

# [PATCH 2.6.18 V8] drivers: add LCD support

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-10/msg01342.html>

---

- *From:* Miguel Ojeda Sandonis <[maxextreme@xxxxxxxxx](mailto:maxextreme@xxxxxxxxx)>
  - *Date:* Wed, 4 Oct 2006 18:06:15 +0000
- 

miguelojeda-2.6.18-add-lcd-display-support.patch.v8

Andrew, I think this can be the final review.  
Nomenclature changed as requested. Please apply.

## Brief

-----

- Adds a "auxdisplay/" folder in "drivers/" for auxiliary display drivers.
- Adds a LCD 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 as a device driver.  
(uses ks0108 LCD Controller driver)
- Adds a new ioctl() magic number/range (0xFF) in the list for the cfag12864b device.
- Adds the usual Documentation, ABI, includes, Makefiles, Kconfigs, MAINTAINERS, CREDITS...

## Changelog (newer first)

-----

V8 - Nomenclature changed as discussed (mainly by Randy Dunlap and Stefan Richter; also Alan Cox, Alexander van Heukelum, Andrew James, Andreas Schwab...)

- Changed tree to a neutral "drivers/auxdisplay/"
- Changed references from "LCD Display" to "LCD"
- Other minor reviews

V7 - Fixed mistakes found by Randy Dunlap

- removed kcalloc/kfree calls, do it at init/exit
- added Documentation/ABI/testing/cfag12864b
- changed %i to %Zd at printk calls to avoid gcc warnings
- removed image startup
- coding style
- improved MODULE\_DESCRIPTIONs
- added parens to some macros
- changed KERN\_DEBUG to KERN\_INFO for "inited" and "exited" messages
- added single spaces around arithmetic operators
- added several spacebars

## [PATCH 2.6.18 V8] drivers: add LCD support

- fixed several spelling issues

### 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

- lots about coding style

### V1 - Fixed mistakes found by Alexey Dobriyan

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

### V0 - Original patch

### Patched files Index

---

#### CREDITS | 8

Documentation/ABI/testing/cfag12864b | 50 ++

Documentation/auxdisplay/auxdisplay | 37 +

Documentation/auxdisplay/cfag12864b | 404 ++++++

Documentation/auxdisplay/ks0108 | 59 ++

[PATCH 2.6.18 V8] drivers: add LCD support

Documentation/ioctl-number.txt | 2  
MAINTAINERS | 6  
drivers/Kconfig | 2  
drivers/Makefile | 1  
drivers/auxdisplay/Kconfig | 103 +++++  
drivers/auxdisplay/Makefile | 7  
drivers/auxdisplay/cfag12864b.c | 600 +++++  
drivers/auxdisplay/ks0108.c | 171 +++++  
drivers/auxdisplay/lcdclass.c | 79 +++  
include/linux/cfag12864b.h | 58 ++  
include/linux/ks0108.h | 36 +  
include/linux/lcdclass.h | 33 +  
17 files changed, 1656 insertions(+)

miguelojeda-2.6.18-add-lcd-display-support.patch.v8

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

---  
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 2006-10-04 17:10:27.000000000 +0000  
@@ -2535,6 +2535,14 @@ S: Subiaco, 6008  
S: Perth, Western Australia  
S: Australia

+N: Miguel Ojeda Sandonis  
+E: maxextreme@xxxxxxxx  
+D: Author: LCD and LCD Controller drivers (ks0108, cfag12864b)  
+D: Maintainer: Auxiliary display drivers tree (drivers/auxdisplay/\*)  
+S: C/ Mieses 20, 9-B  
+S: Valladolid 47009  
+S: Spain

+

N: Greg Page

E: gpage@xxxxxxxxxxxxxx

D: IPX development and support

diff -uprN -X dontdiff linux-2.6.18-vanilla/Documentation/ABI/testing/cfag12864b

linux-2.6.18/Documentation/ABI/testing/cfag12864b

--- linux-2.6.18-vanilla/Documentation/ABI/testing/cfag12864b 1970-01-01 00:00:00.000000000 +0000

+++ linux-2.6.18/Documentation/ABI/testing/cfag12864b 2006-10-04 16:52:21.000000000 +0000

@@ -0,0 +1,50 @@

+What: drivers/auxdisplay/cfag12864b driver

+Date: 2006-10-04

+KernelVersion: 2.6.18

+Contact: Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>

+Description:

+ The cfag12864b LCD driver defines two ways to communicate

+ with the lcd display.

+

+ 1. Use seek and write syscalls. Bytes written appear on

+ the screen without any formatting on the position pointed

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ by the file offset.
+
+ It is hardware dependent. It should be use to modify
+ specific display's pixels to achieve higher refreshing
+ rates.
+
+ 2. Use ioctl syscall. The magic number is 0xFF.
+ There are four ioctls:
+
+ 2.0. off _IO(0xFF,0)
+
+ Power off display.
+
+ It doesn't clear the display.
+ It doesn't stop the controllers.
+
+ 2.1. on _IO(0xFF,1)
+
+ Power on display.
+
+ 2.2. clear _IO(0xFF,2)
+
+ Clear the display.
+
+ 2.3. format _IOW(0xFF,3,void *)
+
+ Read the given buffer, transform it to the hardware
+ dependent format and show it on the screen.
+
+ The argument must point to a userspace buffer of
+ size 128*64 bytes (the display's size).
+
+ Each buffer's byte (unsigned) represent a pixel:
+ 0 = pixel will turn off
+ >0 = pixel will turn on
+
+ For more information and examples, see
+ Documentation/auxdisplay/cfag12864b
+
+Users:
diff -uprN -X dontdiff linux-2.6.18-vanilla/Documentation/auxdisplay/auxdisplay
linux-2.6.18/Documentation/auxdisplay/auxdisplay
--- linux-2.6.18-vanilla/Documentation/auxdisplay/auxdisplay 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/Documentation/auxdisplay/auxdisplay 2006-10-04 16:53:12.000000000 +0000
@@ -0,0 +1,37 @@
+ =====
+ Auxiliary Display Drivers Documentation
+ =====
+
+License: GPLv2
+Author & Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
```

[PATCH 2.6.18 V8] drivers: add LCD support

+Date: 2006-10-04

+

+-----

+0. INDEX

+-----

+

+ 1. NEW DISPLAY DRIVERS

+ 2. GENERAL TIPS

+

+

+

+-----

+1. NEW DISPLAY DRIVERS

+-----

+

+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 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/auxdisplay/cfag12864b

linux-2.6.18/Documentation/auxdisplay/cfag12864b

--- linux-2.6.18-vanilla/Documentation/auxdisplay/cfag12864b 1970-01-01 00:00:00.000000000 +0000

+++ linux-2.6.18/Documentation/auxdisplay/cfag12864b 2006-10-04 16:56:25.000000000 +0000

@@ -0,0 +1,404 @@

+ =====

+ cfag12864b LCD Driver Documentation

+ =====

+

+License: GPLv2

+Author & Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>

+Date: 2006-10-04

+

+

+

+-----

+0. INDEX

+-----

+

+ 1. DRIVER INFORMATION

[PATCH 2.6.18 V8] drivers: add LCD support

- + 2. DEVICE INFORMATION
- + 3. WIRING
- + 4. USER-SPACE PROGRAMMING
- + 4.1. ioctl and a 128\*64 boolean matrix
- + 4.2. Direct writing
- + 5. USEFUL FILES
- + 5.1. cfag12864b.h
- + 5.2. bmpwriter.h

+

+

+-----  
+1. DRIVER INFORMATION

+-----

+

+This driver support one cfag12864b display at time.

+

+

+-----  
+2. DEVICE INFORMATION

+-----

+

+Manufacturer: Crystalfontz

+Device Name: Crystalfontz 12864b LCD Series

+Device Code: cfag12864b

+Webpage: <http://www.crystalfontz.com>

+Device Webpage: <http://www.crystalfontz.com/products/12864b/>

+Type: LCD (Liquid Crystal Display)

+Width: 128

+Height: 64

+Colors: 2 (B/N)

+Controller: ks0108

+Controllers: 2

+Pages: 8 each controller

+Addresses: 64 each page

+

+

+

+-----

+3. WIRING

+-----

+

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

+

+The common wiring is done to the parallel port:

+

+[http://www.skippari.net/lcd/sekalaista/crystalfontz\\_cfag12864B-TMI-V.png](http://www.skippari.net/lcd/sekalaista/crystalfontz_cfag12864B-TMI-V.png)

+

+You can get help at Crystalfontz and LCDInfo forums.

+

+

+

```
+-----  
+4. USERSPACE PROGRAMMING  
+-----  
+  
+This interface is described briefly at Documentation/ABI/testing/cfag12864b  
+  
+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_IOCTL); /* Turn off (don't clear) */  
+ ioctl(fdisplay,CFAG12864B_IOCTL); /* Turn on */  
+ ioctl(fdisplay,CFAG12864B_IOCTL); /* Clear the display */  
+  
+For writing to the display, you have two options:  
+  
+  
+4.1. ioctl & 128*64 boolean matrix  
+-----  
+  
+This method is easier, but you have to update the entire display  
+each time you want to change it.  
+  
+Just calling the ioctl:  
+  
+ ioctl(fdisplay, CFAG12864B_IOCTL, Buffer);  
+  
+Your drawing should appear on the display.  
+  
+Note:  
+  
+ CFAG12864B_FORMATSIZE ==  
+ CFAG12864B_WIDTH * CFAG12864B_HEIGHT ==  
+ 128 * 64  
+  
+The buffer should be a 128*64 unsigned char array:  
+  
+ unsigned char Buffer[CFAG12864B_FORMATSIZE];  
+  
+To fill this buffer, you have several ways. Examples:  
+  
+a)  
+ Fill a bidimensional 128x64 buffer (each byte representing a pixel  
+ on the display), copy it to the first one and call the ioctl:  
+
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ unsigned char Buffer[CFAG12864B_FORMATSIZE];
+ unsigned char MyDrawing[CFAG12864B_WIDTH][CFAG12864B_HEIGHT];
+
+ /* Create drawing ... */
+
+ for(i = 0; i < CFAG12864B_WIDTH; i++)
+ for(j = 0; j < CFAG12864B_HEIGHT; j++)
+ Buffer[i + j * CFAG12864B_WIDTH] = MyDrawing[i][j];
+
+ ioctl(fdisplay, CFAG12864B_IOCTLFORMAT, Buffer);
+
+ b)
+ Create helper functions/macros to change a specific bit
+ in the final buffer. This way you don't need an extra buffer,
+ and it is faster. When you are ready, call the ioctl:
+
+ #define SET(x,y) Buffer[x + y * CFAG12864B_WIDTH] = 1;
+ #define UNSET(x,y) Buffer[x + y * CFAG12864B_WIDTH] = 0;
+
+ unsigned char Buffer[CFAG12864B_FORMATSIZE];
+
+ /* Create drawing ... */
+
+ ioctl(fdisplay, CFAG12864B_IOCTLFORMAT, Buffer);
+
+
+4.2. Direct writing
+-----
+
+ This method 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 | _____ | _____ |
```

## [PATCH 2.6.18 V8] drivers: add LCD support

```
+ <---- 64 ---->
+
+You will understand how the device work executing some commands:
+
+ # echo -n A > /dev/cfag12864b0
+ # echo -n a > /dev/cfag12864b0
+ # 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);
+
+
+-----
+5. USEFUL FILES
+-----
+
+
+5.1 cfag12864b.h
+-----
+
+You can use a copy of this header in your userspace 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: 2006-09-30
+ */
+
+#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))
```



[PATCH 2.6.18 V8] drivers: add LCD support

```
+
+#define Bit(n) ((unsigned char)(1<<(n)))
+
+void BMP2Format(
+ unsigned char _Src[BMP_SIZE],
+ unsigned char _Dest[CFAG12864B_FORMATSIZE])
+{
+ 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;
+ }
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ }
+
+ // 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",
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ argv[0], c, BPP);
+ fclose(fbmp);
+ return -8;
+ }
+
+ // Get bitmap data
+ 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/auxdisplay/ks0108
linux-2.6.18/Documentation/auxdisplay/ks0108
--- linux-2.6.18-vanilla/Documentation/auxdisplay/ks0108 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/Documentation/auxdisplay/ks0108 2006-10-04 17:04:30.000000000 +0000
@@ -0,0 +1,59 @@
+ =====
+ ks0108 LCD Controller Driver Documentation
+ =====
+
+ +License: GPLv2
+ +Author & Maintainer: Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ +Date: 2006-10-04
+
+
+
+ +-----
+ +0. INDEX
+ +-----
+
```

[PATCH 2.6.18 V8] drivers: add LCD support

+ 1. DRIVER INFORMATION  
+ 2. DEVICE INFORMATION  
+ 3. WIRING

+  
+

+-----  
+1. DRIVER INFORMATION

+-----  
+

+This driver support the ks0108 LCD controller.

+  
+

+-----  
+2. DEVICE INFORMATION

+-----  
+

+Manufacturer: Samsung

+Device Name: KS0108 LCD Controller

+Device Code: ks0108

+Webpage: -

+Device Webpage: -

+Type: LCD Controller (Liquid Crystal Display Controller)

+Width: 64

+Height: 64

+Colors: 2 (B/N)

+Pages: 8

+Addresses: 64 each page

+  
+

+  
+

+-----  
+3. WIRING

+-----  
+

+The driver supports data parallel port wiring.

+  
+

+If you aren't creating a LCD related hardware, you should check  
+your LCD specific wiring information in the same folder, not this one.

+  
+

+Wiring example of a cfag12864b LCD which has two ks0108 controllers:

+  
+

+[http://www.skippari.net/lcd/sekalaista/crystalfontz\\_cfag12864B-TMI-V.png](http://www.skippari.net/lcd/sekalaista/crystalfontz_cfag12864B-TMI-V.png)

+  
+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 2006-10-04 17:04:56.000000000 +0000

@@ -191,3 +191,5 @@ Code Seq# Include File Comments

<<mailto:aherrman@xxxxxxxxxx>>

[PATCH 2.6.18 V8] drivers: add LCD support

```
0xF3 00-3F video/sisfb.h sisfb (in development)
<mailto:thomas@xxxxxxxxxxxxxxxx>
+0xFF 00-1F linux/cfag12864b.h cfag12864b LCD driver
+ <mailto:maxextreme@xxxxxxxx>
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/auxdisplay/cfag12864b.c
linux-2.6.18/drivers/auxdisplay/cfag12864b.c
--- linux-2.6.18-vanilla/drivers/auxdisplay/cfag12864b.c 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/auxdisplay/cfag12864b.c 2006-10-04 16:50:28.000000000 +0000
@@ -0,0 +1,600 @@
+/*
+ * Filename: cfag12864b.c
+ * Version: 0.1.0
+ * Description: cfag12864b LCD driver
+ * License: GPLv2
+ * Depends: lcdclass ks0108
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>
+ * Date: 2006-10-04
+ *
+ * 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/lcdclass.h>
+#include <linux/ks0108.h>
+#include <linux/cfag12864b.h>
+#include <asm/uaccess.h>
+
+#define CFAG12864B_NAME "cfag12864b"
+
+/*
+ * Device
+ */
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+
+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;
+static unsigned char * cfag12864b_buffer;
+static unsigned char * cfag12864b_formatbuffer;
+DECLARE_MUTEX(cfag12864b_mutex);
+
+/*
+ * cfag12864b Commands
+ *
+ * E = Enable signal
+ * Everytime E switch from low to high,
+ * cfag12864b/ks0108 reads the command/data.
+ *
+ * CS1 = First ks0108controller.
+ * If high, the first ks0108 receives commands/data.
+ *
+ * CS2 = Second ks0108 controller
+ * If high, the second ks0108 receives commands/data.
+ *
+ * DI = Data/Instruction
+ * If low, cfag12864b will expect commands.
+ * If high, cfag12864b will expect data.
+ *
+ */
+
+#define bit(n) (((unsigned char)1)<<(n))
+
+#define CFAG12864B_BIT_E (0)
+#define CFAG12864B_BIT_CS1 (2)
+#define CFAG12864B_BIT_CS2 (1)
+#define CFAG12864B_BIT_DI (3)
+
+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();
+}
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+}  
+  
+static void cfag12864b_e(unsigned char state)  
+{  
+ cfag12864b_setbit(state, CFAG12864B_BIT_E);  
+}  
+  
+static void cfag12864b_cs1(unsigned char state)  
+{  
+ cfag12864b_setbit(state, CFAG12864B_BIT_CS1);  
+}  
+  
+static void cfag12864b_cs2(unsigned char state)  
+{  
+ cfag12864b_setbit(state, CFAG12864B_BIT_CS2);  
+}  
+  
+static void cfag12864b_di(unsigned char state)  
+{  
+ cfag12864b_setbit(state, CFAG12864B_BIT_DI);  
+}  
+  
+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);  
+}  
+
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+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);
+ 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;
+}
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+}
+
+static unsigned char calcontroller(unsigned int offset)
+{
+ if (offset < CFAG12864B_PAGES * CFAG12864B_ADDRESSES)
+ return 0;
+ return 1;
+}
+
+static unsigned char calpage(unsigned int offset)
+{
+ normalizeoffset(&offset);
+ return offset / CFAG12864B_ADDRESSES;
+}
+
+/*
+ * cfag12864b Internal Commands (don't lock)
+ */
+
+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;
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ 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 = calccontroller(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;
+
+ for (i = 0; i < CFAG12864B_CONTROLLERS; i++)
+ for (j = 0; j < CFAG12864B_PAGES; j++)
+ for (k = 0; k < CFAG12864B_ADDRESSES; k++) {
+ cfag12864b_buffer[(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])
+ cfag12864b_buffer[(i * CFAG12864B_PAGES + j) * CFAG12864B_ADDRESSES + k] |= bit(n);
+ }
+
+ cfag12864b_write_nolock(0, cfag12864b_buffer, CFAG12864B_SIZE);
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+}  
+  
+/*  
+ * cfag12864b Exported Commands (do lock)  
+ */  
+  
+void cfag12864b_on(void)  
+{  
+ 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;
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+
+ cfag12864b_format_nolock(src);
+
+ up(&cfag12864b_mutex);
+}
+
+EXPORT_SYMBOL_GPL(cfag12864b_on);
+EXPORT_SYMBOL_GPL(cfag12864b_off);
+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;
+
+ result = copy_from_user(cfag12864b_formatbuffer, 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 none;
+ }
+
+ cfag12864b_format_nolock(cfag12864b_formatbuffer);
+
+ ret = 0;
+
+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;
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ 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;
+
+ if (down_interruptible(&cfag12864b_mutex))
+ return -ERESTARTSYS;
+
+ if (*offset > CFAG12864B_SIZE) {
+ ret = 0;
+ goto none;
+ }
+ if (*offset + count > CFAG12864B_SIZE)
+ count = CFAG12864B_SIZE - *offset;
+
+ result = copy_from_user(cfag12864b_buffer, buffer, count);
+ if (result != 0) {
+ printk(KERN_ERR CFAG12864B_NAME ": FOP write: ERROR: "
+ "can't copy memory from user\n");
+ ret = -EFAULT;
+ goto none;
+ }
+
+ cfag12864b_write_nolock(*offset, cfag12864b_buffer, count);
+
+ *offset += count;
+ ret = count;
+
+}
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+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);
+ }
+
+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
+ */
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+
+static int __init cfag12864b_init(void)
+{
+ int result;
+ int ret = -EINVAL;
+
+
+ cfag12864b_buffer = kmalloc(sizeof(unsigned char) * CFAG12864B_SIZE,
+ GFP_KERNEL);
+ if (cfag12864b_buffer == NULL) {
+ printk(KERN_ERR CFAG12864B_NAME ": ERROR: "
+ "can't alloc buffer (%i bytes)\n", CFAG12864B_SIZE);
+ ret = -ENOMEM;
+ goto none;
+ }
+
+
+ cfag12864b_formatbuffer = kmalloc(sizeof(unsigned char) * CFAG12864B_MATRIXSIZE,
+ GFP_KERNEL);
+ if (cfag12864b_formatbuffer == NULL) {
+ printk(KERN_ERR CFAG12864B_NAME ": ERROR: "
+ "can't alloc format buffer (%i bytes)\n", CFAG12864B_MATRIXSIZE);
+ ret = -ENOMEM;
+ goto bufferallocated;
+ }
+
+
+ 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 formatbufferallocated;
+ }
+
+
+ cfag12864b_major = MAJOR(cfag12864b_device);
+ cfag12864b_minor = cfag12864b_firstminor;
+ cfag12864b_device = MKDEV(cfag12864b_major, cfag12864b_minor);
+
+
+ cfag12864b_clear_nolock();
+ cfag12864b_on_nolock();
+
+
+ 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");
+ ret = result;
+ goto regionallocated;
+ }
+ }
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+
+ if (class_device_create(lcdclass_class, NULL, cfag12864b_device, NULL,
+ "cfag12864b%d", cfag12864b_minor) == NULL) {
+ printk(KERN_ERR CFAG12864B_NAME ": ERROR: "
+ "unable to create a device for the LCD class\n");
+ ret = -EINVAL;
+ goto cdevadded;
+ }
+
+ printk(KERN_DEBUG CFAG12864B_NAME ": Inited\n");
+
+ return 0;
+
+cdevadded:
+ cdev_del(&cfag12864b_chardevice);
+
+regionallocated:
+ unregister_chrdev_region(cfag12864b_device, cfag12864b_ndevices);
+
+formatbufferallocated:
+ kfree(cfag12864b_formatbuffer);
+
+bufferallocated:
+ kfree(cfag12864b_buffer);
+
+none:
+ return ret;
+}
+
+static void __exit cfag12864b_exit(void)
+{
+ cfag12864b_off_nolock();
+
+ class_device_destroy(lcdclass_class, cfag12864b_device);
+ cdev_del(&cfag12864b_chardevice);
+ unregister_chrdev_region(cfag12864b_device, cfag12864b_ndevices);
+ kfree(cfag12864b_formatbuffer);
+ kfree(cfag12864b_buffer);
+
+ printk(KERN_DEBUG 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 LCD driver");
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/auxdisplay/Kconfig
linux-2.6.18/drivers/auxdisplay/Kconfig
--- linux-2.6.18-vanilla/drivers/auxdisplay/Kconfig 1970-01-01 00:00:00.000000000 +0000
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+++ linux-2.6.18/drivers/auxdisplay/Kconfig 2006-10-04 16:43:51.000000000 +0000
@@ -0,0 +1,103 @@
+#
+# For a description of the syntax of this configuration file,
+# see Documentation/kbuild/kconfig-language.txt.
+#
+# Auxiliary display drivers configuration.
+#
+
+menu "Auxiliary Display support"
+
+config LCDCLASS
+tristate "LCD support"
+default n
+---help---
+If you have a LCD (Liquid Crystal Display), say Y.
+
+To compile this as a module, choose M here:
+the module will be called lcdclass.
+
+Most LCD drivers use an I/O port (like the parallel port)
+so you will need to say Y or M for them.
+
+If unsure, say N.
+
+comment "Parallel port dependent:"
+
+config KS0108
+tristate "KS0108 LCD Controller"
+depends on LCDCLASS && PARPORT
+default n
+---help---
+If you have a LCD 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).
+
+To compile this as a module, choose M here:
+the module will be called ks0108.
+
+If unsure, say N.
+
+config KS0108_PORT
+hex "Parallel port where the LCD is connected"
+depends on KS0108
+default 0x378
+---help---
+The address of the parallel port where the LCD is connected.
```

## [PATCH 2.6.18 V8] drivers: add LCD support

- +
- + 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" or "cat /proc/ioports". You can see there how
- + many parallel ports are present and which address each one 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
- + value using the module parameters.
- +
- +config CFAG12864B
- + tristate "CFAG12864B LCD"
- + depends on KS0108
- + default n
- + ----help----
- + If you have a Crystalfontz 128x64 2-color LCD, 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/crystalfontz\\_cfag12864B-TMI-V.png](http://www.skippari.net/lcd/sekalaista/crystalfontz_cfag12864B-TMI-V.png)
- +
- + Also, you can find help in Crystalfontz and LCDStudio forums.
- + Check Documentation/lcddisplay/cfag12864b for more information.
- +
- + To compile this as a module, choose M here:
- + the module will be called cfag12864b.
- +
- + If unsure, say N.

[PATCH 2.6.18 V8] drivers: add LCD support

```
+
+endmenu
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/auxdisplay/ks0108.c
linux-2.6.18/drivers/auxdisplay/ks0108.c
--- linux-2.6.18-vanilla/drivers/auxdisplay/ks0108.c 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/auxdisplay/ks0108.c 2006-10-04 16:50:14.000000000 +0000
@@ -0,0 +1,171 @@
+/*
+ * Filename: ks0108.c
+ * Version: 0.1.0
+ * Description: ks0108 LCD Controller driver
+ * License: GPLv2
+ * Depends: parport
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ * Date: 2006-10-04
+ *
+ * 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 V8] drivers: add LCD support

```
+module_param(ks0108_port, uint, S_IRUGO);
+MODULE_PARM_DESC(ks0108_port, "Parallel port where the LCD is connected");
+
+static unsigned int ks0108_delay = CONFIG_KS0108_DELAY;
+module_param(ks0108_delay, uint, S_IRUGO);
+MODULE_PARM_DESC(ks0108_delay, "Delay between each control writing (microseconds)");
+
+/*
+ * Device
+ */
+
+static struct parport *ks0108_parport;
+static struct pardevice *ks0108_pardevice;
+
+/*
+ * ks0108 Exported cmds (don't lock)
+ *
+ * You should lock in the top driver: This functions should not
+ * get race conditions in any way. Locking for each byte here would be
+ * so slow and useless.
+ *
+ * There are not bit definitions because they are not flags,
+ * just arbitrary combinations defined by the documentation for each
+ * function in the ks0108 LCD controller. If you want to know what means
+ * a specific combination, look at the function's name.
+ *
+ * The ks0108_writecontrol bits need to be reverted ^(0,1,3) because
+ * the parallel port also revert them with a "not" logic gate.
+ */
+
+#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));
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+}
+
+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));
+}
+
+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;
+ }
+
+}
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ printk(KERN_DEBUG KS0108_NAME ": Initd - ks0108_port=0x%X ks0108_delay=%i\n",
+ ks0108_port, ks0108_delay);
+ return 0;
+
+registered:
+ parport_unregister_device(ks0108_pardevice);
+
+none:
+ return ret;
+}
+
+static void __exit ks0108_exit(void)
+{
+ parport_release(ks0108_pardevice);
+ parport_unregister_device(ks0108_pardevice);
+
+ printk(KERN_DEBUG 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 LCD Controller driver");
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/auxdisplay/lcdclass.c
linux-2.6.18/drivers/auxdisplay/lcdclass.c
--- linux-2.6.18-vanilla/drivers/auxdisplay/lcdclass.c 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/auxdisplay/lcdclass.c 2006-10-04 17:27:50.000000000 +0000
@@ -0,0 +1,79 @@
+/*
+ * Filename: lcdclass.c
+ * Version: 0.1.0
+ * Description: LCD Class
+ * License: GPLv2
+ * Depends: -
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>
+ * Date: 2006-10-04
+ *
+ * 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.
+ *
+ */
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ * 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/device.h>
+#include <linux/lcdclass.h>
+
+#define LCDCLASS_NAME "LCD"
+
+/*
+ * Exported Display Data
+ */
+
+struct class *lcdclass_class;
+EXPORT_SYMBOL_GPL(lcdclass_class);
+
+/*
+ * Module Init & Exit
+ */
+
+static int __init lcdclass_init(void)
+{
+ int ret = -EINVAL;
+
+ lcdclass_class = class_create(THIS_MODULE, LCDCLASS_NAME);
+ if (IS_ERR(lcdclass_class)) {
+ printk(KERN_ERR LCDCLASS_NAME ": ERROR: "
+ "can't create %s class\n", LCDCLASS_NAME);
+ goto none;
+ }
+
+ printk(KERN_DEBUG LCDCLASS_NAME ": Inited\n");
+
+ return 0;
+
+none:
+ return ret;
+}
+
+static void __exit lcdclass_exit(void)
+{
+ class_destroy(lcdclass_class);
+
+ printk(KERN_DEBUG LCDCLASS_NAME ": Exited\n");
+}
+}
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+
+module_init(lcdclass_init);
+module_exit(lcdclass_exit);
+
+MODULE_LICENSE("GPL");
+MODULE_AUTHOR("Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>");
+MODULE_DESCRIPTION("LCD Class");
+
diff -uprN -X dontdiff linux-2.6.18-vanilla/drivers/auxdisplay/Makefile
linux-2.6.18/drivers/auxdisplay/Makefile
--- linux-2.6.18-vanilla/drivers/auxdisplay/Makefile 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/drivers/auxdisplay/Makefile 2006-10-04 16:44:19.000000000 +0000
@@ -0,0 +1,7 @@
+#
+# Makefile for the kernel auxiliary displays device drivers.
+#
+
+obj-$(CONFIG_LCDCLASS) += lcdclass.o
+obj-$(CONFIG_KS0108) += ks0108.o
+obj-$(CONFIG_CFAG12864B) += cfag12864b.o
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 2006-10-04 16:37:23.000000000 +0000
@@ -74,4 +74,6 @@ source "drivers rtc/Kconfig"

source "drivers/dma/Kconfig"

+source "drivers/auxdisplay/Kconfig"
+
+endmenu
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 2006-10-04 16:42:41.000000000 +0000
@@ -76,3 +76,4 @@ obj-$(CONFIG_CRYPTOD) += crypto/
obj-$(CONFIG_SUPERH) += sh/
obj-$(CONFIG_GENERIC_TIME) += clocksource/
obj-$(CONFIG_DMA_ENGINE) += dma/
+obj-y += auxdisplay/
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 2006-10-04 16:51:26.000000000 +0000
@@ -0,0 +1,58 @@
+/*
+ * Filename: cfag12864b.h
+ * Version: 0.1.0
+ * Description: cfag12864b LCD driver header
+ * License: GPLv2
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxx>
+ * Date: 2006-10-04
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ *
+ * 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_
+
+#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_IOC_OFF (_IO((CFAG12864B_IOC_MAGIC),0))
+#define CFAG12864B_IOC_ON (_IO((CFAG12864B_IOC_MAGIC),1))
+#define CFAG12864B_IOC_CLEAR (_IO((CFAG12864B_IOC_MAGIC),2))
+#define CFAG12864B_IOC_FORMAT (_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
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
--- 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 2006-10-04 16:51:40.000000000 +0000
@@ -0,0 +1,36 @@
+/*
+ * Filename: ks0108.h
+ * Version: 0.1.0
+ * Description: ks0108 LCD Controller driver header
+ * License: GPLv2
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ * Date: 2006-10-04
+ *
+ * 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 _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/lcdclass.h linux-2.6.18/include/linux/lcdclass.h
--- linux-2.6.18-vanilla/include/linux/lcdclass.h 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.18/include/linux/lcdclass.h 2006-10-04 16:51:06.000000000 +0000
@@ -0,0 +1,33 @@
+/*
+ * Filename: lcdclass.h
+ * Version: 0.1.0
+ * Description: LCD Class Header
+ * License: GPLv2
+ *
+ * Author: Copyright (C) Miguel Ojeda Sandonis <maxextreme@xxxxxxxxxx>
+ * Date: 2006-10-04
```

[PATCH 2.6.18 V8] drivers: add LCD support

```
+ *
+ * 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 _LCDCLASS_H_
+#define _LCDCLASS_H_
+
+#include <linux/device.h>
+
+extern struct class *lcdclass_class;
+
+#endif /* _LCDCLASS_H_ */
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 2006-10-04 17:10:15.000000000 +0000
@@ -435,6 +435,12 @@ W: http://people.redhat.com/sgrubb/audit
T: git kernel.org/pub/scm/linux/kernel/git/dwmw2/audit-2.6.git
S: Maintained

+AUXILIARY DISPLAY DRIVERS
+P: Miguel Ojeda Sandonis
+M: maxextreme@xxxxxxxxxx
+L: linux-kernel@xxxxxxxxxxxxxxxxxxx
+S: Maintained
+
+AX.25 NETWORK LAYER
+P: Ralf Baechle
+M: ralf@xxxxxxxxxxxxxxxxxx
+
+
+To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
+the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
+More majordomo info at http://vger.kernel.org/majordomo-info.html
+Please read the FAQ at http://www.tux.org/lkml/
```