

2. 작업내용

IIO로 안해도 될 듯 하네요... 인터럽트가 있어서 값이 버퍼에 쌓이는게 아니라면 필요 없을 듯..
센서에 인터럽트가 없기에 그냥 읽을 수 있도록 하면 될 듯 합니다.
HWMON으로.....

2.1 HTU21 센서 작업

```
$ cd drivers/hwmon/
```

```
$ vi Kconfig
```

```
config SENSORS_GPIO_FAN
```

```
    tristate "GPIO fan"
```

```
    depends on GENERIC_GPIO
```

```
    help
```

```
        If you say yes here you get support for fans connected to GPIO lines.
```

```
        This driver can also be built as a module.  If so, the module  
        will be called gpio-fan.
```

```
config SENSORS_HTU21
```

```
    tristate "Measurement Specialties HTU21D humidity/temperature sensors"
```

```
    depends on I2C
```

```
    help
```

```
        If you say yes here you get support for the Measurement Specialties  
        HTU21D humidity and temperature sensors.
```

```
        This driver can also be built as a module.  If so, the module  
        will be called htu21.
```

```
config SENSORS_CORETEMP
```

```
    tristate "Intel Core/Core2/Atom temperature sensor"
```

```
    depends on X86 && PCI && EXPERIMENTAL
```

```
    help
```

```
        If you say yes here you get support for the temperature  
        sensor inside your CPU. Most of the family 6 CPUs  
        are supported. Check Documentation/hwmon/coretemp for details.
```

```
$ vi Makefile
```

```
obj-$(CONFIG_SENSORS_GPIO_FAN) += gpio-fan.o  
obj-$(CONFIG_SENSORS_HTU21) += htu21.o  
obj-$(CONFIG_SENSORS_ULTRA45) += ultra45_env.o
```

아래 소스를 kernel 소스의 driver/hwmon에 복사해서 사용...

<http://lxr.free-electrons.com/source/drivers/hwmon/htu21.c?v=3.12#L1>

소스에서 아래 부분 수정

```
$ vi htu21.c
```

```
static struct i2c_driver htu21_driver = {  
    .class = I2C_CLASS_HWMON,  
    .driver = {  
        .name = "htu21",  
    },  
    .probe = htu21_probe,  
    .remove = htu21_remove,  
    .id_table = htu21_id,  
};  
  
static int __init htu21_init(void)  
{  
    return i2c_add_driver(&htu21_driver);  
}  
module_init(htu21_init);  
  
static void __exit htu21_exit(void)  
{  
    i2c_del_driver(&htu21_driver);  
}  
module_exit(htu21_exit);
```

```
$ vi htu21.c
```

```
static int htu21_update_measurements(struct i2c_client *client)  
{  
    int ret = 0;
```

```

struct htu21 *htu21 = i2c_get_clientdata(client);

mutex_lock(&htu21->lock);

if (time_after(jiffies, htu21->last_update + HZ / 2) ||
    !htu21->valid) {
    ret = swab16(i2c_smbus_read_word_data(client,
                                          HTU21_T_MEASUREMENT_HM));

    // printk("Read Temp = %d\n",ret);
    if (ret < 0)
        goto out;
    htu21->temperature = htu21_temp_ticks_to_millicelsius(ret);
    ret = swab16(i2c_smbus_read_word_data(client,
                                          HTU21_RH_MEASUREMENT_HM));

    // printk("Read RH = %d\n",ret);
    if (ret < 0)
        goto out;
    htu21->humidity = htu21_rh_ticks_to_per_cent_mille(ret);
    htu21->last_update = jiffies;
    htu21->valid = true;
}
out:
mutex_unlock(&htu21->lock);

return ret >= 0 ? 0 : ret;
}

```

값이 정확하지 않은 것 같다. 소스 보면서 확인을 해봅시다.

```
cat /sys/class/hwmon/hwmon0/device/humidity1_input
```

```
cat /sys/class/hwmon/hwmon0/device/temp1_input
```

```
# cat /sys/class/hwmon/hwmon0/device/humidity1_input
```

```
59361
# cat /sys/class/hwmon/hwmon0/device/humidity1_input
58186
# cat /sys/class/hwmon/hwmon0/device/temp1_input
26380
# cat /sys/class/hwmon/hwmon0/device/temp1_input
26359
# cat /sys/class/hwmon/hwmon0/device/temp1_input
26359
# cat /sys/class/hwmon/hwmon0/device/temp1_input
26348
```

2.2 BH1721 센서 작업

아래 소스 사용 하시오. 헤더파일도 복사해야 합니다.

<https://github.com/barakinflorida/Vibrant-open/blob/master/drivers/hwmon/bh1721.c>

<https://github.com/barakinflorida/Vibrant-open/blob/master/include/linux/bh1721.h>

```
config SENSORS_HTU21
    tristate "Measurement Specialties HTU21D humidity/temperature sensors"
    depends on I2C
    help
        If you say yes here you get support for the Measurement Specialties
        HTU21D humidity and temperature sensors.

        This driver can also be built as a module.  If so, the module
        will be called htu21.

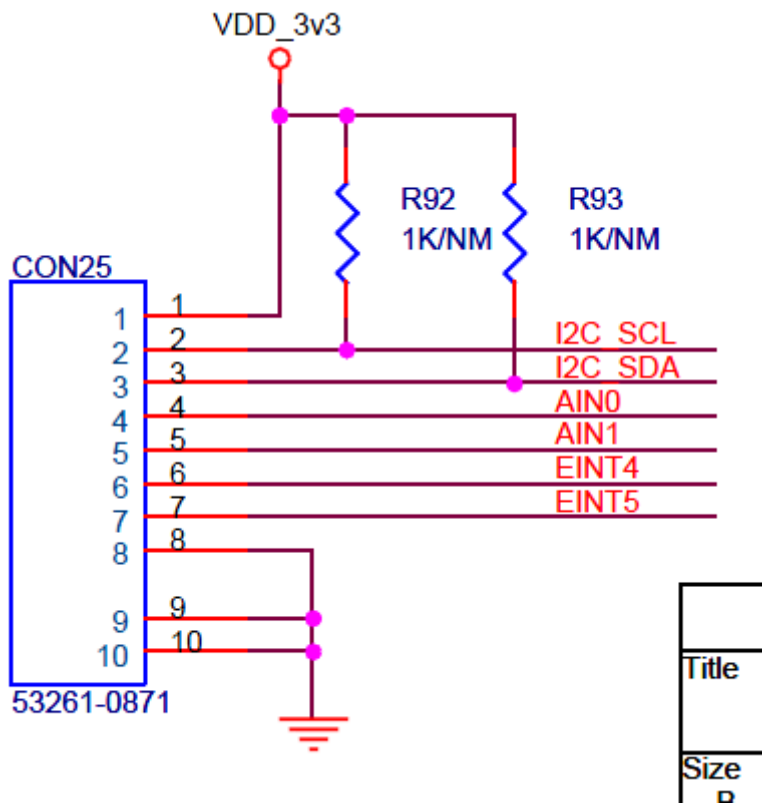
config SENSORS_BH1721
    depends on I2C
    tristate "BH1721 driver"
    default n
    help
```

Say Y here if you use BH1721.

This option enables light sensor using
ROHM BH1721 device driver.

Say N here if you do not use BH1721.

```
obj-$(CONFIG_SENSORS_GL520SM) += gl520sm.o
obj-$(CONFIG_SENSORS_GPIO_FAN) += gpio-fan.o
obj-$(CONFIG_SENSORS_HTU21) += htu21.o
obj-$(CONFIG_SENSORS_BH1721) += bh1721.o
obj-$(CONFIG_SENSORS_ULTRA45) += ultra45_env.o
```



Platform.... Probe시에 6번 핀을 High로 되도록 GPIO설정을 하고 진행 합니다.

bh1721_platform_data

dvi_gpio...

둘 중 하나, slave address 가 맞질 않거나, dvi 가 high가 되어야지만 동작하는 그런..?

```
include/linux/bh1721.h
```

```
#ifndef _BH1721_H_
```

```
#define _BH1721_H_
```

```
struct bh1721_platform_data {
```

```
    void (*reset)(void);
```

```
    int dvi_gpio;
```

```
};
```

```
#endif
```

```
arch/arm/mach-s3c2416/mach-mango2450.c
```

```
static struct bh1721_platform_data bh1721_pdata = {
```

```
    .dvi_gpio = S3C2410_GPF(4),
```

```
};
```

```
static struct i2c_board_info mango2450_i2c0_board_info[] __initdata = {
```

```
{
```

```
    I2C_BOARD_INFO("wm8960", 0x1a),
```

```
},
```

```
{
```

```
    I2C_BOARD_INFO("htu21", 0x40),
```

```
},
```

```
{
```

```
    I2C_BOARD_INFO("bh1721", 0x23),
```

```
    .platform_data = &bh1721_pdata,
```

```
},
```

```
};
```

```
drivers/hwmon/bh1721.c
```

```
static int bh1721_probe(struct i2c_client *client,
                        const struct i2c_device_id *id)
{
    struct bh1721_platform_data *pdata;
    int ret;
    printk("[%s] bh1721 started...",__func__);
    chip = kzalloc(sizeof(struct bh1721_chip), GFP_KERNEL);
    if (!chip)
        return -ENOMEM;

    pdata = client->dev.platform_data;

    #if 0 //by crazyboy 20150824 by treego
        chip->reset = pdata->reset;
    #endif

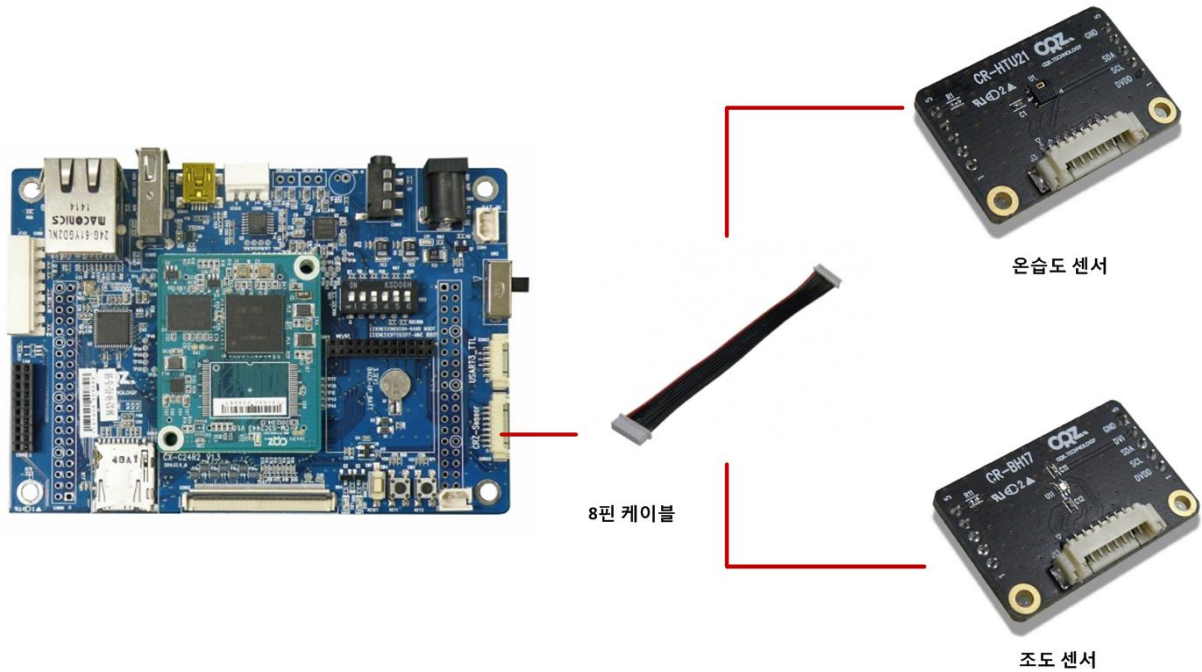
    gpio_request(pdata->dvi_gpio,"bh1721 dvi gpio");
    gpio_direction_output(pdata->dvi_gpio,1);
}
```

```
# cat /sys/class/hwmon/hwmon1/device/illuminance
296
# cat /sys/class/hwmon/hwmon1/device/illuminance
9
# cat /sys/class/hwmon/hwmon1/device/illuminance
9
# cat /sys/class/hwmon/hwmon1/device/illuminance
9
# cat /sys/class/hwmon/hwmon1/device/illuminance
297
```

3. 카페올리는 글 작성

1. 연결방법

아래와 같이 CRZ-Sensor 커넥터에 온습도센서 또는 조도 센서를 연결 합니다.



2. 테스트 방법

소스 다운로드:

http://crztech.iptime.org:8080/Release/mango24R2_S3C2450/linux/m2450_kernel3.0.22_mrvt18787_Qt_150825/

2.1. 온습도(HTU21) 센서 테스트방법

온습도 센서를 연결 후, 부팅 한 이후, 아래 명령으로 센서 값을 확인 합니다.

- 온도 센서 값 확인 (26380 -> 26.3도)

```
# cat /sys/class/hwmon/hwmon0/device/temp1_input
```

```
26380
```

- 습도 센서 값 확인 (58186 -> 58.3%)

```
# cat /sys/class/hwmon/hwmon0/device/humidity1_input
```

```
58186
```

온도는 센서에 손으로 잡으면, 온도가 올라가는 것을 확인 할 수 있고, 습도는 센서에 입김을 불어서 습도가 올라가는 것을 확인 할 수 있습니다.

2.2. 조도(BH1721) 센서 테스트방법

조도 센서를 연결 후, 부팅 한 이후, 아래 명령으로 센서 값을 확인 합니다.

- 평상시 값

```
# cat /sys/class/hwmon/hwmon1/device/illuminance
```

```
296
```

- 손으로 센서 막고 확인 값

```
# cat /sys/class/hwmon/hwmon1/device/illuminance
```

```
9
```

손으로 센서를 막으면 빛이 차단되기 때문에 값이 변하는 것을 확인 할 수 있습니다.