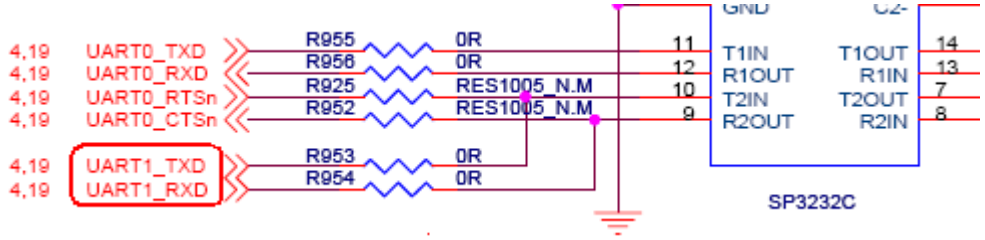
망고100 보드로 놀아보자-15

UART,Keypad,이더넷 드라이버
안드로이드에서 키 처리
http://cafe.naver.com/embeddedcrazyboys

DEBUG PORT 포팅

$(topdir)./config 파일에서
CONFIG_S3C_LOWLEVEL_UART_PORT=1
# CONFIG_SPLIT_ROOT_FILESYSTEM is not set



Kernel configuration에서
"kernel hacking-> S3C UART to use for low-level debug"
에서 1 선택

```
# cat /proc/devices
Character devices:
  1 mem
  2 pty
  3 ttyp
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  7 vcs
 10 misc
 13 input
 14 sound
 29 fb
 81 video4linux
 89 i2c
 90 mtd
116 alsa
128 ptm
136 pts
180 usb
189 usb_device
204 s3c2410_serial
```

./arch/arm/plat-s3c/include/plat/uncompress.h 소스에 포함
#define uart_base S3C_PA_UART + (S3C_UART_OFFSET * CONFIG_S3C_LOWLEVEL_UART_PORT)

.dev_name = "s3c2410_serial",
./drivers/serial/samsung.c 에서 디바이스 이름을 수정 할 수 있습니다.

```
# cat /proc/tty/drivers
/dev/tty             /dev/tty        5       0 system:/dev/tty
/dev/console         /dev/console    5       1 system:console
/dev/ptmx            /dev/ptmx       5       2 system
/dev/vc/0            /dev/vc/0       4       0 system:vtmaster
ttySAC               /dev/s3c2410_serial 204 64-67 serial
serial               /dev/ttyS       4 64-67 serial
pty_slave            /dev/pts      136 0-1048575 pty:slave
pty_master           /dev/ptm      128 0-1048575 pty:master
pty_slave            /dev/ttyp       3 0-255 pty:slave
pty_master           /dev/pty        2 0-255 pty:master
unknown              /dev/tty        4 1-63 console
```

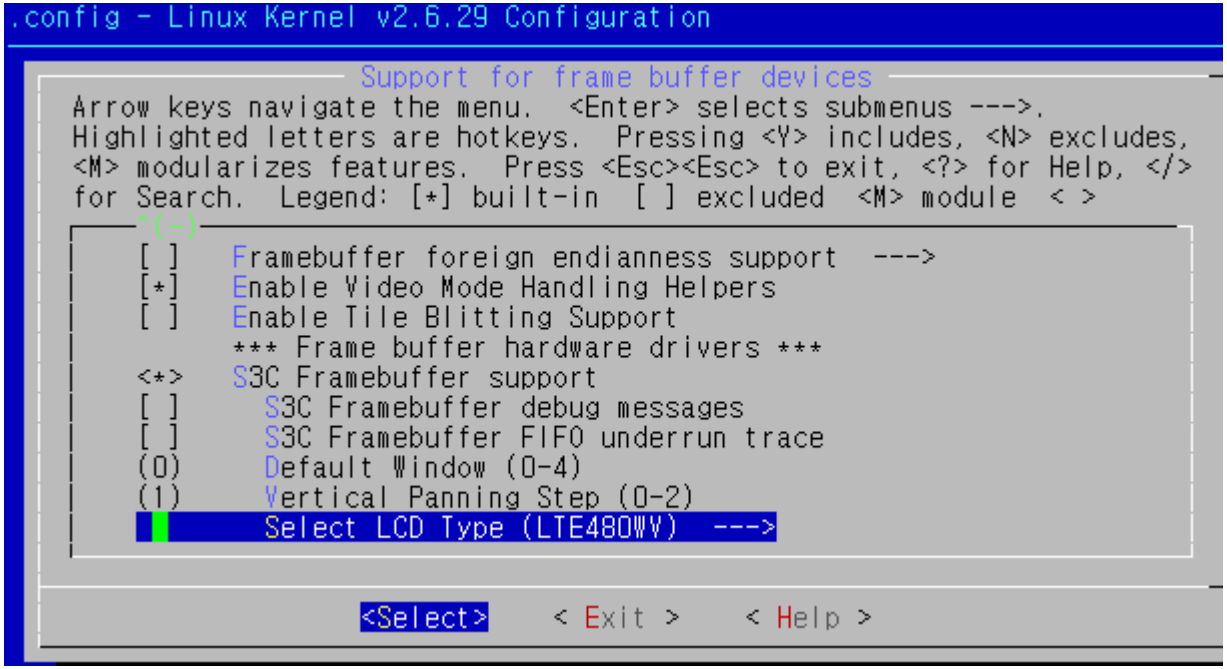
LCD 드라이버 수정-1

#make menuconfig

“Device Drivers->Graphic Support

->Support for frame buffer devices

->S3C Framebuffer support 선택



LCD 드라이버 수정-2

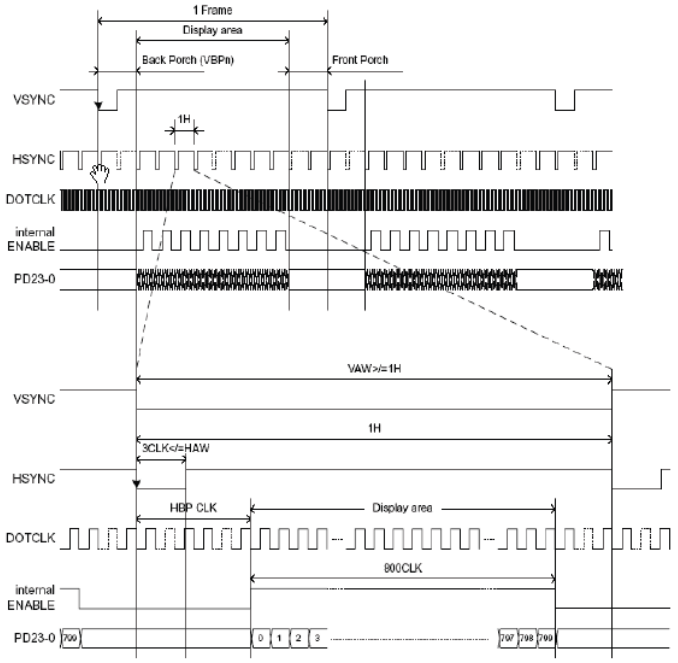

```
void lcd_power_ctrl(s32 value)
{
	int err;

	if (value) {
		if (gpio_is_valid(S5PC1XX_GPH0(2))) {
			err = gpio_request(S5PC1XX_GPH0(2), "GPH0");

			if (err) {
				printk(KERN_ERR "failed to request GPH0 for "
					"lcd reset control\n");
			}
			gpio_direction_output(S5PC1XX_GPH0(2), 1);
		}
	}
	else {
		if (gpio_is_valid(S5PC1XX_GPH0(2))) {
			err = gpio_request(S5PC1XX_GPH0(2), "GPH0");

			if (err) {
				printk(KERN_ERR "failed to request GPH0 for "
					"lcd reset control\n");
			}
			gpio_direction_output(S5PC1XX_GPH0(2), 0);
		}
	}
	gpio_free(S5PC1XX_GPH0(2));
	lcd_power = value;
}
```

./drivers/video/samsung/s3cfb_lte480wv.c
에서 lcd_power_ctrl함수를 수정
LCD 에 전원을 인가한다.

LCD 드라이버 수정-3

```
static struct s3cfb_lcd lte480wv = {
        .width = 800,
        .height = 480,
        .bpp = 24,
        .freq = 60,

        .timing = {
                .h_fp = 8,
                .h_bp = 13,
                .h_sw = 3,
                .v_fp = 5,
                .v_fpe = 1,
                .v_bp = 7,
                .v_bpe = 1,
                .v_sw = 1,
        },

        .polarity = {
                .rise_vclk = 0,
                .inv_hsync = 1,
                .inv_vsync = 1,
                .inv_vden = 0,
        },
};
```

drivers/video/samsung/s3cfb_lte480wv.c
에서 LCD datasheet 타이밍만 맞추면 됨



7-1. Vertical timing

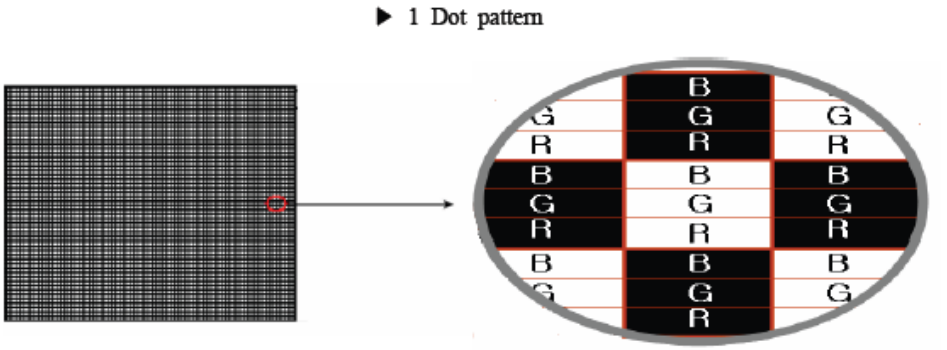
Signal	Symbol	Min.	Typ.	Max.	Unit	Note
Frame Frequency	fFRM	-	60	-	Hz	
Vertical Back porch	VBP	-	8	-	H	*Note
Vertical Front porch	VFP	-	5	-	H	*Note

7-2. Horizontal timing

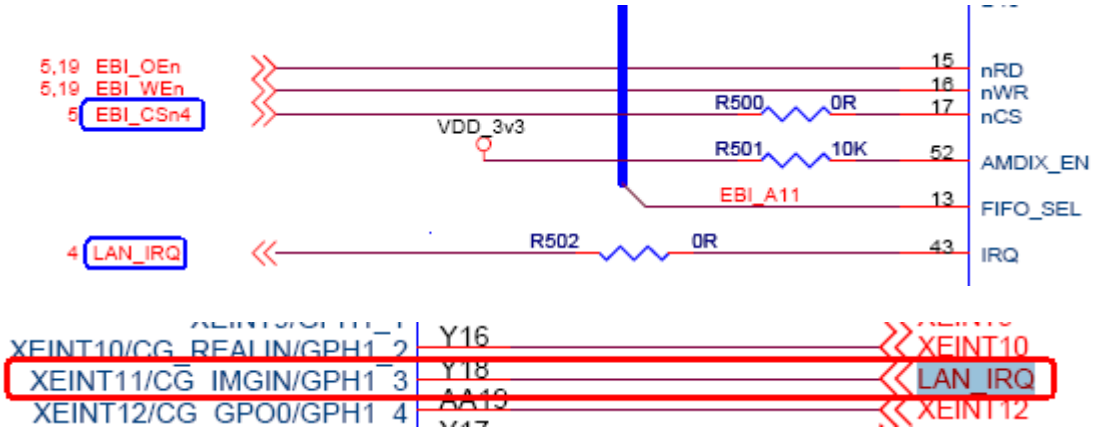
Signal	Symbol	Min.	Typ.	Max.	Unit	Note
Horizontal Back porch	HBP	-	16	-	DOTCLK	*Note
Horizontal Front porch	HFP	-	8	-	DOTCLK	*Note
DOTCLK Frequence	f_{DOTCLK}	-	24.5	-	MHz	@fFRM=60Hz

LCD 드라이버 수정-4

```
static int s3cfb_init_global(void)
{
	fbdev->output = OUTPUT_RGB;
	fbdev->rgb_mode = MODE_BGR_P;

	fbdev->wq_count = 0;
	init_waitqueue_head(&fbdev->wq);
	mutex_init(&fbdev->lock);

	s3cfb_set_output(fbdev);
	s3cfb_set_display_mode(fbdev);
	s3cfb_set_polarity(fbdev);
	s3cfb_set_timing(fbdev);
	s3cfb_set_lcd_size(fbdev);

	return 0;
}
```



이더넷 드라이브 포팅-1



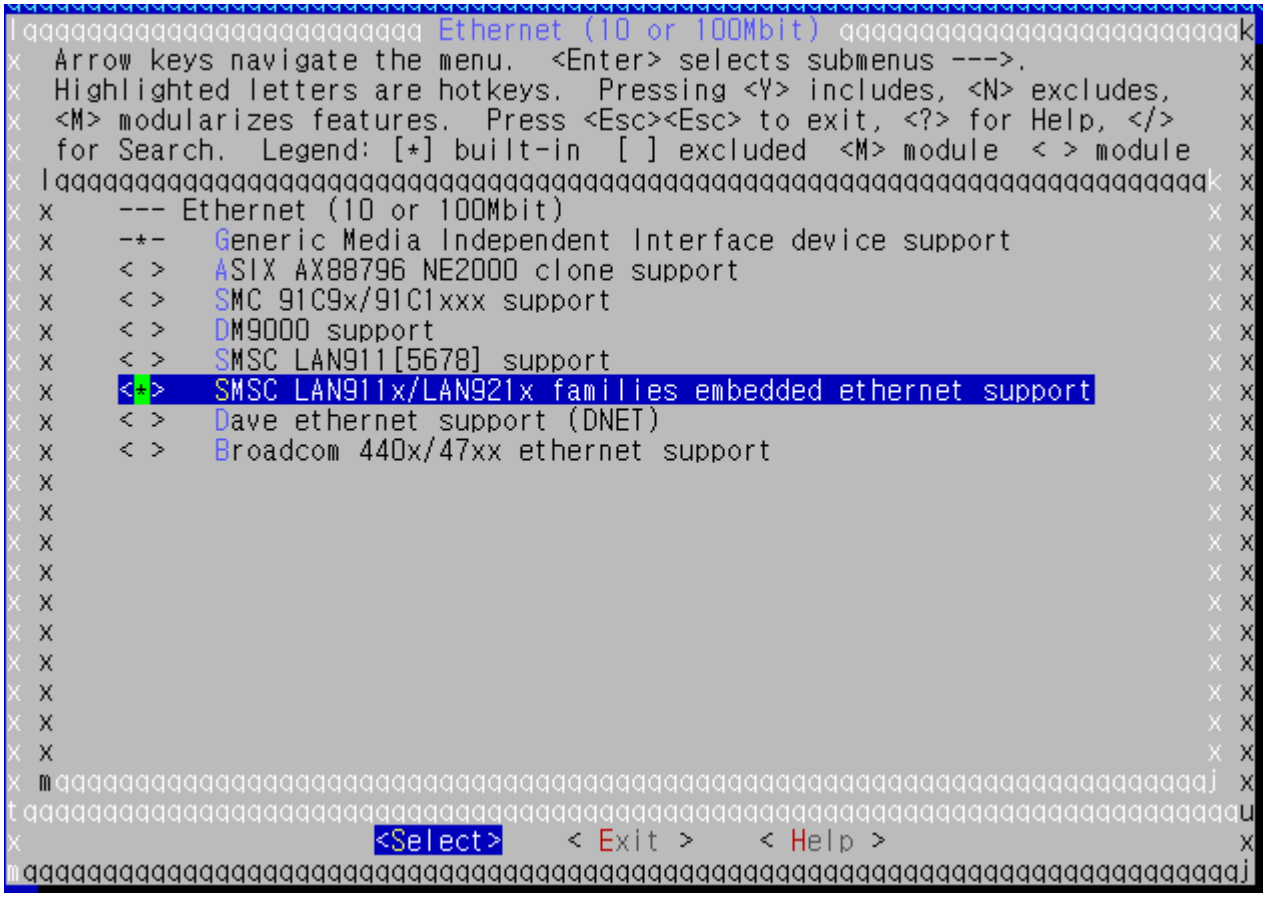

```
static struct resource s3c_smsc911x_resources[] = {
    [0] = {
        .start  = S5PC1XX_PA_SMSC9220,
        .end    = S5PC1XX_PA_SMSC9220 + SZ_1M - 1,
        .flags  = IORESOURCE_MEM,
    },
    [1] = {
        .start = IRQ_EINT11,
        .end   = IRQ_EINT11,
        .flags = IORESOURCE_IRQ,
     },
};
```

arch/arm/plat-s5pc1xx/devs.c 에서 수정

#define S5PC1XX_PA_SMSC9220 (0xA0000000)
/arch/arm/mach-s5pc100/include/mach/map.h

0xA000_0000	0xA800_0000	128MB	SMC Bank 4

이더넷 드라이버 포팅 -2



"Device driver->Network Device support에서 선택
Configuration 파일 drivers/net/Kconfig 위치

이더넷 드라이버 -3

```
static void __init mango100_smsc911x_set(void)
{
        unsigned int tmp;

        tmp = __raw_readl(S5PC1XX_GPK0CON);
        tmp &=~S5PC1XX_GPK0_4_MASK;
        tmp |=(S5PC1XX_GPK0_4_SROM_CSn4);
        __raw_writel(tmp, S5PC1XX_GPK0CON);

        tmp = __raw_readl(S5PC1XX_SROM_BW);
        tmp &= ~(S5PC1XX_SROM_BW_BYTE_ENABLE4_MASK | S5PC1XX_SROM_BW_WAIT_ENABLE4_M
ASK |
                S5PC1XX_SROM_BW_ADDR_MODE4_MASK | S5PC1XX_SROM_BW_DATA_WIDTH4_MASK)
;
        tmp |= S5PC1XX_SROM_BW_DATA_WIDTH4_16BIT;
        tmp |= S5PC1XX_SROM_BW_ADDR_MODE4_BYTE_ADDR;

        __raw_writel(tmp, S5PC1XX_SROM_BW);

        __raw_writel((0x0<<28)|(0x4<<24)|(0xd<<16)|(0x1<<12)|(0x4<<8)|(0x6<<4)|(0x0
<<0), S5PC1XX_SROM_BC4);
}
```

```
static struct platform_device *mango100_devices[] __initdata = {
#ifdef CONFIG_FB_S3C
        &s3c_device_fb,
#endif
        &s3c_device_rtc,

        &s3c_device_i2c0,

        &s3c_device_nand,
#ifdef CONFIG_TOUCHSCREEN_S3C
        &s3c_device_ts,
#endif
        &s3c_device_smsc911x,
        &mango_battery,
```

./arch/arm/mach-s5pc100/mach-mango100.c
디바이스 등록 소스는
./arch/arm/plat-s5pc1xx/devs.c
에 위치

이더넷 디바이스 확인 방법

```
# cat /proc/iomem
20000000-2fffffff : System RAM
  2002f000-20477fff : Kernel text
  20478000-205526eb : Kernel data
a0000000-a00fffff : smsc911x
  a0000000-a00ffffe : smsc911x
```

```
# ls /sys/devices/platform/smsc911x/
driver       mdio_bus     net          subsystem
ffffffff:01  modalias     power        uevent
# ls -al /sys/devices/platform/smsc911x/
drwxr-xr-x    6 0        0               0 Apr 21 19:56 .
drwxr-xr-x   40 0        0               0 Apr 21 19:56 ..
lrwxrwxrwx    1 0        0               0 Apr 22 01:06 driver -> ../../../bus/
platform/drivers/smsc911x
drwxr-xr-x    3 0        0               0 Apr 21 19:56 ffffffff:01
drwxr-xr-x    3 0        0               0 Apr 21 19:56 mdio_bus
-r--r--r--    1 0        0            4096 Apr 22 01:06 modalias
drwxr-xr-x    3 0        0               0 Apr 21 19:56 net
drwxr-xr-x    2 0        0               0 Apr 21 19:56 power
lrwxrwxrwx    1 0        0               0 Apr 22 01:06 subsystem -> ../../../b
us/platform
-rw-r--r--    1 0        0            4096 Apr 21 19:56 uevent
```

```
# cat /proc/net/dev
Inter-|   Receive                                                |  Transmit
 face |bytes    packets errs drop fifo frame compressed multicast|bytes    packe
ts errs drop fifo colls carrier compressed
    lo:       0       0    0    0    0     0          0         0        0
 0    0    0    0     0       0          0
  eth0:  763466   11632    0    0    0     0          0     11623     2052
16    0    0    0     0       0          0
```

이더넷 드라이버 확인 방법

```
# ls -al /sys/devices/platform/smsc911x/driver/
drwxr-xr-x    2 0        0               0 Apr 22 01:06 .
drwxr-xr-x   38 0        0               0 Apr 22 01:06 ..
--w-------    1 0        0            4096 Apr 22 03:17 bind
lrwxrwxrwx    1 0        0               0 Apr 22 03:17 smsc911x -> ../../../..
/devices/platform/smsc911x
--w-------    1 0        0            4096 Apr 22 03:17 uevent
--w-------    1 0        0            4096 Apr 22 03:17 unbind
```

이더넷 드라이버 irq 등록확인

```
# cat /proc/interrupts
           CPU0
 20:        694    s3c-uart   s5pc100-uart
 22:       1631    s3c-uart   s5pc100-uart
 37:          0   s3c_vic_eint   s5p-tvout
 43:      23801   s3c_vic_eint   eth0
 51:          0          VIC  I2S PCM Stereo out
```

S5pc100 user manual에는 11번째에 EINT11이다, 근데 등록은 32를 더한 값인 43번째 등록이 되어 있다.
이유는 커널이 32번까지 **spurious** 인터럽트로 등록하여 사용한다.

11	**EINT11**	EXT_INT[11]

#define IRQ_EINT11 S5PC1XX_IRQ_VIC0(11)
./arch/arm/plat-s5pc1xx/include/plat/irqs.h

Keypad driver

```
Keyboards
  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
    --- Keyboards
    < >   AT keyboard
    < >   Sun Type 4 and Type 5 keyboard
    < >   DECstation/VAXstation LK201/LK401 keyboard
    < >   XT keyboard
    < >   Newton keyboard
    < >   Stowaway keyboard
    < >   GPIO Buttons
    <*>   Mango-100 board keypad support
    [*]     Linux Output Event
    < >   S3C keypad support
          Select keypad port
                    <Select>    < Exit >    < Help >
```

./driver/input/keyboard/Kconfig 에 추가
.config 파일에
CONFIG_KEYPAD_MANGO100_LINUX_EVENT=y

```
config KEYPAD_MANGO100
        tristate "Mango-100 board keypad support"
        depends on (CPU_S3C6400 || CPU_S3C6410 ||CPU_S5PC100)
        default n
        help
          Say Y here if you want to use the MangoC100 keypad.
          To compile this driver as a module, choose M here: the
          module will be called mangoc100-keypad.

config KEYPAD_MANGO100_LINUX_EVENT
        bool "Linux Output Event"
        depends on KEYPAD_MANGO100
        default n
        help
          Say Y here if you want to use the Mango100 keypad for Linux Event.
          To compile this driver as a module, choose M here: the
          module will be called mangoc100-keypad.
```

Keypad driver

```
[icanjji@localhost kernel]$ ls  drivers/input/keyboard/mango*
drivers/input/keyboard/mango100_keycode.h
drivers/input/keyboard/mango100_keypad.c
drivers/input/keyboard/mango100_keypad.h
drivers/input/keyboard/mango100_keypad.o
drivers/input/keyboard/mango100_keypad_sysfs.c
drivers/input/keyboard/mango100_keypad_sysfs.h
drivers/input/keyboard/mango100_keypad_sysfs.o
```

./driver/input/keyboard/ 파일을 추가한다.

```
obj-$(CONFIG_KEYPAD_S3C)                += s3c-keypad.o
obj-$(CONFIG_KEYPAD_MANGO100)    += mango100_keypad.o mango100_keypad_sysfs.o
```

./driver/input/keyboard/Makefile 에 추가

Keypad driver

```
static int __devinit    mango100_keypad_probe(struct platform_device *pdev)
{
        int      rc;
        int key, code;

        // struct init
        memset(&mango100_keypad, 0x00, sizeof(mango100_keypad_t));

        // create sys_fs
        if((rc = mango100_keypad_sysfs_create(pdev)))   {
                printk("%s : sysfs_create_group fail!!\n", __FUNCTION__);
                return  rc;
        }
```

Driver/input/keyboard/mango100-keypad.c
등록확인은 아래와 같이 확인 할 수 있다.

```
# ls -al /sys/devices/platform/mango100-keypad.0
drwxr-xr-x    3 0        0               0 Apr 21 19:56 .
drwxr-xr-x   40 0        0               0 Apr 21 19:56 ..
lrwxrwxrwx    1 0        0               0 Apr 22 02:25 driver -> ../../../bus/
platform/drivers/mango100-keypad
-r--r--r--    1 0        0            4096 Apr 22 02:25 hdmi_connect_state
-r--r--r--    1 0        0            4096 Apr 22 02:25 modalias
drwxr-xr-x    2 0        0               0 Apr 21 19:56 power
lrwxrwxrwx    1 0        0               0 Apr 22 02:25 subsystem -> ../../../b
us/platform
-rw-r--r--    1 0        0            4096 Apr 21 19:56 uevent
```

Keypad driver

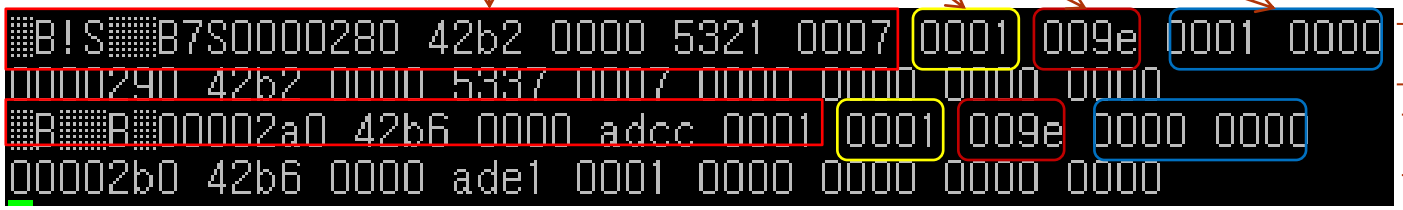
```
struct input_event {
        struct timeval time;
        __u16 type;
        __u16 code;
        __s32 value;
};
```

./include/linux/input.h에 정의

```
#define EV_KEY                  0x01
#define KEY_BACK                158     /* AC Back */
```

키를 누르면 value값은 1을 가진다

hexdump /dev/input/event0



Keypad driver

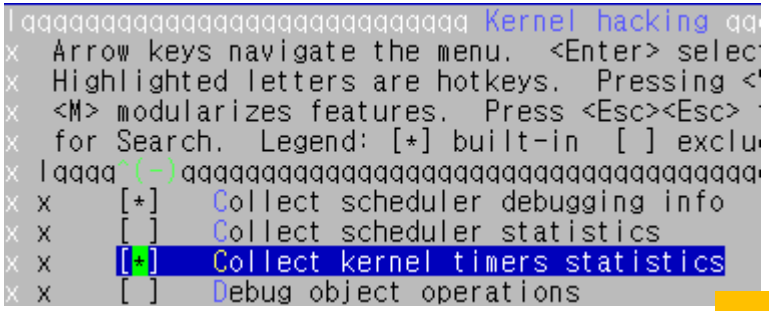
```
mango100_keypad.driver->name    = DEVICE_NAME;
        mango100_keypad.driver->phys    = "mango100-keypad/input0";
        mango100_keypad.driver->open    = mango100_keypad_open;
        mango100_keypad.driver->close   = mango100_keypad_close;

        mango100_keypad.driver->id.bustype      = BUS_HOST;
        mango100_keypad.driver->id.vendor       = 0x16B4;
        mango100_keypad.driver->id.product      = 0x0701;
        mango100_keypad.driver->id.version      = 0x0001;
```

```
#define BUS_HOST            0x19
./include/linux/input.h
```

./drivers/input/keyboard/mango100_keypad.c 소스에 mango100_keypad_prob함수에 정의

```
# cat /proc/bus/input/devices
I: Bus=0019 Vendor=16b4 Product=0701 Version=0001
N: Name="mango100-keypad"
P: Phys=mango100-keypad/input0
S: Sysfs=/class/input/input0
U: Uniq=
H: Handlers=kbd event0
B: EV=3
B: KEY=40000800 0 0 0 0
```

Keypad driver



"Kernel hacking" 설정
CONFIG_TIMER_STATS=y 로 설정

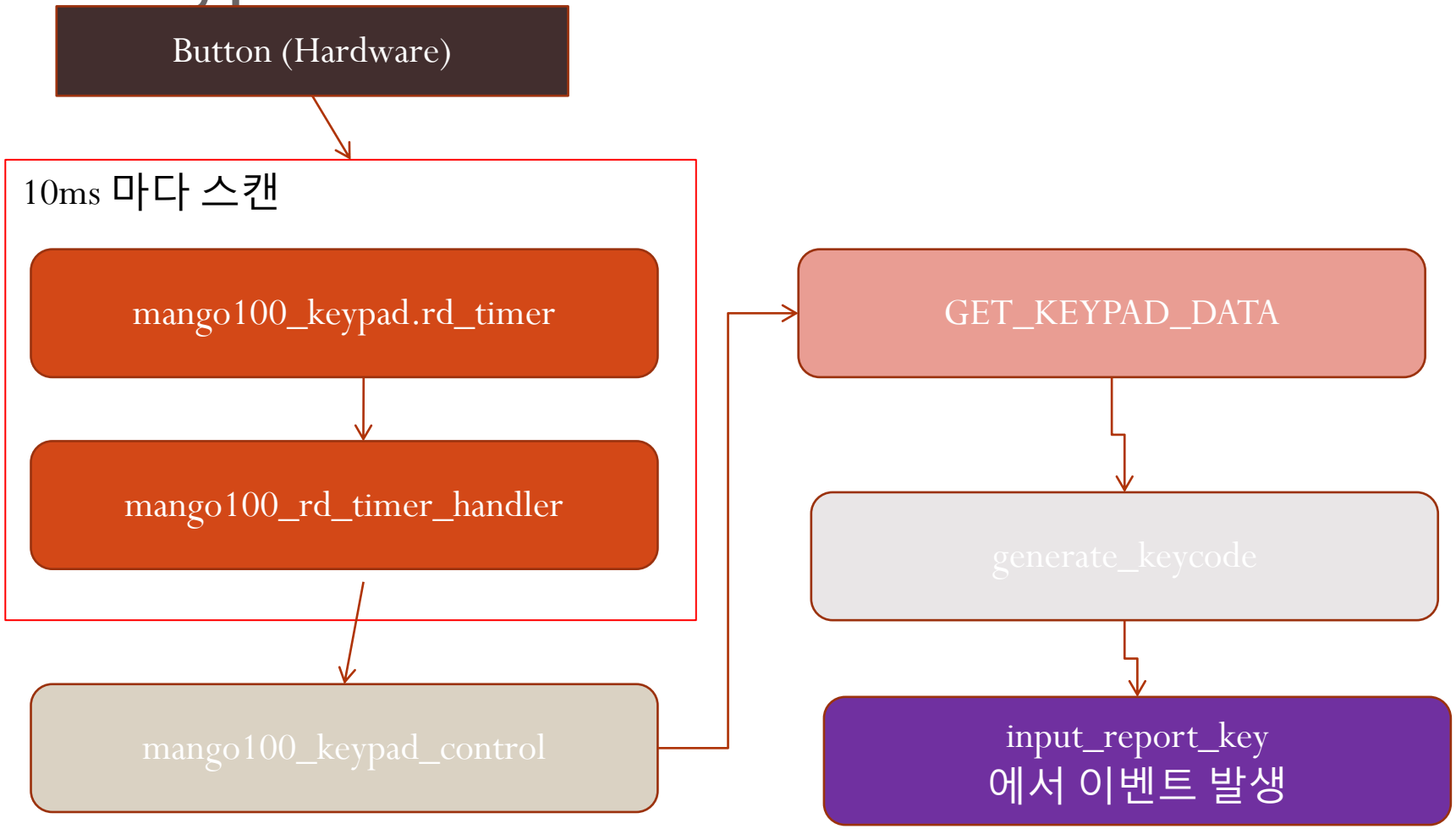
To activate the collection of stats

```
# echo 1 > /proc/timer_stats
# cat /proc/timer_stats
Timer Stats Version: v0.2
Sample period: 2.911 s
 291,     1 swapper        mango100_keypad_config (mango100_rd_timer_handler)
   2,     1 swapper        schedule_delayed_work_on (delayed_work_timer_fn)
   8,  1867 er.ServerThread  futex_wait (hrtimer_wakeup)
   3,     1 swapper        phy_start_machine (phy_timer)
   3,  1832 zygote          queue_delayed_work (delayed_work_timer_fn)
   1,     0 swapper        page_writeback_init (wb_timer_fn)
308 total events, 105.805 events/sec
```

 # echo 0 > /proc/timer_stats

To stop collecting stats

Keypad driver flow



안드로이드 Key event 처리

Key button Map 정의는 vendor\sec\mango100\mango100-keypad.kl
frameworks/base/libs/EventHub.cpp에 scan_dir,open_device 정의

KeyInputQueue(1853): InputDeviceReader.run()

```
static const char *device_path = "/dev/input";
bool EventHub::openPlatformInput(void)
{
..
   res = scan_dir(device_path);
..
}
```

```
int EventHub::scan_dir(const char *dirname)
{
while((de = readdir(dir))) {
      strcpy(filename, de->d_name);
        open_device(devname);
    }
}
```

int EventHub::open_device(const char *deviceName)

안드로이드 Key event 처리

export ANDROID_ROOT /system
/init.rc에 정의

```
int EventHub::open_device(const char *deviceName)
{
…
const char* root = getenv("ANDROID_ROOT");
        property_get("persist.sys.keylayout", keylayout, "qwerty");
        snprintf(keylayoutFilename, sizeof(keylayoutFilename),
                "%s/usrkeylayout/%s.kl", root, keylayout);
 strcpy(devname, keylayout);
        bool defaultKeymap = access(keylayoutFilename, R_OK);
        if (defaultKeymap) {
            snprintf(keylayoutFilename, sizeof(keylayoutFilename),
                    "%s/usr/keylayout/%s.kl", root, "qwerty");
            strcpy(devname, "qwerty");
        }
        LOGI("2:devname = %s, keylayout =%s, keylayoutFilename = %s",
                devname, keylayout, keylayoutFilename);
        device->layoutMap->load(keylayoutFilename);

….

}
```

I/EventHub(1853): 2:devname = qwerty,
keylayout =qwerty,
keylayoutFilename = /system/usr/keylayout/qwert
-#logcat 명령으로 디버깅 메시지 출력

frameworks/base/libs/ui/EventHub.cpp파일에
EventHub::open_device함수에 정의

안드로이드 Key event 처리

```
int EventHub::open_device(const char *deviceName)
{
…
 LOGI("New keyboard: publicID=%d device->id=0x%x devname='%s' propName='%s' keylayout='%s'\n",
            publicID, device->id, name, propName, keylayoutFilename);
   }

   LOGI("New device: path=%s name=%s id=0x%x (of 0x%x) index=%d fd=%d classes=0x%x\n",
       deviceName, name, device->id, mNumDevicesById, mFDCount, fd, device->classes);

   LOGV("Adding device %s %p at %d, id = %d, classes = 0x%x\n",
       deviceName, device, mFDCount, devid, device->classes);
}
```

frameworks/base/libs/ui/EventHub.cpp 에 open_device 정의

#logcat 명령으로 출력 결과

```
I/EventHub( 1853): New keyboard: publicID=65537 device->id=0x10001
devname='mango100-keypad' propName='hw.keyboards.65537.devname' keylayout='/system/usr/keylayout/qwerty.kl'
I/EventHub( 1853): New device: path=/dev/input/event0 name=mango100-keypad id=0x10001 (of 0x2) index=2 fd=58 classes=0x1
```

안드로이드 Key event 처리

```
Thread mThread = new Thread("InputDeviceReader") {
    public void run() {
        if (DEBUG) Log.v(TAG, "InputDeviceReader.run()");
        android.os.Process.setThreadPriority(
                android.os.Process.THREAD_PRIORITY_URGENT_DISPLAY);

        RawInputEvent ev = new RawInputEvent();
        while (true) {
            try {
                InputDevice di;

                // block, doesn't release the monitor
                readEvent(ev);

                boolean send = false;
                boolean configChanged = false;

                if (true) {
                    Log.i(TAG, "Input event: dev=0x"
                            + Integer.toHexString(ev.deviceId)
                            + " type=0x" + Integer.toHexString(ev.type)
                            + " scancode=" + ev.scancode
                            + " keycode=" + ev.keycode
                            + " value=" + ev.value);
                }
```

Key Event 처리 Thread

#logcat 메시지
I/KeyInputQueue(1853): Input event: dev=0x10001 type=0x1 scancode=158 keycode=4 value=1

Mango100-keypad.kl과 scancode값 일치
BACK 키 수행이 됨
./frameworks/base/services/java/com/android/server/KeyInputQueue.java 파일에 정의

안드로이드 Key event 처리

frameworks/base/services/java/com/android/server/KeyInputQueue.java

```
Thread mThread = new Thread("InputDeviceReader") {
.....
while (true) {
        try {
            InputDevice di;
            // block, doesn't release the monitor
            readEvent(ev);
.....
```

frameworks/base/services/jni/com_android_server_KeyInputQueue.cpp

```
static JNINativeMethod gInputMethods[] = {
    /* name, signature, funcPtr */
    { "readEvent",          "(Landroid/view/RawInputEvent;)Z",
            (void*) android_server_KeyInputQueue_readEvent },
```

```
static jboolean
android_server_KeyInputQueue_readEvent(JNIEnv* env, jobject clazz,
                                       jobject event)
{
    gLock.lock();
    sp<EventHub> hub = gHub;
    if (hub == NULL) {
        hub = new EventHub;
        gHub = hub;
    }
    gLock.unlock();

    int32_t deviceId;
    int32_t type;
    int32_t scancode, keycode;
    uint32_t flags;
    int32_t value;
    nsecs_t when;
    bool res = hub->getEvent(&deviceId, &type, &scancode, &keycode,
            &flags, &value, &when);

    env->SetIntField(event, gInputOffsets.mDeviceId, (jint)deviceId);
    env->SetIntField(event, gInputOffsets.mType, (jint)type);
    env->SetIntField(event, gInputOffsets.mScancode, (jint)scancode);
    env->SetIntField(event, gInputOffsets.mKeycode, (jint)keycode);
    env->SetIntField(event, gInputOffsets.mFlags, (jint)flags);
    env->SetIntField(event, gInputOffsets.mValue, value);
    env->SetLongField(event, gInputOffsets.mWhen,
                      (jlong)(nanoseconds_to_milliseconds(when)));

    return res;
}
```

안드로이드 Key event 처리

```
public final boolean dispatch(Callback receiver, DispatcherState state,
        Object target) {
    switch (mAction) {
....
case ACTION_UP:
        if (DEBUG) Log.v(TAG, "Key up to " + target + " in " + state
                + ": " + this);
        if (state != null) {
            state.handleUpEvent(this);
        }
        return receiver.onKeyUp(mKeyCode, this);
```

```
public void handleUpEvent(KeyEvent event) {
        final int keyCode = event.getKeyCode();
        if (DEBUG) Log.v(TAG, "Handle key up " + event + ": " + this);
```

./frameworks/base/core/java/android/view/KeyEvent.java 에서 dispatch를 한다.

#logcat 수행 결과

```
KeyEvent( 1910): Key up to android.inputmethodservice.InputMethodService$InputMethodSessionImpl@43da0b40 in
android.view.KeyEvent$DispatcherState@43d92cf8: KeyEvent{action=1 code=4 repeat=0 meta=0 scancode=158 mFlags=8}
V/KeyEvent( 1910): Handle key up KeyEvent{action=1 code=4 repeat=0 meta=0 scancode=158 mFlags=8}: a
ndroid.view.KeyEvent$DispatcherState@43d92cf8
D/MountListener( 1853): handleEvent ums_connected
```

안드로이드 Key Event 처리

Window Manager
distpatchKey(KeyEvent..)
frameworks/base/services/java/com/android/server/WindowManagerService.java

Thread mThread = new Thread("InputDeviceReader")
readEvent(ev);
di = newInputDevice(ev.deviceId);

frameworks/base/services/java/com/android/server/KeyInputQueue.java

```
static JNINativeMethod gInputMethods[] = {
    /* name, signature, funcPtr */
    { "readEvent",       "(Landroid/view/RawInputEvent;)Z",
            (void*) android_server_KeyInputQueue_readEvent },
```

frameworks/base/services/jni/com_android_server_KeyInputQueue.cpp

static const char *device_path = "/dev/input";
bool EventHub::getEvent(int32_t* outDeviceId, int32_t* outType,
bool EventHub::openPlatformInput(void)
frameworks/base/libs/ui/EventHub.cpp