/\*

u8g\_dev\_ks0108\_128x64.c

Universal 8bit Graphics Library

Copyright (c) 2011, olikraus@gmail.com

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list

of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this

list of conditions and the following disclaimer in the documentation and/or other

materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND

CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,

INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF

ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ADDRESS = 0 (Command Mode)

0x03f Display On

0x0c0 Start Display at line 0

0x040 | y write to y address (y:0..63)

0x0b8 | x write to page [0..7]

u8g\_Init8Bit(u8g, dev, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw, reset)

u8g\_Init8Bit(u8g, dev, 8, 9, 10, 11, 4, 5, 6, 7, 18, 14, 15, 17, 16, U8G\_PIN\_NONE)

\*/

#include "u8g.h"

#define WIDTH 128

#define HEIGHT 64

#define PAGE\_HEIGHT 8

static const uint8\_t u8g\_dev\_ks0108\_128x64\_init\_seq[] PROGMEM = {

U8G\_ESC\_CS(0), /\* disable chip \*/

U8G\_ESC\_ADR(0), /\* instruction mode \*/

U8G\_ESC\_RST(1), /\* do reset low pulse with (1\*16)+2 milliseconds \*/

U8G\_ESC\_CS(1), /\* enable chip 1 \*/

0x03f, /\* display on \*/

0x0c0, /\* start at line 0 \*/

U8G\_ESC\_DLY(20), /\* delay 20 ms \*/

U8G\_ESC\_CS(2), /\* enable chip 2 \*/

0x03f, /\* display on \*/

0x0c0, /\* start at line 0 \*/

U8G\_ESC\_DLY(20), /\* delay 20 ms \*/

U8G\_ESC\_CS(0), /\* disable all chips \*/

U8G\_ESC\_END /\* end of sequence \*/

};

uint8\_t u8g\_dev\_ks0108\_128x64\_fn(u8g\_t \*u8g, u8g\_dev\_t \*dev, uint8\_t msg, void \*arg)

{

switch(msg)

{

case U8G\_DEV\_MSG\_INIT:

u8g\_InitCom(u8g, dev, U8G\_SPI\_CLK\_CYCLE\_NONE);

u8g\_WriteEscSeqP(u8g, dev, u8g\_dev\_ks0108\_128x64\_init\_seq);

break;

case U8G\_DEV\_MSG\_STOP:

break;

case U8G\_DEV\_MSG\_PAGE\_NEXT:

{

u8g\_pb\_t \*pb = (u8g\_pb\_t \*)(dev->dev\_mem);

u8g\_SetAddress(u8g, dev, 0); /\* command mode \*/

u8g\_SetChipSelect(u8g, dev, 2);

u8g\_WriteByte(u8g, dev, 0x0b8 | pb->p.page); /\* select current page (KS0108b) \*/

u8g\_WriteByte(u8g, dev, 0x040 ); /\* set address 0 \*/

u8g\_SetAddress(u8g, dev, 1); /\* data mode \*/

u8g\_WriteSequence(u8g, dev, 64, pb->buf);

u8g\_SetChipSelect(u8g, dev, 0);

u8g\_SetAddress(u8g, dev, 0); /\* command mode \*/

u8g\_SetChipSelect(u8g, dev, 1);

u8g\_WriteByte(u8g, dev, 0x0b8 | pb->p.page); /\* select current page (KS0108b) \*/

u8g\_WriteByte(u8g, dev, 0x040 ); /\* set address 0 \*/

u8g\_SetAddress(u8g, dev, 1); /\* data mode \*/

u8g\_WriteSequence(u8g, dev, 64, 64+(uint8\_t \*)pb->buf);

u8g\_SetChipSelect(u8g, dev, 0);

}

break;

}

return u8g\_dev\_pb8v1\_base\_fn(u8g, dev, msg, arg);

}

U8G\_PB\_DEV(u8g\_dev\_ks0108\_128x64, WIDTH, HEIGHT, PAGE\_HEIGHT, u8g\_dev\_ks0108\_128x64\_fn, U8G\_COM\_PARALLEL);

U8G\_PB\_DEV(u8g\_dev\_ks0108\_128x64\_fast, WIDTH, HEIGHT, PAGE\_HEIGHT, u8g\_dev\_ks0108\_128x64\_fn, U8G\_COM\_FAST\_PARALLEL);