

# 1. 07.charlcd

2024/7/18 Table of Contents

07.charlcd

目的

構成データ

キャラクタLCD制御

デバイスドライバ

例題 cctl

課題1 cclock

## 1.1. 目的

組込みアプリケーション開発 07.charlcd

## 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 | 07.charlcd/
4 | | clclock.c*          <----- 課題1 デバイス制御用ソース
5 | | clctl.c*           <----- 例題 デバイス制御用ソース
6 | | drivers/
7 | | | clcd/
8 | | | | cgram/
9 | | | | | default.h* <-----
10 | | | | | swedish.h* <-----
11 | | | | | charmap.h* <-----
12 | | | | | commands.h* <-----
13 | | | | | compat.h* <-----
14 | | | | | config.h* <-----
15 | | | | | hardware.h* <-----
16 | | | | | hd44780.c* <----- モジュールソース
17 | | | | | hd44780.h* <----- モジュールヘッダ
18 | | | | | Kconfig* <-----
19 | | | | | lcd-linux.c* <----- ドライバソース
20 | | | | | lcd-linux.h* <----- ドライバヘッダ
21 | | | | | Makefile* <----- ドライバ用Makefile
22 | | | | | Makefile-2.6*
23 | | | | | Makefile* <----- デバイス制御用Makefile
```

## 1.3. キャラクタLCD制御

### 1.3.1. デバイス仕様

キャラクタLCDのハードウェア仕様

- デバイスファイルに書き込んだ文字列がキャラクタLCDパネルに表示
- 画面制御は制御文字とエスケープシーケンスで行う



制御コードの一部を示します。

制御コード	意味
\f	画面消去
\r	復帰

制御コード	意味
\n	改行
ESC aN	カーソル制御 (0:非表示 1:表示)
ESC bN	ブリンクカーソル制御 (0:非表示 1:表示)

## 1.4. デバイスドライバ

 insmod は lcd-linux.ko -> hd44780.ko の順番で

### 1.4.1. ソース

#### lcd-linux.c

▼ 07.charlcd/drivers/clcd/lcd-linux.c

[lcd-linux.c](#)

#### hd44780.c

▼ 07.charlcd/drivers/clcd/hd44780.c

[hd44780.c](#)

#### Makefile

▼ 07.charlcd/drivers/clcd/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 := lcd-linux.o hd44780.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/07.charlcd/drivers
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 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

▼ \$ make modules

```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers
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/07.charlcd/drivers/cl
5 CC [M] /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/cl
6 /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/clcd/hd44780
7 static int hd44780_handle_custom_ioctl(unsigned int num, unsigned long arg, unsigned int user_space)
8 ~~~~~
9 /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/clcd/hd44780
10 static void hd44780_read_cgram_char(unsigned char index, unsigned char *pixels)
11 ~~~~~
12 /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/clcd/hd44780
13 static void hd44780_read_char(unsigned int offset, unsigned short *data)
14 ~~~~~
15 Building modules, stage 2.
16 MODPOST 2 modules
17 make[2]: 警告: ファイル '/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.char
18 CC /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/cl
19 LD [M] /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/cl
20 CC /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/cl
21 LD [M] /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/cl
22 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。
23 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/07.charlcd/drivers
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/07.charlcd/drivers/cl
6 INSTALL /media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd/drivers/cl
7 DEPMOD 4.9.133-at27
8 depmod: WARNING: could not open modules.order at /work/linux/nfsroot/lib/modules/4.9.133-at27: No such file or dir
9 depmod: WARNING: could not open modules.builtin at /work/linux/nfsroot/lib/modules/4.9.133-at27: No such file or c
10 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/07.charlcd/drivers
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

 insmod は lcd-linux.ko -> hd44780.ko の順番

▼ #insmod lcd-linux.ko -> #insmod hd44780.ko

```
1 | root@armadillo:/lib/modules/4.9.133-at27/extra# insmod lcd-linux.ko
2 | root@armadillo:/lib/modules/4.9.133-at27/extra# insmod hd44780.ko
3 | root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
4 | Module                Size  Used by
5 | hd44780                16250  0
6 | lcd_linux              51414  2 hd44780
7 | d7seg                   3226   0
8 | motor_hwpwm            4415   0
9 | buttons                 3065   0
10 | leds                   2103   0
```

## 1.4.3. デバイスファイル

"/dev/hd44780"

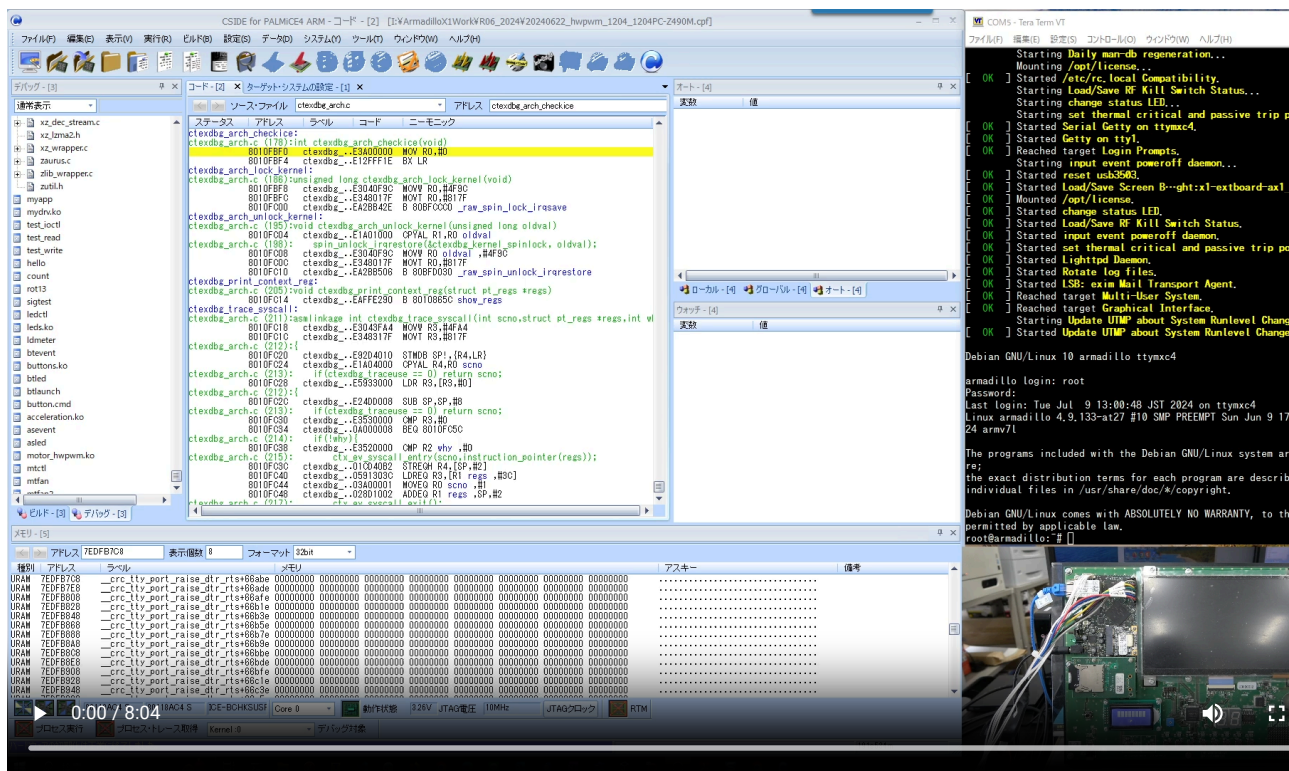
▼ hd44780 デバイスファイルによるキャラクタLCD表示

```
1 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne 'Hello\r\nWorld!' > /dev/hd44780
2 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne '\f' > /dev/hd44780
3 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne '\x1bb1' > /dev/hd44780
4 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne '\x1bb0' > /dev/hd44780
5 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne '\x1ba1' > /dev/hd44780
6 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne '\x1bb1' > /dev/hd44780
7 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne '\x1ba0' > /dev/hd44780
8 | root@armadillo:/lib/modules/4.9.133-at27/extra# echo -ne '\x1bb0' > /dev/hd44780
9 | root@armadillo:/lib/modules/4.9.133-at27/extra#
10 |
```

## 1.4.4. 実行している様子

▼ hd44780 デバイスファイルによるキャラクタLCD表示している動画

<https://youtu.be/iw0JfRs6LUE>



## 1.5. 例題 clctl

- 引数がない場合、キャラクタLCDの内容を消去
- 引数を1つ以上指定すると、キャラクタLCDの内容を消去した後、指定された文字列を表示
- 制御コードは、\r と \n のみをサポート

### 1.5.1. ソース

#### clctl.c

▼ 07.charlcd/clctl.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  // キャラクタLCD制御用ファイル
8  #define CLCD_FILE      "/dev/hd44780"
9  #define DATALEN      80
10
11 int main(int argc, char *argv[])
12 {
13     int fd;
14     int ret, n;
15     int i, j;
16     char data[DATALEN];
17     int main_ret = 0;
18
19     // キャラクタLCD制御用ファイルをオープンします。
20     fd = open(CLCD_FILE, O_RDWR);
21     // オープンに失敗したら、main関数をエラー終了します。
22     if (fd < 0){
23         perror("open");
24         return 1;
25     }
26
27     // 画面消去コードを文字列に連結します。
28     n = sprintf(data, "\f");
29     // 引数が1つ以上指定されていたら、キャラクタLCDに表示するための文字列を
30     // 連結します。
31     if (argc > 1){
32         for (i = 1; i < argc; i++){
33             for (j = 0; argv[i][j]; j++){
34                 // '\の输入があった場合は、制御コードかどうか判断します。
35                 if (argv[i][j] == '\\'){
36                     switch (argv[i][j + 1]){
37                         // '\の次の文字が'r'の場合は、復帰コードを連結します。
38                         case 'r':
39                             data[n++] = '\r';
40                             j++;
41                             break;
42                         // '\の次の文字が'n'の場合は、改行コードを連結します。
43                         case 'n':
44                             data[n++] = '\n';
45                             j++;
46                             break;
47                         // '\の次の文字が'r'でも'n'でもない場合は、'\を表示
48                         // するように連結します。
49                         default:
50                             data[n++] = '\\';
51                     }
52                 } else {
53                     // 1文字連結します。
54                     data[n++] = argv[i][j];
55                 }
56             }
57             data[n++] = ' ';
58         }
59         n -= 1;
60     }
61
62     // 連結した文字列をキャラクタLCDに表示します。
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     }
69
70     // キャラクタLCD制御用ファイルをクローズします。
71     close(fd);
72     return main_ret;
73 }

```

## Makefile

### ▼ 07.charlcd/Makefile

```

1  CC = arm-linux-gnueabi-hf-gcc
2  #TARGET = clctl clclock
3  TARGET = clctl
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/07.charlcd$ make clean
2  rm -f clctl

```

### make

#### ▼ \$ make

```

1  atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd$ make
2  arm-linux-gnueabi-hf-gcc -gdwarf-2 -O0 clctl.c -o clctl

```

### sudo make install

#### ▼ \$ sudo make install

```

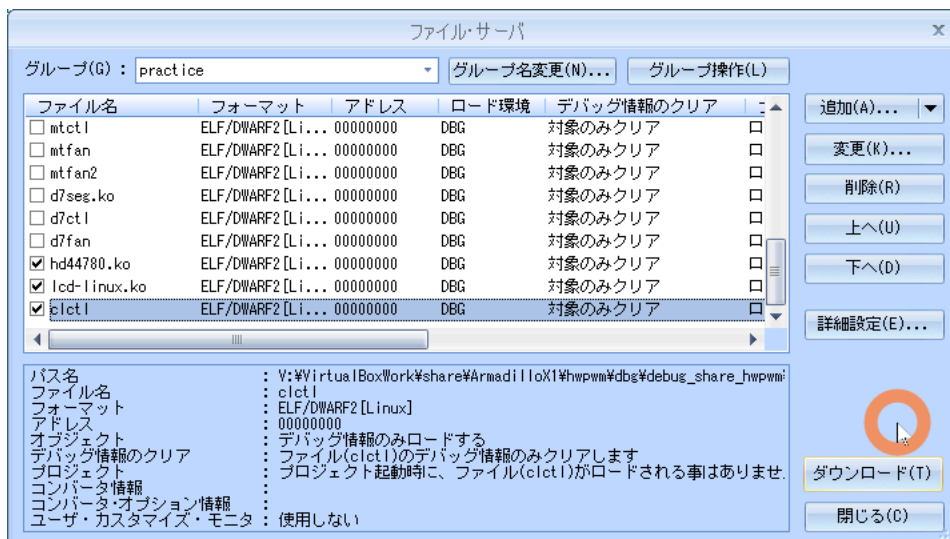
1  atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd$ sudo make install
2  [sudo] atmark のパスワード:
3  cp -p clctl /work/linux/nfsroot/debug/04_practice
4  cp -p clctl /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
5  cp -p clctl.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 lcd-linux.ko -> #insmod hd44780.ko

```

1 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod lcd-linux.ko
2 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod hd44780.ko
3 root@armadillo:/lib/modules/4.9.133-at27/extra# lsmod
4 Module                               Size  Used by
5 hd44780                             16250  0
6 lcd_linux                          51414  2 hd44780
7 d7seg                               3226  0
8 motor_hwpwm                        4415  0
9 buttons                            3065  0
10 leds                               2103  0

```

## 実行結果

▼ root@armadillo:/debug/04\_practice# ./clctl

```

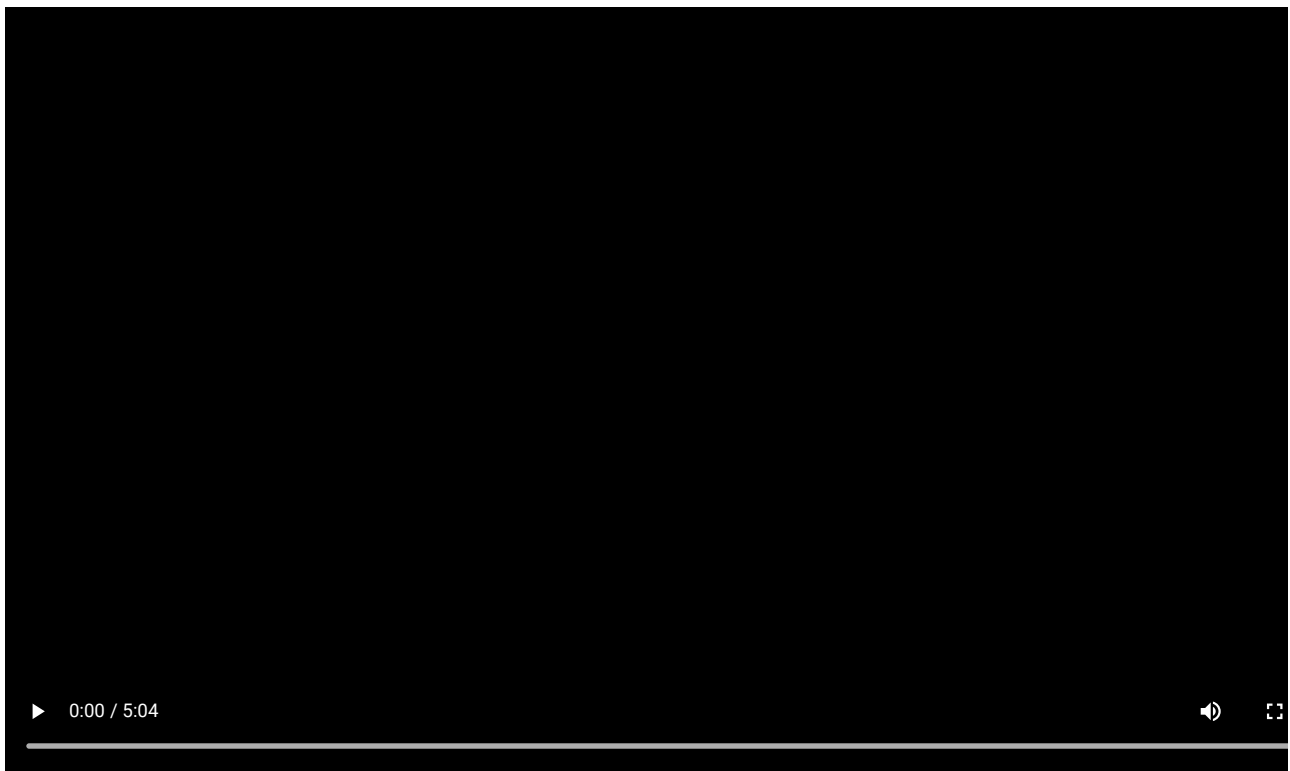
1 | root@armadillo:/debug/04_practice# ./clctl 12345678\\r\\nabcdefg

```

## 実行している様子

▼ clctl を実行している動画

<https://youtu.be/zg64i9R1as8>



## 1.6. 課題1 clclock

<sup>0</sup>							
0	2	0	2	4	0	7	7
1	1	2	:	3	4	:	5

### 1.6.1. ソース

#### clclock.c

▼ 07.charlcd/clclock.c

```

1  #include <stdio.h>
2  #include <fcntl.h>
3  #include <time.h>
4  #include <sys/time.h>
5  #include <sys/types.h>
6  #include <sys/stat.h>
7  #include <unistd.h>
8
9  // キャラクタLCD制御用ファイル
10 #define CLCD_FILE      "/dev/hd44780"
11
12 int main(void)
13 {
14
15
16
17
18
19     // キャラクタLCD制御用ファイルをオープンします。
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51     // 1秒間待ちます。
52     sleep(1);
53
54
55 CLCD_ERR:
56     // キャラクタLCD制御用ファイルをクローズします。
57
58     return main_ret;
59 }

```

## Makefile

### ▼ 07.charlcd/Makefile

```

1 CC = arm-linux-gnueabi-gcc
2 #TARGET = clctl clclock
3 TARGET = clclock
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/07.charlcd$ make c
2 rm -f clclock

```

### make

▼ \$ make

```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd$ make
2 arm-linux-gnueabi-gcc -gdwarf-2 -O0 clclock.c -o clclock

```

### sudo make install

▼ \$ sudo make install

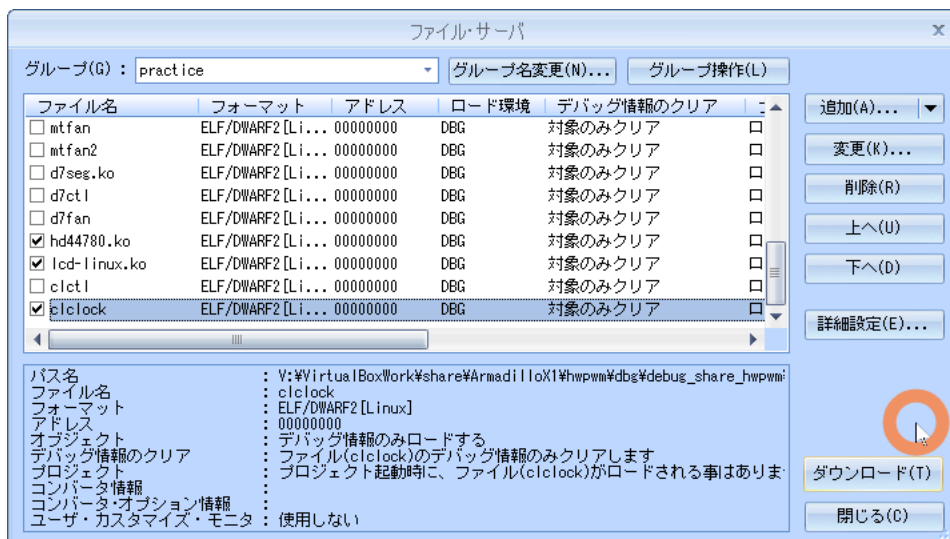
```

1 atmark@atde8:/media/sf_ArmadilloX1/hwpwm/work/R06_2024/Application_debug/text/practice-example/07.charlcd$ sudo m
2 cp -p clclock /work/linux/nfsroot/debug/04_practice
3 cp -p clclock /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice
4 cp -p clclock.c /media/sf_ArmadilloX1/hwpwm/dbg/debug_share_hwpwm/R06_2024/04_practice

```

## C SIDE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 lcd-linux.ko

```
1 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod lcd-linux.ko
2 root@armadillo:/lib/modules/4.9.133-at27/extra# insmod hd44780.ko
```

## 実行

▼ root@armadillo:/debug/04\_practice# ./clclock

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

## 実行している様子

▼ clclock を実行

<https://youtu.be/ztpdkQt-bHo>



### 1.6.3. ヒント

---

gettimeofday, localtime