

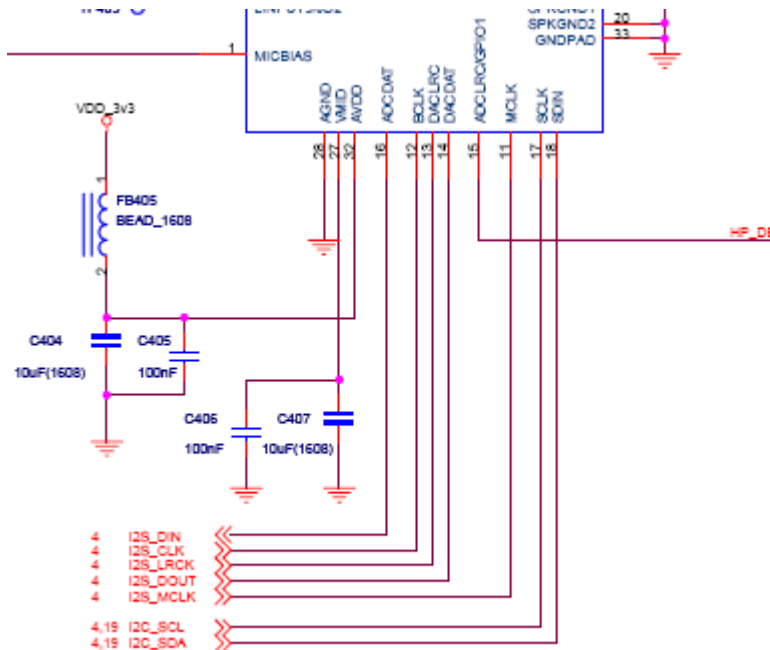
망고100 보드로 놀아보자-16

Codec,battery,powermanager 커널 드라이버

<http://cafe.naver.com/embeddedcrazyboys>

<http://www.mangoboard.com>

codec driver(wm8960)



WM8960 디바이스를
I2C 0번 채널로 디바이스를 컨트롤하고
Data는 I2S0번 채널로 전송토록 설계

12,19	I2C_SDA	↔	L26	Xi2c0SDA/GPD3
12,19	I2C_SCL	↔	F25	Xi2c0SCL/GPD4
18,19	I2C1_SDA	↔	K26	Xi2c1SDA/SPDIF_OUT/GPD5
18,19	I2C1_SCL	↔	K24	Xi2c1SCL/SPDIF_CLK/GPD6
12	I2S_CLK	↔	AB1	Xi2sSCLK0
12	I2S_MCLK	↔	Y1	Xi2sCDCLK0
12	I2S_LRCK	↔	U7	Xi2sLRCK0
12	I2S_DIN	↔	U12	Xi2sSDIO
12	I2S_DOUT	↔	R7	Xi2sSDO0_0
			W2	Xi2sSDO0_1
			W1	Xi2sSDO0_2

codec driver(wm8960)

```
Device Drivers
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, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

Generic Driver Options --->
connector - unified userspace <-> kernel-space linker --->
<-> Memory Technology Device (MTD) support --->
<-> Parallel port support --->
[*] Block devices --->
[*] Misc devices --->
<-> ATA/ATAPI/MFM/RLL support --->
SCSI device support --->
<-> Serial ATA (prod) and Parallel ATA (experimental) drivers --->
[ ] Multiple devices driver support (RAID and LVM) --->
[*] Network device support --->
[ ] ISDN support --->
Input device support --->
Character devices --->
[*] I2C support --->
[ ] SPI support --->
--> GPIO Support --->
<-> Dallas's 1-wire support --->
<-> Power supply class support --->
<-> Hardware Monitoring support --->
<-> generic Thermal sysfs driver --->
v(1)
<Select> < Exit > < Help >
```

if

```
config SND_SOC_ALL_CODECS
tristate "Build all ASoC CODEC drivers"
select SND_SOC_AC97_CODEC if SND_SOC_AC97_BUS
select SND_SOC_AD1980 if SND_SOC_AC97_BUS
select SND_SOC_AD73311 if I2C
select SND_SOC_AK4535 if I2C
select SND_SOC_CS4270 if I2C
select SND_SOC_PCM3008
select SND_SOC_SSM2602 if I2C
select SND_SOC_TLV320AIC23 if I2C
select SND_SOC_TLV320AIC26 if SPI_MASTER
select SND_SOC_TLV320AIC3X if I2C
select SND_SOC_TWL4030 if TWL4030_CORE
select SND_SOC_UDA134X
select SND_SOC_UDA1380 if I2C
select SND_SOC_WM8350 if MFD_WM8350
select SND_SOC_WM8510 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8580 if I2C
select SND_SOC_WM8728 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8731 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8750 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8753 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8900 if I2C
select SND_SOC_WM8903 if I2C
select SND_SOC_WM8960 if I2C
```

CONFIG_I2C=y 로 설정이 되어 있어야 SND_SOC_WM8960 이 활성화 됨
./ sound/soc/codecs/Kconfig 에 추가
cafe.naver.com/embeddedcrazyboys

codec driver(wm8960)

```
snd-soc-wm8900-objs := wm8900.o
snd-soc-wm8903-objs := wm8903.o
snd-soc-wm8960-objs := wm8960.o
snd-soc-wm8971-objs := wm8971.o
snd-soc-wm8990-objs := wm8990.o
snd-soc-wm9712-objs := wm9712.o
snd-soc-wm9713-objs := wm9713.o

obj-$(CONFIG_SND_SOC_AC97_CODEC) += snd-soc-ac97.o
obj-$(CONFIG_SND_SOC_AD1980) += snd-soc-ad1980.o
obj-$(CONFIG_SND_SOC_AD73311) += snd-soc-ad73311.o
obj-$(CONFIG_SND_SOC_AK4535) += snd-soc-ak4535.o
obj-$(CONFIG_SND_SOC_CS4270) += snd-soc-cs4270.o
"sound/soc/codecs/Makefile" 57L, 2335C
obj-$(CONFIG_SND_SOC_L3) += snd-soc-l3.o
obj-$(CONFIG_SND_SOC_PCM3008) += snd-soc-pcm3008.o
obj-$(CONFIG_SND_SOC_SSM2602) += snd-soc-ssm2602.o
obj-$(CONFIG_SND_SOC_TLV320A1C23) += snd-soc-tlv320a1c23.o
obj-$(CONFIG_SND_SOC_TLV320A1C26) += snd-soc-tlv320a1c26.o
obj-$(CONFIG_SND_SOC_TLV320A1C3X) += snd-soc-tlv320a1c3x.o
obj-$(CONFIG_SND_SOC_TWL4030) += snd-soc-twl4030.o
obj-$(CONFIG_SND_SOC_UDA134X) += snd-soc-uda134x.o
obj-$(CONFIG_SND_SOC_UDA1380) += snd-soc-uda1380.o
obj-$(CONFIG_SND_SOC_WM8350) += snd-soc-wm8350.o
obj-$(CONFIG_SND_SOC_WM8510) += snd-soc-wm8510.o
obj-$(CONFIG_SND_SOC_WM8580) += snd-soc-wm8580.o
obj-$(CONFIG_SND_SOC_WM8728) += snd-soc-wm8728.o
obj-$(CONFIG_SND_SOC_WM8731) += snd-soc-wm8731.o
obj-$(CONFIG_SND_SOC_WM8750) += snd-soc-wm8750.o
obj-$(CONFIG_SND_SOC_WM8753) += snd-soc-wm8753.o
obj-$(CONFIG_SND_SOC_WM8900) += snd-soc-wm8900.o
obj-$(CONFIG_SND_SOC_WM8903) += snd-soc-wm8903.o
obj-$(CONFIG_SND_SOC_WM8960) += snd-soc-wm8960.o
obj-$(CONFIG_SND_SOC_WM8971) += snd-soc-wm8971.o
```

snd-soc-wm8960.o==wm8960.o 동일
이유는 사운드 드라이버 소스가 2개이상 인
경우를 대비

./ sound/soc/codecs/Makefile 에 추가
Wm8960.c 드라이버 파일을
./sound/soc/codecs/에 추가

codec driver(wm8960)

```
config SND_S5P_MANG0100
    tristate "Soc Audio support Mango100"
    depends on SND_SAMSUNG_SOC && (MACH_MANG0100)
    select SND_S3C24XX_SOC
    help
        Say Y if you want to add support for SoC audio on the MANG0100.

choice
    prompt "Select MAnGo100 Audio Port Type"
    depends on SND_S5P_MANG0100

config SND_MANG0100_WM8960
    bool "WM8960 Driver"
    select SND_SOC_WM8960
    select SND_S5P_SOC_I2S
    select SND_S3C_I2SV2_SOC

config SND_MANG0100_HDMI_SPDIF
    bool "HDMI SPDIF Driver"
    select SND_S5P_SPDIF
```

sound/soc/s3c24xx/Kconfig 파일에 위의 내용 추가

codec driver(wm8960)

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaa ALSA for SoC audio support aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
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, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < > module
l
x --- ALSA for SoC audio support
x <+> Samsung SoC Audio Drivers
x <+> Soc Audio support Mango100
x Select Mango100 Audio Port Type (WM8960 Driver) --->
x < > SoC AC97 Audio support for LN2440SBC - ALC650
x < > SoC I2S Audio support UDA134X wired to a S3C24XX
x < > HDMI-SPDIF on SMDKS5P
x < > Build all ASoC CODEC drivers
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
m
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
<Select> < Exit > < Help >
```

#make menuconfig 명령으로 확인 할 있음

codec driver(wm8960)

```
obj-$(CONFIG_SND_S3C24XX_SOC) += snd-soc-s3c24xx.o
obj-$(CONFIG_SND_LPAM_SOC) += snd-soc-lpam.o
obj-$(CONFIG_SND_S3C24XX_SOC_I2S) += snd-soc-s3c24xx-i2s.o
obj-$(CONFIG_SND_S3C2443_SOC_AC97) += snd-soc-s3c2443-ac97.o
obj-$(CONFIG_SND_S3C2412_SOC_I2S) += snd-soc-s3c2412-i2s.o
obj-$(CONFIG_SND_S5P_SOC_AC97) += snd-soc-s5p-ac97.o
obj-$(CONFIG_SND_S5P_SOC_I2S) += snd-soc-s5p-i2s.o
obj-$(CONFIG_SND_S5P_SOC_I2S_LP) += snd-soc-s5p-i2s-lp.o
obj-$(CONFIG_SND_S3C_I2SV2_SOC) += snd-soc-s3c-i2s-v2.o
obj-$(CONFIG_SND_S3C_SOC_PCM) += snd-soc-s3c-pcm.o
obj-$(CONFIG_SND_S5P_SPDIF) += snd-soc-s5p-spdif.o

# S3C24XX Machine Support
snd-soc-neo1973-wm8753-objs := neo1973_wm8753.o
snd-soc-smdk2443-wm9710-objs := smdk2443_wm9710.o
snd-soc-ln2440sbc-alc650-objs := ln2440sbc_alc650.o
snd-soc-s3c24xx-uda134x-objs := s3c24xx_uda134x.o
snd-soc-smdks5p-wm9713-objs := smdks5p_wm9713.o
snd-soc-smdks5p-wm8580-objs := smdks5p_wm8580.o
snd-soc-smdks5plp-wm8580-objs := smdks5plp_wm8580.o
snd-soc-mango100-wm8960-objs := mango100_wm8960.o
snd-soc-universal-ak4671-objs := universal_ak4671.o
snd-soc-smdks5p-spdif-objs := smdks5p_hdmi_spdif.o
snd-soc-mango100-spdif-objs := mango100_hdmi_spdif.o

obj-$(CONFIG_SND_S3C24XX_SOC_NEO1973_WM8753) += snd-soc-neo1973-wm8753.o
obj-$(CONFIG_SND_S3C24XX_SOC_SMDK2443_WM9710) += snd-soc-smdk2443-wm9710.o
obj-$(CONFIG_SND_S3C24XX_SOC_LN2440SBC_ALC650) += snd-soc-ln2440sbc-alc650.o
obj-$(CONFIG_SND_S3C24XX_SOC_S3C24XX_UDA134X) += snd-soc-s3c24xx-uda134x.o
obj-$(CONFIG_SND_S5P_SOC_SMDK_WM9713) += snd-soc-smdks5p-wm9713.o
obj-$(CONFIG_SND_S5P_SOC_WM8580) += snd-soc-smdks5p-wm8580.o
obj-$(CONFIG_SND_S5P_SOC_WM8580_LP) += snd-soc-smdks5plp-wm8580.o
obj-$(CONFIG_SND_S5P_SOC_UNIVERSAL_AK4671) += snd-soc-universal-ak4671.o
obj-$(CONFIG_SND_SMDKS5P_HDMI_SPDIF) += snd-soc-smdks5p-spdif.o
obj-$(CONFIG_SND_MANGO100_WM8960) += snd-soc-mango100-wm8960.o
obj-$(CONFIG_SND_MANGO100_HDMI_SPDIF) += snd-soc-mango100-spdif.o
```

sound/soc/s3c24xx/Makefile을 위와 같이 수정

sound/soc/s3c24xx/mango100_hdmi_spdif.c

sound/soc/s3c24xx/mango100_wm8960.c

을 sound/soc/s3c24xx 디렉토리에 파일을 만들어서 추가

codec driver(wm8960)

```
static struct snd_soc_dai_link smdk_dai[] = {
{
    .name = "WM8960 I2S",
    .stream_name = "Tx/Rx",
    .cpu_dai = &s5p_i2s_dai[0],
    .codec_dai = &wm8960_dai,
    .init = smdk_wm8960_init,
    .ops = &smdk_i2s_ops,
},
};

static struct snd_soc_card smdk = {
    .name = "smdk",
    .platform = &s3c24xx_soc_platform,
    .dai_link = smdk_dai,
    .num_links = ARRAY_SIZE(smdk_dai),
};

static struct wm8960_setup_data smdk_wm8960_setup = {
    .i2c_bus = 0,
    .i2c_address = 0x1a,
};

static struct snd_soc_device smdk_snd_devdata = {
    .card = &smdk,
    .codec_dev = &soc_codec_dev_wm8960,
    .codec_data = &smdk_wm8960_setup,
};
```

```
# cat pcm
00-00: Tx/Rx WM8960-0 : : playback 1 : capture 1
# pwd
/proc/asound
```

```
drwxr-xr-x 3 0 0 0 Apr 22 17:52 a
drwxr-xr-x 3 0 0 0 Apr 22 17:52 c
lrwxrwxrwx 1 0 0 0 Apr 22 17:54 d
audio
drwxr-xr-x 3 0 0 0 Apr 22 17:52 d
-rw-r--r-- 1 0 0 4096 Apr 22 17:54 i
drwxr-xr-x 3 0 0 0 Apr 22 17:52 m
-r--r--r-- 1 0 0 4096 Apr 22 17:54 n
drwxr-xr-x 3 0 0 0 Apr 22 17:52 p
drwxr-xr-x 3 0 0 0 Apr 22 17:52 p
drwxr-xr-x 2 0 0 0 Apr 22 17:52 p
lrwxrwxrwx 1 0 0 0 Apr 22 17:54 s
./../class/sound
-rw-r--r-- 1 0 0 4096 Apr 22 17:52 u
# pwd
/sys/devices/platform/soc-audio/driver/soc-audio/sound/car
```

```
/sys/bus/i2c/devices/0-001a/driver
# ls -al
drwxr-xr-x 2 0 0 0 Apr 22 17:38 .
drwxr-xr-x 7 0 0 0 Apr 22 17:37 ..
lrwxrwxrwx 1 0 0 0 Apr 22 17:39 0-001a ->
evices/platform/s3c2410-i2c.0/i2c-adapter/i2c-0/0-001a
--w----- 1 0 0 4096 Apr 22 17:39 bind
--w----- 1 0 0 4096 Apr 22 17:39 uevent
--w----- 1 0 0 4096 Apr 22 17:39 unbind
```


codec driver(wm8960)

```
static const struct i2c_device_id wm8960_i2c_id[] = {
    { "wm8960", 0 },
    { }
};
MODULE_DEVICE_TABLE(i2c, wm8960_i2c_id);

static struct i2c_driver wm8960_i2c_driver = {
    .driver = {
        .name = "WM8960 I2C Codec",
        .owner = THIS_MODULE,
    },
    .probe =    wm8960_i2c_probe,
    .remove =  __devexit_p(wm8960_i2c_remove),
    .id_table = wm8960_i2c_id,
};
```

Sound/soc/codecs/wm8960.c 파일 참조

```
drwxr-xr-x  3 0      0      0 Apr 22 18:26 .
drwxr-xr-x  5 0      0      0 Apr 22 18:26 ..
lrwxrwxrwx  1 0      0      0 Apr 22 18:39 bus -> ../../../../bus/
i2c
lrwxrwxrwx  1 0      0      0 Apr 22 18:39 driver -> ../../../../b
us/i2c/drivers/WM8960 I2C Codec
-r--r--r--  1 0      0      0 Apr 22 18:39 modalias
-r--r--r--  1 0      0      0 Apr 22 18:39 name
drwxr-xr-x  2 0      0      0 Apr 22 18:26 power
lrwxrwxrwx  1 0      0      0 Apr 22 18:39 subsystem -> ../../../../
./bus/i2c
-rw-r--r--  1 0      0      0 Apr 22 18:26 uevent
# pwd
/sys/class/i2c-adapter/i2c-0/0-001a/driver/0-001a
```

codec driver(wm8960)

The WM8960 is controlled by writing to registers through a 2-wire serial control interface. A control word consists of 16 bits. The first 7 bits (B15 to B9) are address bits that select which control register is accessed. The remaining 9 bits (B8 to B0) are data bits, corresponding to the 9 bits in each control register. Many devices can be controlled by the same bus, and each device has a unique 7-bit address (this is not the same as the 7-bit address of each register in the WM8960).

The device address is 0011010 (0x34h).

```
static struct wm8960_setup_data smdk_wm8960_setup = {  
    .i2c_bus = 0,  
    .i2c_address = 0x1a,  
};
```

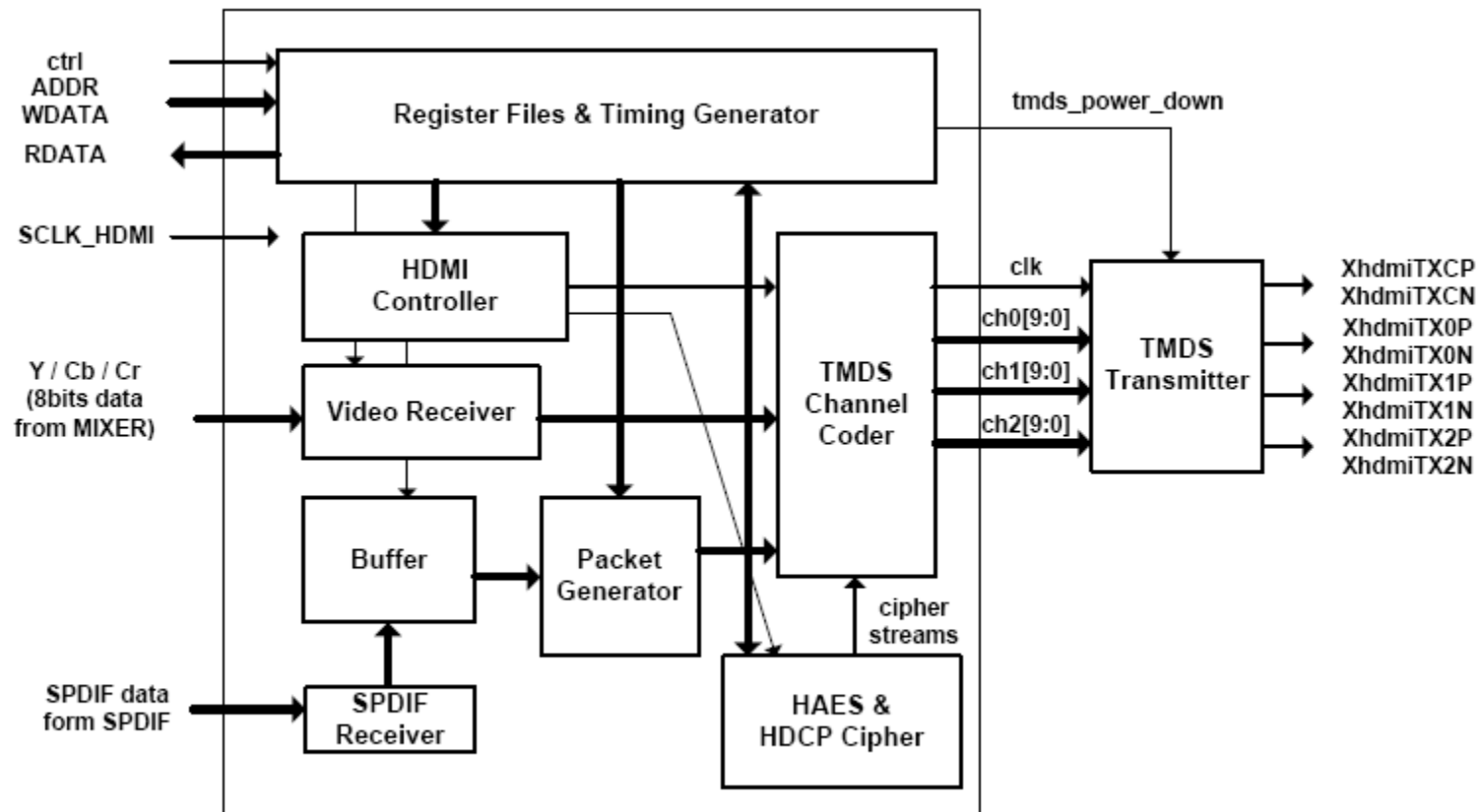
sound/soc/s3c24xx/mango100_wm8960.c에 정의

```
static struct i2c_board_info i2c_devs0[] __initdata = {  
    { I2C_BOARD_INFO("wm8960", 0x1a), }, // WM8960  
};
```

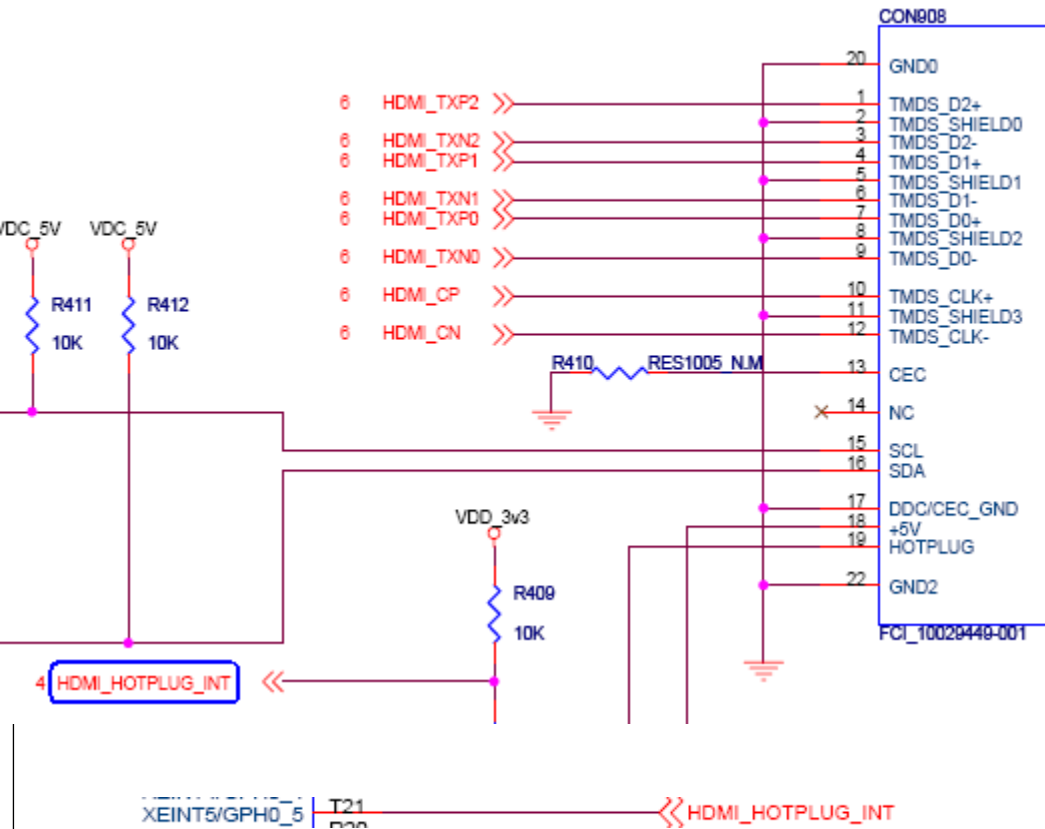
../arch/arm/mach-s5pc100/mach-mango100.c 에 정의

```
static void s3c24xx_i2c_message_start(struct s3c24xx_i2c *i2c,  
                                     struct i2c_msg *msg)  
{  
    unsigned int addr = (msg->addr & 0x7f) << 1;
```

HDMI Driver



HDMI Driver



```

static struct resource s5p_tvout_resources[] = {
..
[8] = {
    .start = IRQ_EINT5,
    .end = IRQ_EINT5,
    .flags = IORESOURCE_IRQ
}
./arch/arm/plat-s5pc1xx/devs.c에 수정,
드라이버 소스는
drivers/media/video/samsung/tv20/s5pc100/hdmi_s5pc100.c
  
```

HDMI S/PDIF Driver

S/PDIF의 약자는 Sony/Philips Digital InterFace의 약자입니다.
소니와 필립스사에서 디지털 오디오 전송을 위해서 만든 표준 인터페이스로써,
신호선 1개와 그라운드선 1개 이렇게 2가닥을 사용하는 방식입니다.

기존의 디지털 오디오 신호의 경우 동기(싱크신호)신호를 넣어야 했고
또 신호의 간섭이 심하여 일정 이상의 거리를 전송하기 어려웠던 반면에
S/PDIF는 약 20 미터 정도까지 일정하게 데이터를 전달할 수 있어
CDP, 사운드 카드, CD-ROM drive, A/V 리시버에 다양하게 사용되고 있습니다.

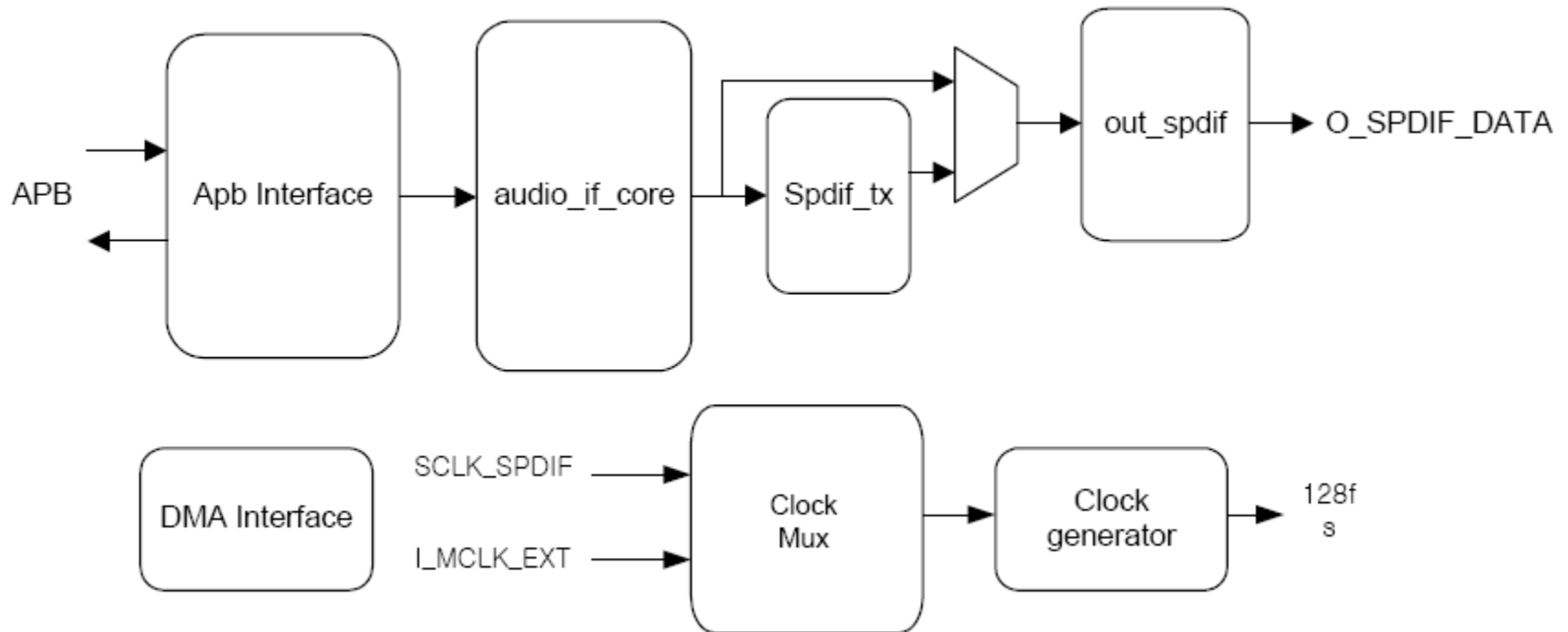
사람이 들을 수 있는 신호는 모두 아날로그이며, 컴퓨터는 궁극적으로
디지털 신호만을 입출력합니다. 컴퓨터의 경우도 아날로그 출력은
디지털 신호를 아날로그로 컨버트(Digital to Analog Converter)하여
출력하는 것이고, 반대로 아날로그 신호를 저장하려면
ADC(Analog to Digital Converter)를 사용해야 합니다.

예를 들면 컴퓨터의 MP3 데이터(디지털)을 MD(디지털)로 저장하려면
디지털 신호를 사용하면 디지털 -> 디지털로 전달하면 되지만
아날로그 신호의 경우는 사운드 카드에서
디지털 -> DAC -> ADC -> 디지털로 받게 되어 음질이 많이 손실됩니다.
때문에 디지털 인터페이스를 사용하는 것입니다.

일반적으로 디지털 기기간 원본의 손실없이 전달하기 위해
아날로그 신호전송 보다 디지털 전송을 선호하고 있습니다.
참고로 S/PDIF의 케이블간의 저항의 권장치는 75오옴입니다.

HDMI S/PDIF Driver

S5pc100 cpu가 기본 제공



HDMI S/PDIF Driver

```
config SND_S5P_MANG0100
    tristate "Soc Audio support Mango100"
    depends on SND_SAMSUNG_SOC && (MACH_MANG0100)
    select SND_S3C24XX_SOC
    help
        Say Y if you want to add support for SoC audio on the MANG0100.

choice
    prompt "Select MAnGo100 Audio Port Type"
    depends on SND_S5P_MANG0100

config SND_MANG0100_WM8960
    bool "WM8960 Driver"
    select SND_SOC_WM8960
    select SND_S5P_SOC_I2S
    select SND_S3C_I2SV2_SOC

config SND_MANG0100_HDMI_SPDIF
    bool "HDMI SPDIF Driver"
    select SND_S5P_SPDIF
```

sound/soc/s3c24xx/Kconfig 파일에 위의 내용 추가

HDMI S/PDIF Driver

ALSA device list:

#0: smdks5p (HDMI-SPDIF)

커널 로그에서 위와 같이 메시지가 출력

```
static struct snd_soc_dai_link smdks5p_dai[] = {
    {
        .name = "HDMI-SPDIF",
        .stream_name = "HDMI-SPDIF Playback",
        .cpu_dai = &s5p_spdif_dai,
        .codec_dai = &s5p_hdmi_spdif_dai[0],
        .init = smdks5p_spdif_init,
        .ops = &smdks5p_spdif_ops,
    },
};
```

```
static struct snd_soc_card smdks5p = {
    .name = "smdks5p",
    .platform = &s3c24xx_soc_platform,
    .dai_link = smdks5p_dai,
    .num_links = ARRAY_SIZE(smdks5p_dai),
};
```

```
# ls -al
dr-xr-xr-x  4 0          0          0 Apr 22 21:14 .
dr-xr-xr-x 63 0          0          0 Jan  1  1970 ..
dr-xr-xr-x  4 0          0          0 Apr 22 21:14 card0
-r--r--r--  1 0          0          0 Apr 22 21:14 cards
-r--r--r--  1 0          0          0 Apr 22 21:14 devices
dr-xr-xr-x  2 0          0          0 Apr 22 21:14 oss
-r--r--r--  1 0          0          0 Apr 22 21:14 pcm
lrwxrwxrwx  1 0          0          0 Apr 22 21:14 smdks5p -> card0
-r--r--r--  1 0          0          0 Apr 22 21:14 timers
-r--r--r--  1 0          0          0 Apr 22 21:14 version
```

```
# cat cards
0 [smdks5p          ]: HDMI-SPDIF - smdks5p
smdks5p (HDMI-SPDIF)

# pwd
/proc/asound
# cat devices
2:          : timer
3: [ 0- 0]: digital audio playback
4: [ 0- 0]: digital audio capture
5: [ 0]   : control

# cat pcm
00-00: HDMI-SPDIF Playback HDMI-SPDIF Codec-0 : : playback 1 : capture 1
```

```
struct snd_soc_dai s5p_hdmi_spdif_dai[] = {
    {
        .name = "HDMI-SPDIF Codec",
        .id = 0,
        .playback = {
            .stream_name = "Playback",
            .channels_min = 1,
            .channels_max = 2,
            .rates = S5P_HDMI_SPDIF_RATES,
            .formats = S5P_HDMI_SPDIF_FORMATS,
        },
        .capture = {
            .stream_name = "Capture",
            .channels_min = 2,
            .channels_max = 2,
            .rates = S5P_HDMI_SPDIF_RATES,
            .formats = S5P_HDMI_SPDIF_FORMATS,
        },
        .ops = {
            .hw_params = NULL,
            .set_fmt = NULL,
            .set_clkdiv = NULL,
            .set_pll = NULL,
        },
    },
};
```

Battery Driver

```
config BATTERY_MANGO_DUMMY
      tristate "Mango Dummy Battery"
      help
        Mango Dummy Battery Driver
```

Power supply class support

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, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

```
--- Power supply class support
[ ] Power supply debug
< > Generic PDA/phone power driver
< > Battery driver (S3C)
< > APM emulation for class batteries
< > DS2760 battery driver (HP IPAQ & others)
< > BQ27200 battery driver
< > Maxim MAX17040 Fuel Gauge
[*] Mango Dummy Battery
```

drivers/power/Kconfig 에 dummy battery 를 추가

Battery Driver

```
ifeq ($(CONFIG_SYSFS),y)
power_supply-objs += power_supply_sysfs.o
endif

ifeq ($(CONFIG_LEDS_TRIGGERS),y)
power_supply-objs += power_supply_leds.o
endif

ifeq ($(CONFIG_POWER_SUPPLY_DEBUG),y)
EXTRA_CFLAGS += -DDEBUG
endif

obj-$(CONFIG_POWER_SUPPLY) += power_supply.o

obj-$(CONFIG_PDA_POWER) += pda_power.o
obj-$(CONFIG_BATTERY_S3C) += s3c_fake_battery.o
obj-$(CONFIG_APM_POWER) += apm_power.o
obj-$(CONFIG_WM8350_POWER) += wm8350_power.o

obj-$(CONFIG_BATTERY_DS2760) += ds2760_battery.o
obj-$(CONFIG_BATTERY_PMU) += pmu_battery.o
obj-$(CONFIG_BATTERY_OLPC) += olpc_battery.o
obj-$(CONFIG_BATTERY_TOSA) += tosa_battery.o
obj-$(CONFIG_BATTERY_WM97XX) += wm97xx_battery.o
obj-$(CONFIG_BATTERY_BQ27X00) += bq27x00_battery.o
obj-$(CONFIG_BATTERY_DA9030) += da9030_battery.o
obj-$(CONFIG_CHARGER_PCF50633) += pcf50633-charger.o
obj-$(CONFIG_BATTERY_MAX17040) += max17040_battery.o
obj-$(CONFIG_BATTERY_MANGO_DUMMY) += mango_dummy_battery.o
```

drivers/power/Makefile 에 추가
cafe.naver.com/embeddedcrazyboys

Battery Driver

```
static struct platform_driver dummy_battery_device = {
    .probe         = dummy_battery_probe,
    .remove        = dummy_battery_remove,
    .driver = {
        .name = "dummy-battery"
    }
};
```

```
drwxr-xr-x  4 0      0      0 Apr 22 21:11 .
drwxr-xr-x 39 0      0      0 Apr 22 21:11 ..
lrwxrwxrwx  1 0      0      0 Apr 23 01:36 driver -> ../../../../bus/platform/drivers/dummy-battery
-r--r--r--  1 0      0      0 Apr 23 01:36 modalias
drwxr-xr-x  2 0      0      0 Apr 22 21:11 power
drwxr-xr-x  5 0      0      0 Apr 22 21:11 power_supply
lrwxrwxrwx  1 0      0      0 Apr 23 01:36 subsystem -> ../../../../bus/platform
-rw-r--r--  1 0      0      0 Apr 22 21:11 uevent
# cat modalias
platform:dummy-battery
# ls power
wakeup
# ls power_supply/
ac      battery  usb
```

drivers/power/mango_dummy_battery.c 드라이버 소스 생성 추가

Battery Driver

```
#define POWER_SUPPLY_PATH "/sys/class/power_supply"
```

실행 결과

```
# ls /sys/class/power_supply/  
ac      battery  usb
```

Éclair 소스에서 framework/base/services/jni/com_android_server_BatteryService.cpp
에 POWER_SUPPLY_PATH 정의

```
struct dummy_battery_data {  
    struct power_supply  battery;  
    struct power_supply  ac;  
    struct power_supply  usb;  
  
    #if defined(CONFIG_HAS_WAKELOCK)  
        int              locked;  
    #endif  
  
    int                usb_online;  
    int                ac_online;  
} *dummy_data;
```

커널 drivers/power/mango_dummy_battery.c

```
struct Fields {  
    // members  
    jfieldID mAcOnline;  
    jfieldID mUsbOnline;  
    jfieldID mBatteryStatus;  
    jfieldID mBatteryHealth;  
    jfieldID mBatteryPresent;  
    jfieldID mBatteryLevel;  
    jfieldID mBatteryVoltage;  
    jfieldID mBatteryTemperature;  
    jfieldID mBatteryTechnology;  
};
```

Éclair 소스에서 framework/base/services/jni/com_android_server_BatteryService.cpp

Battery Driver

```
int register_android_server_BatteryService(JNIEnv* env)
```

```
{  
    DIR* dir = opendir(POWER_SUPPLY_PATH);  
    while ((entry = readdir(dir))) {  
        const char* name = entry->d_name;  
  
        char buf[20];  
        // Look for "type" file in each subdirectory  
        snprintf(path, sizeof(path), "%s/%s/type", POWER_SUPPLY_PATH, name);  
        int length = readFromFile(path, buf, sizeof(buf));  
        if (length > 0) {  
            if (strcmp(buf, "Mains") == 0) {  
                snprintf(path, sizeof(path), "%s/%s/online", POWER_SUPPLY_PATH, name);  
                if (access(path, R_OK) == 0)  
                    gPaths.acOnlinePath = strdup(path);  
            }  
        }  
    }  
}
```

```
# cat /sys/class/power_supply/Mains/online  
1
```

```
else if (strcmp(buf, "USB") == 0) {  
    snprintf(path, sizeof(path), "%s/%s/online", POWER_SUPPLY_PATH, name);  
    if (access(path, R_OK) == 0)  
        gPaths.usbOnlinePath = strdup(path);  
}
```

```
# cat /sys/class/power_supply/usb/online  
0
```

```
else if (strcmp(buf, "Battery") == 0) {  
    snprintf(path, sizeof(path), "%s/%s/status", POWER_SUPPLY_PATH, name);  
    if (access(path, R_OK) == 0)  
        gPaths.batteryStatusPath = strdup(path);  
}
```

```
# cat /sys/class/power_supply/battery/status  
Charging
```

Battery Driver

```
    snprintf(path, sizeof(path), "%s/%s/health", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0)
    gPaths.batteryHealthPath = strdup(path);
snprintf(path, sizeof(path), "%s/%s/present", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0)
    gPaths.batteryPresentPath = strdup(path);
snprintf(path, sizeof(path), "%s/%s/capacity", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0)
    gPaths.batteryCapacityPath = strdup(path);

snprintf(path, sizeof(path), "%s/%s/voltage_now", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0) {
    gPaths.batteryVoltagePath = strdup(path);
    // voltage_now is in microvolts, not millivolts
    gVoltageDivisor = 1000;
} else {
    snprintf(path, sizeof(path), "%s/%s/batt_vol", POWER_SUPPLY_PATH, name);
    if (access(path, R_OK) == 0)
        gPaths.batteryVoltagePath = strdup(path);
}

snprintf(path, sizeof(path), "%s/%s/temp", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0) {
    gPaths.batteryTemperaturePath = strdup(path);
} else {
    snprintf(path, sizeof(path), "%s/%s/batt_temp", POWER_SUPPLY_PATH, name);
    if (access(path, R_OK) == 0)
        gPaths.batteryTemperaturePath = strdup(path);
}
```

```
/sys/class/power_supply/battery/capacity
/sys/class/power_supply/battery/device/
/sys/class/power_supply/battery/health
/sys/class/power_supply/battery/power/
/sys/class/power_supply/battery/present
/sys/class/power_supply/battery/status
/sys/class/power_supply/battery/subsystem/
/sys/class/power_supply/battery/technology
/sys/class/power_supply/battery/temp
/sys/class/power_supply/battery/type
/sys/class/power_supply/battery/uevent
/sys/class/power_supply/battery/voltage_now
```

Battery Driver

```
data->battery.properties = dummy_battery_props;
data->battery.num_properties = ARRAY_SIZE(dummy_battery_props);
data->battery.get_property = dummy_battery_get_property;
data->battery.name = "Battery";
data->battery.type = POWER_SUPPLY_TYPE_BATTERY;

data->ac.properties = dummy_ac_props;
data->ac.num_properties = ARRAY_SIZE(dummy_ac_props);
data->ac.get_property = dummy_ac_get_property;
data->ac.name = "Mains";
data->ac.type = POWER_SUPPLY_TYPE_MAINS;

data->usb.properties = dummy_usb_props;
data->usb.num_properties = ARRAY_SIZE(dummy_usb_props);
data->usb.get_property = dummy_usb_get_property;
data->usb.name = "USB";
data->usb.type = POWER_SUPPLY_TYPE_USB;

data->ac_online = 1;
data->usb_online = 0;
```

drivers/power/mango_dummy_battery.c에서
dummy_battery_probe 함수에 Name을 지정하고, 속성을 지정

```
# ls /sys/class/power_supply/
Battery Mains USB
```


안드로이드 PowerServiceManager

```
private void setTimeoutLocked(long now, int nextState)
{
    if ( mDoneBooting) {
```



```
private void setTimeoutLocked(long now, int nextState)
{
    if ( false&mDoneBooting) {
```

frameworks/base/services/java/com/android/server/PowerManagerService.java파일에서
안드로이드 부팅 후 Sleep으로 진입을 하지 않음