

# 1. YAFFS 파일 시스템

[주의사항]

EZ-X5 의 CD에 제공하는 커널은 YAFFS 파일을 커널 자체에 포함시킨 상태로 제공되는 것입니다.

이후 설명은 YAFFS 파일 시스템이 탑재 되지 않은 상태에서 설명이므로 주의 하시기 바랍니다.

기존의 임베디드 시스템에 리눅스를 채용하는 이용 중 하나가 파일 시스템이다. 응용 프로그래머의 부담이 최소화 되기 때문이다.

대부분의 NOR 시스템의 파일 시스템은 MTD 와 JFFS2 시스템을 묶어서 사용한다. 하지만 실제 사용 중에 JFFS2 시스템의 불안정성이 문제가 된다. 특히 용량이 커지면 쓰기 속도에 문제가 발생하고 점유 메모리가 크게 늘어난다.

이런 단점을 해소하기 위해서 EZ-X5 는 MTD + YAFFS 를 도입하였다. 메모리 사용 효율은 JFFS2 보다 떨어지나 부팅시 마운트 속도가 매우 빠르고 NAND 플래시 시스템에서는 JFFS2 보다 월등히 앞선 성능을 보인다.

우선 진행하기 이전에 YAFFS에 대해서 간단하게 살펴 보고자 한다.

## YAFFS

YAFFS 는 MTD 기반의 파일 시스템이다. 즉 MTD의 블록 디바이스 시스템에 파일 시스템을 구성하기 위한 라이브러리이다. 하드 디스크에서 사용되는 대표적인 파일 시스템으로는 ext2 가 있다. 이와 동일한 개념으로 보면 된다.

YAFFS 관련 공식 사이트는 <http://www.aleph1.co.uk/armlinux/projects/yaffs/> 이다  
YAFFS 관련 공식 사이트 보다 다음 사이트에서 파일을 다운로드 하기로 한다.

<http://husaberg.toby-churchill.com/balloon/linux-2.6.8.1-tcl1/fs/yaffs/>

## 2.1. Yaffs 설치 및 컴파일

위 사이트에서 다운로드해서 받는 방법이 어렵다면 CD에 제공된 /sw/yaffs/yaffs.tar.gz 를 사용한다.

여기서는 커널에 YAFFS를 포팅하는 것을 설명하기로 한다.

모듈로 하고자 할 경우에는 YAFFS 관련 공식 사이트에 접속하여 관련 파일을 다운로드하여 작업하면 된다.

다운로드 받은 YAFFS 관련 파일을 커널의 fs 디렉토리에 넣는다.

현재 작업은 제공된 CD에 있는 /sw/yaffs/yaffs.tar.gz 를 이용한다.

### **./fs/Kconfig** 를 수정한다.

```

config YAFFS_FS
    tristate "YAFFS filesystem support"
    depends on MTD_NAND
    help
        If you want YAFFS support on NAND flash devices you either need to have
        NAND mtd support or customise the NAND interface.

        YAFFS is a journaling filesystem tailored to the idiosyncrasies of
        NAND flash memory. For further details see <http://www.aleph1.co.uk>

        YAFFS is also available as a module (= code which can be
        inserted in and removed from the running kernel whenever you want).
        The module is called nfsd. If you want to compile it as a module,
        say M here and read <file:Documentation/modules.txt>. If unsure,
        say N.
462
463 config ██████████
464     tristate "kernel automounter version 4 support (also supports v3)"
465     help
466         The automounter is a tool to automatically mount remote file systems
467         on demand. This implementation is partially kernel-based to reduce
468         overhead in the already-mounted case; this is unlike the BSD
469         automounter (amd), which is a pure user space daemon.
470
471         To use the automounter you need the user-space tools from
472         <ftp://ftp.kernel.org/pub/linux/daemons/autofs/v4/>; you also
473         want to answer Y to "NFS file system support", below.
474
475         To compile this support as a module, choose M here: the module will be
476         called autofs4. You will need to add "alias autofs autofs4" to your
477         modules configuration file.
478
479         If you are not a part of a fairly large, distributed network or
480         don't have a laptop which needs to dynamically reconfigure to the
481         local network, you probably do not need an automounter, and can say
482         N here.
483
484 config YAFFS_FS
485     tristate "YAFFS filesystem support"
486     depends on MTD_NAND
487     help
488         If you want YAFFS support on NAND flash devices you either need to have
489         NAND mtd support or customise the NAND interface.
490
491         YAFFS is a journaling filesystem tailored to the idiosyncrasies of
492         NAND flash memory. For further details see <http://www.aleph1.co.uk>
493
494         YAFFS is also available as a module (= code which can be
495         inserted in and removed from the running kernel whenever you want).
496         The module is called nfsd. If you want to compile it as a module,
497         say M here and read <file:Documentation/modules.txt>. If unsure,
498         say N.
499
500 menu "CD-ROM/DVD Filesystems"
501
502 config ISO9660_FS
503     tristate "ISO 9660 CDRom file system support"
504     help
505         This is the standard file system used on CD-ROMs. It was previously
506         known as "High Sierra File System" and is called "hfsfs" on other
507         Unix systems. The so-called Rock-Ridge extensions which allow for
508         long Unix filenames and symbolic links are also supported by this
509         driver. If you have a CD-ROM drive and want to do more with it than
510         just listen to audio CDs and watch its LEDs, say Y (and read
511         <file:Documentation/filesystems/isofs.txt> and the CD-ROM-HOWTO,

```

./fs/Makefile 를 수정한다.

obj-\$(CONFIG\_YAFFS\_FS) += yaffs/

```

81 obj-$(CONFIG_JFFS2_FS)      += jffs2/
82 obj-$(CONFIG_AFFS_FS)      += affs/
83 obj-$(CONFIG_ROMFS_FS)     += romfs/
84 obj-$(CONFIG_QNX4FS_FS)    += qnx4/
85 obj-$(CONFIG_AUTofs_FS)    += autofs/
86 obj-$(CONFIG_ )           += autofs4/
87 obj-$(CONFIG_ADFS_FS)     += adfs/
88 obj-$(CONFIG_UDF_FS)      += udf/
89 obj-$(CONFIG_SUN_OPENPROMFS) += openpromfs/
90 obj-$(CONFIG_JFS_FS)      += jfs/
91 obj-$(CONFIG_XFS_FS)      += xfs/
92 obj-$(CONFIG_AFS_FS)      += afs/
93 obj-$(CONFIG_BEFS_FS)     += befs/
94 obj-$(CONFIG_YAFFS_FS)    += yaffs/
    
```

# make menuconfig

```

Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus -->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [ ] excluded <M> module < >
module capable
^(-)
 I2C support -->
 Multimedia devices -->
 File systems --->
 Profiling support -->

File systems
Arrow keys navigate the menu. <Enter> selects submenus -->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [ ] excluded <M> module < >
module capable
^(-)
 < > Minix fs support
 < * > ROM file system support
 [ ] Quota support
 < > Kernel automounter support
 < > Kernel automounter version 4 support (also supports v3)
 < * > YAFFS filesystem support
 CD-ROM/DVD Filesystems -->
 DOS/FAT/NT Filesystems -->
 Pseudo filesystems -->
 Miscellaneous filesystems -->
 Network File Systems -->
 Partition Types -->
 Native Language Support -->

<Select> < Exit > < Help >
    
```

File systems --->

< \* > YAFFS filesystem support

이제 커널을 컴파일 한다.

```
# make dep
# make clean
# make zImage
```

타겟 보드에 커널을 다운로드하고 부팅한다.

부팅 후 다음 명령으로 확인 할 수 있다.

파일 시스템 확인

```
[root@falinux ~]$ cat /proc/filesystems
nodev    sysfs
nodev    rootfs
nodev    bdev
nodev    proc
nodev    sockfs
nodev    usbfs
nodev    usbdevfs
nodev    futexfs
nodev    tmpfs
nodev    pipefs
nodev    eventpollfs
nodev    devpts
        ext3
        ext2
nodev    ramfs
        msdos
        vfat
nodev    nfs
nodev    nfs4
        romfs
        yaffs
nodev    rpc_pipefs
[root@falinux ~]$
```

마운트 확인

```
root@falinux ~]$ df
Filesystem      1k-blocks    Used Available Use% Mounted on
/dev/ram0        11895        7317    3964  65% /
/dev/mtdblock2  58352         128    58224   0% /app
[root@falinux ~]$
```

만약 위와 같이 Auto 마운트가 안되었다면 아래와 같이 직접 명령을 써 주면 된다.

```
[root@falinux ~]$ mount -t yaffs /dev/mtdblock2 /app
root@falinux ~]$ df
Filesystem      1k-blocks    Used Available Use% Mounted on
/dev/ram0        11895        7317    3964  65% /
/dev/mtdblock2  58352         128    58224   0% /app
[root@falinux ~]$
```

mtdblock0 는 커널, mtdblock1은 램디스크가 사용하고 있다. 따라서 사용자 영역인 mtdblock2를 마운트하여 사용하면 된다.