

1. 06.d7seg

2024/7/18 Table of Contents

06.d7seg

目的

構成データ

d7seg制御

デバイスドライバ

例題 d7ctl

課題1 d7fan

1.1. 目的

組込みアプリケーション開発 06.d7seg

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

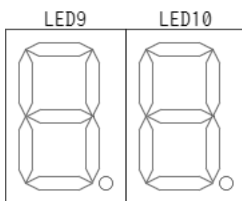
▼ .../share/ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice/ の構成

```
1 user@1204PC-Z490M:/mnt/v/VirtualBoxWork/share/ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice$ tr
2 ./
3 | 06.d7seg/
4 | | d7ctl.c*          <----- 例題 デバイス制御用ソース
5 | | d7fan.c*          <----- 課題1 デバイス制御用ソース
6 | | drivers/
7 | | | buttons/
8 | | | d7seg/
9 | | | | d7seg.c*      <----- ドライバソース
10 | | | | Makefile*    <----- ドライバ用Makefile
11 | | | leds/
12 | | | motor/
13 | | Makefile*        <----- デバイス制御用Makefile
```

1.3. d7seg制御

1.3.1. デバイス仕様

d7segのハードウェア仕様



デバイスファイル /dev/d7seg に値を書き込むことで表示

1.4. デバイスドライバ

1.4.1. ソース

d7seg.c

▼ 06.d7seg/drivers/d7seg.c

```

1  #include <linux/module.h>
2  #include <linux/platform_device.h>
3  #include <linux/fs.h>
4  #include <linux/io.h>
5  #include <linux/uaccess.h>
6  #include <linux/cdev.h>
7
8  #include <asm/armadilloX1-ext-cpld.h>
9
10 #define D7SEG_MAJOR          121
11 #define D7SEG_MINOR          0
12 #define D7SEG_NUM            1
13
14 #define D7SEG_ENABLE          0x0C
15 #define D7SEG_DISABLE        0x00
16
17 static struct platform_device *pdev;
18
19 static dev_t dev;
20
21 static struct cdev *chrdev;
22
23 /* write用の関数ハンドラ */
24 static ssize_t d7seg_write(struct file *file, const char __user *buf, size_t count, loff_t *ppos)
25 {
26     int ret;
27     unsigned char dat = 0;
28
29     if (count > 1)
30         count = 1;
31
32     // ユーザバッファからデータを取り出す
33     ret = copy_from_user(&dat, buf, count);
34     if (ret)
35         return -EFAULT;
36
37     // LED点灯状態レジスタにデータを書き込む
38     cpld_write(CPLD_WRITE_7SEG_DATA, dat);
39
40     return count;
41 }
42
43 /* システムコールと関数の登録 */
44 static const struct file_operations d7seg_fops = {
45     .owner  = THIS_MODULE,
46     .write  = d7seg_write,
47 };
48
49 /* クラス構造体登録 */
50 static struct class d7seg_class = {
51     .owner  = THIS_MODULE,
52     .name   = "d7seg",
53 };
54
55 /* デバイスファイルの作成関数 */
56 static int d7seg_probe(struct platform_device *pdev)
57 {
58     struct device *device;
59
60     // デバイスファイルを作成する
61     device = device_create(&d7seg_class, NULL, MKDEV(D7SEG_MAJOR, D7SEG_MINOR), NULL, "d7seg");
62     if (IS_ERR(device)){
63         dev_err(&pdev->dev, KERN_ERR "failed to create device.\n");
64         return PTR_ERR(device);
65     }
66 }

```

```

65     }
66
67     // 7セグメントLEDを有効化する
68     cpld_write(CPLD_WRITE_7SEG_CTRL, D7SEG_ENABLE);
69
70     // LED点灯状態レジスタにデータを書き込む
71     cpld_write(CPLD_WRITE_7SEG_DATA, 0x00);
72
73     return 0;
74 }
75
76 /* デバイスファイルの削除関数 */
77 static int d7seg_remove(struct platform_device *pdev)
78 {
79     // 7セグメントLEDを無効化する
80     cpld_write(CPLD_WRITE_7SEG_CTRL, D7SEG_DISABLE);
81
82     // デバイスファイルを消去する
83     device_destroy(&d7seg_class, MKDEV(D7SEG_MAJOR, D7SEG_MINOR));
84
85     return 0;
86 }
87
88 /* プラットフォーム構造体 */
89 static struct platform_driver d7seg_driver = {
90     .probe      = d7seg_probe,
91     .remove     = d7seg_remove,
92     .driver     = {
93         .name    = "armadillo-x1-extension-d7seg",
94         .owner   = THIS_MODULE,
95     },
96 };
97
98 /* 初期化関数 */
99 static int __init d7seg_init(void)
100 {
101     int ret = 0;
102
103     // Class登録
104     ret = class_register(&d7seg_class);
105     if (ret)
106         goto err1;
107
108     // プラットフォームドライバ登録
109     ret = platform_driver_register(&d7seg_driver);
110     if (ret)
111         goto err2;
112
113     // デバイス番号の登録
114     dev = MKDEV(D7SEG_MAJOR, D7SEG_MINOR);
115     ret = register_chrdev_region(dev, D7SEG_NUM, "d7seg");
116     if (ret < 0)
117         goto err3;
118
119     chrdev = cdev_alloc();
120     if (!chrdev)
121         goto err4;
122     chrdev->owner = THIS_MODULE;
123
124     /* ファイル操作関数を登録 */
125     chrdev->ops = &d7seg_fops;
126     ret = cdev_add(chrdev, dev, (unsigned int)D7SEG_NUM);
127     if (ret < 0)
128         goto err5;
129
130 }

```

```

130 // デバイスを登録
131 pdev = platform_device_register_simple("armadillo-x1-extension-d7seg", -1, NULL, 0);
132 if (IS_ERR(pdev)) {
133     ret = (int)pdev;
134     goto err5;
135 }
136
137 return 0;
138
139 err5: cdev_del(chrdev);
140 err4: unregister_chrdev_region(dev, (unsigned int)D7SEG_NUM);
141 err3: platform_driver_unregister(&d7seg_driver);
142 err2: class_unregister(&d7seg_class);
143 err1:
144     return ret;
145 }
146
147 /* 終了関数 */
148 static void __exit d7seg_exit(void)
149 {
150     platform_device_unregister(pdev);
151     cdev_del(chrdev);
152     unregister_chrdev_region(dev, D7SEG_NUM);
153     platform_driver_unregister(&d7seg_driver);
154     class_unregister(&d7seg_class);
155 }
156
157 module_init(d7seg_init);
158 module_exit(d7seg_exit);
159
160 MODULE_DESCRIPTION("Seven-segment display driver");
161 MODULE_LICENSE("GPL");

```

Makefile

▼ 06.d7seg/drivers/d7seg/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 := d7seg.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

▼ \$ make clean

```
1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg/drivers/c
2 | make: *** ターゲット 'clear' を make するルールがありません。 中止。
```

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

▼ \$ make modules

```
1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg/drivers/c
2 | make -C /home/atmark/linux-4.9-x1-at27_dbg M='pwd' ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- 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/06.d7seg/drivers/d7se
5 | Building modules, stage 2.
6 | MODPOST 1 modules
7 | make[2]: 警告: ファイル '/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7se
8 | CC /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg/drivers/d7se
9 | LD [M] /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg/drivers/d7se
10 | make[2]: 警告: 時刻のずれを検出。不完全なビルド結果になるかもしれません。
11 | make[1]: ディレクトリ '/home/atmark/linux-4.9-x1-at27_dbg' から出ます
```

sudo make modules_install

▼ \$ sudo make modules_install

```
1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg/drivers/c
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/06.d7seg/drivers/d7se
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

▼ \$ sudo make myinstall

```
1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg/drivers/c
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:/debug/04_practice# cd /lib/modules/4.9.133-at27/extra/
```


insmod

▼ #insmod d7seg.ko

```
1 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod d7seg.ko
2 root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
3 Module                Size  Used by
4 d7seg                  3226  0
5 motor_hwpwm            4415  0
6 buttons                3065  0
7 leds                   2103  0
```

1.4.3. デバイスファイル

"/dev/d7seg"

 echo -e "\x00" > /dev/d7seg してもバイナリデータじゃないので反応しない

1.5. 例題 d7ctl

- d7ctl の引数には 1桁目の数字 (0~9) または2桁分の数字 (00~99)
- 引数が1桁 (0~9) の場合、LED9は0、LED10は数字を表示
- 引数が2桁 (00~99) の場合、LED9は2桁目、LED10は1桁目の数字を表示

1.5.1. ソース

d7ctl.c

▼ 06.d7seg/d7ctl.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <fcntl.h>
4  #include <string.h>
5  #include <sys/types.h>
6  #include <sys/stat.h>
7  #include <unistd.h>
8
9  // 7セグメントLED制御用ファイル
10 #define D7SEG_FILE      "/dev/d7seg"
11
12 // 本アプリの説明を、コンソール上に表示し、アプリを終了します。
13 void usage()
14 {
15     printf("usage: d7ctl d1dd (d:0-9)\n");
16     exit(EXIT_FAILURE);
17 }
18
19 int main(int argc, char *argv[])
20 {
21     int i;
22     unsigned char buf = 0;
23     unsigned char c;
24     int fd;
25     int main_ret = 0;
26
27     // 引数が1つ指定されていない場合、または、引数に指定した値が2桁より大きい
28     // (7セグメントLEDに収まらない)場合は、コンソール上に本アプリの説明を表示
29     // し、アプリを終了します。
30     if ((argc != 2) || (strlen(argv[1]) > 2)){
31         usage();
32     }
33
34     // 7セグメントLEDに表示する数字を、バッファに格納します。
35     for (i = 0; i < 2; i++){
36         // 引数がNULLの場合は、バッファに何も格納しません。
37         c = argv[1][i];
38         if (!c){
39             break;
40         }
41         // バッファに入っている1桁目の値を2桁目に左シフトします。
42         buf <<= 4;
43         // 引数に0~9を指定している場合、1桁目のバッファに値を格納します。
44         if (c >= '0' && c <= '9'){
45             buf |= (c - '0');
46         }
47         // 引数に0~9以外を指定している場合、コンソール上に本アプリの説明を
48         // 表示し、アプリを終了します。
49         else {
50             usage();
51         }
52     }
53
54     // 7セグメントLED制御用ファイルをオープンします。
55     fd = open(D7SEG_FILE, O_WRONLY);
56     // オープンに失敗したら、main関数をエラー終了します。
57     if (fd < 0){
58         perror("failed to open d7seg");
59         main_ret = 1;
60         goto D7SEG_ERR;
61     }
62
63     // 7セグメントLEDを表示します。
64     if (write(fd, &buf, 1) < 0){

```



```

65 // 書き込みに失敗したら、main関数をエラー終了します。
66 perror("failed to write d7seg");
67 main_ret = 1;
68 goto D7SEG_ERR;
69 }
70
71 D7SEG_ERR:
72 // 7セグメントLED制御用ファイルをクローズします。
73 close(fd);
74
75 return main_ret;
76 }

```

Makefile

▼ 06.d7seg/Makefile

```

1 CC = arm-linux-gnueabi-gcc
2 #TARGET = d7ctl d7fan
3 TARGET = d7ctl
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)
15
16 .PHONY: clean

```

1.5.2. 動作確認

make clean

▼ \$ make clean

```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg$ make clean
2 rm -f d7ctl

```

make

▼ \$ make

```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg$ make
2 arm-linux-gnueabi-gcc -gdwarf-2 -O0 d7ctl.c -o d7ctl

```

sudo make install

▼ \$ sudo make install

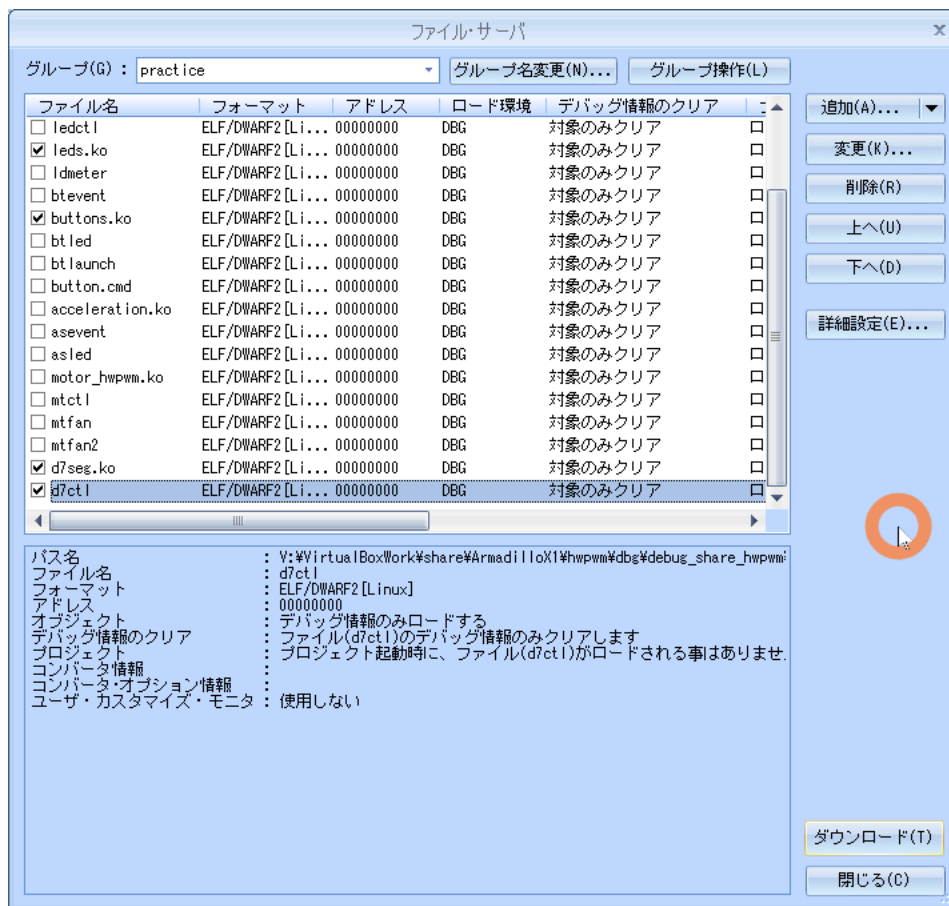
```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg$ sudo make install
2 cp -p d7ctl /work/linux/nfsroot/debug/04_practice
3 cp -p d7ctl /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
4 cp -p d7ctl.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)

▼ # insmod d7seg.ko

```
1 | root@armadillo:~# cd /lib/modules/4.9.133-at27/extra/  
2 | root@armadillo:/lib/modules/4.9.133-at27/extra# insmod d7seg.ko  
3 | root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod  
4 | Module                Size  Used by  
5 | d7seg                  3226  0  
6 | root@armadillo:/lib/modules/4.9.133-at27/extra#
```

実行結果

▼ root@armadillo:/debug/04_practice# ./d7ctl

```
1 | root@armadillo:/debug/04_practice# ./d7ctl 99
```

実行している様子

▼ d7ctl を実行している動画

<https://youtu.be/aR0KXLVeaVU>



1.6. 課題1 d7fan

mtfan2 に スイッチオフタイマーを追加

タイマー仕様

- SW4を押下 → タイムアウト時間を10秒にセットしてタイマ起動
- SW5を押下 → タイムアウト時間を20秒にセットしてタイマ起動
- SW6を押下 → タイムアウト時間を30秒にセットしてタイマ起動
- SW7を押下 → タイマ取消

タイマの動作

- タイムアウトしたら扇風機を停止
- モータ停止中はSW4～SW7の操作を無効

7セグメントLEDにタイマの状態を表示

- タイマ非動作時は、7セグメントLEDに0を表示
- タイマ動作中は、7セグメントLEDに残り時間（秒）を表示

1.6.1. ソース

d7fan.c

▼ 06.d7seg/d7fan.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <fcntl.h>
4  #include <linux/input.h>
5  #include <signal.h>
6  #include <string.h>
7  #include <errno.h>
8  #include <sys/types.h>
9  #include <sys/stat.h>
10 #include <unistd.h>
11 #include <sys/time.h>
12
13 // ボタン制御用ファイル
14 #define BUTTON_FILE      "/dev/input/event3"
15 // LED制御用ファイル
16 #define LED_FILE "/sys/class/leds/led_ext/brightness"
17 // モータ制御用ファイル
18 #define MOTOR_FILE      "/sys/class/motor/motor0/motor_rotation"
19 // 7セグメントLED制御用ファイル
20 #define D7SEG_FILE      "/dev/d7seg"
21 #define DATALEN        16
22 #define MOTOR_ON        1
23 #define MOTOR_OFF       0
24 #define TIMER_INTERVAL  1
25 #define TIMER_ON        1
26 #define TIMER_OFF       0
27
28 int fd_mt, fd_led, fd_d7seg;
29 int motor_state;
30 int timer_state;
31 struct timeval timer_expire;
32
33 void change_motor(int rotation)
34 {
35
36
37
38
39
40     // モータの状態変化を書き込みます。
41
42
43
44
45
46
47
48
49
50 }
51
52 void change_led(int num)
53 {
54
55
56
57
58
59     // LEDの状態変化を書き込みます。
60
61
62
63
64

```

```

65
66 }
67
68 void change_d7seg(int num)
69 {
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84 // 7セグメントLEDの状態変化を書き込みます。
85
86
87
88
89
90
91 }
92
93 void check_timer(int signum)
94 {
95
96
97
98
99
100
101
102
103
104
105
106
107
108 // 1秒後にシグナルが送信されるように設定します。
109
110
111 // モータの回転を停止させます。
112
113 // LEDを消灯させます。
114
115
116
117 // 7セグメントLEDの表示を0とします。
118
119
120 }
121
122 void start_timer(int tout)
123 {
124
125
126
127
128
129
130

```

```

130
131
132
133
134
135     // 7セグメントLEDの表示を更新します。
136
137     // 1秒後にシグナルが送信されるように設定します。
138
139 }
140
141 void stop_timer(void)
142 {
143
144
145     // 7セグメントLEDの表示を0とします。
146
147     // アラームを終了します。
148
149 }
150
151 int main(void)
152 {
153
154
155
156
157
158     // モータ制御用ファイルをオープンします。
159
160
161
162
163
164
165
166     // LED制御用ファイルをオープンします。
167
168
169
170
171
172
173
174     // ボタン制御用ファイルをオープンします。
175
176
177
178
179
180
181
182     // 7セグメントLED制御用ファイルをオープンします。
183
184
185
186
187
188
189
190     // SIGALRM受信時のシグナルハンドラを登録します。
191
192
193
194
195

```

195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260

```

260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284 // モータ制御用ファイルをクローズします。
285
286 // LED制御用ファイルをクローズします。
287
288 // ボタン制御用ファイルをクローズします。
289
290 // 7セグメントLED制御用ファイルをクローズします。
291
292 return 0;
293 }

```

Makefile

▼ 06.d7seg/Makefile

```

1 CC = arm-linux-gnueabi-gcc
2 #TARGET = d7ctl d7fan
3 TARGET = d7fan
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)
15
16 .PHONY: clean

```

1.6.2. 動作確認

make clean

▼ \$ make clean


```

1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg$ make cle
2 | rm -f d7fan

```

make

▼ \$ make

```

1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg$ make
2 | arm-linux-gnueabi-gcc -gdwarf-2 -O0 d7fan.c -o d7fan

```

sudo make install

▼ \$ sudo make install

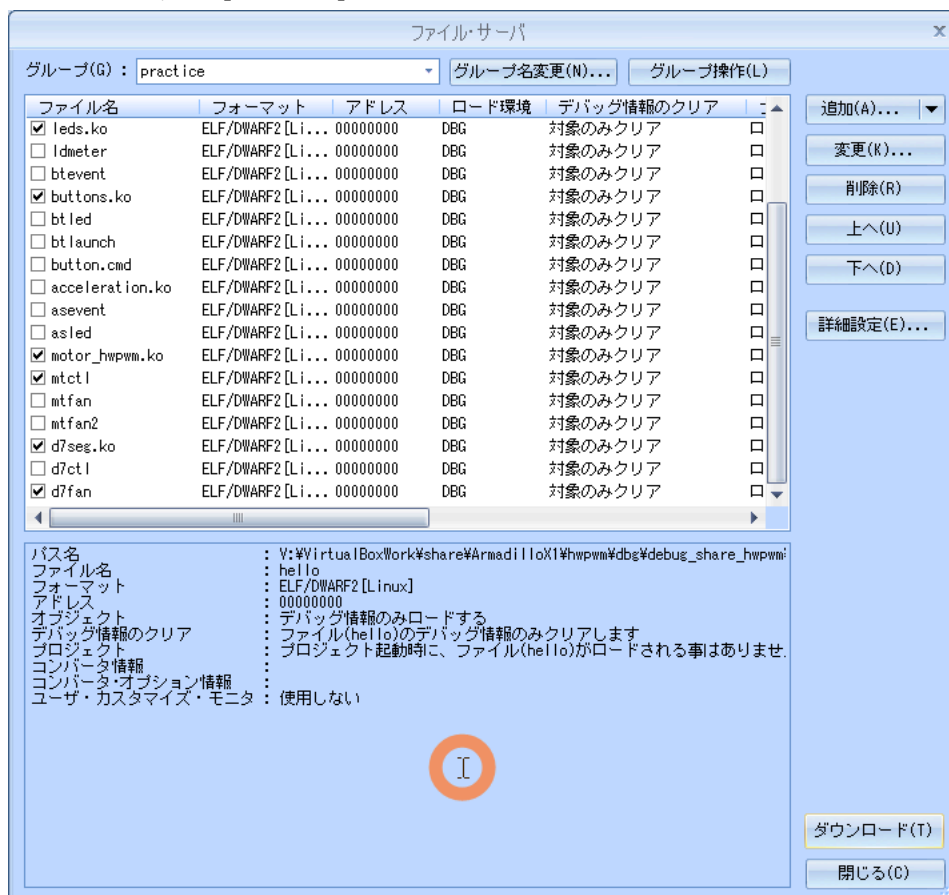
```

1 | atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/06.d7seg$ sudo mak
2 | [sudo] atmark のパスワード:
3 | cp -p d7fan /work/linux/nfsroot/debug/04_practice
4 | cp -p d7fan /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
5 | cp -p d7fan.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)

▼ # insmod leds.ko, # insmod buttons.ko, # insmod motor_hwpwm.ko, # insmod d7seg.ko

```
1 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod d7seg.ko
2 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod leds.ko
3 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod buttons.ko
4 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod motor_hwpwm.ko
5 root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
6 Module                Size  Used by
7 motor_hwpwm            4415  0
8 buttons                3065  0
9 leds                   2103  0
10 d7seg                  3226  0
```

実行

▼ root@armadillo:/debug/04_practice# ./d7fan

```
1 | root@armadillo:/debug/04_practice# ./d7fan
```

実行している様子

▼

<https://youtu.be/7wtLI7YoFok>

