

[PATCH 2.6.18 V6] drivers: add lcd display support

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-09/msg09054.html>

- *From:* Miguel Ojeda Sandonis <maxextreme@xxxxxxxxx>
 - *Date:* Sat, 30 Sep 2006 13:22:53 +0000
-

miguelojeda-2.6.18-add-lcd-display-support.patch

Changelog (newer first)

V6 – Fixed mistakes found by Andrew Morton

- added locking
- removed "add const to class_create" part, as it is already accepted in 2.6.18-git10
- removed PRINT_PREFIX macro, use KERN_... instead
- reverted bitops to standard |= and &=
- added const to fop's static struct
- added error code checking for class_device_create()
- removed "if(foo > maxfoo) foo=maxfoo;", use min(foo,maxfoo) instead
- coding style:
- used #defined constants directly, not static constants = DEFINED_CONSTANT
- removed blank lines
- changed pointers declarations to "T *name"
- changed for's coding style
- added single spaces around arithmetic operators
- changed counter's nomenclature to i,j,k
- shorter function definitions
- removed newlines at some function calls
- udelay(2)

V5 – Fixed mistakes found by Joe Perches

- coding style
- const, spelling mistake

V4 – Fixed mistakes found by Pavel Machek

- no zero initialization
- GPL version
- use bitop generic functions

V3 – Apply for 2.6.18

- added const to class_create (2.6.18 hasn't add-const-to-class_create.patch)

V2 – Fixed mistakes found by Greg KH

- coding style

[PATCH 2.6.18 V6] drivers: add lcd display support

V1 – Fixed mistakes found by Alexey Dobriyan

- coding style
- init() and exit() error management

V0 – Original patch

Brief

- Adds const to class_create (2.6.18 hasn't add-const-to-class_create.patch)
- Adds a "lcddisplay" class for registering LCD devices.
- Adds support for the ks0108 LCD Controller as a device driver.
(uses lcddisplay class and parport interface)
- Adds support for the cfag12864b LCD Display as a device driver.
(uses ks0108 device driver)
- Adds a new ioctl() magic number/range (0xFF) in the list for the cfag12864b device.
- Adds the usual Documentation, includes, Makefiles, Kconfigs, MAINTAINERS, CREDITS...

Patched files Index

```
patching file drivers/Kconfig
patching file drivers/lcddisplay/cfag12864b.c
patching file drivers/lcddisplay/cfag12864b_image.h
patching file drivers/lcddisplay/Kconfig
patching file drivers/lcddisplay/ks0108.c
patching file drivers/lcddisplay/lcddisplay.c
patching file drivers/lcddisplay/Makefile
patching file drivers/Makefile
patching file include/linux/cfag12864b.h
patching file include/linux/ks0108.h
patching file include/linux/lcddisplay.h
patching file Documentation/lcddisplay/cfag12864b
patching file Documentation/lcddisplay/lcddisplay
patching file Documentation/ioctl-number.txt
patching file CREDITS
patching file MAINTAINERS
```

Please review.

miguelojeda-2.6.18-add-lcd-display-support.patch

Signed-off-by: Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>

```
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/Kconfig linux-2.6.18/drivers/Kconfig
--- linux-2.6.18-vanilla/drivers/Kconfig 2006-09-20 03:42:06.000000000 +0000
+++ linux-2.6.18/drivers/Kconfig 2006-09-27 19:15:12.000000000 +0000
```

[PATCH 2.6.18 V6] drivers: add lcd display support

[PATCH 2.6.18 V6] drivers: add lcd display support

```
@@ -74,4 +74,6 @@ source "drivers rtc/Kconfig"

source "drivers/dma/Kconfig"

+source "drivers/lcddisplay/Kconfig"
+
+endmenu
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/lcddisplay/cfag12864b.c
linux-2.6.18/drivers/lcddisplay/cfag12864b.c
--- linux-2.6.18-vanilla/drivers/lcddisplay/cfag12864b.c 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/lcddisplay/cfag12864b.c 2006-09-30 10:56:32.000000000 +0000
@@ -0,0 +1,592 @@
+/*
+ * Filename: cfag12864b.c
+ * Version: 0.1.0
+ * Description: cfag12864b LCD Display Driver
+ * License: GPLv2 or later
+ * Depends: lcddisplay ks0108
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ * Date: 2006-09-28
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
+ */
+
+#include <linux/init.h>
+#include <linux/module.h>
+#include <linux/kernel.h>
+#include <linux/fs.h>
+#include <linux/cdev.h>
+#include <linux/device.h>
+#include <linux/lcddisplay.h>
+#include <linux/ks0108.h>
+#include <linux/cfag12864b.h>
+#include <asm/uaccess.h>
+
+#define CFAG12864B_NAME "cfag12864b"
+
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```

+/*
+ * Device
+ */
+
+static const unsigned int cfag12864b_firstminor;
+static const unsigned int cfag12864b_ndevices = 1;
+static int cfag12864b_major;
+static int cfag12864b_minor;
+static dev_t cfag12864b_device;
+struct cdev cfag12864b_chardevice;
+DECLARE_MUTEX(cfag12864b_mutex);
+
+/*
+ * cfag12864b Commands
+ */
+
+#define bit(n) (((unsigned char)1)<<(n))
+
+static unsigned char cfag12864b_state;
+
+static void cfag12864b_set(void)
+{
+ ks0108_writecontrol(cfag12864b_state);
+}
+
+static void cfag12864b_setbit(unsigned char state, unsigned char n)
+{
+ if (state)
+ cfag12864b_state |= bit(n);
+ else
+ cfag12864b_state &= ~bit(n);
+ cfag12864b_set();
+}
+
+static void cfag12864b_e(unsigned char state)
+{
+ cfag12864b_setbit(state, 0);
+}
+
+static void cfag12864b_cs1(unsigned char state)
+{
+ cfag12864b_setbit(state, 2);
+}
+
+static void cfag12864b_cs2(unsigned char state)
+{
+ cfag12864b_setbit(state, 1);
+}
+
+static void cfag12864b_di(unsigned char state)
+{
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ cfag12864b_setbit(state, 3);
+}
+
+static void cfag12864b_setcontrollers(unsigned char first, unsigned char second)
+{
+ if (first)
+ cfag12864b_cs1(0);
+ else
+ cfag12864b_cs1(1);
+
+ if (second)
+ cfag12864b_cs2(0);
+ else
+ cfag12864b_cs2(1);
+}
+
+static void cfag12864b_controller(unsigned char which)
+{
+ if (which == 0)
+ cfag12864b_setcontrollers(1, 0);
+ else if (which == 1)
+ cfag12864b_setcontrollers(0, 1);
+}
+
+static void cfag12864b_displaystate(unsigned char state)
+{
+ cfag12864b_di(0);
+ cfag12864b_e(1);
+ ks0108_displaystate(state);
+ cfag12864b_e(0);
+}
+
+static void cfag12864b_address(unsigned char address)
+{
+ cfag12864b_di(0);
+ cfag12864b_e(1);
+ ks0108_address(address);
+ cfag12864b_e(0);
+}
+
+static void cfag12864b_page(unsigned char page)
+{
+ cfag12864b_di(0);
+ cfag12864b_e(1);
+ ks0108_page(page);
+ cfag12864b_e(0);
+}
+
+static void cfag12864b_startline(unsigned char startline)
+{
+ cfag12864b_di(0);
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ cfag12864b_e(1);
+ ks0108_startline(startline);
+ cfag12864b_e(0);
+}
+
+static void cfag12864b_writebyte(unsigned char byte)
+{
+ cfag12864b_di(1);
+ cfag12864b_e(1);
+ ks0108_writedata(byte);
+ cfag12864b_e(0);
+}
+
+static void cfag12864b_nop(void)
+{
+ cfag12864b_startline(0);
+}
+
+/*
+ * Auxiliary
+ */
+
+static void normalizeoffset(unsigned int * offset)
+{
+ if (*offset >= CFAG12864B_PAGES*CFAG12864B_ADDRESSES)
+ *offset -= CFAG12864B_PAGES*CFAG12864B_ADDRESSES;
+}
+
+static unsigned char calcaddress(unsigned int offset)
+{
+ normalizeoffset(&offset);
+ return offset%CFAG12864B_ADDRESSES;
+}
+
+static unsigned char calcontroller(unsigned int offset)
+{
+ if (offset < CFAG12864B_PAGES*CFAG12864B_ADDRESSES)
+ return 0;
+ return 1;
+}
+
+static unsigned char calcpage(unsigned int offset)
+{
+ normalizeoffset(&offset);
+ return offset/CFAG12864B_ADDRESSES;
+}
+
+/*
+ * cfag12864b Internal Commands (don't lock)
+ */
+
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+void cfag12864b_on_nolock(void)
+{
+ cfag12864b_setcontrollers(1, 1);
+ cfag12864b_displaystate(1);
+}
+
+void cfag12864b_off_nolock(void)
+{
+ cfag12864b_setcontrollers(1, 1);
+ cfag12864b_displaystate(0);
+}
+
+void cfag12864b_clear_nolock(void)
+{
+ unsigned char i,j;
+
+ cfag12864b_setcontrollers(1, 1);
+ for (i = 0; i < CFAG12864B_PAGES; i++) {
+ cfag12864b_page(i);
+ cfag12864b_address(0);
+ for (j = 0; j < CFAG12864B_ADDRESSES; j++)
+ cfag12864b_writebyte(0);
+ }
+}
+
+void cfag12864b_write_nolock(unsigned short offset, const unsigned char *buffer,
+ unsigned short count)
+{
+ unsigned short i;
+
+ /* Invalid values: They get updated at the first cycle */
+ unsigned char controller = 0xFF;
+ unsigned char page = 0xFF;
+ unsigned char address = 0xFF;
+
+ unsigned char tmpcontroller, tmppage, tmpaddress;
+
+ if (offset > CFAG12864B_SIZE)
+ return;
+ if (count + offset > CFAG12864B_SIZE)
+ count = CFAG12864B_SIZE - offset;
+
+ for (i = 0; i < count; i++, offset++, address++) {
+ tmpcontroller = calcontroller(offset);
+ tmppage = calcpage(offset);
+ tmpaddress = calcaddress(offset);
+
+ if (controller != tmpcontroller) {
+ controller = tmpcontroller;
+ cfag12864b_controller(controller);
+ cfag12864b_nop();

```

```

+ }
+ if (page != tmppage) {
+ page = tmppage;
+ cfag12864b_page(page);
+ cfag12864b_nop();
+ }
+
+ /* Safe method, still quick */
+ cfag12864b_address(tmpaddress);
+ cfag12864b_nop();
+
+ /* Dummy */
+ cfag12864b_nop();
+
+ cfag12864b_writebyte(buffer[i]);
+ }
+}
+
+void cfag12864b_format_nolock(unsigned char *src)
+{
+ unsigned short i,j,k,n;
+ unsigned char *dest;
+
+ dest = kmalloc(sizeof(unsigned char) * CFAG12864B_SIZE, GFP_KERNEL);
+ if (dest == NULL) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "format: ERROR: "
+ "can't alloc memory %i bytes\n",
+ sizeof(unsigned char) * CFAG12864B_SIZE);
+ return;
+ }
+
+ for (i = 0; i < CFAG12864B_CONTROLLERS; i++)
+ for (j = 0; j < CFAG12864B_PAGES; j++)
+ for (k = 0; k < CFAG12864B_ADDRESSES; k++) {
+ dest[(i * CFAG12864B_PAGES + j) * CFAG12864B_ADDRESSES + k] = 0;
+ for (n=0; n < 8; n++)
+ if (src[i * CFAG12864B_ADDRESSES + k + (j * 8 + n) * CFAG12864B_WIDTH])
+ dest[(i * CFAG12864B_PAGES + j) * CFAG12864B_ADDRESSES + k] |= bit(n);
+ }
+
+ cfag12864b_write_nolock(0, dest, CFAG12864B_SIZE);
+
+ kfree(dest);
+}
+
+/*
+ * cfag12864b Exported Commands (do lock)
+ */
+
+void cfag12864b_on(void)
+{

```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ if(down_interruptible(&cfag12864b_mutex))
+ return;
+
+ cfag12864b_on_nolock();
+
+ up(&cfag12864b_mutex);
+}
+
+void cfag12864b_off(void)
+{
+ if(down_interruptible(&cfag12864b_mutex))
+ return;
+
+ cfag12864b_off_nolock();
+
+ up(&cfag12864b_mutex);
+}
+
+void cfag12864b_clear(void)
+{
+ if(down_interruptible(&cfag12864b_mutex))
+ return;
+
+ cfag12864b_clear_nolock();
+
+ up(&cfag12864b_mutex);
+}
+
+void cfag12864b_write(unsigned short offset, const unsigned char *buffer,
+ unsigned short count)
+{
+ if(down_interruptible(&cfag12864b_mutex))
+ return;
+
+ cfag12864b_write_nolock(offset,buffer,count);
+
+ up(&cfag12864b_mutex);
+}
+
+void cfag12864b_format(unsigned char *src)
+{
+ if(down_interruptible(&cfag12864b_mutex))
+ return;
+
+ cfag12864b_format_nolock(src);
+
+ up(&cfag12864b_mutex);
+}
+
+EXPORT_SYMBOL_GPL(cfag12864b_on);
+EXPORT_SYMBOL_GPL(cfag12864b_off);
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+EXPORT_SYMBOL_GPL(cfag12864b_clear);
+EXPORT_SYMBOL_GPL(cfag12864b_write);
+EXPORT_SYMBOL_GPL(cfag12864b_format);
+
+/*
+ * cfag12864b ioctls (don't lock because ioctl fop do)
+ */
+
+static int cfag12864b_fpioctlformat(void __user * arg)
+{
+ int result;
+ int ret = -ENOTTY;
+ unsigned char *tmpbuffer;
+
+ tmpbuffer = kmalloc(
+ sizeof(unsigned char)*CFAG12864B_MATRIXSIZE,GFP_KERNEL);
+ if (tmpbuffer == NULL) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "FOP ioctl: ERROR: "
+ "can't alloc memory %i bytes\n",
+ sizeof(unsigned char)*CFAG12864B_MATRIXSIZE);
+ goto none;
+ }
+
+ result = copy_from_user(tmpbuffer, arg,
+ sizeof(unsigned char) * CFAG12864B_MATRIXSIZE);
+ if (result != 0) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "FOP ioctl: ERROR: "
+ "can't copy memory from user\n");
+ goto bufferallocated;
+ }
+
+
+ cfag12864b_format_nolock(tmpbuffer);
+
+
+ ret = 0;
+
+bufferallocated:
+ kfree(tmpbuffer);
+
+none:
+ return ret;
+}
+
+/*
+ * cfag12864b_fops (do lock)
+ */
+
+static loff_t cfag12864b_fopseek(struct file *filp, loff_t offset, int whence)
+{
+ loff_t ret = -EINVAL;
+
+ if(down_interruptible(&cfag12864b_mutex))
```

```

+ return -ERESTARTSYS;
+
+ switch(whence) {
+ case SEEK_SET:
+ ret = offset;
+ break;
+ case SEEK_CUR:
+ ret = filp->f_pos + offset;
+ break;
+ case SEEK_END:
+ ret = CFAG12864B_SIZE + offset;
+ break;
+ }
+
+ if (ret < 0) {
+ ret = -EINVAL;
+ goto none;
+ }
+
+ filp->f_pos = ret;
+
+none:
+ up(&cfag12864b_mutex);
+ return ret;
+}
+
+
+static ssize_t cfag12864b_fopwrite(struct file *filp,
+ const char __user *buffer, size_t count, loff_t *offset)
+{
+ int ret = -EINVAL;
+ int result;
+ unsigned char *tmpbuffer;
+
+ if(down_interruptible(&cfag12864b_mutex))
+ return -ERESTARTSYS;
+
+ if (*offset > CFAG12864B_SIZE) {
+ ret = 0;
+ goto none;
+ }
+ if (*offset + count > CFAG12864B_SIZE)
+ count = CFAG12864B_SIZE-*offset;
+
+ tmpbuffer = kmalloc(count, GFP_KERNEL);
+ if (tmpbuffer == NULL) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "FOP write: ERROR: "
+ "can't alloc memory %i bytes\n",count);
+ ret = -ENOMEM;
+ goto none;
+ }

```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+
+ result = copy_from_user(tmpbuffer, buffer, count);
+ if (result != 0) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "FOP write: ERROR: "
+ "can't copy memory from user\n");
+ ret = -EFAULT;
+ goto bufferallocated;
+ }
+
+ cfag12864b_write_nolock(*offset, tmpbuffer, count);
+
+ *offset += count;
+ ret = count;
+
+bufferallocated:
+ kfree(tmpbuffer);
+
+none:
+ up(&cfag12864b_mutex);
+ return ret;
+}
+
+static int cfag12864b_fopioctl(struct inode * inode, struct file * filp,
+ unsigned int cmd, unsigned long arg)
+{
+ int ret = -ENOTTY;
+
+ if(down_interruptible(&cfag12864b_mutex))
+ return -ERESTARTSYS;
+
+ if (_IOC_TYPE(cmd) != CFAG12864B_IOC_MAGIC)
+ goto none;
+ if (_IOC_NR(cmd) > CFAG12864B_IOC_MAXNR)
+ goto none;
+
+ switch(cmd) {
+ case CFAG12864B_IOCON:
+ cfag12864b_on_nolock();
+ ret = 0;
+ break;
+ case CFAG12864B_IOCOFF:
+ cfag12864b_off_nolock();
+ ret = 0;
+ break;
+ case CFAG12864B_IOCCLEAR:
+ cfag12864b_clear_nolock();
+ ret = 0;
+ break;
+ case CFAG12864B_IOCFORMAT:
+ ret = cfag12864b_fopioctlformat((void __user *)arg);
+ }
+ }
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+
+none:
+ up(&cfag12864b_mutex);
+ return ret;
+}
+
+static const struct file_operations cfag12864b_fops =
+{
+ .owner = THIS_MODULE,
+ .llseek = cfag12864b_fopseek,
+ .write = cfag12864b_fopwrite,
+ .ioctl = cfag12864b_fopioctl,
+};
+
+/*
+ * Module Init & Exit
+ */
+
+static int __init cfag12864b_init(void)
+{
+
+ #include "cfag12864b_image.h"
+
+ int result;
+ int ret = -EINVAL;
+
+ result = alloc_chrdev_region(&cfag12864b_device, cfag12864b_firstminor,
+ cfag12864b_ndevices, CFAG12864B_NAME);
+ if (result < 0) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "ERROR: "
+ "can't alloc the char device region\n");
+ ret = result;
+ goto none;
+ }
+
+ cfag12864b_major = MAJOR(cfag12864b_device);
+ cfag12864b_minor = cfag12864b_firstminor;
+ cfag12864b_device = MKDEV(cfag12864b_major, cfag12864b_minor);
+
+ cfag12864b_clear_nolock();
+ cfag12864b_on_nolock();
+ cfag12864b_write_nolock(0, cfag12864b_image, CFAG12864B_SIZE);
+
+ cdev_init(&cfag12864b_chardevice, &cfag12864b_fops);
+ cfag12864b_chardevice.owner = THIS_MODULE;
+ cfag12864b_chardevice.ops = &cfag12864b_fops;
+ result = cdev_add(&cfag12864b_chardevice, cfag12864b_device,
+ cfag12864b_ndevices);
+ if (result < 0) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "ERROR: "
+ "unable to add a new char device\n");

```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ ret = result;
+ goto regionallocated;
+ }
+
+ if(class_device_create(lcddisplay_class, NULL, cfag12864b_device, NULL,
+ "cfag12864b%d", cfag12864b_minor) == NULL) {
+ printk(KERN_ERR CFAG12864B_NAME ": " "ERROR: "
+ "unable to create a device class\n");
+ ret = -EINVAL;
+ goto cdevadded;
+ }
+
+ printk(KERN_INFO CFAG12864B_NAME ": " "Initied\n");
+
+ return 0;
+
+cdevadded:
+ cdev_del(&cfag12864b_chardevice);
+
+regionallocated:
+ unregister_chrdev_region(cfag12864b_device, cfag12864b_ndevices);
+
+none:
+ return ret;
+}
+
+static void __exit cfag12864b_exit(void)
+{
+ cfag12864b_off_nolock();
+
+ class_device_destroy(lcddisplay_class, cfag12864b_device);
+ cdev_del(&cfag12864b_chardevice);
+ unregister_chrdev_region(cfag12864b_device, cfag12864b_ndevices);
+
+ printk(KERN_INFO CFAG12864B_NAME ": " "Exited\n");
+}
+
+module_init(cfag12864b_init);
+module_exit(cfag12864b_exit);
+
+MODULE_LICENSE("GPL");
+MODULE_AUTHOR("Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>");
+MODULE_DESCRIPTION("cfag12864b");
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/lcddisplay/cfag12864b_image.h
linux-2.6.18/drivers/lcddisplay/cfag12864b_image.h
--- linux-2.6.18-vanilla/drivers/lcddisplay/cfag12864b_image.h 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/lcddisplay/cfag12864b_image.h 2006-09-28 19:59:11.000000000 +0000
@@ -0,0 +1,95 @@
+/*
+ * Filename: cfag12864b_image.h
+ * Version: 0.1.0
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ * Description: cfag12864b LCD Display Driver Startup Image
+ * License: GPLv2 or later
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>
+ * Date: 2006-09-28
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
+ *
+ */
+
+#ifndef _CFAG12864B_IMAGE_H_
+#define _CFAG12864B_IMAGE_H_
+
+const unsigned char cfag12864b_image[] = {
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0x00,0xE0,0xF8,0xFC,0xFE,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFE,0xCE,
+0xEE,0xFC,0xFC,0xF8,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0x00,0xFF,0x87,0xFB,0x69,0x23,0x3F,0x3F,0x3F,0x07,0x21,0x79,0xF9,0xF1,
+0x83,0xFF,0xFF,0xFF,0xFF,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0x06,0x18,0x30,
+0x18,0x06,0xFF,0x00,0x00,0xFD,0x00,0x00,0x78,0xCC,0x84,0xCC,0xFC,0x00,0x00,0x7C,
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0xC0,0xFF,0x3D,0x38,0xF0,0xD0,0xD0,0xD0,0xD0,0x48,0x68,0x24,0x14,0x14,
+0x00,0x3F,0xFF,0xF7,0xEF,0xEF,0xFC,0xF0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,0x02,0x02,0x01,0x00,0x00,0x00,
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xE0,0x30,0xF8,
+0xFE,0x7F,0x03,0x00,0x00,0x00,0x00,0x01,0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0x00,0x03,0x1F,0x7F,0xFF,0xFF,0xDF,0x3F,0xFE,0xF8,0xF0,0xE0,0x80,0x00,
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1E,0x33,0x21,0x21,
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xF0,0xFC,0xFF,0xFF,0xFF,0x07,
+0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x1F,0xFF,0xFF,0xFC,0x03,0xFF,0xFF,0xFF,0xFE,
+0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF0,0x00,0x00,0x00,
+0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x7E,0x0F,0x0F,0x0F,0x1B,0x37,0xE7,0xC0,
+0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
+0x00,0x00,0x00,0x00,0xE0,0x00,0x10,0xFB,0xFF,0xFD,0xFF,0xFB,0xFF,0x7F,0x3F,
+0x5F,0x86,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x08,0x08,0x08,
+0x00,0x1E,0xF2,0x02,0x02,0x02,0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
```


[PATCH 2.6.18 V6] drivers: add lcd display support

```
+#
+# Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+#
+
+menu "LCD Display support"
+
+config LCDDISPLAY
+ tristate "LCD Display support"
+ default n
+ ---help---
+ If you have a LCD display, say Y.
+
+ To compile this as a module, choose M here:
+ module will be called lccdisplay.
+
+ Most LCD drivers use a I/O port (like the parallel port)
+ so you will need to say Y or M at them if you want to see
+ more options in this menu.
+
+ If unsure, say N.
+
+ Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+
+comment "Parallel port dependent:"
+
+config KS0108
+ tristate "KS0108 LCD Controller"
+ depends on LCDDISPLAY && PARPORT
+ default n
+ ---help---
+ If you have a LCD display controlled by one or more KS0108
+ controllers, say Y. You will need also another more specific
+ driver for your LCD.
+
+ Depends on Parallel Port support. If you say Y at
+ parport, you will be able to compile this as a module (M)
+ and built-in as well (Y). If you said M at parport,
+ you will be able only to compile this as a module (M).
+
+ To compile this as a module, choose M here:
+ module will be called ks0108.
+
+ If unsure, say N.
+
+ Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+
+config KS0108_PORT
+ hex "Parallel port where the LCD is connected to"
+ depends on KS0108
+ default 0x378
+ ---help---
```

[PATCH 2.6.18 V6] drivers: add lcd display support

- + The address of the parallel port where the LCD is connected to.
- +
- + The first standard parallel port address is 0x378.
- + The second standard parallel port address is 0x278.
- + The third standard parallel port address is 0x3BC.
- +
- + You can specify a different address if you need.
- +
- + If you don't know what I'm talking about, load the parport module,
- + and execute "dmesg". You can see there how many parallel ports
- + where detected and which address everyone has.
- +
- + Usually you only need to use 0x378.
- +
- + If you compile this as a module, you can still override this
- + using the module parameters.
- +
- +config KS0108_DELAY
- + int "Delay between each control writing (microseconds)"
- + depends on KS0108
- + default "2"
- + ----help----
- + Amount of time the ks0108 should wait between each control write
- + to the parallel port.
- +
- + If your driver seems to miss random writings, increment this.
- +
- + If you don't know what I'm talking about, ignore it.
- +
- + If you compile this as a module, you can still override this
- + using the module parameters.
- +
- +config CFAG12864B
- + tristate "CFAG12864B LCD Display"
- + depends on KS0108
- + default n
- + ----help----
- + If you have a Crystalfontz 128x64 2-color LCD display,
- + cfag12864b Series, say Y. You also need the ks0108 LCD
- + Controller driver.
- +
- + For help about how to wire your LCD to the parallel port,
- + check this image: [http://www.skippari.net/lcd/sekalaista](http://www.skippari.net/lcd/sekalaista/crystalfontz_cfag12864B-TMI-V.png)
- + /crystalfontz_cfag12864B-TMI-V.png
- +
- + To compile this as a module, choose M here:
- + module will be called cfag12864b.
- +
- + If unsure, say N.
- +
- + Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+
+endmenu
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/lcddisplay/ks0108.c
linux-2.6.18/drivers/lcddisplay/ks0108.c
--- linux-2.6.18-vanilla/drivers/lcddisplay/ks0108.c 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/lcddisplay/ks0108.c 2006-09-30 12:41:31.000000000 +0000
@@ -0,0 +1,160 @@
+/*
+ * Filename: ks0108.c
+ * Version: 0.1.0
+ * Description: ks0108 LCD Display Controller Driver
+ * License: GPLv2 or later
+ * Depends: parport
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ * Date: 2006-09-28
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
+ */
+
+#include <linux/init.h>
+#include <linux/module.h>
+#include <linux/kernel.h>
+#include <linux/fs.h>
+#include <linux/delay.h>
+#include <linux/parport.h>
+#include <linux/ks0108.h>
+#include <asm/io.h>
+#include <asm/uaccess.h>
+
+#define KS0108_NAME "ks0108"
+
+/*
+ * Module Parameters
+ */
+
+static unsigned int ks0108_port = CONFIG_KS0108_PORT;
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+module_param(ks0108_port, uint, S_IRUGO);
+
+static unsigned int ks0108_delay = CONFIG_KS0108_DELAY;
+module_param(ks0108_delay, uint, S_IRUGO);
+
+/*
+ * Device
+ */
+
+static struct parport *ks0108_parport;
+static struct pardevice *ks0108_pardevice;
+
+/*
+ * ks0108 Exported cmds (don't lock)
+ *
+ * you _should_ lock in the top driver, so
+ * this functions _should not_ get race conditions in any way.
+ * Locking for each byte here would be so slow and useless.
+ */
+
+#define bit(n) (((unsigned char)1)<<(n))
+
+void ks0108_writedata(unsigned char byte)
+{
+ parport_write_data(ks0108_parport, byte);
+}
+
+void ks0108_writecontrol(unsigned char byte)
+{
+ udelay(ks0108_delay);
+ parport_write_control(ks0108_parport, byte ^ (bit(0) | bit(1) | bit(3)));
+}
+
+void ks0108_displaystate(unsigned char state)
+{
+ ks0108_writedata((state ? bit(0) : 0) | bit(1) | bit(2) | bit(3) | bit(4) | bit(5));
+}
+
+void ks0108_startline(unsigned char startline)
+{
+ ks0108_writedata(min(startline,(unsigned char)63) | bit(6) | bit(7));
+}
+
+void ks0108_address(unsigned char address)
+{
+ ks0108_writedata(min(address,(unsigned char)63) | bit(6));
+}
+
+void ks0108_page(unsigned char page)
+{
+ ks0108_writedata(min(page,(unsigned char)7) | bit(3) | bit(4) | bit(5) | bit(7));
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+}
+
+EXPORT_SYMBOL_GPL(ks0108_writedata);
+EXPORT_SYMBOL_GPL(ks0108_writecontrol);
+EXPORT_SYMBOL_GPL(ks0108_displaystate);
+EXPORT_SYMBOL_GPL(ks0108_startline);
+EXPORT_SYMBOL_GPL(ks0108_address);
+EXPORT_SYMBOL_GPL(ks0108_page);
+
+/*
+ * Module Init & Exit
+ */
+
+static int __init ks0108_init(void)
+{
+ int result;
+ int ret = -EINVAL;
+
+ ks0108_parport = parport_find_base(ks0108_port);
+ if (ks0108_parport == NULL) {
+ printk(KERN_ERR KS0108_NAME ": " "ERROR: "
+ "parport didn't find %i port\n",ks0108_port);
+ goto none;
+ }
+
+ ks0108_pardevice = parport_register_device(ks0108_parport, KS0108_NAME,
+ NULL, NULL, NULL, PARPORT_DEV_EXCL, NULL);
+ if (ks0108_pardevice == NULL) {
+ printk(KERN_ERR KS0108_NAME ": " "ERROR: "
+ "parport didn't register new device\n");
+ goto none;
+ }
+
+ result = parport_claim(ks0108_pardevice);
+ if (result != 0) {
+ printk(KERN_ERR KS0108_NAME ": " "ERROR: "
+ "can't claim %i parport, maybe in use\n",ks0108_port);
+ ret = result;
+ goto registered;
+ }
+
+ printk(KERN_INFO KS0108_NAME ": " "Inited - ks0108_port=0x%X ks0108_delay=%i\n", ks0108_port,
+ ks0108_delay);
+ return 0;
+
+registered:
+ parport_unregister_device(ks0108_pardevice);
+
+none:
+ return ret;
+}
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+
+static void __exit ks0108_exit(void)
+{
+ parport_release(ks0108_pardevice);
+ parport_unregister_device(ks0108_pardevice);
+
+ printk(KERN_INFO KS0108_NAME ": " "Exited\n");
+}
+
+module_init(ks0108_init);
+module_exit(ks0108_exit);
+
+MODULE_LICENSE("GPL");
+MODULE_AUTHOR("Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>");
+MODULE_DESCRIPTION("ks0108");
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/lcddisplay/lcddisplay.c
linux-2.6.18/drivers/lcddisplay/lcddisplay.c
--- linux-2.6.18-vanilla/drivers/lcddisplay/lcddisplay.c 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/lcddisplay/lcddisplay.c 2006-09-28 19:59:18.000000000 +0000
@@ -0,0 +1,79 @@
+/*
+ * Filename: lcddisplay.c
+ * Version: 0.1.0
+ * Description: LCD Display Class
+ * License: GPLv2 or later
+ * Depends: -
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>
+ * Date: 2006-09-28
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
+ */
+
+#include <linux/init.h>
+#include <linux/module.h>
+#include <linux/kernel.h>
+#include <linux/fs.h>
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+#include <linux/device.h>
+#include <linux/lcddisplay.h>
+
+#define LCDDISPLAY_NAME "lcddisplay"
+
+/*
+ * Exported Display Data
+ */
+
+struct class *lcddisplay_class;
+EXPORT_SYMBOL_GPL(lcddisplay_class);
+
+/*
+ * Module Init & Exit
+ */
+
+static int __init lcddisplay_init(void)
+{
+ int ret = -EINVAL;
+
+ lcddisplay_class = class_create(THIS_MODULE, LCDDISPLAY_NAME);
+ if (IS_ERR(lcddisplay_class)) {
+ printk(KERN_ERR LCDDISPLAY_NAME ": " "ERROR: "
+ "can't create %s class\n", LCDDISPLAY_NAME);
+ goto none;
+ }
+
+ printk(KERN_INFO LCDDISPLAY_NAME ": " "Inited\n");
+
+ return 0;
+
+none:
+ return ret;
+}
+
+static void __exit lcddisplay_exit(void)
+{
+ class_destroy(lcddisplay_class);
+
+ printk(KERN_INFO LCDDISPLAY_NAME ": " "Exited\n");
+}
+
+module_init(lcddisplay_init);
+module_exit(lcddisplay_exit);
+
+MODULE_LICENSE("GPL");
+MODULE_AUTHOR("Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>");
+MODULE_DESCRIPTION("lcddisplay");
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/lcddisplay/Makefile
linux-2.6.18/drivers/lcddisplay/Makefile
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
--- linux-2.6.18-vanilla/drivers/lcddisplay/Makefile 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/lcddisplay/Makefile 2006-09-27 19:15:13.000000000 +0000
@@ -0,0 +1,7 @@
+#
+# Makefile for the kernel LCD Display device drivers.
+#
+
+obj-$(CONFIG_LCDDISPLAY) += lcddisplay.o
+obj-$(CONFIG_KS0108) += ks0108.o
+obj-$(CONFIG_CFAG12864B) += cfag12864b.o
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/Makefile linux-2.6.18/drivers/Makefile
--- linux-2.6.18-vanilla/drivers/Makefile 2006-09-20 03:42:06.000000000 +0000
+++ linux-2.6.18/drivers/Makefile 2006-09-27 19:15:13.000000000 +0000
@@ -76,3 +76,4 @@ obj-$(CONFIG_CRYPT0) += crypto/
obj-$(CONFIG_SUPERH) += sh/
obj-$(CONFIG_GENERIC_TIME) += clocksource/
obj-$(CONFIG_DMA_ENGINE) += dma/
+obj-$(CONFIG_LCDDISPLAY) += lcddisplay/
diff -uprN -X dontdiff linux-2.6.18-vanilla/include/linux/cfag12864b.h
linux-2.6.18/include/linux/cfag12864b.h
--- linux-2.6.18-vanilla/include/linux/cfag12864b.h 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/include/linux/cfag12864b.h 2006-09-28 19:59:57.000000000 +0000
@@ -0,0 +1,58 @@
+/*
+ * Filename: cfag12864b.h
+ * Version: 0.1.0
+ * Description: cfag12864b LCD Display Driver Header
+ * License: GPLv2 or later
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ * Date: 2006-09-28
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
+ */
+
+#ifndef _CFAG12864B_H_
+#define _CFAG12864B_H_
+
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+#include <linux/ioctl.h>
+
+#define CFAG12864B_WIDTH 128
+#define CFAG12864B_HEIGHT 64
+#define CFAG12864B_MATRIXSIZE CFAG12864B_WIDTH*CFAG12864B_HEIGHT
+
+#define CFAG12864B_CONTROLLERS 2
+#define CFAG12864B_PAGES 8
+#define CFAG12864B_ADDRESSES 64
+#define CFAG12864B_SIZE CFAG12864B_CONTROLLERS * \
+ CFAG12864B_PAGES * \
+ CFAG12864B_ADDRESSES
+
+#define CFAG12864B_IOC_MAGIC 0xFF
+#define CFAG12864B_IOC_MAXNR 0x03
+
+#define CFAG12864B_IOCOFF _IO(CFAG12864B_IOC_MAGIC,0)
+#define CFAG12864B_IOCON _IO(CFAG12864B_IOC_MAGIC,1)
+#define CFAG12864B_IOCLEAR _IO(CFAG12864B_IOC_MAGIC,2)
+#define CFAG12864B_IOCFORMAT _IOW(CFAG12864B_IOC_MAGIC,3,void *)
+
+extern void cfag12864b_on(void);
+extern void cfag12864b_off(void);
+extern void cfag12864b_clear(void);
+extern void cfag12864b_write(unsigned short offset,
+ const unsigned char *buffer, unsigned short count);
+extern void cfag12864b_format(unsigned char *src);
+
+#endif /* _CFAG12864B_H_ */
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/include/linux/ks0108.h linux-2.6.18/include/linux/ks0108.h
--- linux-2.6.18-vanilla/include/linux/ks0108.h 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/include/linux/ks0108.h 2006-09-28 19:59:51.000000000 +0000
@@ -0,0 +1,36 @@
+/*
+ * Filename: ks0108.h
+ * Version: 0.1.0
+ * Description: ks0108 LCD Display Controller Driver Header
+ * License: GPLv2 or later
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>
+ * Date: 2006-09-28
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
+ *
+ */
+
+#ifndef _KS0108_H_
+#define _KS0108_H_
+
+extern void ks0108_writedata(unsigned char byte);
+extern void ks0108_writecontrol(unsigned char byte);
+extern void ks0108_displaystate(unsigned char state);
+extern void ks0108_startline(unsigned char startline);
+extern void ks0108_address(unsigned char address);
+extern void ks0108_page(unsigned char page);
+
+#endif /* _KS0108_H_ */
diff -uprN -X dontdiff linux-2.6.18-vanilla/include/linux/lcddisplay.h
linux-2.6.18/include/linux/lcddisplay.h
--- linux-2.6.18-vanilla/include/linux/lcddisplay.h 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/include/linux/lcddisplay.h 2006-09-28 19:59:44.000000000 +0000
@@ -0,0 +1,34 @@
+/*
+ * Filename: lcddisplay.h
+ * Version: 0.1.0
+ * Description: Display Class Header
+ * License: GPLv2 or later
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ * Date: 2006-09-28
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the Free Software
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
+ *
+ */
+
+#ifndef _LCDDISPLAY_H_
+#define _LCDDISPLAY_H_
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+
+#include <linux/device.h>
+
+extern struct class *lcd_display_class;
+
+#endif /* _LCDDISPLAY_H_ */
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/Documentation/lcd_display/cfag12864b
linux-2.6.18/Documentation/lcd_display/cfag12864b
--- linux-2.6.18-vanilla/Documentation/lcd_display/cfag12864b 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/Documentation/lcd_display/cfag12864b 2006-09-30 12:20:28.000000000 +0000
@@ -0,0 +1,371 @@
+
+=====
+ cfag12864b Driver Documentation
+=====
+
+License: GPL
+Author & Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+Date: 2006-09-30
+
+
+
+-----
+0. INDEX
+-----
+
+1. DEVICE INFORMATION
+2. WIRING
+3. USER-SPACE PROGRAMMING
+3.1. ioctl and a 128*64 boolean matrix
+3.2. Direct writing
+4. USEFUL FILES
+4.1. cfag12864b.h
+4.2. bmpwriter.h
+
+
+
+-----
+1. DEVICE INFORMATION
+-----
+
+Manufacturer: Crystalfontz
+Webpage: http://www.crystalfontz.com
+Device Webpage: http://www.crystalfontz.com/products/12864b/
+Type: LCD Display
+Width: 128
+Height: 64
+Colors: 2
+Controller: ks0108
+Controllers: 2
+Pages: 8 each controller
```

+Addresses: 64 each page

+

+

+

+-----

+2. WIRING

+-----

+

+The cfag12864b LCD Display Series don't have a official wiring.

+

+The common wiring is done to the parallel port:

+

+http://www.skippari.net/lcd/sekalaista/crystalfontz_cfag12864B-TMI-V.png

+

+You can get help at Crystalfontz and LCDInfo forums.

+

+

+

+-----

+3. USER-SPACE PROGRAMMING

+-----

+

+Include a copy of the provided header:

+

+ #include "cfag12864b.h"

+

+Open the device for writing, /dev/cfag12864b0:

+

+ int fd = open("/dev/cfag12864b0",O_WRONLY);

+

+Then use simple ioctl calls to control it:

+

+ ioctl(fdisplay,CFAG12864B_IOCOFF); /* Turn off (don't clear) */

+ ioctl(fdisplay,CFAG12864B_IOCON); /* Turn on */

+ ioctl(fdisplay,CFAG12864B_IOCCLLEAR); /* Clear the display */

+

+For writing to the display, you have two options:

+

+

+3.1. ioctl & 128*64 boolean matrix

+-----

+

+This method is easier, but you have to update the entire display

+each time you want to change it.

+

+Note:

+

+ CFAG12864B_FORMATSIZE ==

+ CFAG12864B_WIDTH * CFAG12864B_HEIGHT ==

+ 128 * 64

+

```
+Declare the matrix and other one:
+
+ unsigned char MyDrawing[CFAG12864B_WIDTH][CFAG12864B_HEIGHT];
+
+ unsigned char Buffer[CFAG12864B_FORMATSIZE];
+
+Copy the 2d matrix to the buffer , like:
+
+ for(i = 0; i < CFAG12864B_WIDTH; i++)
+ for(j = 0; j < CFAG12864B_HEIGHT; j++)
+ Buffer[i + j * CFAG12864B_WIDTH] = MyDrawing[i][j];
+
+Call the ioctl:
+
+ ioctl(fdisplay, CFAG12864B_IOCTLFORMAT, Buffer);
+
+Voila! Your drawing should appear on the screen.
+
+
+
+3.2. Direct writing
+-----
+
+This methods allows you to change each byte of the device,
+so you can achieve a higher update rate updating only the pixels
+you are going to change.
+
+The device size is 1024 == CFAG12864B_SIZE.
+
+You can write and seek the device. The first 512 bytes write to
+the first k0108 controller (left display half) and the last 512 bytes
+write to the second ks0108 controller (right display half).
+
+Each controller is divided into 8 pages. Each page has 64 bytes.
+
+ Controller 0 Controller 1
+-----
+Page 0 |_____||_____||
+Page 1 |_____||_____||
+Page 2 |_____||_____||
+Page 3 |_____||_____||
+Page 4 |_____||_____||
+Page 5 |_____||_____||
+Page 6 |_____||_____||
+Page 7 |_____||_____||
+<---- 64 ---->
+
+You will understand how the device work executing some commands:
+
+# echo -n A > /dev/cfag12864b0
+# echo -n a > /dev/cfag12864b0
```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ # echo AAAAAA > /dev/cfag12864b0
+ # echo 000000 > /dev/cfag12864b0
+ # echo Hello world! > /dev/cfag12864b0
+ # echo Hello world! Hello world! > /dev/cfag12864b0
+
+After you understand it, code your functions to change specific bytes.
+
+Use write() and lseek() system calls, like:
+
+lseek(fddisplay, ipage * CFAG12864B_HEIGHT, SEEK_SET);
+lseek(fddisplay, icontroller * CFAG12864B_SIZE / 2, SEEK_SET);
+
+write(fddisplay, bufpage, CFAG12864B_HEIGHT);
+write(fddisplay, bufcontroller, CFAG12864B_SIZE / 2);
+write(fddisplay, bufdisplay, CFAG12864B_SIZE);
+
+-----
+4. USEFUL FILES
+-----
+
+
+4.1 cfag12864b.h
+-----
+
+You can use a copy of this header in your user-space programs.
+
+---
+/*
+ * Filename: cfag12864b.h
+ * Version: 0.1.0
+ * Description: cfag12864b LCD Display Driver Header for user-space apps
+ *
+ * Author: Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>
+ * Date: 2006-09-12
+ */
+
+#ifndef _CFAG12864B_H_
+#define _CFAG12864B_H_
+
+#include <sys/ioctl.h>
+
+#define CFAG12864B_WIDTH 128
+#define CFAG12864B_HEIGHT 64
+#define CFAG12864B_FORMATSIZE CFAG12864B_WIDTH*CFAG12864B_HEIGHT
+#define CFAG12864B_SIZE 1024
+
+#define CFAG12864B_IOC_MAGIC 0xFF
+
+#define CFAG12864B_IOCOFF_IO(CFAG12864B_IOC_MAGIC,0)
+#define CFAG12864B_IOCOFF_IO(CFAG12864B_IOC_MAGIC,1)
```


[PATCH 2.6.18 V6] drivers: add lcd display support

```
+{
+ const unsigned int Width = CFAG12864B_WIDTH;
+ const unsigned int Height = CFAG12864B_HEIGHT;
+ const unsigned int Bits = 8;
+
+ unsigned int Y,X,Bit;
+
+ for(Y=0; Y<Height; ++Y)
+ for(X=0; X<Width/Bits; ++X)
+ for(Bit=0; Bit<Bits; ++Bit)
+ _Dest[X * Bits + Bit + (Height - Y - 1) * Width] =
+ _Src[Y * Width / Bits + X] & Bit(Bits - Bit - 1) ? 0 : 1;
+}
+
+int main(int argc, char * argv[])
+{
+ const unsigned int Width = CFAG12864B_WIDTH;
+ const unsigned int Height = CFAG12864B_HEIGHT;
+ const unsigned int Size = CFAG12864B_SIZE;
+ const unsigned int BPP = 1;
+ const unsigned int HeaderSize = 0x3E;
+ const unsigned int BMPSize = BMP_SIZE;
+
+ unsigned char c;
+ unsigned int i,j;
+ union dword n;
+
+ unsigned char Buffer_BMP[BMP_SIZE];
+ unsigned char Buffer_Matrix[CFAG12864B_FORMATSIZE];
+
+ int fdisplay;
+ FILE * fbmp;
+
+ // Check args
+ if(argc!=3) {
+ printf("%s: Bad number of arguments. Expected 3\n",
+ argv[0]);
+ return -1;
+ }
+
+ // Open file
+ fbmp = fopen(argv[2],"rb");
+ if(fbmp==NULL) {
+ printf("%s: Can't open %s\n",argv[0], argv[2]);
+ return -2;
+ }
+
+ // Check file size
+ fseek(fbmp,0,SEEK_END);
+ i=ftell(fbmp);
+ if(i!=HeaderSize+Size) {
```

```

+ printf("%s: Bad file size. %i instead of %i\n",
+ argv[0], i, HeaderSize+Size);
+ fclose(fbmp);
+ return -3;
+ }
+
+ // Check both magic BMP bytes
+ fseek(fbmp,0,SEEK_SET);
+ c = fgetc(fbmp);
+ if(c!='B') {
+ printf("%s: Bad first magic byte. '%c' instead of 'B'\n",
+ argv[0], c);
+ fclose(fbmp);
+ return -4;
+ }
+ c = fgetc(fbmp);
+ if(c!='M') {
+ printf("%s: Bad second magic byte. '%c' instead of 'M'\n",
+ argv[0], c);
+ fclose(fbmp);
+ return -5;
+ }
+
+ // Check this is a 128x64 1-bpp BMP file
+ fseek(fbmp,0x12,SEEK_SET);
+ for(i=0; i<4; ++i)
+ n.u8[i] = fgetc(fbmp);
+ if(n.u32!=Width) {
+ printf("%s: Bad width. %i instead of %i\n",
+ argv[0], n.u32, Width);
+ fclose(fbmp);
+ return -6;
+ }
+ for(i=0; i<4; ++i)
+ n.u8[i] = fgetc(fbmp);
+ if(n.u32!=Height) {
+ printf("%s: Bad width. %i instead of %i\n",
+ argv[0], n.u32, Height);
+ fclose(fbmp);
+ return -7;
+ }
+ fseek(fbmp,0x1C,SEEK_SET);
+ c = fgetc(fbmp);
+ if(c!=BPP) {
+ printf("%s: Bad bpp. %i instead of %i\n",
+ argv[0], c, BPP);
+ fclose(fbmp);
+ return -8;
+ }
+
+ // Get bitmap data

```

[PATCH 2.6.18 V6] drivers: add lcd display support

```
+ fseek(fbmp,0x3E,SEEK_SET);
+ fread(Buffer_BMP,1,BMPSize,fbmp);
+ fclose(fbmp);
+
+ // Transform BMP data to 2D matrix
+ BMP2Format(Buffer_BMP,Buffer_Matrix);
+
+ // Open file
+ fdisplay = open(argv[1],O_WRONLY);
+ if(fdisplay < 0) {
+ printf("%s: Can't open %s\n", argv[0], argv[1]);
+ return -9;
+ }
+
+ // Send matrix
+ ioctl(fdisplay,CFAG12864B_IOCTLFORMAT,Buffer_Matrix);
+
+ // Close file
+ close(fdisplay);
+
+ return 0;
+}
+---
+
+
+EOF
diff -uprN -X dontdiff linux-2.6.18-vanilla/Documentation/lcddisplay/lcddisplay
linux-2.6.18/Documentation/lcddisplay/lcddisplay
--- linux-2.6.18-vanilla/Documentation/lcddisplay/lcddisplay 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/Documentation/lcddisplay/lcddisplay 2006-09-30 12:07:59.000000000 +0000
@@ -0,0 +1,37 @@
+ =====
+ LCD Display Drivers Documentation
+ =====
+
+License: GPL
+Author & Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+Date: 2006-09-30
+
+-----
+0. INDEX
+-----
+
+1. NEW DISPLAY DRIVERS
+2. GENERAL TIPS
+
+
+-----
+1. NEW DISPLAY DRIVERS
+-----
```

[PATCH 2.6.18 V6] drivers: add lcd display support

+
+Feel free to send me new display drivers. I will try to do my best.
+
+If you don't get any answer, send your patch directly to the linux-kernel ml.
+
+
+-----
+2. GENERAL TIPS
+-----
+
+- Divide your driver into the controller driver, like ks0108,
+ and the specific LCD display series driver, like cfag12864b.
+
+- Claim for your IO ports in the controller driver.
+
+EOF
diff -uprN -X dontdiff linux-2.6.18-vanilla/Documentation/ioctl-number.txt
linux-2.6.18/Documentation/ioctl-number.txt
--- linux-2.6.18-vanilla/Documentation/ioctl-number.txt 2006-09-20 03:42:06.000000000 +0000
+++ linux-2.6.18/Documentation/ioctl-number.txt 2006-09-27 19:15:13.000000000 +0000
@@ -191,3 +191,5 @@ Code Seq# Include File Comments
<mailto:aherrman@xxxxxxxxxx>
0xF3 00-3F video/sisfb.h sisfb (in development)
<mailto:thomas@xxxxxxxxxxxxxxxxxx>
+0xFF 00-1F linux/cfag12864b.h cfag12864b LCD Display Driver
+ <mailto:maxextreme@xxxxxxxxxx>
diff -uprN -X dontdiff linux-2.6.18-vanilla/CREDITS linux-2.6.18/CREDITS
--- linux-2.6.18-vanilla/CREDITS 2006-09-20 03:42:06.000000000 +0000
+++ linux-2.6.18/CREDITS 2006-09-27 19:15:13.000000000 +0000
@@ -2535,6 +2535,14 @@ S: Subiaco, 6008
S: Perth, Western Australia
S: Australia

+N: Miguel Ojeda Sandonis
+E: maxextreme@xxxxxxxxxx
+D: Author: LCD Display Drivers (ks0108, cfag12864b)
+D: Maintainer: LCD Display Drivers Tree (drivers/lcddisplay/*)
+S: C/ Mieses 20, 9-B
+S: Valladolid 47009
+S: Spain
+
N: Greg Page
E: gpage@xxxxxxxxxxxxxxxx
D: IPX development and support
diff -uprN -X dontdiff linux-2.6.18-vanilla/MAINTAINERS linux-2.6.18/MAINTAINERS
--- linux-2.6.18-vanilla/MAINTAINERS 2006-09-20 03:42:06.000000000 +0000
+++ linux-2.6.18/MAINTAINERS 2006-09-27 19:15:13.000000000 +0000
@@ -1707,6 +1707,11 @@ M: James.Bottomley@xxxxxxxxxxxxxxxxxxxxxxxxxx
L: linux-scsi@xxxxxxxxxxxxxxxxxx
S: Maintained

[PATCH 2.6.18 V6] drivers: add lcd display support

+LCD DISPLAY DRIVERS

+P: Miguel Ojeda Sandonis

+M: maxextreme@xxxxxxxxxx

+S: Maintained

+

LED SUBSYSTEM

P: Richard Purdie

M: rpurdie@xxxxxxxxxx

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>