

1. 02.led

2024/7/18 Table of Contents

02.led

目的

構成データ

LED制御

デバイスドライバ

例題 ledctl

課題 負荷表示器 Idmeter

1.1. 目的

組込みアプリケーション開発 02.led

1.2. 構成データ

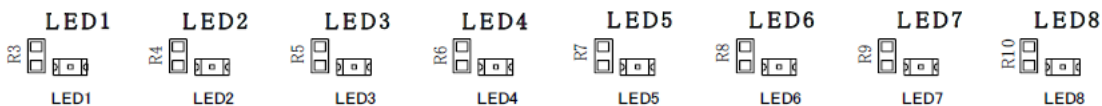
1.2.1. /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Apllication_debug/text/practiceディレクトリ

▼ .../share/ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice/ の構成

```
1 user@1204PC-Z490M:/mnt/v/VirtualBoxWork/share/ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice$ tr
2 ./
3 | 02.led/
4 | | drivers/
5 | | | leds/
6 | | | | leds.c*      <----- ドライバソース
7 | | | | Makefile*    <----- ドライバ用Makefile
8 | | | ldmeteter.c*    <----- 例題 デバイス制御用ソース
9 | | | ledctl.c*       <----- 課題 デバイス制御用ソース
10 | | | Makefile*      <----- デバイス制御用Makefile
11 |
12 |
```

1.3. LED制御

1.3.1. デバイス仕様



デバイスファイル: "/sys/class/leds/led_ext/brightness"

b7	b6	b5	b4	b3	b2	b1	b0
led8	led7	led6	led5	led4	led3	led2	led1

デバイスファイルに 1~255 の値を書き込むことで点灯、0は消灯

1.4. デバイスドライバ

1.4.1. ソース

leds.c

▼ 02.led/drivers/leds.c

```

1  #include <linux/module.h>
2  #include <linux/kernel.h>
3  #include <linux/init.h>
4  #include <linux/platform_device.h>
5  #include <linux/io.h>
6  #include <linux/leds.h>
7  #include <linux/err.h>
8
9  #include <asm/armadilloX1-ext-cpld.h>
10
11 // デバイス書き込み関数(leds_set)
12 static void leds_set(struct led_classdev *led_cdev,
13                     enum led_brightness brightness)
14 {
15     // LEDを光らせます。(cpld_write)
16     cpld_write(CPLD_WRITE_LED, brightness);
17 }
18
19 // LEDクラスデバイス構造体
20 static struct led_classdev leds = {
21     .name = "led_ext",
22     .brightness_set = leds_set,
23     .max_brightness = 255,
24 };
25
26 // probe関数(leds_probe)
27 static int leds_probe(struct platform_device *pdev)
28 {
29     // LED点灯状態レジスタを初期化します。(cpld_write)
30     cpld_write(CPLD_WRITE_LED, 0x00);
31
32     // LEDクラスデバイスを登録します。(led_classdev_register)
33     // ->/sys/class/leds/配下にファイルを作成し、/sys/class/leds/led_ext/brightnessに
34     //   値を書き込むとデバイス書き込み関数(leds_set)が呼び出されるように
35     //   登録します。
36     return led_classdev_register(NULL, &leds);
37 }
38
39 // remove関数(leds_remove)
40 static int leds_remove(struct platform_device *pdev)
41 {
42     // LEDクラスデバイスを解除します。(led_classdev_unregister)
43     led_classdev_unregister(&leds);
44
45     return 0;
46 }
47
48 // プラットフォームドライバ
49 static struct platform_driver leds_driver = {
50     .probe = leds_probe,
51     .remove = leds_remove,
52     .driver = {
53         .name = "armadillo-x1-extension-leds",
54         .owner = THIS_MODULE,
55     },
56 };
57
58 static struct platform_device *pdev;
59
60 // 初期化関数(leds_init)
61 static int __init leds_init(void)
62 {
63     int ret;
64

```

```

65 // プラットフォームデバイスを登録します。(platform_device_register_simple)
66 // ->プラットフォーム依存のデバイス情報を登録します。
67 pdev = platform_device_register_simple("armadillo-x1-extension-leds", -1, NULL, 0);
68 if (IS_ERR(pdev)) {
69     ret = (int)pdev;
70     goto err_ret;
71 }
72
73 // プラットフォームドライバを登録します。(platform_driver_register)
74 // ->登録したプラットフォームデバイスのリソース情報を取得し、
75 // プラットフォームドライバとして、probe関数とremove関数を登録します。
76 ret = platform_driver_register(&leds_driver);
77 if (ret < 0)
78     goto err_platform_device_unregister;
79
80 return 0;
81
82 err_platform_device_unregister:
83     platform_device_unregister(pdev);
84 err_ret:
85     return ret;
86 }
87
88 // 終了関数(leds_exit)
89 static void __exit leds_exit(void)
90 {
91     // プラットフォームドライバを解除します。(platform_driver_unregister)
92     platform_driver_unregister(&leds_driver);
93     // プラットフォームデバイスを解除します。(platform_device_unregister)
94     platform_device_unregister(pdev);
95 }
96
97 // 初期化の際に、初期化関数が呼ばれるように登録します。
98 module_init(leds_init);
99 // 終了する際に、終了関数が呼ばれるように登録します。
100 module_exit(leds_exit);
101
102 // MODULE_LICENSEは"GPL"とします。
103 MODULE_LICENSE("GPL");

```

Makefile

▼ /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds/Makefile

```

1  KERNELDIR = /home/atmark/linux-4.9-x1-at27_dbg
2  ARCH = arm
3  PREFIX = arm-linux-gnueabi-
4  MOD_PATH = /work/linux/nfsroot
5
6  EXTRA_CFLAGS += -gdwarf-2 -O0
7
8  obj-m := leds.o
9
10 modules:
11     $(MAKE) -C $(KERNELDIR) M=`pwd` ARCH=$(ARCH) CROSS_COMPILE=$(PREFIX) modules
12
13 modules_install:
14     $(MAKE) -C $(KERNELDIR) M=`pwd` ARCH=$(ARCH) INSTALL_MOD_PATH=$(MOD_PATH) modules_install
15
16 myinstall:
17     cp -p *.ko /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
18     cp -p *.c /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
19
20 clean:
21     $(MAKE) -C $(KERNELDIR) M=`pwd` clean

```

1.4.2. 動作確認

make clean

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds\$ make clean

```

1  atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds$ make clean
2  make -C /home/atmark/linux-4.9-x1-at27_dbg M=`pwd` clean
3  make[1]: ディレクトリ '/home/atmark/linux-4.9-x1-at27_dbg' に入ります
4  CLEAN /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds/.
5  CLEAN /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds/
6  make[1]: ディレクトリ '/home/atmark/linux-4.9-x1-at27_dbg' から出ます

```

make modules

⚠ 「make[2]: 警告: ファイル '/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds/leds.o' の修正時刻 20 は未来の時刻です」と表示された場合は chrony を ATDE8 と ArmadilloX1 にインストールすると解決する

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds\$ make modules

```

1  atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds$ make modules
2  make -C /home/atmark/linux-4.9-x1-at27_dbg M=`pwd` ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- modules
3  make[1]: ディレクトリ '/home/atmark/linux-4.9-x1-at27_dbg' に入ります
4  CC [M] /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds/leds.o
5  Building modules, stage 2.
6  MODPOST 1 modules
7  LD [M] /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds/leds.ko
8  make[1]: ディレクトリ '/home/atmark/linux-4.9-x1-at27_dbg' から出ます
9  atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds$

```

sudo make modules_install

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds\$ sudo make modules_install

```

1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/lec
2 | [sudo] atmark のパスワード:
3 | make -C /home/atmark/linux-4.9-x1-at27_dbg M=`pwd` ARCH=arm INSTALL_MOD_PATH=/work/linux/nfsroot modules_install
4 | make[1]: ディレクトリ '/home/atmark/linux-4.9-x1-at27_dbg' に入ります
5 |   INSTALL /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds/1
6 |   DEPMOD  4.9.133-at27
7 | depmod: WARNING: could not open modules.order at /work/linux/nfsroot/lib/modules/4.9.133-at27: No such file or dir
8 | depmod: WARNING: could not open modules.builtin at /work/linux/nfsroot/lib/modules/4.9.133-at27: No such file or c
9 | make[1]: ディレクトリ '/home/atmark/linux-4.9-x1-at27_dbg' から出ます

```

sudo make myinstall

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/leds\$ sudo make myinstall

```

1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led/drivers/lec
2 | cp -p *.ko /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
3 | cp -p *.c /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice

```

cd

▼ root@armadillo:/# cd /lib/modules/4.9.133-at27/extra/

```

1 | root@armadillo:/# cd /lib/modules/4.9.133-at27/extra/

```

insmod

▼ root@armadillo:/lib/modules/4.9.133-at27/extra# insmod leds.ko

```

1 | root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
2 | Module                Size  Used by
3 |
4 | root@armadillo:/lib/modules/4.9.133-at27/extra# insmod leds.ko
5 | leds: loading out-of-tree module taints kernel.
6 |
7 | root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
8 | Module                Size  Used by
9 | leds                  2103  0

```

1.4.3. デバイスファイル

/sys/class/leds/led_ext/brightness

▼ echo 255 > /sys/class/leds/led_ext/brightness

```

1 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo 255 > /sys/class/leds/led_ext/brightness

```

1.4.4. 実行している様子

▼ LEDデバイスファイルによるLED制御を実行している動画

<https://youtu.be/K0L9yKmVo8k>



1.5. 例題 ledctl

- 引数が指定されない時は LED の点灯状態を表示
- 引数を1つ指定すると LED の点灯状態を変更

1.5.1. ソース

ledctl.c

▼ 02.led/ledctl.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <fcntl.h>
4  #include <sys/types.h>
5  #include <sys/stat.h>
6  #include <unistd.h>
7
8  // LED制御用ファイル
9  #define LED_FILE      "/sys/class/leds/led_ext/brightness"
10 #define DATALEN      16
11
12 int main(int argc, char *argv[])
13 {
14     int fd;
15     int val;
16     char *p;
17     int ret, n;
18     int main_ret = 0;
19     char data[DATALEN];
20
21     // LED制御用ファイルをオープンします。
22     fd = open(LED_FILE, O_RDWR);
23     // オープンに失敗したら、main関数をエラー終了します。
24     if (fd < 0){
25         perror("open");
26         return 1;
27     }
28
29     switch (argc){
30     case 1:
31         // LEDの点灯状態をリードします。
32         ret = read(fd, data, DATALEN);
33         // リードに失敗したら、main関数をエラー終了します。
34         if (ret < 0){
35             perror("read");
36             main_ret = 1;
37             goto LED_ERR;
38         }
39         // LEDの点灯状態を表示します。
40         data[ret] = '\0';
41         printf("value: %s", data);
42         break;
43     case 2:
44         // 文字列を数値に変換します。
45         val = strtol(argv[1], &p, 0);
46         // 変換に失敗したら、main関数をエラー終了します。
47         if (*p != '\0'){
48             fprintf(stderr, "invalid number\n");
49             main_ret = 1;
50             goto LED_ERR;
51         }
52
53         // 数値が2バイトの範囲内でなければ、main関数をエラー終了します。
54         if (val < 0 || val > 255){
55             fprintf(stderr, "out of range\n");
56             main_ret = 1;
57             goto LED_ERR;
58         }
59
60         // 数値を文字列に変換します。
61         n = sprintf(data, "%d", val);
62         // LEDを点灯/消灯させます。
63         ret = write(fd, data, n);
64         // ライトに失敗したら、main関数をエラー終了します。

```



```

65         if (ret < 0){
66             perror("write");
67             main_ret = 1;
68             goto LED_ERR;
69         }
70         break;
71     default:
72         // 引数の数が指定どおりでない場合、main関数をエラー終了します。
73         fprintf(stderr, "Usage: %s [value]\n", argv[0]);
74         main_ret = 1;
75     }
76
77 LED_ERR:
78     // LED制御用ファイルをクローズします。
79     close(fd);
80     return main_ret;
81 }

```

Makefile

▼ 02.led/Makefile

```

1  CC = arm-linux-gnueabi-gcc
2  #TARGET = ledctl ldmetter
3  TARGET = ledctl <----- その都度変更
4  CFLAGS = -gdwarf-2 -O0
5
6  all: $(TARGET)
7
8  install:
9      cp -p $(TARGET) /work/linux/nfsroot/debug/04_practice
10     cp -p $(TARGET) /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
11     cp -p $(TARGET).c /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
12
13 clean:
14     rm -f $(TARGET)

```

1.5.2. 動作確認

make clean

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led\$ make clean

```

1  atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led$ make clean
2  rm -f ledctl

```

make

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led\$ make

```

1  atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led$ make
2  arm-linux-gnueabi-gcc -gdwarf-2 -O0 ledctl.c -o ledctl

```

sudo make install

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led\$ sudo make install

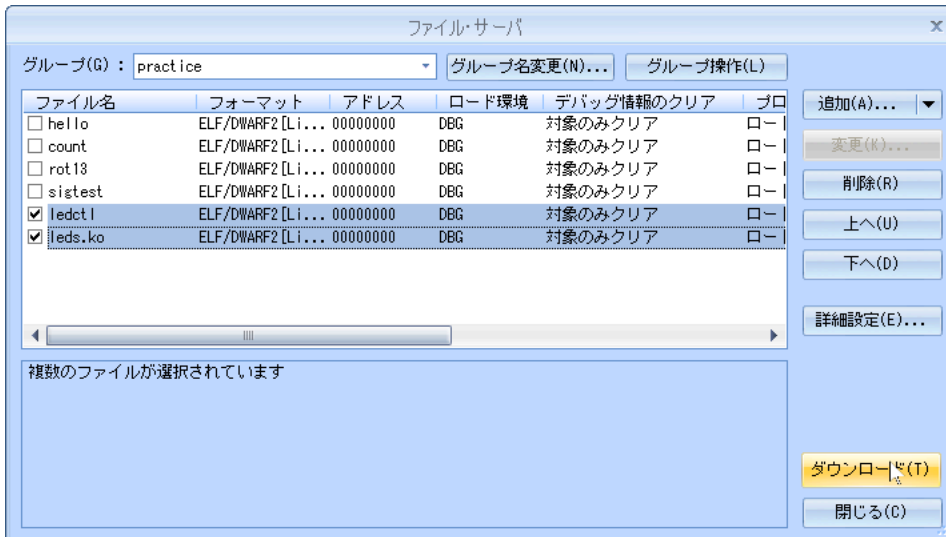
```

1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led$ sudo make
2 | [sudo] atmark のパスワード:
3 | cp -p ledctl /work/linux/nfsroot/debug/04_practice
4 | cp -p ledctl /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
5 | cp -p ledctl.c /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice

```

CSIDEでロード

▼ メニュー「ファイル」-「ロード」



insmod (既にinsmod 済みなら割愛)

▼ root@armadillo:/lib/modules/4.9.133-at27/extra# insmod leds.ko

```

1 | root@armadillo:/debug/04_practice# cd /lib/modules/4.9.133-at27/extra/
2 |
3 | root@armadillo:/lib/modules/4.9.133-at27/extra# ls
4 | leds.ko
5 |
6 | root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
7 | Module                Size Used by
8 |
9 | root@armadillo:/lib/modules/4.9.133-at27/extra# insmod leds.ko
10 |
11 | root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
12 | Module                Size Used by
13 | leds                  2103  0
14 |
15 | root@armadillo:/lib/modules/4.9.133-at27/extra#

```

実行している様子

▼ ledctl による LED 制御を実行している動画

<https://youtu.be/zoULMecAvbw>



1.6. 課題 負荷表示器 Idmeter

- システムの負荷をLEDに表示
- システムの負荷情報は /proc/loadavg ファイルから取得
- 1分間の平均値を取得してLED上に表示
- LED表示は1秒ごとに更新

負荷情報の取得

▼ root@armadillo:/debug/04_practice# cat /proc/loadavg

```
1 | root@armadillo:/debug/04_practice# cat /proc/loadavg
2 | 0.00 0.00 0.03 1/133 3076
```

実行可能なプロセス数の1分間の平均値、5分間、15分間、現在実行中のプロセス数/存在する全プロセス数、最新のプロセスID

負荷情報に対するLEDの状態

1min Avg	LED
0.00-0.12	ALL OFF
0.12-0.24	1
0.24-0.36	1,2
0.36-0.48	1-3
0.48-0.60	1-4
0.60-0.72	1-5
0.72-0.84	1-6
0.84-0.96	1-7
0.96 over	ALL ON

1.6.1. ソース

ldmeter.c

▼ 02.led/ldmeter.c

```

1  #include <stdio.h>
2  #include <fcntl.h>
3  #include <sys/types.h>
4  #include <sys/stat.h>
5  #include <unistd.h>
6
7  // LED制御用ファイル
8  #define LED_FILE "/sys/class/leds/led_ext/brightness"
9  // 負荷情報ファイル
10 #define LOADAVG_FILE "/proc/loadavg"
11
12 int main(void)
13 {
14
15
16
17
18
19     // 負荷情報ファイルをオープンします。
20
21
22
23
24
25
26
27     // LED制御用ファイルをオープンします。
28
29
30
31
32
33
34
35
36
37     // 負荷情報をリードします。
38
39
40
41
42
43
44
45
46
47     // 負荷情報を数値に変換します。
48
49
50
51
52
53
54
55
56     // 負荷情報から、どのLEDを点灯させるのかを決定します。
57
58
59
60
61
62
63
64     // LEDを点灯させます。
65

```

```

65
66         // 1秒間待ちます。
67         sleep(1);
68
69
70 LOAD_ERR:
71     // 負荷情報ファイルをクローズします。
72
73 LED_ERR:
74     // LED制御用ファイルをクローズします。
75
76     return main_ret;
77 }
78

```

Makefile

▼ 02.led/Makefile

```

1 CC = arm-linux-gnueabi-gcc
2 #TARGET = ledctl ldmetr
3 TARGET = ldmetr <----- その都度変更
4 CFLAGS = -gdwarf-2 -O0
5
6 all: $(TARGET)
7
8 install:
9     cp -p $(TARGET) /work/linux/nfsroot/debug/04_practice
10    cp -p $(TARGET) /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
11    cp -p $(TARGET).c /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
12
13 clean:
14     rm -f $(TARGET)

```

1.6.2. 動作確認

make clean

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led\$ make clean

```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led$ make clean
2 rm -f ldmetr

```

make

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led\$ make

```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led$ make
2 arm-linux-gnueabi-gcc -gdwarf-2 -O0 ldmetr.c -o ldmetr

```

sudo make install

▼ atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led\$ sudo make install

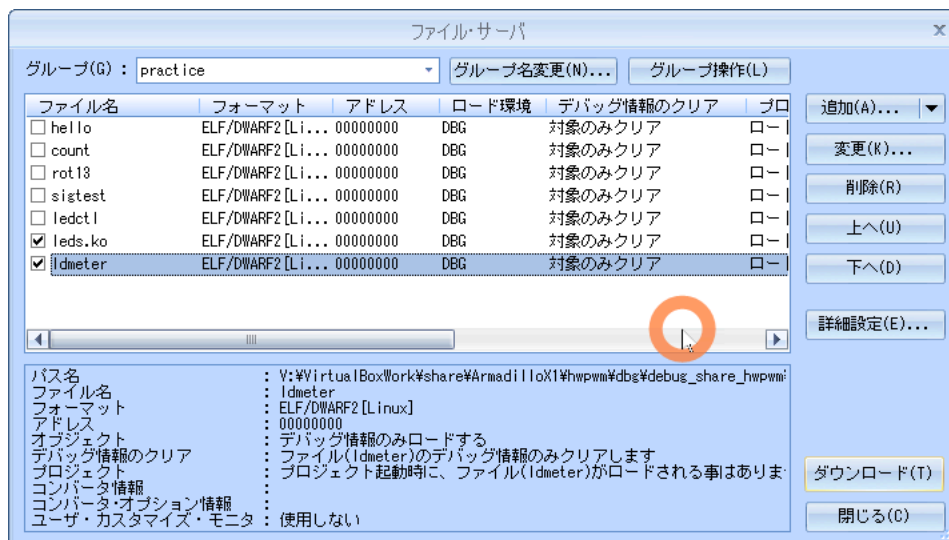
```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/02.led$ sudo make
2 [sudo] atmark のパスワード:
3 cp -p ldmetr /work/linux/nfsroot/debug/04_practice
4 cp -p ldmetr /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
5 cp -p ldmetr.c /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice

```

CSIDEでロード

▼ メニュー「ファイル」 - 「ロード」



insmod (既にinsmod済みなら割愛)

▼ root@armadillo:/lib/modules/4.9.133-at27/extra# insmod leds.ko

```
1 root@armadillo:~# cd /lib/modules/4.9.133-at27/extra/
2
3 root@armadillo:/lib/modules/4.9.133-at27/extra# ls
4 leds.ko
5
6 root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
7 Module                Size  Used by
8
9 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod leds.ko
10
11 root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
12 Module                Size  Used by
13 leds                  2103  0
14 root@armadillo:/lib/modules/4.9.133-at27/extra#
```

実行

./ldmeter &

▼ root@armadillo:/debug/04_practice# ./ldmeter &

```
1 root@armadillo:/debug/04_practice# ./ldmeter &
2 [1] 3046
```

CPU へ負荷

▼ root@armadillo:/debug/04_practice# while true; do echo dummy > /dev/null ; done

```
1 root@armadillo:/debug/04_practice# while true; do echo dummy > /dev/null ; done
2 ^Cdummy <----- ctrl + c
```

kill 3046

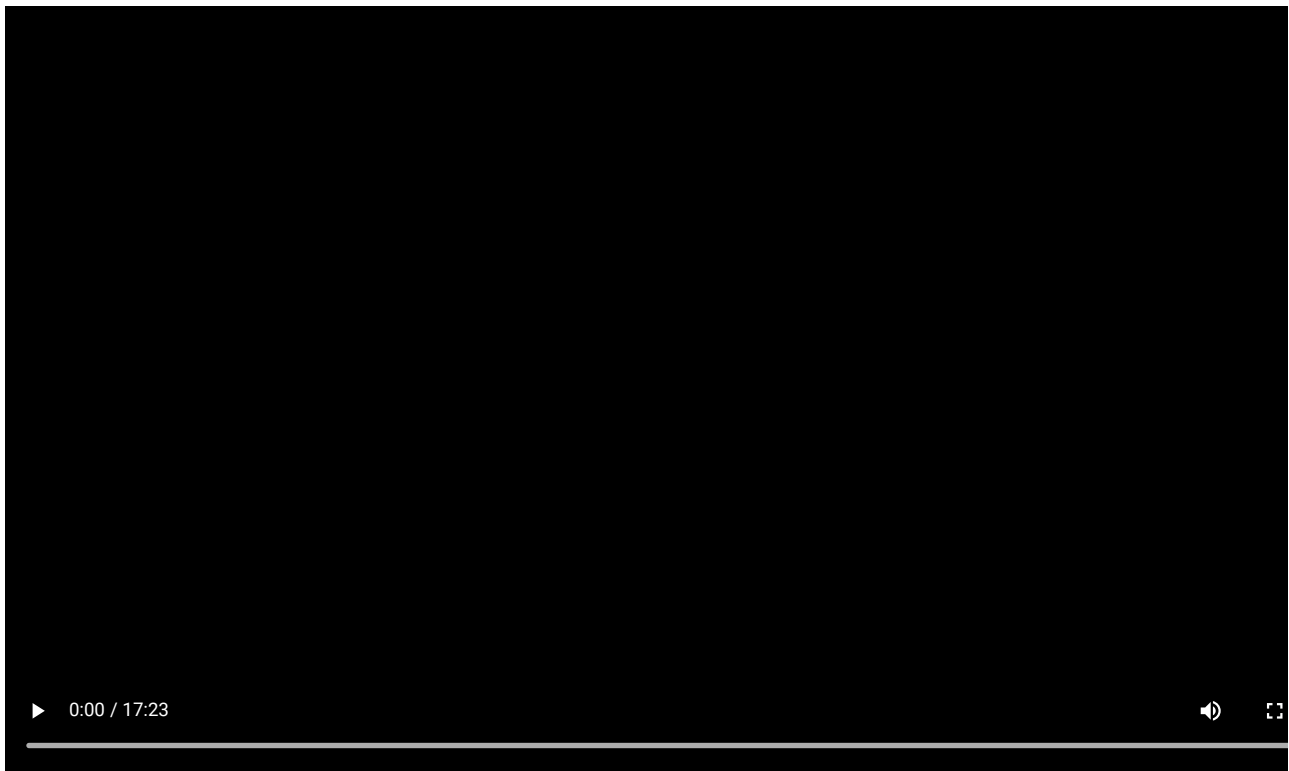
▼ root@armadillo:/debug/04_practice# kill 3046

```
1 root@armadillo:/debug/04_practice# ps
2   PID TTY          TIME CMD
3  2980 pts/0    00:03:06 bash
4  3046 pts/0    00:00:00 ldmeter
5  3055 pts/0    00:00:00 ps
6
7 root@armadillo:/debug/04_practice# kill 3046
```

実行している様子

▼ ldmeter による LED レベルメータ表示の動画

<https://youtu.be/vxxYnZIT9bM>



ヒント

sscanf, sleep