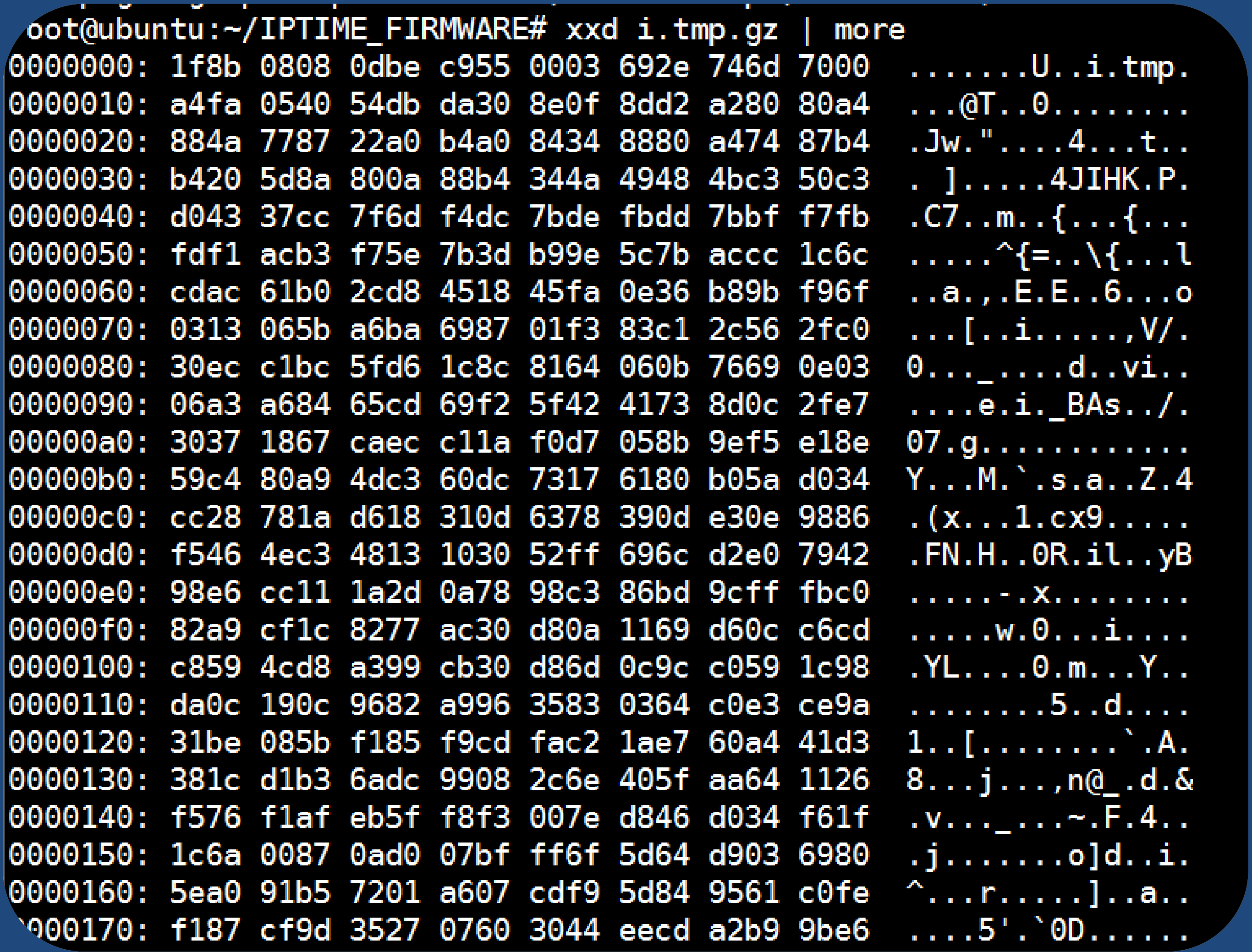
```
root@ip-172-31-4-170:~/mongii/IPTIME# dd skip=65592 if=./g104_kr_7_60.bin of=./i.tmp.gz bs=1
1859720+0 records in
1859720+0 records out
1859720 bytes (1.9 MB) copied, 2.05117 s, 907 kB/s
root@ip-172-31-4-170:~/mongii/IPTIME#
root@ip-172-31-4-170:~/mongii/IPTIME# file i.tmp.gz
i.tmp.gz: gzip compressed data, was "i.tmp", from Unix, last modified: Tue Apr 12
07:55:31 2011
root@ip-172-31-4-170:~/mongii/IPTIME#
root@ip-172-31-4-170:~/mongii/IPTIME# ls -al
total 3780
drwxr-xr-x  2 root root    4096 Jun 25 15:11 .
drwxr-xr-x 26 root root    4096 Jun 25 14:52 ..
-rw-r--r--  1 root root   65592 Jun 25 15:09 bootloader
-rw-r--r--  1 root root 1925312 Jun 25 14:47 g104_kr_7_60.bin
-rw-r--r--  1 root root 1859720 Jun 25 15:11 i.tmp.gz
root@ip-172-31-4-170:~/mongii/IPTIME#
```

-e : extraction



```
root@ubuntu:~/IPTIME_FIRMWARE# binwalk --help

Binwalk v1.0
Craig Heffner, http://www.devttys0.com

Usage: binwalk [OPTIONS] [FILE1] [FILE2] [FILE3] ...

        -o, --offset=<int>          Start scan at this file offset
        -l, --length=<int>          Number of bytes to scan
        -b, --align=<int>           Set byte alignment [default: 1]
        -m, --magic=<file>          Specify an alternate magic file to use
        -i, --include=<filter>      Include matches that are normally excluded and that have <filter> in their description
        -x, --exclude=<filter>      Exclude matches that have <filter> in their description
        -y, --search=<filter>       Only search for matches that have <filter> in their description
        -g, --grep=<text>           Grep results for the specified text
        -R, --raw-bytes=<string>    Search for a sequence of raw bytes instead of using the default magic signatures
        -f, --file=<file>           Log results to file
        -D, --dd=<type:ext[:cmd]>   Extract entries whose descriptions match <type>, give them file extension <ext>, and execute <cmd>
        -e, --extract=[file]        Automatically extract known file types. Load rules from file, if specified.
        -r, --rm                    Cleanup extracted files and zero-size files
        -d, --delay                 Delay file extraction for files with known footers
        -a, --all                   Include all short signatures
        -I, --show-invalid          Show results marked as invalid
        -A, --opcodes               Scan for executable code
        -C, --cast                  Cast file contents as various data types
        -k, --keep-going            Show all matching results at a given offset, not just the first one
        -q, --quiet                 Supress output to stdout
        -v, --verbose               Be verbose (specify twice for very verbose)
        -u, --update                Update magic signature files
        -h, --help                  Show help output

root@ubuntu:~/IPTIME_FIRMWARE#
```

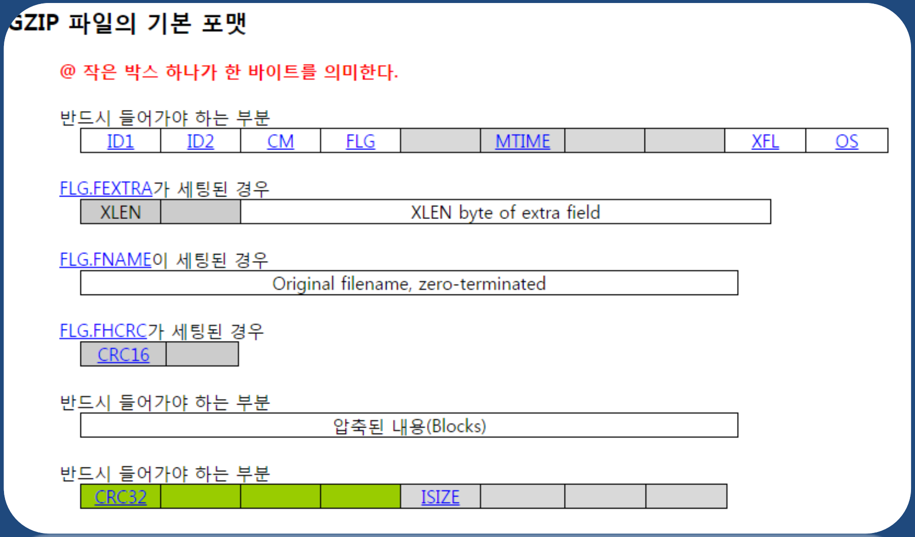
i.tmp.gz 분석



```
oot@ubuntu:~/IPTIME_FIRMWARE# xxd i.tmp.gz | more
0000000: 1f8b 0808 0dbe c955 0003 692e 746d 7000  .......U..i.tmp.
0000010: a4fa 0540 54db da30 8e0f 8dd2 a280 80a4  ...@T..0........
0000020: 884a 7787 22a0 b4a0 8434 8880 a474 87b4  .Jw."....4...t..
0000030: b420 5d8a 800a 88b4 344a 4948 4bc3 50c3  . ].....4JIHK.P.
0000040: d043 37cc 7f6d f4dc 7bde fbdd 7bbf f7fb  .C7..m..{...{...
0000050: fdf1 acb3 f75e 7b3d b99e 5c7b accc 1c6c  .....^{=..\{...l
0000060: cdac 61b0 2cd8 4518 45fa 0e36 b89b f96f  ..a.,.E.E..6...o
0000070: 0313 065b a6ba 6987 01f3 83c1 2c56 2fc0  ...[..i.....,V/.
0000080: 30ec c1bc 5fd6 1c8c 8164 060b 7669 0e03  0..._....d..vi..
0000090: 06a3 a684 65cd 69f2 5f42 4173 8d0c 2fe7  ....e.i._BAs../.
00000a0: 3037 1867 caec c11a f0d7 058b 9ef5 e18e  07.g............
00000b0: 59c4 80a9 4dc3 60dc 7317 6180 b05a d034  Y...M.`.s.a..Z.4
00000c0: cc28 781a d618 310d 6378 390d e30e 9886  .(x...1.cx9.....
00000d0: f546 4ec3 4813 1030 52ff 696c d2e0 7942  .FN.H..0R.il..yB
00000e0: 98e6 cc11 1a2d 0a78 98c3 86bd 9cff fbc0  .....-.x........
00000f0: 82a9 cf1c 8277 ac30 d80a 1169 d60c c6cd  .....w.0...i....
0000100: c859 4cd8 a399 cb30 d86d 0c9c c059 1c98  .YL....0.m...Y..
0000110: da0c 190c 9682 a996 3583 0364 c0e3 ce9a  ........5..d....
0000120: 31be 085b f185 f9cd fac2 1ae7 60a4 41d3  1..[........`.A.
0000130: 381c d1b3 6adc 9908 2c6e 405f aa64 1126  8...j...,n@_.d.&
0000140: f576 f1af eb5f f8f3 007e d846 d034 f61f  .v..._...~.F.4..
0000150: 1c6a 0087 0ad0 07bf ff6f 5d64 d903 6980  .j.......o]d..i.
0000160: 5ea0 91b5 7201 a607 cdf9 5d84 9561 c0fe  ^...r.....]..a..
000170: f187 cf9d 3527 0760 3044 eecd a2b9 9be6  ....5'.`0D......
```

i.tmp.gz 분석

- http://andromedarabbit.net/project/Zip/GzipFileFormat.html



GZIP 파일의 기본 포맷

@ 작은 박스 하나가 한 바이트를 의미한다.

반드시 들어가야 하는 부분

ID1	ID2	CM	FLG		MTIME			XFL	OS

FLG.FEXTRA가 세팅된 경우

XLEN		XLEN byte of extra field

FLG.FNAME이 세팅된 경우

Original filename, zero-terminated

FLG.FHCRC가 세팅된 경우

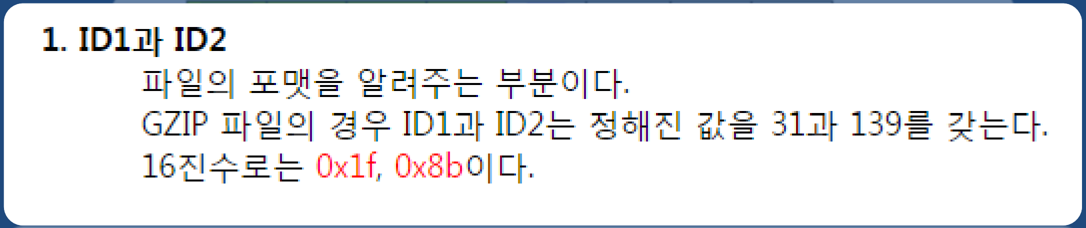
CRC16	

반드시 들어가야 하는 부분

압축된 내용(Blocks)

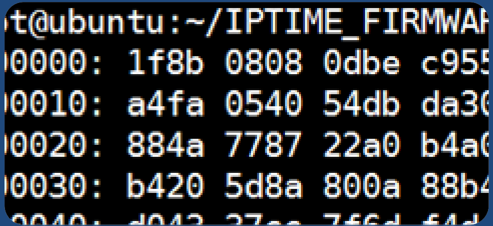
반드시 들어가야 하는 부분

CRC32				ISIZE			



1. ID1과 ID2

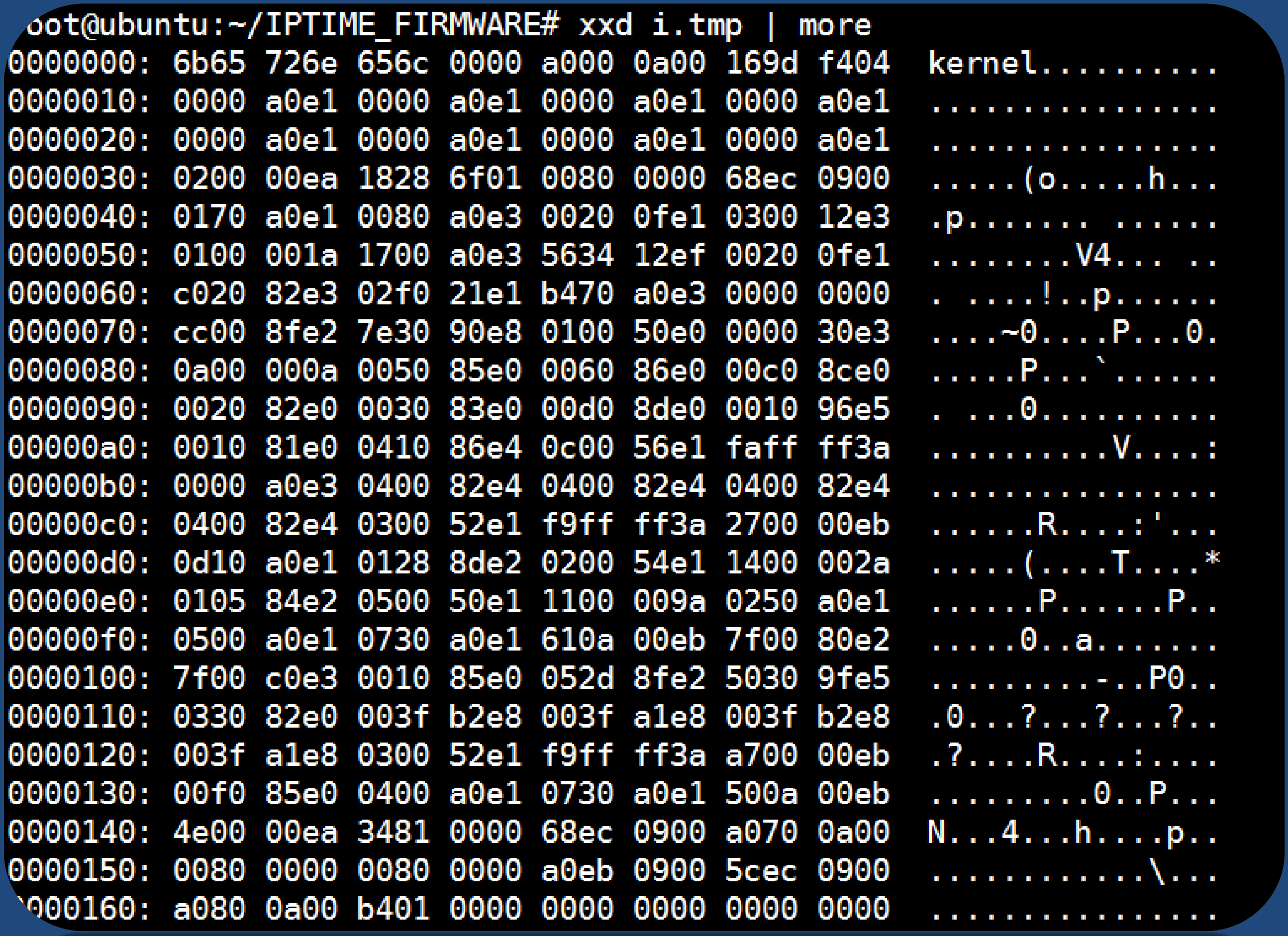
파일의 포맷을 알려주는 부분이다.
GZIP 파일의 경우 ID1과 ID2는 정해진 값을 31과 139를 갖는다.
16진수로는 0x1f, 0x8b이다.



```
t@ubuntu:~/IPTIME_FIRMWA
0000: 1f8b 0808 0dbe c95
0010: a4fa 0540 54db da3
0020: 884a 7787 22a0 b4a
0030: b420 5d8a 800a 88b
```

i.tmp 분석

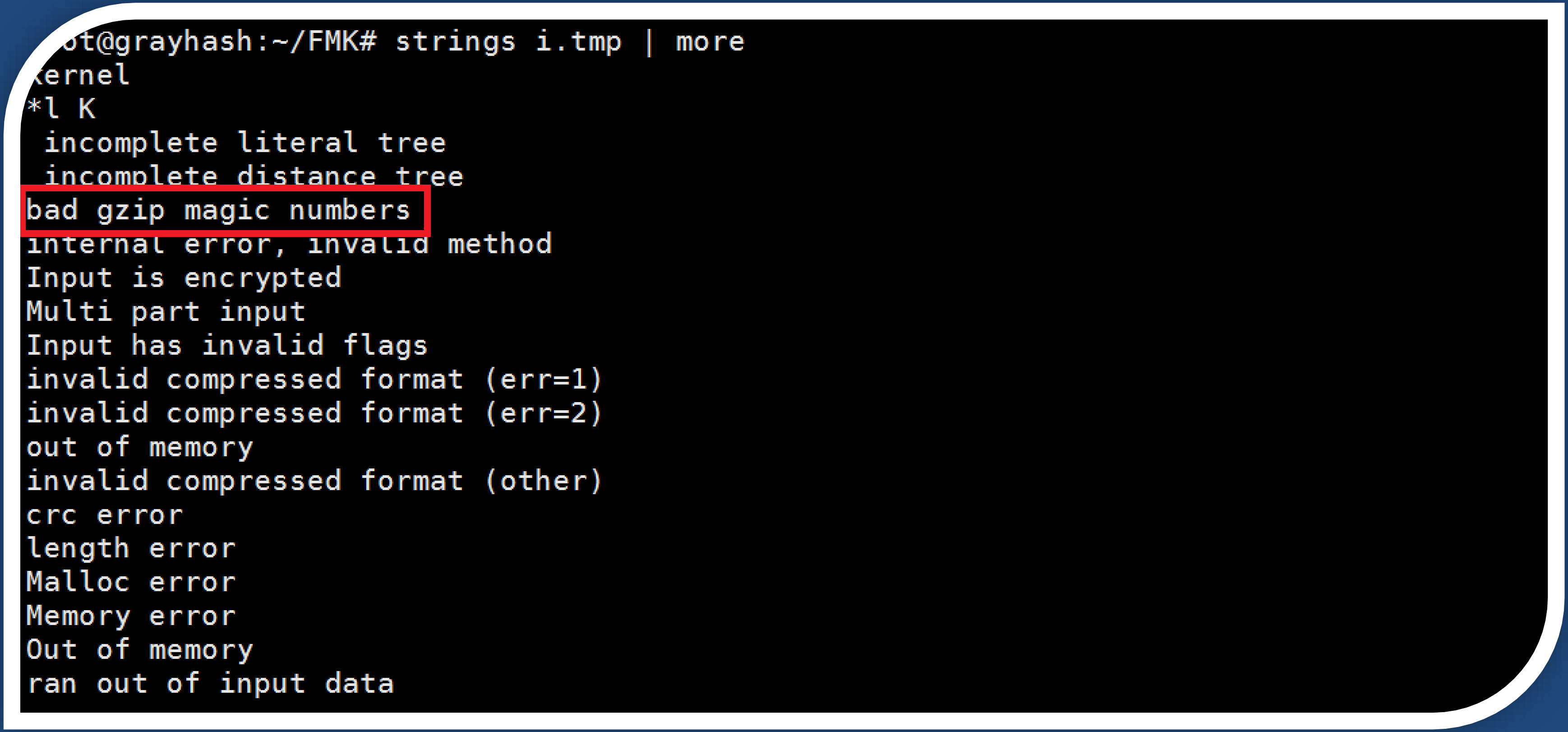
- gzip -d i.tmp.gz



```
oot@ubuntu:~/IPTIME_FIRMWARE# xxd i.tmp | more
0000000: 6b65 726e 656c 0000 a000 0a00 169d f404  kernel..........
0000010: 0000 a0e1 0000 a0e1 0000 a0e1 0000 a0e1  ................
0000020: 0000 a0e1 0000 a0e1 0000 a0e1 0000 a0e1  ................
0000030: 0200 00ea 1828 6f01 0080 0000 68ec 0900  .....(o.....h...
0000040: 0170 a0e1 0080 a0e3 0020 0fe1 0300 12e3  .p....... ......
0000050: 0100 001a 1700 a0e3 5634 12ef 0020 0fe1  ........V4... ..
0000060: c020 82e3 02f0 21e1 b470 a0e3 0000 0000  . ....!..p......
0000070: cc00 8fe2 7e30 90e8 0100 50e0 0000 30e3  ....~0....P...0.
0000080: 0a00 000a 0050 85e0 0060 86e0 00c0 8ce0  .....P...`......
0000090: 0020 82e0 0030 83e0 00d0 8de0 0010 96e5  . ...0..........
00000a0: 0010 81e0 0410 86e4 0c00 56e1 faff ff3a  ..........V....:
00000b0: 0000 a0e3 0400 82e4 0400 82e4 0400 82e4  ................
00000c0: 0400 82e4 0300 52e1 f9ff ff3a 2700 00eb  ......R....:'...
00000d0: 0d10 a0e1 0128 8de2 0200 54e1 1400 002a  .....(....T....*
00000e0: 0105 84e2 0500 50e1 1100 009a 0250 a0e1  ......P......P..
00000f0: 0500 a0e1 0730 a0e1 610a 00eb 7f00 80e2  .....0..a.......
0000100: 7f00 c0e3 0010 85e0 052d 8fe2 5030 9fe5  .........-..P0..
0000110: 0330 82e0 003f b2e8 003f a1e8 003f b2e8  .0...?...?...?..
0000120: 003f a1e8 0300 52e1 f9ff ff3a a700 00eb  .?....R....:....
0000130: 00f0 85e0 0400 a0e1 0730 a0e1 500a 00eb  .........0..P...
0000140: 4e00 00ea 3481 0000 68ec 0900 a070 0a00  N...4...h....p..
0000150: 0080 0000 0080 0000 a0eb 0900 5cec 0900  ............\...
000160: a080 0a00 b401 0000 0000 0000 0000 0000  ................
```

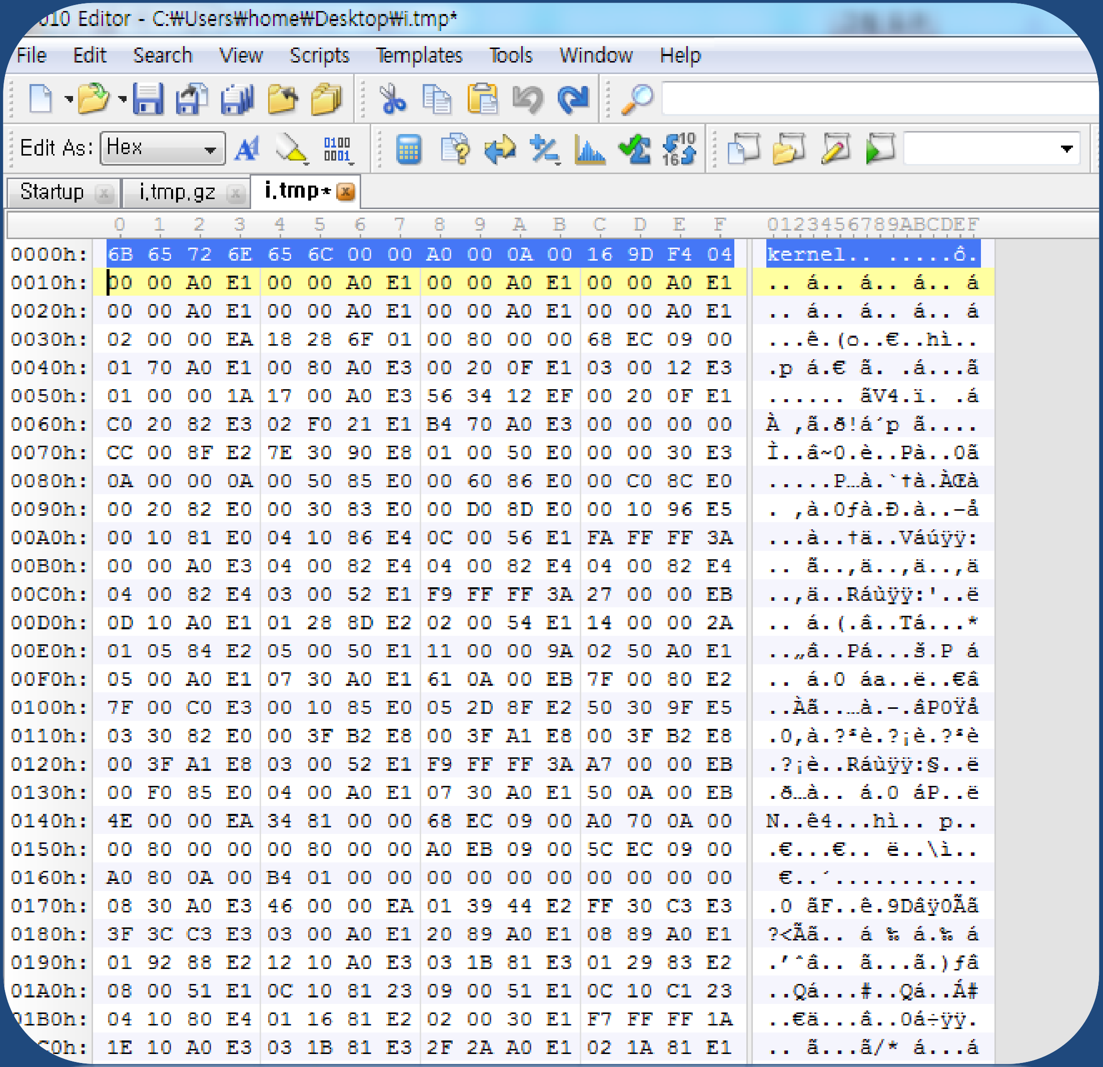
문자열 확인

- gzip 해제 코드가 들어있는 것을 알 수 있음
  – misc.c



```
ot@grayhash:~/FMK# strings i.tmp | more
ernel
*l K
 incomplete literal tree
 incomplete distance tree
bad gzip magic numbers
internal error, invalid method
Input is encrypted
Multi part input
Input has invalid flags
invalid compressed format (err=1)
invalid compressed format (err=2)
out of memory
invalid compressed format (other)
crc error
length error
Malloc error
Memory error
Out of memory
ran out of input data
```

헤더로 추정되는 값 삭제

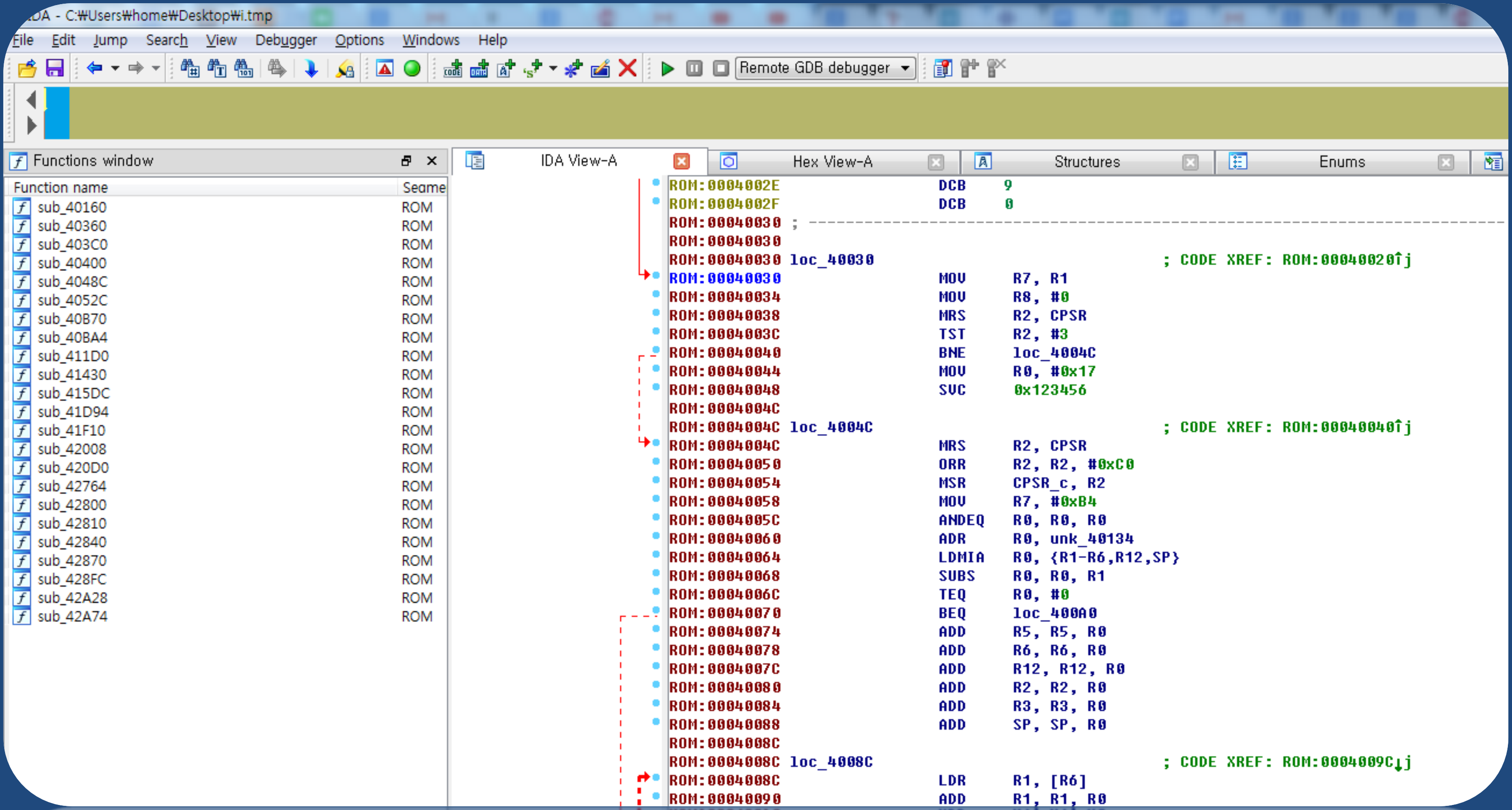


010 Editor - C:₩Users₩home₩Desktop₩i.tmp*
File Edit Search View Scripts Templates Tools Window Help
Edit As: Hex
Startup
i.tmp.gz
i.tmp*

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 6B 65 72 6E 65 6C 00 00 A0 00 0A 00 16 9D F4 04  kernel.. .....ô.
0010h: 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1  .. á.. á.. á.. á
0020h: 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1  .. á.. á.. á.. á
0030h: 02 00 00 EA 18 28 6F 01 00 80 00 00 68 EC 09 00  ...ê.(o..€..hì..
0040h: 01 70 A0 E1 00 80 A0 E3 00 20 0F E1 03 00 12 E3  .p á.€ ã. .á...ã
0050h: 01 00 00 1A 17 00 A0 E3 56 34 12 EF 00 20 0F E1  ...... ãV4.ï. .á
0060h: C0 20 82 E3 02 F0 21 E1 B4 70 A0 E3 00 00 00 00  À ‚ã.ð!á´p ã....
0070h: CC 00 8F E2 7E 30 90 E8 01 00 50 E0 00 00 30 E3  Ì..â~0.è..Pà..0ã
0080h: 0A 00 00 0A 00 50 85 E0 00 60 86 E0 00 C0 8C E0  .....P…à.`†à.ÀŒà
0090h: 00 20 82 E0 00 30 83 E0 00 D0 8D E0 00 10 96 E5  . ‚à.0ƒà.Ð.à..–å
00A0h: 00 10 81 E0 04 10 86 E4 0C 00 56 E1 FA FF FF 3A  ...à..†ä..Váúÿÿ:
00B0h: 00 00 A0 E3 04 00 82 E4 04 00 82 E4 04 00 82 E4  .. ã..‚ä..‚ä..‚ä
00C0h: 04 00 82 E4 03 00 52 E1 F9 FF FF 3A 27 00 00 EB  ..‚ä..Ráùÿÿ:'..ë
00D0h: 0D 10 A0 E1 01 28 8D E2 02 00 54 E1 14 00 00 2A  .. á.(.â..Tá...*
00E0h: 01 05 84 E2 05 00 50 E1 11 00 00 9A 02 50 A0 E1  ..„â..Pá...š.P á
00F0h: 05 00 A0 E1 07 30 A0 E1 61 0A 00 EB 7F 00 80 E2  .. á.0 áa..ë..€â
0100h: 7F 00 C0 E3 00 10 85 E0 05 2D 8F E2 50 30 9F E5  ..Àã..…à.-.âP0Ÿå
0110h: 03 30 82 E0 00 3F B2 E8 00 3F A1 E8 00 3F B2 E8  .0‚à.?²è.?¡è.?²è
0120h: 00 3F A1 E8 03 00 52 E1 F9 FF FF 3A A7 00 00 EB  .?¡è..Ráùÿÿ:§..ë
0130h: 00 F0 85 E0 04 00 A0 E1 07 30 A0 E1 50 0A 00 EB  .ð…à.. á.0 áP..ë
0140h: 4E 00 00 EA 34 81 00 00 68 EC 09 00 A0 70 0A 00  N..ê4...hì.. p..
0150h: 00 80 00 00 00 80 00 00 A0 EB 09 00 5C EC 09 00  .€...€.. ë..\ì..
0160h: A0 80 0A 00 B4 01 00 00 00 00 00 00 00 00 00 00   €..´..........
0170h: 08 30 A0 E3 46 00 00 EA 01 39 44 E2 FF 30 C3 E3  .0 ãF..ê.9Dâÿ0Ãã
0180h: 3F 3C C3 E3 03 00 A0 E1 20 89 A0 E1 08 89 A0 E1  ?<Ãã.. á ‰ á.‰ á
0190h: 01 92 88 E2 12 10 A0 E3 03 1B 81 E3 01 29 83 E2  .'ˆâ.. ã...ã.)ƒâ
01A0h: 08 00 51 E1 0C 10 81 23 09 00 51 E1 0C 10 C1 23  ..Qá...#..Qá..Á#
01B0h: 04 10 80 E4 01 16 81 E2 02 00 30 E1 F7 FF FF 1A  ..€ä...â..0á÷ÿÿ.
01C0h: 1E 10 A0 E3 03 1B 81 E3 2F 2A A0 E1 02 1A 81 E1  .. ã...ã/* á...á
```

IDA로 확인

- piggy.gz 압축 해제 코드


DA - C:₩Users₩home₩Desktop₩i.tmp
File Edit Jump Search View Debugger Options Windows Help
Remote GDB debugger
Functions window
IDA View-A
Hex View-A
Structures
Enums

Function name	Segme
sub_40160	ROM
sub_40360	ROM
sub_403C0	ROM
sub_40400	ROM
sub_4048C	ROM
sub_4052C	ROM
sub_40B70	ROM
sub_40BA4	ROM
sub_411D0	ROM
sub_41430	ROM
sub_415DC	ROM
sub_41D94	ROM
sub_41F10	ROM
sub_42008	ROM
sub_420D0	ROM
sub_42764	ROM
sub_42800	ROM
sub_42810	ROM
sub_42840	ROM
sub_42870	ROM
sub_428FC	ROM
sub_42A28	ROM
sub_42A74	ROM

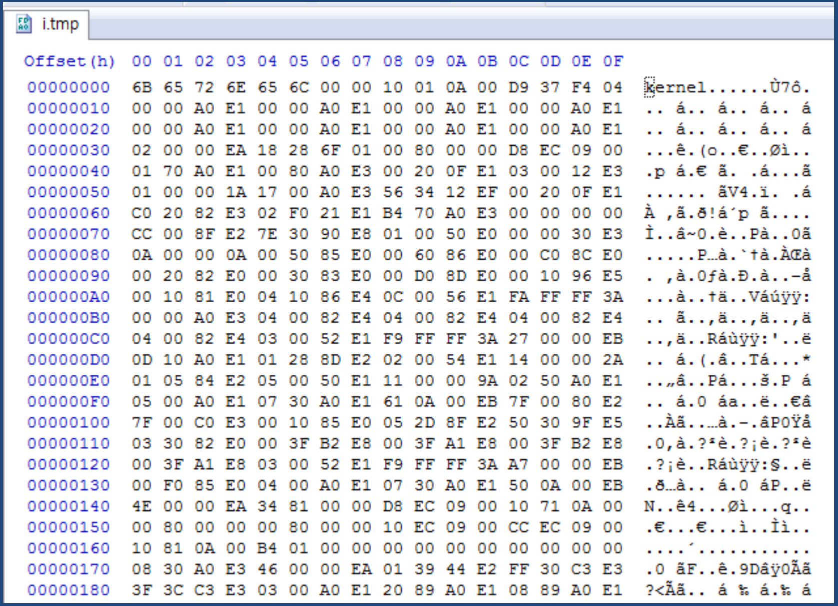
```
ROM:0004002E                 DCB    9
ROM:0004002F                 DCB    0
ROM:00040030 ; ---------------------------------------------------------------------------
ROM:00040030
ROM:00040030 loc_40030                               ; CODE XREF: ROM:00040020↑j
ROM:00040030                 MOV    R7, R1
ROM:00040034                 MOV    R8, #0
ROM:00040038                 MRS    R2, CPSR
ROM:0004003C                 TST    R2, #3
ROM:00040040                 BNE    loc_4004C
ROM:00040044                 MOV    R0, #0x17
ROM:00040048                 SVC    0x123456
ROM:0004004C
ROM:0004004C loc_4004C                               ; CODE XREF: ROM:00040040↑j
ROM:0004004C                 MRS    R2, CPSR
ROM:00040050                 ORR    R2, R2, #0xC0
ROM:00040054                 MSR    CPSR_c, R2
ROM:00040058                 MOV    R7, #0xB4
ROM:0004005C                 ANDEQ  R0, R0, R0
ROM:00040060                 ADR    R0, unk_40134
ROM:00040064                 LDMIA  R0, {R1-R6,R12,SP}
ROM:00040068                 SUBS   R0, R0, R1
ROM:0004006C                 TEQ    R0, #0
ROM:00040070                 BEQ    loc_400A0
ROM:00040074                 ADD    R5, R5, R0
ROM:00040078                 ADD    R6, R6, R0
ROM:0004007C                 ADD    R12, R12, R0
ROM:00040080                 ADD    R2, R2, R0
ROM:00040084                 ADD    R3, R3, R0
ROM:00040088                 ADD    SP, SP, R0
ROM:0004008C
ROM:0004008C loc_4008C                               ; CODE XREF: ROM:0004009C↓j
ROM:0004008C                 LDR    R1, [R6]
ROM:00040090                 ADD    R1, R1, R0
```

i.tmp의 구조

```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk i.tmp

DECIMAL         HEX             DESCRIPTION
--------------------------------------------------------------------------------
11936           0x2EA0          gzip compressed data, from Unix, last
modified: Thu Apr 15 01:49:36 2010, max compression
655664          0xA0130         gzip compressed data, was "initrd",
from Unix, last modified: Tue Apr 12 07:55:27 2011, max compression

root@ip-172-31-4-170:~/mongii/IPTIME#
```

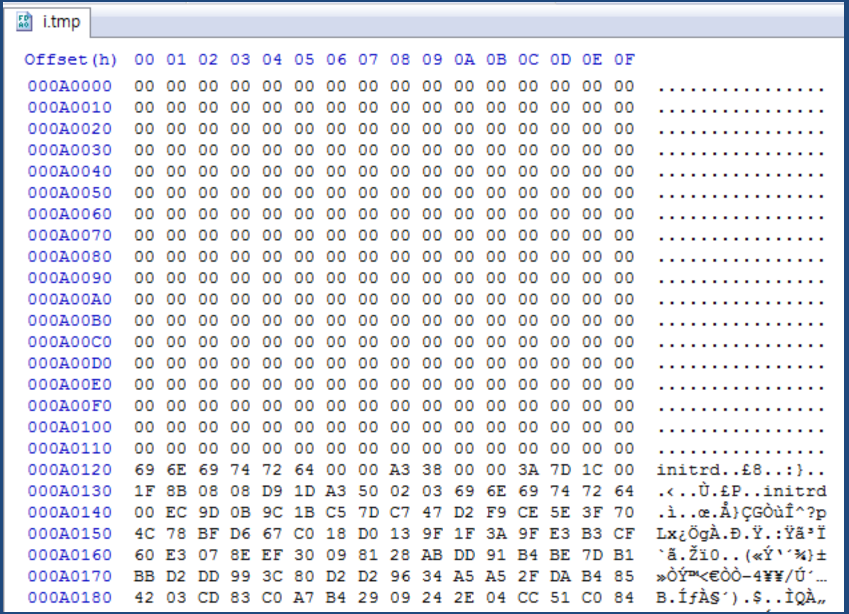
i.tmp의 구조

- Iptime의 부트로더에서 사용하는 이미지 파일
- kernel과 initrd를 포함하고 있다.



i.tmp

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  6B 65 72 6E 65 6C 00 00 10 01 0A 00 D9 37 F4 04  kernel......Ù7ô.
00000010  00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1  .. á.. á.. á.. á
00000020  00 00 A0 E1 00 00 A0 E1 00 00 A0 E1 00 00 A0 E1  .. á.. á.. á.. á
00000030  02 00 00 EA 18 28 6F 01 00 80 00 00 D8 EC 09 00  ...ê.(o..€..Øì..
00000040  01 70 A0 E1 00 80 A0 E3 00 20 0F E1 03 00 12 E3  .p á.€ ã. .á...ã
00000050  01 00 00 1A 17 00 A0 E3 56 34 12 EF 00 20 0F E1  ...... ãV4.ï. .á
00000060  C0 20 82 E3 02 F0 21 E1 B4 70 A0 E3 00 00 00 00  À ,ã.ð!á´p ã....
00000070  CC 00 8F E2 7E 30 90 E8 01 00 50 E0 00 00 30 E3  Ì..â~0.è..Pà..0ã
00000080  0A 00 00 0A 00 50 85 E0 00 60 86 E0 00 C0 8C E0  .....P…à.`†à.ÀŒà
00000090  00 20 82 E0 00 30 83 E0 00 D0 8D E0 00 10 96 E5  . ,à.0ƒà.Ð.à..–å
000000A0  00 10 81 E0 04 10 86 E4 0C 00 56 E1 FA FF FF 3A  ...à..†ä..Váúÿÿ:
000000B0  00 00 A0 E3 04 00 82 E4 04 00 82 E4 04 00 82 E4  .. ã..,ä..,ä..,ä
000000C0  04 00 82 E4 03 00 52 E1 F9 FF FF 3A 27 00 00 EB  ..,ä..Ráùÿÿ:'..ë
000000D0  0D 10 A0 E1 01 28 8D E2 02 00 54 E1 14 00 00 2A  .. á.(.â..Tá...*
000000E0  01 05 84 E2 05 00 50 E1 11 00 00 9A 02 50 A0 E1  ..„â..Pá...š.P á
000000F0  05 00 A0 E1 07 30 A0 E1 61 0A 00 EB 7F 00 80 E2  .. á.0 áa..ë..€â
00000100  7F 00 C0 E3 00 10 85 E0 05 2D 8F E2 50 30 9F E5  ..Àã...…à.-.âP0Ÿå
00000110  03 30 82 E0 00 3F B2 E8 00 3F A1 E8 00 3F B2 E8  .0,à.?²è.?¡è.?²è
00000120  00 3F A1 E8 03 00 52 E1 F9 FF FF 3A A7 00 00 EB  .?¡è..Ráùÿÿ:§..ë
00000130  00 F0 85 E0 04 00 A0 E1 07 30 A0 E1 50 0A 00 EB  .ð…à.. á.0 áP..ë
00000140  4E 00 00 EA 34 81 00 00 D8 EC 09 00 10 71 0A 00  N..ê4...Øì...q..
00000150  00 80 00 00 00 80 00 00 10 EC 09 00 CC EC 09 00  .€...€...ì..Ìì..
00000160  10 81 0A 00 B4 01 00 00 00 00 00 00 00 00 00 00  ....´...........
00000170  08 30 A0 E3 46 00 00 EA 01 39 44 E2 FF 30 C3 E3  .0 ãF..ê.9DâÿOÃã
00000180  3F 3C C3 E3 03 00 A0 E1 20 89 A0 E1 08 89 A0 E1  ?<Ãã.. á ‰ á.‰ á
```



i.tmp

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

000A0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0120  69 6E 69 74 72 64 00 00 A3 38 00 00 3A 7D 1C 00  initrd..£8..:}..
000A0130  1F 8B 08 08 D9 1D A3 50 02 03 69 6E 69 74 72 64  .‹..Ù.£P..initrd
000A0140  00 EC 9D 0B 9C 1B C5 7D C7 47 D2 F9 CE 5E 3F 70  .ì..œ.Å}ÇGÒùÎ^?p
000A0150  4C 78 BF D6 67 C0 18 D0 13 9F 1F 3A 9F E3 B3 CF  Lx¿ÖgÀ.Ð.Ÿ.:Ÿã³Ï
000A0160  60 E3 07 8E EF 30 09 81 28 AB DD 91 B4 BE 7D B1  `ã.Žï0..(«Ý'´¾}±
000A0170  BB D2 DD 99 3C 80 D2 D2 96 34 A5 A5 2F DA B4 85  »ÒÝ™<€ÒÒ–4¥¥/Ú´…
000A0180  42 03 CD 83 C0 A7 B4 29 09 24 2E 04 CC 51 C0 84  B.ÍƒÀ§´).$..ÌQÀ„
```

Root File System 파일 추출

Initrd 추출

- binwalk -e i.tmp

- # file initrd
  - initrd: Linux rev 1.0 ext2 filesystem data (mounted or unclean), UUID=fbc0cc35-5c72-4ef0-bc05-5d6b9bdc8e50

- mkdir FILE_SYSTEM
- mount initrd ./FILE_SYSTEM

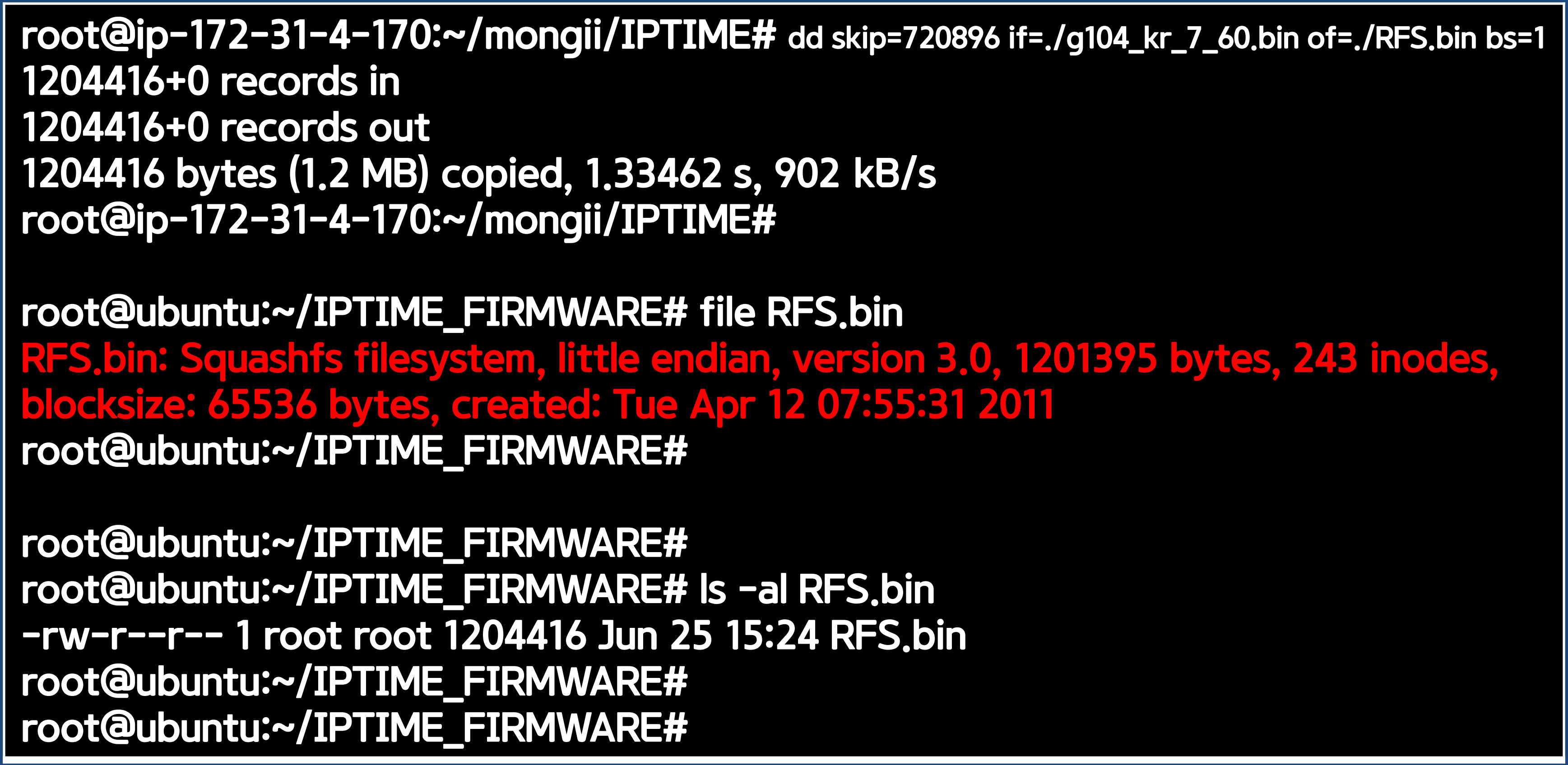
Initrd 추출

```
root@ip-172-31-4-170:~/mongii/IPTIME# cd FILE_SYSTEM/
root@ip-172-31-4-170:~/mongii/IPTIME/FILE_SYSTEM# ls -al
total 26
drwxr-xr-x 12 root root  1024 Apr 12  2011 .
drwxr-xr-x  3 root root  4096 Jun 25 15:22 ..
lrwxrwxrwx  1 root root    11 Apr 12  2011 bin -> /cramfs/bin
drwxr-xr-x  2  510  504  1024 Apr 12  2011 cramfs
drwxr-xr-x  3  510  504  1024 Apr 12  2011 dev
drwxr-xr-x  5  510  504  1024 Apr 12  2011 etc
drwxr-xr-x  3  510  504  1024 Apr 12  2011 home
lrwxrwxrwx  1 root root    11 Apr 12  2011 lib -> /cramfs/lib
drwx------  2 root root 12288 Apr 12  2011 lost+found
lrwxrwxrwx  1 root root    13 Apr 12  2011 ndbin -> /cramfs/ndbin
drwxr-xr-x  2  510  504  1024 Apr 12  2011 proc
drwxr-xr-x  2  510  504  1024 Apr 12  2011 save
lrwxrwxrwx  1 root root    12 Apr 12  2011 sbin -> /cramfs/sbin
drwxr-xr-x  2  510  504  1024 Apr 12  2011 tmp
drwxr-xr-x  2  510  504  1024 Apr 12  2011 upgrade-bin
lrwxrwxrwx  1 root root    11 Apr 12  2011 usr -> /cramfs/usr
drwxr-xr-x  5  510  504  1024 Apr 12  2011 var
root@ip-172-31-4-170:~/mongii/IPTIME/FILE_SYSTEM#
```

Binwalk 결과 재확인

```
root@ip-172-31-4-170:~/mongii/IPTIME# binwalk g104_kr_7_60.bin

DECIMAL         HEX             DESCRIPTION
-------------------------------------------------------------------------------------------------
65592           0x10038         gzip compressed data, was "i.tmp", from Unix, last modified: Tue Apr 12 07:55:31 2011
720896          0xB0000         Squashfs filesystem, little endian, version 3.0, size: 1201395 bytes,  243 inodes, blocksize: 65536 bytes, created: Tue Apr 12 07:55:31 2011

root@ip-172-31-4-170:~/mongii/IPTIME#
```

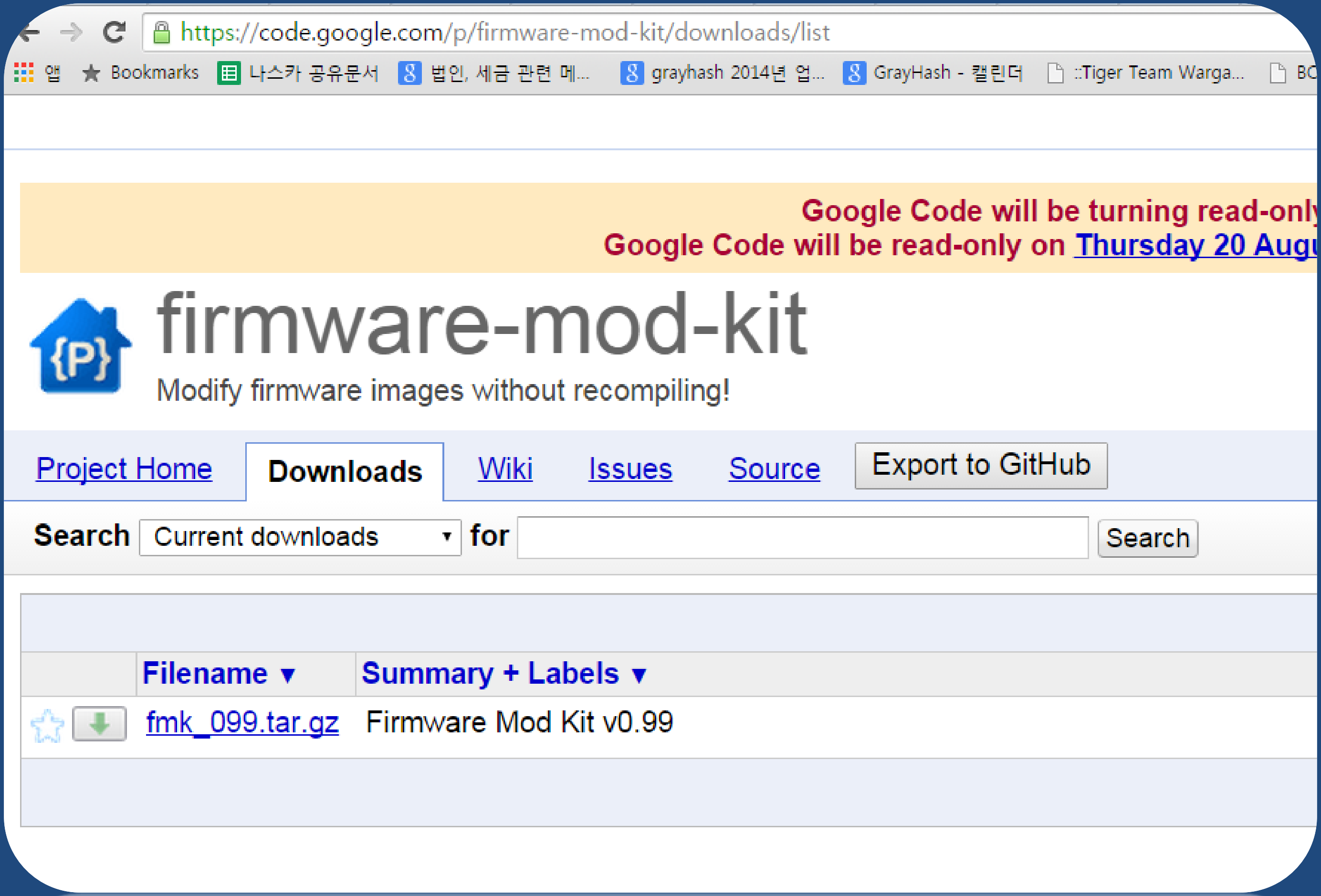
Extraction



```
root@ip-172-31-4-170:~/mongii/IPTIME# dd skip=720896 if=./g104_kr_7_60.bin of=./RFS.bin bs=1
1204416+0 records in
1204416+0 records out
1204416 bytes (1.2 MB) copied, 1.33462 s, 902 kB/s
root@ip-172-31-4-170:~/mongii/IPTIME#

root@ubuntu:~/IPTIME_FIRMWARE# file RFS.bin
RFS.bin: Squashfs filesystem, little endian, version 3.0, 1201395 bytes, 243 inodes,
blocksize: 65536 bytes, created: Tue Apr 12 07:55:31 2011
root@ubuntu:~/IPTIME_FIRMWARE#

root@ubuntu:~/IPTIME_FIRMWARE#
root@ubuntu:~/IPTIME_FIRMWARE# ls -al RFS.bin
-rw-r--r-- 1 root root 1204416 Jun 25 15:24 RFS.bin
root@ubuntu:~/IPTIME_FIRMWARE#
root@ubuntu:~/IPTIME_FIRMWARE#
```

Firmware-mod-kit

- https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/firmware-mod-kit/fmk_099.tar.gz



https://code.google.com/p/firmware-mod-kit/downloads/list

Google Code will be turning read-onl
Google Code will be read-only on Thursday 20 Augu

firmware-mod-kit
Modify firmware images without recompiling!

Filename ▾	Summary + Labels ▾
fmk_099.tar.gz	Firmware Mod Kit v0.99

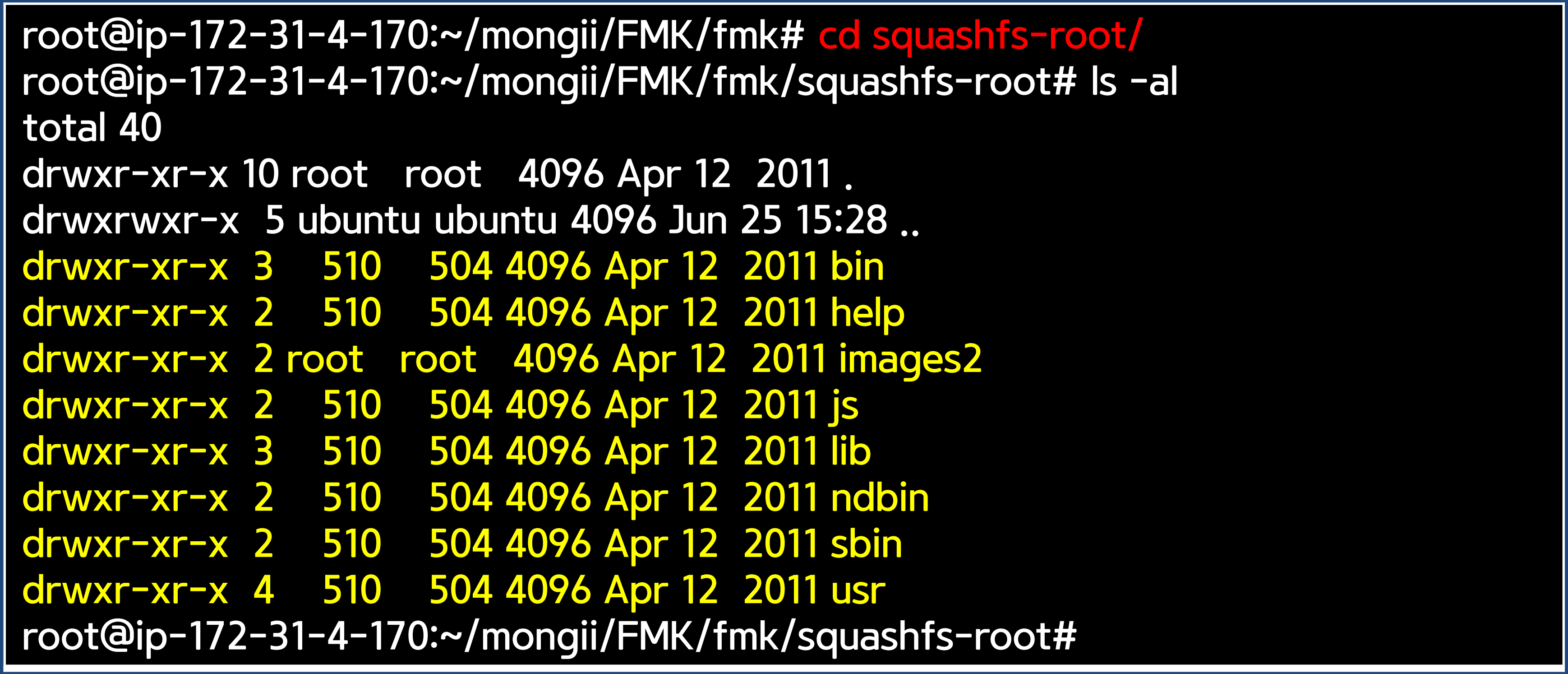
FMK 설치

- # apt-get install git build-essential zlib1g-dev liblzma-dev python-magic
- tar xvfz fmk_099.tar.gz
- cd fmk/src
- ./configure
- make
- cd ..

Squashfs 추출

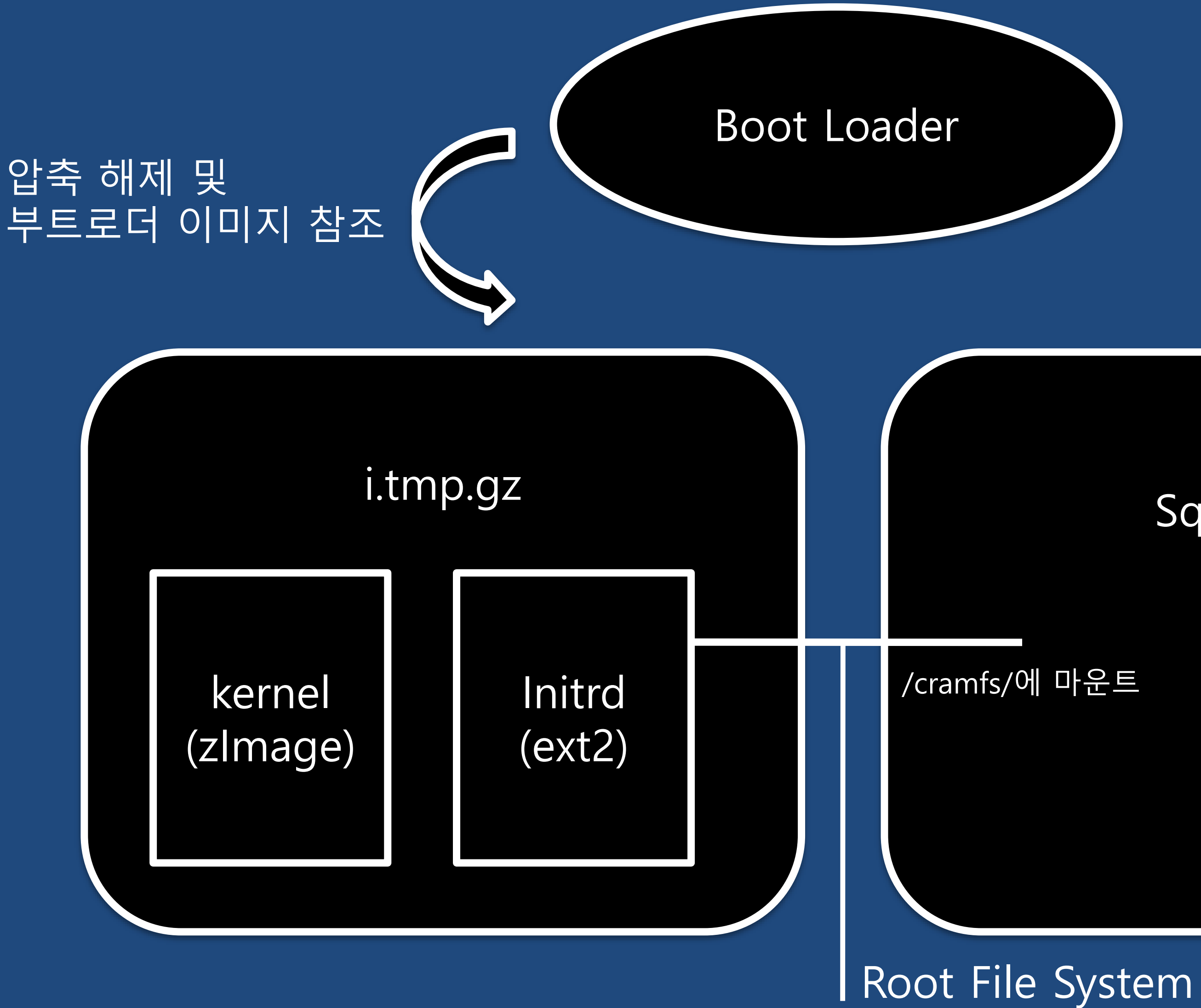
```
root@ip-172-31-4-170:~/mongii/FMK/fmk# ./unsquashfs_all.sh RFS.bin (B0000.squashfs)

Attempting to extract SquashFS .X file system...


Trying ./src/squashfs-2.1-r2/unsquashfs-lzma...
Trying ./src/squashfs-2.1-r2/unsquashfs...
Trying ./src/squashfs-3.0/unsquashfs-lzma...
created 173 files
created 17 directories
created 53 symlinks
created 0 devices
created 0 fifos
File system sucessfully extracted!
MKFS="./src/squashfs-3.0/mksquashfs-lzma"
root@ip-172-31-4-170:~/mongii/FMK/fmk#
```

파일 시스템 추출 결과



```
root@ip-172-31-4-170:~/mongii/FMK/fmk# cd squashfs-root/
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root# ls -al
total 40
drwxr-xr-x 10 root   root   4096 Apr 12  2011 .
drwxrwxr-x  5 ubuntu ubuntu 4096 Jun 25 15:28 ..
drwxr-xr-x  3    510    504 4096 Apr 12  2011 bin
drwxr-xr-x  2    510    504 4096 Apr 12  2011 help
drwxr-xr-x  2 root   root   4096 Apr 12  2011 images2
drwxr-xr-x  2    510    504 4096 Apr 12  2011 js
drwxr-xr-x  3    510    504 4096 Apr 12  2011 lib
drwxr-xr-x  2    510    504 4096 Apr 12  2011 ndbin
drwxr-xr-x  2    510    504 4096 Apr 12  2011 sbin
drwxr-xr-x  4    510    504 4096 Apr 12  2011 usr
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root#
```

Iptime 펌웨어의 구조



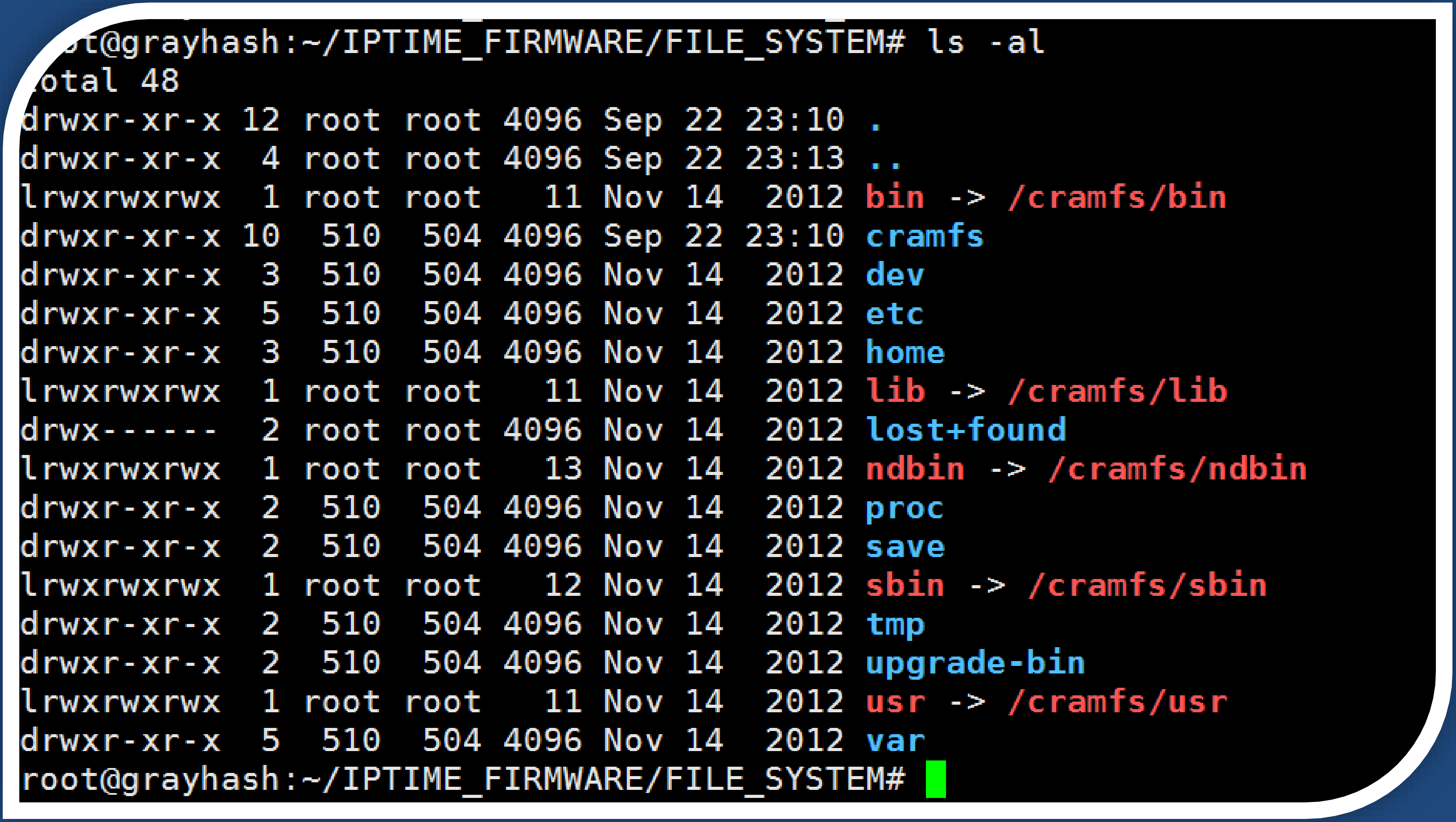
파일 시스템 복원

- initrd 마운트
  - mount initrd FILE_SYSTEM

- Squashfs 파일 추출
  - unsquashfs_all.sh B0000.squashfs

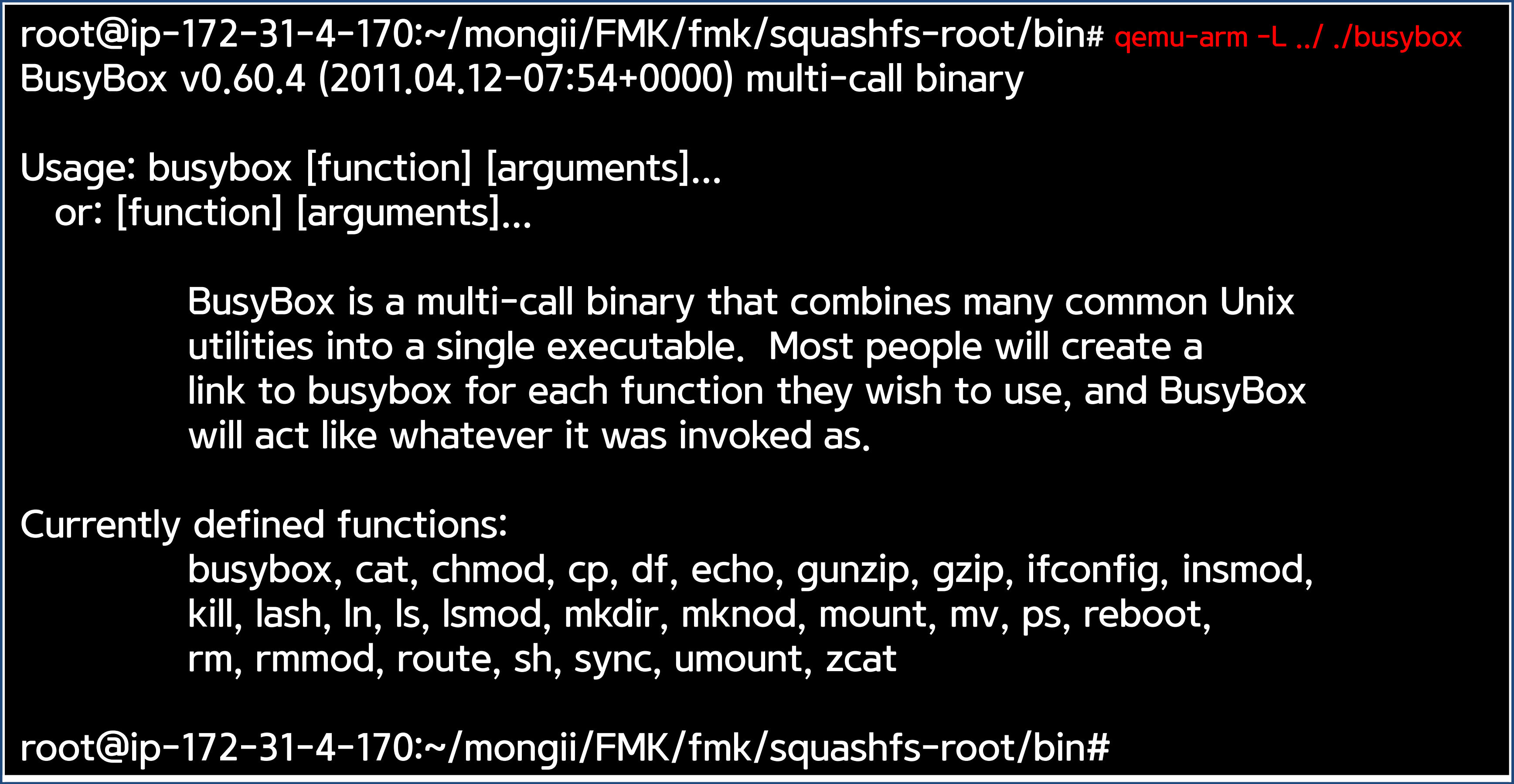
- 합치기
  - mkdir ALL_FILE_SYSTEM
  - cd ALL_FILE_SYSTEM
  - cp XXX/FILE_SYSTEM/* . -Rfpd
  - cp YYY/squashfs-root/* ./cramfs/ -Rfpd

파일 시스템 복원



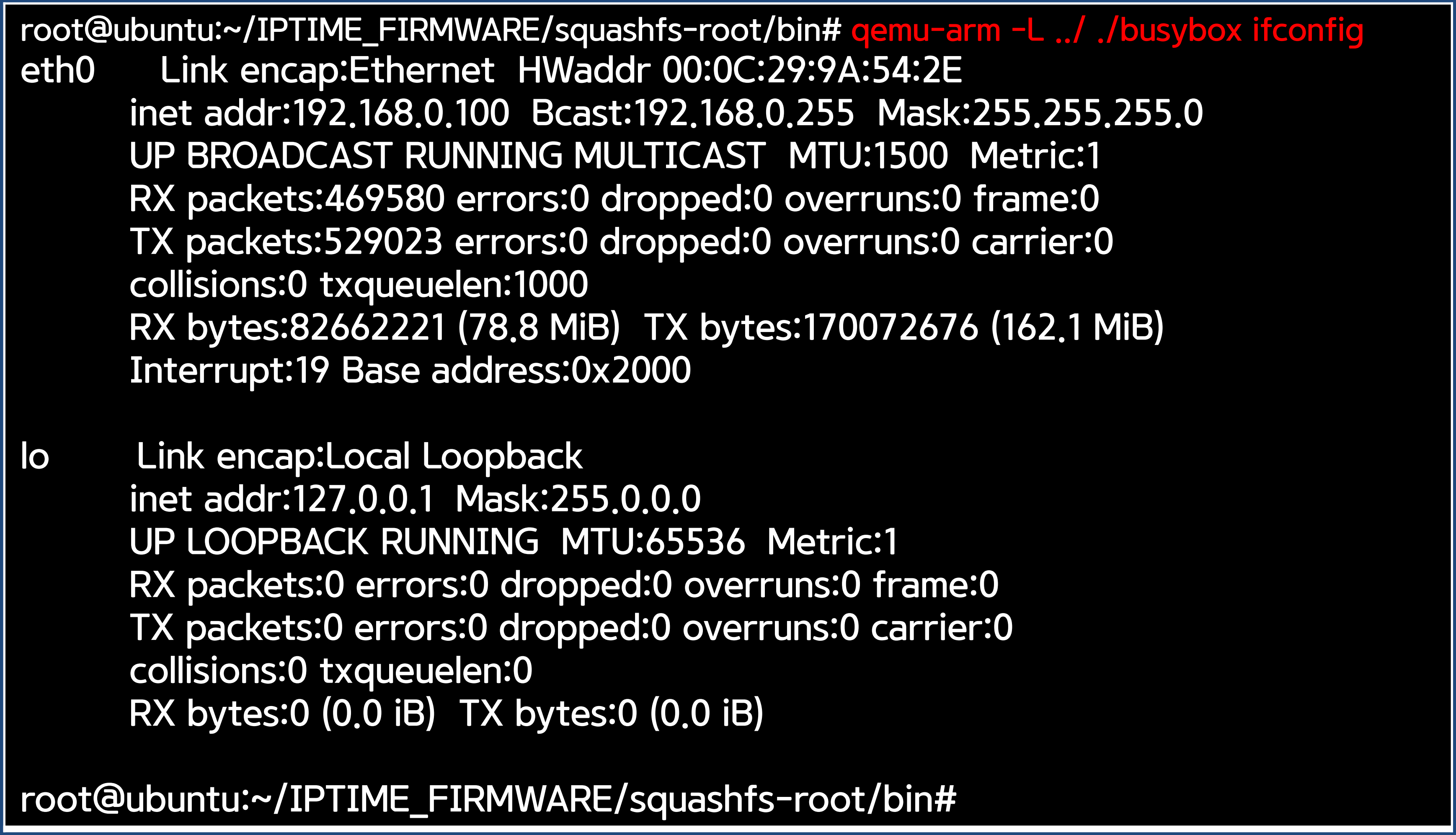
```
  t@grayhash:~/IPTIME_FIRMWARE/FILE_SYSTEM# ls -al
 otal 48
drwxr-xr-x 12 root root 4096 Sep 22 23:10 .
drwxr-xr-x  4 root root 4096 Sep 22 23:13 ..
lrwxrwxrwx  1 root root   11 Nov 14  2012 bin -> /cramfs/bin
drwxr-xr-x 10  510  504 4096 Sep 22 23:10 cramfs
drwxr-xr-x  3  510  504 4096 Nov 14  2012 dev
drwxr-xr-x  5  510  504 4096 Nov 14  2012 etc
drwxr-xr-x  3  510  504 4096 Nov 14  2012 home
lrwxrwxrwx  1 root root   11 Nov 14  2012 lib -> /cramfs/lib
drwx------  2 root root 4096 Nov 14  2012 lost+found
lrwxrwxrwx  1 root root   13 Nov 14  2012 ndbin -> /cramfs/ndbin
drwxr-xr-x  2  510  504 4096 Nov 14  2012 proc
drwxr-xr-x  2  510  504 4096 Nov 14  2012 save
lrwxrwxrwx  1 root root   12 Nov 14  2012 sbin -> /cramfs/sbin
drwxr-xr-x  2  510  504 4096 Nov 14  2012 tmp
drwxr-xr-x  2  510  504 4096 Nov 14  2012 upgrade-bin
lrwxrwxrwx  1 root root   11 Nov 14  2012 usr -> /cramfs/usr
drwxr-xr-x  5  510  504 4096 Nov 14  2012 var
root@grayhash:~/IPTIME_FIRMWARE/FILE_SYSTEM# 
```

Qemu로 돌리기



```
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin# qemu-arm -L ../ ./busybox
BusyBox v0.60.4 (2011.04.12-07:54+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, df, echo, gunzip, gzip, ifconfig, insmod,
                kill, lash, ln, ls, lsmod, mkdir, mknod, mount, mv, ps, reboot,
                rm, rmmod, route, sh, sync, umount, zcat

root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin#
```

Qemu로 돌리기



```
root@ubuntu:~/IPTIME_FIRMWARE/squashfs-root/bin# qemu-arm -L ../ ./busybox ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:9A:54:2E
          inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:469580 errors:0 dropped:0 overruns:0 frame:0
          TX packets:529023 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:82662221 (78.8 MiB)  TX bytes:170072676 (162.1 MiB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  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 iB)  TX bytes:0 (0.0 iB)

root@ubuntu:~/IPTIME_FIRMWARE/squashfs-root/bin#
```

Qemu로 돌리기

```
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin# qemu-arm -L ../ ./timepro.cgi
Content-type: text/html; charset=euc-kr

<html>
<script>


...

                                   if( ipstr == '151.35583.255.199')                          {
                                               return document.getElementsByName(ip+4)[0];
                                   }                                   return 0;              }


</script>
<head><title> </title>
<style></style></head>
</html>
root@ip-172-31-4-170:~/mongii/FMK/fmk/squashfs-root/bin#
```

가상 IPTIME 시스템

- cd 구성한 IPTIME 파일시스템 경로
  - # find . | cpio -o --format=newc > ../rootfs.img

- gzip -c ../rootfs.img > rootfs.img.gz

- zImage : 앞서 실습을 통해 만든 zImage 파일
  - iptime 펌웨어에서 추출한 zImage는 보드 호환이 되지 않음

- qemu-system-arm -M versatilepb -m 128M -kernel zImage -initrd rootfs.img.gz -append "root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200" -nographic

- mount -t proc /proc /proc
- ps -aux

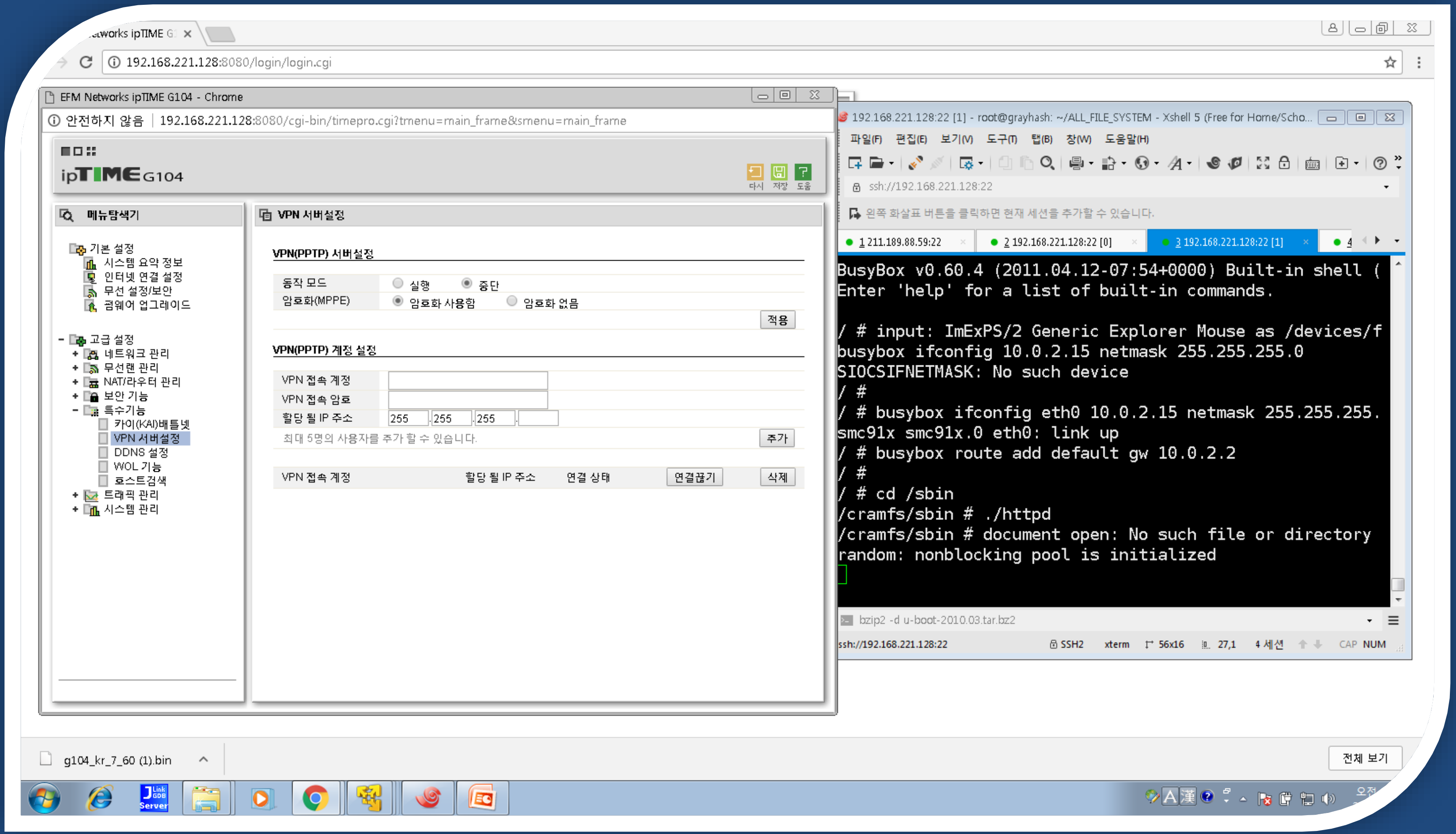
가상 IPTIME 시스템

```
Uncompressing Linux... done, booting the kernel.
Booting Linux on physical CPU 0x0
Linux version 4.1.6 (root@ubuntu) (gcc version 4.4.1 (Sourcery G++ Lite 2009q3-67) ) #1 Thu Aug 20 17:46:08 KST 2015
CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00093177
CPU: VIVT data cache, VIVT instruction cache
Machine: ARM-Versatile PB
Memory policy: Data cache writeback
sched_clock: 32 bits at 24MHz, resolution 41ns, wraps every 89478484971ns
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 32512
Kernel command line: root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200
PID hash table entries: 512 (order: -1, 2048 bytes)
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 121596K/131072K available (3209K kernel code, 139K rwdata, 796K rodata, 120K init, 119K bss, 9476K reserved, 0K cma-reserved)
Virtual kernel memory layout:
    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xffc00000 - 0xfff00000   (3072 kB)
    vmalloc : 0xc8800000 - 0xff000000   ( 872 MB)
    lowmem  : 0xc0000000 - 0xc8000000   ( 128 MB)
    modules : 0xbf000000 - 0xc0000000   (  16 MB)
      .text : 0xc0008000 - 0xc03f1944   (4007 kB)
      .init : 0xc03f2000 - 0xc0410000   ( 120 kB)
      .data : 0xc0410000 - 0xc0432e00   ( 140 kB)
       .bss : 0xc0432e00 - 0xc0450d04   ( 120 kB)
NR_IRQS:224

...


BusyBox v0.60.4 (2015.08.11-09:18+0000) Built-in shell (lash)
Enter 'help' for a list of built-in commands.

input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
/ # input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input2

/ #
```

Network 활성화

```
root@grayhash:~/ALL_FILE_SYSTEM# qemu-system-arm -M versatilepb -m 128M -kernel zImage -
initrd rootfs.img.gz -append "root=/dev/ram rdinit=/bin/sh console=ttyAMA0,115200" -nographic -
redir tcp:8080::80



...


/ #
/ # busybox ifconfig eth0 10.0.2.15 netmask 255.255.255.0
smc91x smc91x.0 eth0: link up
/ # busybox route add default gw 10.0.2.2
/ #
/ # cd /sbin
/cramfs/sbin # ./httpd
/cramfs/sbin #
```

관리자 페이지 접속


192.168.221.128:8080/login/login.cgi
EFM Networks ipTIME G104 - Chrome
안전하지 않음 | 192.168.221.128:8080/cgi-bin/timepro.cgi?tmenu=main_frame&smenu=main_frame
ipTIME G104
메뉴탐색기
기본 설정
시스템 요약 정보
인터넷 연결 설정
무선 설정/보안
펌웨어 업그레이드
고급 설정
네트워크 관리
무선랜 관리
NAT/라우터 관리
보안 기능
특수기능
카이(KAI)배틀넷
VPN 서버설정
DDNS 설정
WOL 기능
호스트검색
트래픽 관리
시스템 관리
VPN 서버설정
g104_kr_7_60 (1).bin
전체 보기
192.168.221.128:22 [1] - root@grayhash: ~/ALL_FILE_SYSTEM - Xshell 5 (Free for Home/Scho...
파일(F) 편집(E) 보기(V) 도구(T) 탭(B) 창(W) 도움말(H)
ssh://192.168.221.128:22
왼쪽 화살표 버튼을 클릭하면 현재 세션을 추가할 수 있습니다.
1 211.189.88.59:22
2 192.168.221.128:22 [0]
3 192.168.221.128:22 [1]
bzip2 -d u-boot-2010.03.tar.bz2
ssh://192.168.221.128:22 SSH2 xterm 56x16 27,1 4 세션 CAP NUM

VPN(PPTP) 서버설정

동작 모드	○ 실행 [✓ 중단]
암호화(MPPE)	[✓ 암호화 사용함] ○ 암호화 없음

적용

VPN(PPTP) 계정 설정

VPN 접속 계정	
VPN 접속 암호	
할당 될 IP 주소	255 . 255 . 255 .

최대 5명의 사용자를 추가 할 수 있습니다. 추가

VPN 접속 계정	할당 될 IP 주소	연결 상태	연결끊기	삭제

```
BusyBox v0.60.4 (2011.04.12-07:54+0000) Built-in shell (
Enter 'help' for a list of built-in commands.

/ # input: ImExPS/2 Generic Explorer Mouse as /devices/f
busybox ifconfig 10.0.2.15 netmask 255.255.255.0
SIOCSIFNETMASK: No such device
/ #
/ # busybox ifconfig eth0 10.0.2.15 netmask 255.255.255.
smc91x smc91x.0 eth0: link up
/ # busybox route add default gw 10.0.2.2
/ #
/ # cd /sbin
/cramfs/sbin # ./httpd
/cramfs/sbin # document open: No such file or directory
random: nonblocking pool is initialized
```

결론

- Flash Memory Dump를 통해 임베디드 기기 내의 Firmware를 추출하고 binary들의 취약점을 분석할 수 있다.
- Flash Memory도 결국 개발자가 다루는 주변 장치 중 하나에 불과하기 때문에 우리가 마음대로 다루는 것이 가능하다.

감사합니다.