/\*

u8g\_dev\_st7687\_c144mvgd.c (1.44" TFT)

Status: Started, but not finished

Universal 8bit Graphics Library

Copyright (c) 2012, 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.

\*/

#include "u8g.h"

#define WIDTH 128

#define HEIGHT 128

#define PAGE\_HEIGHT 8

#ifdef FIRST\_VERSION

/\*

see also: read.pudn.com/downloads115/sourcecode/app/484503/LCM\_Display.c\_\_.htm

http://en.pudn.com/downloads115/sourcecode/app/detail484503\_en.html

\*/

static const uint8\_t u8g\_dev\_st7687\_c144mvgd\_init\_seq[] PROGMEM = {

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

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

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

U8G\_ESC\_RST(15), /\* do reset low pulse with (15\*16)+2 milliseconds (=maximum delay)\*/

0x001, /\* A0=0, SW reset \*/

U8G\_ESC\_DLY(200), /\* delay 200 ms \*/

0x0d7, /\* EEPROM data auto re-load control \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x09f, /\* ARD = 1 \*/

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

U8G\_ESC\_DLY(100), /\* delay 100 ms \*/

0x0e0, /\* EEPROM control in \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000, /\* \*/

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

U8G\_ESC\_DLY(100), /\* delay 100 ms \*/

#ifdef NOT\_REQUIRED

0x0fa, /\* EEPROM function selection 8.1.66 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000, /\* \*/

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

U8G\_ESC\_DLY(100), /\* delay 100 ms \*/

#endif

0x0e3, /\* Read from EEPROM, 8.1.55 \*/

U8G\_ESC\_DLY(100), /\* delay 100 ms \*/

0x0e1, /\* EEPROM control out, 8.1.53 \*/

U8G\_ESC\_DLY(100), /\* delay 100 ms \*/

//0x028, /\* display off \*/

0x011, /\* Sleep out & booster on \*/

U8G\_ESC\_DLY(100), /\* delay 100 ms \*/

0x0c0, /\* Vop setting, 8.1.42 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000, /\* \*/

0x001, /\* 3.6 + 256\*0.04 = 13.84 Volt \*/

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

U8G\_ESC\_DLY(100), /\* delay 100 ms \*/

0x0c3, /\* Bias selection, 8.1.45 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x003,

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

0x0c4, /\* Booster setting 8.1.46 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x007,

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

0x0c5, /\* ??? \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x001,

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

0x0cb, /\* FV3 with Booster x2 control, 8.1.47 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x001,

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

0x036, /\* Memory data access control, 8.1.28 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x080,

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

0x0b5, /\* N-line control, 8.1.37 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x089,

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

0x0d0, /\* Analog circuit setting, 8.1.49 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x01d,

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

0x0b7, /\* Com/Seg Scan Direction, 8.1.38 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x040,

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

0x025, /\* Write contrast, 8.1.17 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x03f,

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

0x03a, /\* Interface pixel format, 8.1.32 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x004, /\* 3: 12 bit per pixel Type A, 4: 12 bit Type B, 5: 16bit per pixel \*/

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

0x0b0, /\* Display Duty setting, 8.1.34 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x07f,

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

0x0f0, /\* Frame Freq. in Temp range A,B,C and D, 8.1.59 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x007,

0x00c,

0x00c,

0x015,

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

0x0f9, /\* Frame RGB Value, 8.1.65 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000,

0x005,

0x008,

0x00a,

0x00c,

0x00e,

0x010,

0x011,

0x012,

0x013,

0x014,

0x015,

0x016,

0x018,

0x01a,

0x01b,

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

0x0f9, /\* Frame RGB Value, 8.1.65 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000,

0x000,

0x000,

0x000,

0x033,

0x055,

0x055,

0x055,

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

0x029, /\* display on \*/

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

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

};

#else

/\*

http://www.waitingforfriday.com/images/e/e3/FTM144D01N\_test.zip

\*/

static const uint8\_t u8g\_dev\_st7687\_c144mvgd\_init\_seq[] PROGMEM = {

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

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

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

U8G\_ESC\_RST(15), /\* do reset low pulse with (15\*16)+2 milliseconds (=maximum delay)\*/

0x011, /\* Sleep out & booster on \*/

U8G\_ESC\_DLY(5), /\* delay 5 ms \*/

0x03a, /\* Interface pixel format, 8.1.32 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x004, /\* 3: 12 bit per pixel Type A, 4: 12 bit Type B, 5: 16bit per pixel \*/

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

0x026, /\* SET\_GAMMA\_CURVE \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x004,

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

0x0f2, /\* GAM\_R\_SEL \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x001, /\* enable gamma adj \*/

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

0x0e0, /\* POSITIVE\_GAMMA\_CORRECT \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x3f,

0x25,

0x1c,

0x1e,

0x20,

0x12,

0x2a,

0x90,

0x24,

0x11,

0x00,

0x00,

0x00,

0x00,

0x00,

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

0x0e1, /\* NEGATIVE\_GAMMA\_CORRECT \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x20,

0x20,

0x20,

0x20,

0x05,

0x00,

0x15,

0xa7,

0x3d,

0x18,

0x25,

0x2a,

0x2b,

0x2b,

0x3a,

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

0x0b1, /\* FRAME\_RATE\_CONTROL1 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x008, /\* DIVA = 8 \*/

0x008, /\* VPA = 8 \*/

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

0x0b4, /\* DISPLAY\_INVERSION \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x007, /\* NLA = 1, NLB = 1, NLC = 1 (all on Frame Inversion) \*/

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

0x0c0, /\* POWER\_CONTROL1 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x00a, /\* VRH = 10: GVDD = 4.30 \*/

0x002, /\* VC = 2: VCI1 = 2.65 \*/

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

0x0c1, /\* POWER\_CONTROL2 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x002, /\* BT = 2: AVDD = 2xVCI1, VCL = -1xVCI1, VGH = 5xVCI1, VGL = -2xVCI1 \*/

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

0x0c5, /\* VCOM\_CONTROL1 \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x050, /\* VMH = 80: VCOMH voltage = 4.5 \*/

0x05b, /\* VML = 91: VCOML voltage = -0.225 \*/

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

0x0c7, /\* VCOM\_OFFSET\_CONTROL \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x040, /\* nVM = 0, VMF = 64: VCOMH output = VMH, VCOML output = VML \*/

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

0x02a, /\* SET\_COLUMN\_ADDRESS \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000, /\* \*/

0x000, /\* \*/

0x000, /\* \*/

0x07f, /\* \*/

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

0x02b, /\* SET\_PAGE\_ADDRESS \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000, /\* \*/

0x000, /\* \*/

0x000, /\* \*/

0x07f, /\* \*/

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

0x036, /\* SET\_ADDRESS\_MODE \*/

U8G\_ESC\_ADR(1), /\* data mode \*/

0x000, /\* Select display orientation \*/

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

0x029, /\* display on \*/

0x02c, /\* write start \*/

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

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

};

#endif

/\* calculate bytes for Type B 4096 color display \*/

static uint8\_t get\_byte\_1(uint8\_t v)

{

v >>= 4;

v &= 0x0e;

return v;

}

static uint8\_t get\_byte\_2(uint8\_t v)

{

uint8\_t w;

w = v;

w &= 3;

w = (w<<2) | w;

v <<= 3;

v &= 0x0e0;

w |= v;

return w;

}

uint8\_t u8g\_dev\_st7687\_c144mvgd\_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\_400NS);

u8g\_WriteEscSeqP(u8g, dev, u8g\_dev\_st7687\_c144mvgd\_init\_seq);

break;

case U8G\_DEV\_MSG\_STOP:

break;

case U8G\_DEV\_MSG\_PAGE\_NEXT:

{

uint8\_t y, i, j;

uint8\_t \*ptr;

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

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

u8g\_SetChipSelect(u8g, dev, 1);

y = pb->p.page\_y0;

ptr = pb->buf;

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

u8g\_WriteByte(u8g, dev, 0x02a ); /\* Column address set 8.1.20 \*/

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

u8g\_WriteByte(u8g, dev, 0x000 ); /\* x0 \*/

u8g\_WriteByte(u8g, dev, WIDTH-1 ); /\* x1 \*/

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

u8g\_WriteByte(u8g, dev, 0x02b ); /\* Row address set 8.1.21 \*/

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

u8g\_WriteByte(u8g, dev, y ); /\* y0 \*/

u8g\_WriteByte(u8g, dev, y+PAGE\_HEIGHT-1 ); /\* y1 \*/

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

u8g\_WriteByte(u8g, dev, 0x02c ); /\* Memory write 8.1.22 \*/

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

for( i = 0; i < PAGE\_HEIGHT; i ++ )

{

for( j = 0; j < WIDTH; j ++ )

{

u8g\_WriteByte(u8g, dev, get\_byte\_1(\*ptr) );

u8g\_WriteByte(u8g, dev, get\_byte\_2(\*ptr) );

ptr++;

}

}

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

u8g\_SetChipSelect(u8g, dev, 0);

}

break;

}

return u8g\_dev\_pb8h8\_base\_fn(u8g, dev, msg, arg);

}

uint8\_t u8g\_st7687\_c144mvgd\_8h8\_buf[WIDTH\*8] U8G\_NOCOMMON ;

u8g\_pb\_t u8g\_st7687\_c144mvgd\_8h8\_pb = { {8, HEIGHT, 0, 0, 0}, WIDTH, u8g\_st7687\_c144mvgd\_8h8\_buf};

u8g\_dev\_t u8g\_dev\_st7687\_c144mvgd\_sw\_spi = { u8g\_dev\_st7687\_c144mvgd\_fn, &u8g\_st7687\_c144mvgd\_8h8\_pb, u8g\_com\_arduino\_sw\_spi\_fn };

u8g\_dev\_t u8g\_dev\_st7687\_c144mvgd\_8bit = { u8g\_dev\_st7687\_c144mvgd\_fn, &u8g\_st7687\_c144mvgd\_8h8\_pb, U8G\_COM\_PARALLEL };