//---------------------------------------------------------

/\*

NHD\_7\_0\_800480EF\_mega.ino

Program for writing to Newhaven Display 7.0" TFT with NHD-5.0-800480TF-20 Controller Board (SSD1963, 8-bit)

(c)2013 Mike LaVine - Newhaven Display International, LLC.

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.

\*/

//---------------------------------------------------------

// The 8 bit data bus is connected to PORTA of the Arduino Mega2560

// 5V voltage regulator on Arduino Mega has been replaced with a 3.3V regulator to provide 3.3V logic

int RS = 30; // RS signal connected to Arduino digital pin 30

int WR = 31; // /WR signal connected to Arduino digital pin 31

int RD = 32; // /RD signal connected to Arduino digital pin 32

int RES = 33; // /RES signal connected to Arduino digital pin 33

int DIS = 34; // DISP signal connected to Arduino digital pin 34

// /CS signal tied to GND

//;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void TFT\_Write\_Command(unsigned char command)

{

digitalWrite(RS, LOW);

PORTA = command;

digitalWrite(WR, LOW);

digitalWrite(WR, HIGH);

}

//;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void TFT\_Write\_Data(unsigned char data)

{

//digitalWrite(RS, HIGH);

PORTA = data;

digitalWrite(WR, LOW);

digitalWrite(WR, HIGH);

}

//====================================================

void TFT\_Command\_Write(unsigned char REG,unsigned char VALUE)

{

TFT\_Write\_Command(REG);

digitalWrite(RS, HIGH);

TFT\_Write\_Data(VALUE);

}

//======================================================

void WindowSet(unsigned int s\_x,unsigned int e\_x,unsigned int s\_y,unsigned int e\_y)

{

TFT\_Write\_Command(0x2a); //SET column address

digitalWrite(RS, HIGH);

TFT\_Write\_Data((s\_x)>>8); //SET start column address

TFT\_Write\_Data(s\_x);

TFT\_Write\_Data((e\_x)>>8); //SET end column address

TFT\_Write\_Data(e\_x);

TFT\_Write\_Command(0x2b); //SET page address

digitalWrite(RS, HIGH);

TFT\_Write\_Data((s\_y)>>8); //SET start page address

TFT\_Write\_Data(s\_y);

TFT\_Write\_Data((e\_y)>>8); //SET end page address

TFT\_Write\_Data(e\_y);

}

void disp()

{

unsigned int i, j;

WindowSet(0,799,0,479); //set start/end column/page address (full screen)

TFT\_Write\_Command(0x2C); //command to begin writing to frame memory

digitalWrite(RS, HIGH);

for(i=0;i<480;i++) //fill screen with blue pixels

{

for(j=0;j<800;j++)

{

TFT\_Write\_Data(0xFF);

TFT\_Write\_Data(0x00);

TFT\_Write\_Data(0x00);

}

}

WindowSet(0,799,0,479); //set start/end column/page address (full screen)

TFT\_Write\_Command(0x2C); //command to begin writing to frame memory

digitalWrite(RS, HIGH);

for(i=0;i<480;i++) //fill screen with green pixels

{

for(j=0;j<800;j++)

{

TFT\_Write\_Data(0x00);

TFT\_Write\_Data(0xFF);

TFT\_Write\_Data(0x00);

}

}

WindowSet(0,799,0,479); //set start/end column/page address (full screen)

TFT\_Write\_Command(0x2C); //command to begin writing to frame memory

digitalWrite(RS, HIGH);

for(i=0;i<480;i++) //fill screen with red pixels

{

for(j=0;j<800;j++)

{

TFT\_Write\_Data(0x00);

TFT\_Write\_Data(0x00);

TFT\_Write\_Data(0xFF);

}

}

}

//======================================================

void setup()

{

DDRA = 0xFF;

PORTA = 0x00;

DDRC = 0xFF;

PORTC = 0x00;

digitalWrite(DIS, HIGH);

digitalWrite(RD, HIGH);

digitalWrite(WR, LOW);

digitalWrite(RES, LOW);

delay(120);

digitalWrite(RES, HIGH);

delay(120);

TFT\_Write\_Command(0x01); //Software reset

delay(120);

TFT\_Write\_Command(0xe2); //set multiplier and divider of PLL

digitalWrite(RS, HIGH);

TFT\_Write\_Data(0x1d);

TFT\_Write\_Data(0x02);

TFT\_Write\_Data(0x04);

TFT\_Command\_Write(0xe0,0x01); //Enable PLL

delay(1);

TFT\_Command\_Write(0xe0,0x03); //Lock PLL

TFT\_Write\_Command(0x01); //Software reset

delay(120);

TFT\_Write\_Command(0xb0); //SET LCD MODE SET TFT 18Bits MODE

digitalWrite(RS, HIGH);

TFT\_Write\_Data(0x08); //SET TFT MODE & hsync+Vsync+DEN MODE

TFT\_Write\_Data(0x80); //SET TFT MODE & hsync+Vsync+DEN MODE

TFT\_Write\_Data(0x03); //SET horizontal size=800-1 HightByte

TFT\_Write\_Data(0x1f); //SET horizontal size=800-1 LowByte

TFT\_Write\_Data(0x01); //SET vertical size=480-1 HightByte

TFT\_Write\_Data(0xdf); //SET vertical size=480-1 LowByte

TFT\_Write\_Data(0x00); //SET even/odd line RGB seq.=RGB

TFT\_Command\_Write(0xf0,0x00); //SET pixel data I/F format=8bit

TFT\_Command\_Write(0x36,0x09); //SET address mode=flip vertical

TFT\_Write\_Command(0xe6); //SET PCLK freq

digitalWrite(RS, HIGH);

TFT\_Write\_Data(0x0f);

TFT\_Write\_Data(0xff);

TFT\_Write\_Data(0xff);

TFT\_Write\_Command(0xb4); //SET HBP

digitalWrite(RS, HIGH);

TFT\_Write\_Data(0x04); //SET HSYNC Total=1056

TFT\_Write\_Data(0x20);

TFT\_Write\_Data(0x00); //SET HBP 88

TFT\_Write\_Data(0x58);

TFT\_Write\_Data(0x80); //SET HSYNC Pulse Width=128=127pixels+1

TFT\_Write\_Data(0x00); //SET Hsync pulse start position

TFT\_Write\_Data(0x00);

TFT\_Write\_Data(0x00); //SET Hsync pulse subpixel start position

TFT\_Write\_Command(0xb6); //SET VBP

digitalWrite(RS, HIGH);

TFT\_Write\_Data(0x02); //SET Vsync total 525

TFT\_Write\_Data(0x0d);

TFT\_Write\_Data(0x00); //SET VBP=32

TFT\_Write\_Data(0x20);

TFT\_Write\_Data(0x01); //SET VSYNC Pulse Width= 0=0lines+1

TFT\_Write\_Data(0x00); //SET Vsync pulse start position

TFT\_Write\_Data(0x00);

TFT\_Write\_Command(0x13); //SET display on

TFT\_Write\_Command(0x38); //SET display on

TFT\_Write\_Command(0x29); //SET display on

delay(10);

}

void loop()

{

disp();

delay(1000);

}