/\*

U8glib.h

C++ Interface

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.

\*/

#ifndef \_CPP\_U8GLIB

#define \_CPP\_U8GLIB

#include <Print.h>

#include "utility/u8g.h"

class U8GLIB : public Print

{

private:

u8g\_t u8g;

u8g\_uint\_t tx, ty; // current position for the Print base class procedures

uint8\_t is\_begin;

void prepare(void) { tx = 0; ty = 0; is\_begin = 0; }

void cbegin(void) { if ( is\_begin == 0 ) { is\_begin = 1; u8g\_Begin(&u8g); } }

uint8\_t initSPI(u8g\_dev\_t \*dev, uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE);

uint8\_t initHWSPI(u8g\_dev\_t \*dev, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE);

uint8\_t initI2C(u8g\_dev\_t \*dev, uint8\_t options);

protected:

uint8\_t init8BitFixedPort(u8g\_dev\_t \*dev, uint8\_t en, uint8\_t cs, uint8\_t di, uint8\_t rw, uint8\_t reset);

private:

uint8\_t init8Bit(u8g\_dev\_t \*dev, uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7, uint8\_t en, uint8\_t cs1, uint8\_t cs2, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE);

uint8\_t initRW8Bit(u8g\_dev\_t \*dev, uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7, uint8\_t cs, uint8\_t a0, uint8\_t wr, uint8\_t rd, uint8\_t reset);

public:

/\* constructor \*/

U8GLIB(void)

{ }

U8GLIB(u8g\_dev\_t \*dev)

{ prepare(); u8g\_Init(&u8g, dev); }

U8GLIB(u8g\_dev\_t \*dev, u8g\_com\_fnptr com\_fn)

{ prepare(); u8g\_InitComFn(&u8g, dev, com\_fn); }

U8GLIB(u8g\_dev\_t \*dev, uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset)

{ initSPI(dev, sck, mosi, cs, a0, reset); }

U8GLIB(u8g\_dev\_t \*dev, uint8\_t cs, uint8\_t a0, uint8\_t reset)

{ initHWSPI(dev, cs, a0, reset); }

U8GLIB(u8g\_dev\_t \*dev, uint8\_t options)

{ initI2C(dev, options); }

U8GLIB(u8g\_dev\_t \*dev, uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7, uint8\_t en, uint8\_t cs1, uint8\_t cs2, uint8\_t di, uint8\_t rw, uint8\_t reset)

{ init8Bit(dev, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw, reset); }

U8GLIB(u8g\_dev\_t \*dev, uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7, uint8\_t cs, uint8\_t a0, uint8\_t wr, uint8\_t rd, uint8\_t reset)

{ initRW8Bit(dev, d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd, reset); }

uint8\_t begin(void) { is\_begin = 1; return u8g\_Begin(&u8g); }

void setPrintPos(u8g\_uint\_t x, u8g\_uint\_t y) { tx = x; ty = y; }

u8g\_t \*getU8g(void) { return &u8g; }

/\* implementation of the write interface to the print class \*/

#if defined(ARDUINO) && ARDUINO >= 100

size\_t write(uint8\_t c) { tx += u8g\_DrawGlyph(&u8g, tx, ty, c); return 1;}

#else

void write(uint8\_t c) { tx += u8g\_DrawGlyph(&u8g, tx, ty, c); }

#endif

/\* screen rotation \*/

void undoRotation(void) { u8g\_UndoRotation(&u8g); }

void setRot90(void) { u8g\_SetRot90(&u8g); }

void setRot180(void) { u8g\_SetRot180(&u8g); }

void setRot270(void) { u8g\_SetRot270(&u8g); }

/\* screen scaling \*/

void undoScale(void) { u8g\_UndoScale(&u8g); }

void setScale2x2(void) { u8g\_SetScale2x2(&u8g); }

/\* picture loop \*/

void firstPage(void) { cbegin(); u8g\_FirstPage(&u8g); }

uint8\_t nextPage(void) { return u8g\_NextPage(&u8g); }

/\* system commands \*/

uint8\_t setContrast(uint8\_t contrast) { cbegin(); return u8g\_SetContrast(&u8g, contrast); }

void sleepOn(void) { u8g\_SleepOn(&u8g); }

void sleepOff(void) { u8g\_SleepOff(&u8g); }

/\* graphic primitives \*/

void setColorEntry(uint8\_t color\_index, uint8\_t r, uint8\_t g, uint8\_t b) { u8g\_SetColorEntry(&u8g, color\_index, r, g, b); }

void setHiColor(uint16\_t rgb) { u8g\_SetHiColor(&u8g, rgb); }

void setHiColorByRGB(uint8\_t r, uint8\_t g, uint8\_t b) { u8g\_SetHiColorByRGB(&u8g, r, g, b); }

void setRGB(uint8\_t r, uint8\_t g, uint8\_t b) { u8g\_SetRGB(&u8g, r, g, b); }

void setColorIndex(uint8\_t color\_index) { u8g\_SetColorIndex(&u8g, color\_index); }

uint8\_t getColorIndex(void) { return u8g\_GetColorIndex(&u8g); }

void setDefaultForegroundColor(void) { u8g\_SetDefaultForegroundColor(&u8g); }

void setDefaultBackgroundColor(void) { u8g\_SetDefaultBackgroundColor(&u8g); }

void setDefaultMidColor(void) { u8g\_SetDefaultMidColor(&u8g); }

u8g\_uint\_t getWidth(void) { return u8g\_GetWidth(&u8g); }

u8g\_uint\_t getHeight(void) { return u8g\_GetHeight(&u8g); }

uint8\_t getMode(void) { return u8g\_GetMode(&u8g); }

void drawPixel(u8g\_uint\_t x, u8g\_uint\_t y) { return u8g\_DrawPixel(&u8g, x, y); }

void drawHLine(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t w) { u8g\_DrawHLine(&u8g, x, y, w); }

void drawVLine(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t h) { u8g\_DrawVLine(&u8g, x, y, h); }

void drawLine(u8g\_uint\_t x1, u8g\_uint\_t y1, u8g\_uint\_t x2, u8g\_uint\_t y2) { u8g\_DrawLine(&u8g, x1, y1, x2, y2); }

void drawFrame(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t w, u8g\_uint\_t h) { u8g\_DrawFrame(&u8g, x, y, w, h); }

void drawRFrame(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t w, u8g\_uint\_t h, u8g\_uint\_t r) { u8g\_DrawRFrame(&u8g, x, y, w, h,r); }

void drawBox(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t w, u8g\_uint\_t h) { u8g\_DrawBox(&u8g, x, y, w, h); }

void drawRBox(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t w, u8g\_uint\_t h, u8g\_uint\_t r) { u8g\_DrawRBox(&u8g, x, y, w, h,r); }

void drawCircle(u8g\_uint\_t x0, u8g\_uint\_t y0, u8g\_uint\_t rad, uint8\_t opt = U8G\_DRAW\_ALL) { u8g\_DrawCircle(&u8g, x0, y0, rad, opt); }

void drawDisc(u8g\_uint\_t x0, u8g\_uint\_t y0, u8g\_uint\_t rad, uint8\_t opt = U8G\_DRAW\_ALL) { u8g\_DrawDisc(&u8g, x0, y0, rad, opt); }

void drawEllipse(u8g\_uint\_t x0, u8g\_uint\_t y0, u8g\_uint\_t rx, u8g\_uint\_t ry, uint8\_t opt = U8G\_DRAW\_ALL) { u8g\_DrawEllipse(&u8g, x0, y0, rx, ry, opt); }

void drawFilledEllipse(u8g\_uint\_t x0, u8g\_uint\_t y0, u8g\_uint\_t rx, u8g\_uint\_t ry, uint8\_t opt = U8G\_DRAW\_ALL) { u8g\_DrawFilledEllipse(&u8g, x0, y0, rx, ry, opt); }

void drawTriangle(uint16\_t x0, uint16\_t y0, uint16\_t x1, uint16\_t y1, uint16\_t x2, uint16\_t y2)

{ u8g\_DrawTriangle(&u8g, x0, y0, x1, y1, x2, y2); }

/\* bitmap handling \*/

void drawBitmap(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t cnt, u8g\_uint\_t h, const uint8\_t \*bitmap)

{ u8g\_DrawBitmap(&u8g, x, y, cnt, h, bitmap); }

void drawBitmapP(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t cnt, u8g\_uint\_t h, const u8g\_pgm\_uint8\_t \*bitmap)

{ u8g\_DrawBitmapP(&u8g, x, y, cnt, h, bitmap); }

void drawXBM(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t w, u8g\_uint\_t h, const uint8\_t \*bitmap)

{ u8g\_DrawXBM(&u8g, x, y, w, h, bitmap); }

void drawXBMP(u8g\_uint\_t x, u8g\_uint\_t y, u8g\_uint\_t w, u8g\_uint\_t h, const u8g\_pgm\_uint8\_t \*bitmap)

{ u8g\_DrawXBMP(&u8g, x, y, w, h, bitmap); }

/\* font handling \*/

void setFont(const u8g\_fntpgm\_uint8\_t \*font) {u8g\_SetFont(&u8g, font); }

int8\_t getFontAscent(void) { return u8g\_GetFontAscent(&u8g); }

int8\_t getFontDescent(void) { return u8g\_GetFontDescent(&u8g); }

int8\_t getFontLineSpacing(void) { return u8g\_GetFontLineSpacing(&u8g); }

u8g\_uint\_t drawStr(u8g\_uint\_t x, u8g\_uint\_t y, const char \*s) { return u8g\_DrawStr(&u8g, x, y, s); }

u8g\_uint\_t drawStr90(u8g\_uint\_t x, u8g\_uint\_t y, const char \*s) { return u8g\_DrawStr90(&u8g, x, y, s); }

u8g\_uint\_t drawStr180(u8g\_uint\_t x, u8g\_uint\_t y, const char \*s) { return u8g\_DrawStr180(&u8g, x, y, s); }

u8g\_uint\_t drawStr270(u8g\_uint\_t x, u8g\_uint\_t y, const char \*s) { return u8g\_DrawStr270(&u8g, x, y, s); }

u8g\_uint\_t drawStrP(u8g\_uint\_t x, u8g\_uint\_t y, const u8g\_pgm\_uint8\_t \*s) { return u8g\_DrawStrP(&u8g, x, y, s); }

u8g\_uint\_t drawStr90P(u8g\_uint\_t x, u8g\_uint\_t y, const u8g\_pgm\_uint8\_t \*s) { return u8g\_DrawStr90P(&u8g, x, y, s); }

u8g\_uint\_t drawStr180P(u8g\_uint\_t x, u8g\_uint\_t y, const u8g\_pgm\_uint8\_t \*s) { return u8g\_DrawStr180P(&u8g, x, y, s); }

u8g\_uint\_t drawStr270P(u8g\_uint\_t x, u8g\_uint\_t y, const u8g\_pgm\_uint8\_t \*s) { return u8g\_DrawStr270P(&u8g, x, y, s); }

void setFontPosBaseline(void) { u8g\_SetFontPosBaseline(&u8g); }

void setFontPosBottom(void) { u8g\_SetFontPosBottom(&u8g); }

void setFontPosCenter(void) { u8g\_SetFontPosCenter(&u8g); }

void setFontPosTop(void) { u8g\_SetFontPosTop(&u8g); }

void setFontRefHeightText(void) { u8g\_SetFontRefHeightText(&u8g); }

void setFontRefHeightExtendedText(void) { u8g\_SetFontRefHeightExtendedText(&u8g); }

void setFontRefHeightAll(void) { u8g\_SetFontRefHeightAll(&u8g); }

void setFontLineSpacingFactor(uint8\_t factor) { u8g\_SetFontLineSpacingFactor(&u8g, factor); }

u8g\_uint\_t getStrPixelWidth(const char \*s) { return u8g\_GetStrPixelWidth(&u8g, s); }

u8g\_uint\_t getStrPixelWidthP(u8g\_pgm\_uint8\_t \*s) { return u8g\_GetStrPixelWidthP(&u8g, s); }

u8g\_uint\_t getStrWidth(const char \*s) { return u8g\_GetStrWidth(&u8g, s); }

u8g\_uint\_t getStrWidthP(u8g\_pgm\_uint8\_t \*s) { return u8g\_GetStrWidthP(&u8g, s); }

void setHardwareBackup(u8g\_state\_cb backup\_cb) { u8g\_SetHardwareBackup(&u8g, backup\_cb); }

#if defined(ARDUINO) && ARDUINO >= 100

// support for the F() macro

u8g\_uint\_t drawStr(u8g\_uint\_t x, u8g\_uint\_t y, const \_\_FlashStringHelper \*s) { return u8g\_DrawStrP(&u8g, x, y, (u8g\_pgm\_uint8\_t \*)s); }

u8g\_uint\_t drawStr90(u8g\_uint\_t x, u8g\_uint\_t y, const \_\_FlashStringHelper \*s) { return u8g\_DrawStr90P(&u8g, x, y, (u8g\_pgm\_uint8\_t \*)s); }

u8g\_uint\_t drawStr180(u8g\_uint\_t x, u8g\_uint\_t y, const \_\_FlashStringHelper \*s) { return u8g\_DrawStr180P(&u8g, x, y, (u8g\_pgm\_uint8\_t \*)s); }

u8g\_uint\_t drawStr270(u8g\_uint\_t x, u8g\_uint\_t y, const \_\_FlashStringHelper \*s) { return u8g\_DrawStr270P(&u8g, x, y, (u8g\_pgm\_uint8\_t \*)s); }

u8g\_uint\_t getStrPixelWidth(const \_\_FlashStringHelper \*s) { return u8g\_GetStrPixelWidthP(&u8g, (u8g\_pgm\_uint8\_t \*)s); }

u8g\_uint\_t getStrWidth(const \_\_FlashStringHelper \*s) { return u8g\_GetStrWidthP(&u8g, (u8g\_pgm\_uint8\_t \*)s); }

#endif

/\* cursor handling \*/

void setCursorFont(const u8g\_pgm\_uint8\_t \*cursor\_font) { u8g\_SetCursorFont(&u8g, cursor\_font); }

void setCursorStyle(uint8\_t encoding) { u8g\_SetCursorStyle(&u8g, encoding); }

void setCursorPos(u8g\_uint\_t cursor\_x, u8g\_uint\_t cursor\_y) { u8g\_SetCursorPos(&u8g, cursor\_x, cursor\_y); }

void setCursorColor(uint8\_t fg, uint8\_t bg) { u8g\_SetCursorColor(&u8g, fg, bg); }

void enableCursor(void) { u8g\_EnableCursor(&u8g); }

void disableCursor(void) { u8g\_DisableCursor(&u8g); }

void drawCursor(void) { u8g\_DrawCursor(&u8g); }

/\* virtual screen \*/

void setVirtualScreenDimension(u8g\_uint\_t width, u8g\_uint\_t height) { u8g\_SetVirtualScreenDimension(&u8g, width, height); }

uint8\_t addToVirtualScreen(u8g\_uint\_t x, u8g\_uint\_t y, U8GLIB &child\_u8g) { return u8g\_AddToVirtualScreen(&u8g, x, y, &child\_u8g.u8g); }

};

class U8GLIB\_DOGS102 : public U8GLIB

{

public:

U8GLIB\_DOGS102(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_dogs102\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGS102(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_dogs102\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_DOGS102\_2X : public U8GLIB

{

public:

U8GLIB\_DOGS102\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_dogs102\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGS102\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_dogs102\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_MINI12864 : public U8GLIB

{

public:

U8GLIB\_MINI12864(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_mini12864\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_MINI12864(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_mini12864\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_MINI12864\_2X : public U8GLIB

{

public:

U8GLIB\_MINI12864\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_mini12864\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_MINI12864\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1701\_mini12864\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_DOGM132 : public U8GLIB

{

public:

U8GLIB\_DOGM132(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm132\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGM132(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm132\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_NHD\_C12832 : public U8GLIB

{

public:

U8GLIB\_NHD\_C12832(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12832\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD\_C12832(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12832\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_NHD\_C12832(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12832\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_NHD\_C12832\_USART : public U8GLIB

{

public:

U8GLIB\_NHD\_C12832\_USART(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12832\_hw\_usart\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_DOGM128 : public U8GLIB

{

public:

U8GLIB\_DOGM128(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm128\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGM128(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm128\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_DOGM128(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm128\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_DOGM128\_2X : public U8GLIB

{

public:

U8GLIB\_DOGM128\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm128\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGM128\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm128\_2x\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_DOGM128\_2X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_dogm128\_2x\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_LM6059 : public U8GLIB

{

public:

U8GLIB\_LM6059(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6059\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_LM6059(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6059\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_LM6059\_2X : public U8GLIB

{

public:

U8GLIB\_LM6059\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6059\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_LM6059\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6059\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_LM6063 : public U8GLIB

{

public:

U8GLIB\_LM6063(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6063\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_LM6063(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6063\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_LM6063\_2X : public U8GLIB

{

public:

U8GLIB\_LM6063\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6063\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_LM6063\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_lm6063\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_64128N : public U8GLIB

{

public:

U8GLIB\_64128N(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_64128n\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_64128N(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_64128n\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_64128N(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_64128n\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_64128N\_2X : public U8GLIB

{

public:

U8GLIB\_64128N\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_64128n\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_64128N\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_64128n\_2x\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_64128N\_2X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_64128n\_2x\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_NHD\_C12864 : public U8GLIB

{

public:

U8GLIB\_NHD\_C12864(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12864\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD\_C12864(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12864\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_NHD\_C12864\_2X : public U8GLIB

{

public:

U8GLIB\_NHD\_C12864\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12864\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD\_C12864\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7565\_nhd\_c12864\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1601\_C128032 : public U8GLIB

{

public:

U8GLIB\_UC1601\_C128032(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1601\_c128032\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1601\_C128032(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1601\_c128032\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1601\_C128032\_2X : public U8GLIB

{

public:

U8GLIB\_UC1601\_C128032\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1601\_c128032\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1601\_C128032\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1601\_c128032\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1608\_240X64 : public U8GLIB

{

public:

U8GLIB\_UC1608\_240X64(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x64\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1608\_240X64(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x64\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1608\_240X64\_2X : public U8GLIB

{

public:

U8GLIB\_UC1608\_240X64\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x64\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1608\_240X64\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x64\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1608\_240X128 : public U8GLIB

{

public:

U8GLIB\_UC1608\_240X128(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x128\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1608\_240X128(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x128\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1608\_240X128\_2X : public U8GLIB

{

public:

U8GLIB\_UC1608\_240X128\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x128\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1608\_240X128\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1608\_240x128\_2x\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1611\_DOGM240 : public U8GLIB

{

public:

U8GLIB\_UC1611\_DOGM240(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_uc1611\_dogm240\_i2c, options)

{}

U8GLIB\_UC1611\_DOGM240(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1611\_dogm240\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1611\_DOGM240(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1611\_dogm240\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_UC1611\_DOGXL240 : public U8GLIB

{

public:

U8GLIB\_UC1611\_DOGXL240(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_uc1611\_dogxl240\_i2c, options)

{}

U8GLIB\_UC1611\_DOGXL240(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1611\_dogxl240\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_UC1611\_DOGXL240(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1611\_dogxl240\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_ST7920\_128X64 : public U8GLIB

{

public:

U8GLIB\_ST7920\_128X64(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_ST7920\_128X64(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_ST7920\_128X64(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t cs2, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw, reset)

{ }

// U8GLIB\_ST7920\_128X64(uint8\_t cs)

// : U8GLIB(&u8g\_dev\_st7920\_128x64\_sw\_spi, cs, U8G\_PIN\_NONE, U8G\_PIN\_NONE)

// { }

};

class U8GLIB\_ST7920\_128X64\_1X : public U8GLIB

{

public:

U8GLIB\_ST7920\_128X64\_1X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_sw\_spi, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_128X64\_1X(uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_hw\_spi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_128X64\_1X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, U8G\_PIN\_NONE, U8G\_PIN\_NONE, di, rw, reset) // cs1 = cs2 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_ST7920\_128X64\_CUSTOM\_1X : public U8GLIB

{

public:

U8GLIB\_ST7920\_128X64\_CUSTOM\_1X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_custom, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_ST7920\_128X64\_4X : public U8GLIB

{

public:

U8GLIB\_ST7920\_128X64\_4X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_4x\_sw\_spi, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_128X64\_4X(uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_4x\_hw\_spi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_128X64\_4X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_4x\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, U8G\_PIN\_NONE, U8G\_PIN\_NONE, di, rw, reset) // cs1 = cs2 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_ST7920\_128X64\_CUSTOM\_4X : public U8GLIB

{

public:

U8GLIB\_ST7920\_128X64\_CUSTOM\_4X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_128x64\_4x\_custom, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_ST7920\_192X32 : public U8GLIB // OBSOLETE, use U8GLIB\_ST7920\_192X32\_1X instead

{

public:

U8GLIB\_ST7920\_192X32(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_ST7920\_192X32(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_ST7920\_192X32(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t cs2, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw, reset)

{ }

};

class U8GLIB\_ST7920\_192X32\_1X : public U8GLIB

{

public:

U8GLIB\_ST7920\_192X32\_1X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_sw\_spi, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_192X32\_1X(uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_hw\_spi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_192X32\_1X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, U8G\_PIN\_NONE, U8G\_PIN\_NONE, di, rw, reset) // cs1 = cs2 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_ST7920\_192X32\_4X : public U8GLIB

{

public:

U8GLIB\_ST7920\_192X32\_4X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_4x\_sw\_spi, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_192X32\_4X(uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_4x\_hw\_spi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_192X32\_4X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_192x32\_4x\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, U8G\_PIN\_NONE, U8G\_PIN\_NONE, di, rw, reset) // cs1 = cs2 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_ST7920\_202X32 : public U8GLIB

{

public:

U8GLIB\_ST7920\_202X32(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_ST7920\_202X32(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_ST7920\_202X32(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t cs2, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw, reset)

{ }

};

class U8GLIB\_ST7920\_202X32\_1X : public U8GLIB

{

public:

U8GLIB\_ST7920\_202X32\_1X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_sw\_spi, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_202X32\_1X(uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_hw\_spi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_202X32\_1X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, U8G\_PIN\_NONE, U8G\_PIN\_NONE, di, rw, reset) // cs1 = cs2 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_ST7920\_202X32\_4X : public U8GLIB

{

public:

U8GLIB\_ST7920\_202X32\_4X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_4x\_sw\_spi, sck, mosi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_202X32\_4X(uint8\_t cs, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_4x\_hw\_spi, cs, U8G\_PIN\_NONE, reset) // a0 = U8G\_PIN\_NONE

{ }

U8GLIB\_ST7920\_202X32\_4X(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7920\_202x32\_4x\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, U8G\_PIN\_NONE, U8G\_PIN\_NONE, di, rw, reset) // cs1 = cs2 = U8G\_PIN\_NONE

{ }

};

class U8GLIB\_LC7981\_160X80 : public U8GLIB

{

public:

U8GLIB\_LC7981\_160X80(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_lc7981\_160x80\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_LC7981\_240X64 : public U8GLIB

{

public:

U8GLIB\_LC7981\_240X64(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_lc7981\_240x64\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_LC7981\_240X128 : public U8GLIB

{

public:

U8GLIB\_LC7981\_240X128(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_lc7981\_240x128\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

// 16 bit mode required: Remove comment from "#define U8G\_16BIT 1" in utility/utility/u8g.h

class U8GLIB\_LC7981\_320X64 : public U8GLIB

{

public:

U8GLIB\_LC7981\_320X64(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_lc7981\_320x64\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_DOGXL160\_BW : public U8GLIB

{

public:

U8GLIB\_DOGXL160\_BW(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_bw\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGXL160\_BW(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_bw\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_DOGXL160\_GR : public U8GLIB

{

public:

U8GLIB\_DOGXL160\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGXL160\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_gr\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_DOGXL160\_2X\_BW : public U8GLIB

{

public:

U8GLIB\_DOGXL160\_2X\_BW(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_2x\_bw\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGXL160\_2X\_BW(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_2x\_bw\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_DOGXL160\_2X\_GR : public U8GLIB

{

public:

U8GLIB\_DOGXL160\_2X\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_2x\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_DOGXL160\_2X\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_uc1610\_dogxl160\_2x\_gr\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_NHD27OLED\_BW : public U8GLIB

{

public:

U8GLIB\_NHD27OLED\_BW(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_bw\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD27OLED\_BW(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_bw\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_NHD27OLED\_BW(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_bw\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, U8G\_PIN\_NONE, cs, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_NHD27OLED\_2X\_BW : public U8GLIB

{

public:

U8GLIB\_NHD27OLED\_2X\_BW(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_2x\_bw\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD27OLED\_2X\_BW(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_2x\_bw\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_NHD27OLED\_2X\_BW(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_2x\_bw\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, U8G\_PIN\_NONE, cs, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_NHD31OLED\_BW : public U8GLIB

{

public:

U8GLIB\_NHD31OLED\_BW(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_bw\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD31OLED\_BW(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_bw\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_NHD31OLED\_BW(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_bw\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, U8G\_PIN\_NONE, cs, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_NHD31OLED\_2X\_BW : public U8GLIB

{

public:

U8GLIB\_NHD31OLED\_2X\_BW(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_2x\_bw\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD31OLED\_2X\_BW(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_2x\_bw\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_NHD31OLED\_GR : public U8GLIB

{

public:

U8GLIB\_NHD31OLED\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD31OLED\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_gr\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_NHD31OLED\_GR(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_gr\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, U8G\_PIN\_NONE, cs, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_NHD31OLED\_2X\_GR : public U8GLIB

{

public:

U8GLIB\_NHD31OLED\_2X\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_2x\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD31OLED\_2X\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1322\_nhd31oled\_2x\_gr\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1306\_128X64 : public U8GLIB

{

public:

U8GLIB\_SSD1306\_128X64(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x64\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X64(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x64\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X64(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x64\_i2c, options)

{ }

};

class U8GLIB\_SSD1306\_ADAFRUIT\_128X64 : public U8GLIB

{

public:

U8GLIB\_SSD1306\_ADAFRUIT\_128X64(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_adafruit\_128x64\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_ADAFRUIT\_128X64(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_adafruit\_128x64\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_ADAFRUIT\_128X64(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_adafruit\_128x64\_i2c, options)

{ }

};

class U8GLIB\_SSD1306\_128X64\_2X : public U8GLIB

{

public:

U8GLIB\_SSD1306\_128X64\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x64\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X64\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x64\_2x\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X64\_2X(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x64\_2x\_i2c, options)

{ }

};

class U8GLIB\_SH1106\_128X64 : public U8GLIB

{

public:

U8GLIB\_SH1106\_128X64(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_sh1106\_128x64\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SH1106\_128X64(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_sh1106\_128x64\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SH1106\_128X64(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_sh1106\_128x64\_i2c, options)

{ }

};

class U8GLIB\_SH1106\_128X64\_2X : public U8GLIB

{

public:

U8GLIB\_SH1106\_128X64\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_sh1106\_128x64\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SH1106\_128X64\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_sh1106\_128x64\_2x\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SH1106\_128X64\_2X(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_sh1106\_128x64\_2x\_i2c, options)

{ }

};

class U8GLIB\_SSD1309\_128X64 : public U8GLIB

{

public:

U8GLIB\_SSD1309\_128X64(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1309\_128x64\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1309\_128X64(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1309\_128x64\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1309\_128X64(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1309\_128x64\_i2c, options)

{ }

};

class U8GLIB\_SSD1306\_128X32 : public U8GLIB

{

public:

U8GLIB\_SSD1306\_128X32(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x32\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X32(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x32\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X32(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x32\_i2c, options)

{ }

};

class U8GLIB\_SSD1306\_128X32\_2X : public U8GLIB

{

public:

U8GLIB\_SSD1306\_128X32\_2X(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x32\_2x\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X32\_2X(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x32\_2x\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1306\_128X32\_2X(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1306\_128x32\_2x\_i2c, options)

{ }

};

class U8GLIB\_NHD27OLED\_GR : public U8GLIB

{

public:

U8GLIB\_NHD27OLED\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD27OLED\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_gr\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_NHD27OLED\_2X\_GR : public U8GLIB

{

public:

U8GLIB\_NHD27OLED\_2X\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_2x\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_NHD27OLED\_2X\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1325\_nhd27oled\_2x\_gr\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1327\_96X96\_GR : public U8GLIB

{

public:

U8GLIB\_SSD1327\_96X96\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1327\_96x96\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1327\_96X96\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1327\_96x96\_gr\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1327\_96X96\_GR(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1327\_96x96\_gr\_i2c, options)

{ }

};

class U8GLIB\_SSD1327\_96X96\_2X\_GR : public U8GLIB

{

public:

U8GLIB\_SSD1327\_96X96\_2X\_GR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1327\_96x96\_2x\_gr\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1327\_96X96\_2X\_GR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1327\_96x96\_2x\_gr\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_SSD1327\_96X96\_2X\_GR(uint8\_t options = U8G\_I2C\_OPT\_NONE)

: U8GLIB(&u8g\_dev\_ssd1327\_96x96\_2x\_gr\_i2c, options)

{ }

};

class U8GLIB\_LD7032\_60x32 : public U8GLIB

{

public:

U8GLIB\_LD7032\_60x32(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ld7032\_60x32\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_LD7032\_60x32(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ld7032\_60x32\_hw\_spi, cs, a0, reset)

{ }

U8GLIB\_LD7032\_60x32(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ld7032\_60x32\_parallel, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

};

class U8GLIB\_HT1632\_24X16 : public U8GLIB

{

public:

U8GLIB\_HT1632\_24X16(uint8\_t wr, uint8\_t data, uint8\_t cs)

: U8GLIB(&u8g\_dev\_ht1632\_24x16, wr, data, cs, U8G\_PIN\_NONE, U8G\_PIN\_NONE)

{ }

};

class U8GLIB\_PCF8812 : public U8GLIB

{

public:

U8GLIB\_PCF8812(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_pcf8812\_96x65\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_PCF8812(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_pcf8812\_96x65\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_PCD8544 : public U8GLIB

{

public:

U8GLIB\_PCD8544(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_pcd8544\_84x48\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_PCD8544(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_pcd8544\_84x48\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_TLS8204\_84X48 : public U8GLIB

{

public:

U8GLIB\_TLS8204\_84X48(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_tls8204\_84x48\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

};

class U8GLIB\_KS0108\_128 : public U8GLIB

{

public:

U8GLIB\_KS0108\_128(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t cs2, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ks0108\_128x64\_fast, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, cs2, di, rw, reset)

{ }

};

class U8GLIB\_SBN1661\_122X32 : public U8GLIB

{

public:

U8GLIB\_SBN1661\_122X32(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs1, uint8\_t cs2, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_sbn1661\_122x32, d0, d1, d2, d3, d4, d5, d6, d7, U8G\_PIN\_NONE, cs1, cs2, di, rw, reset)

{ }

};

class U8GLIB\_T6963\_240X128 : public U8GLIB

{

public:

U8GLIB\_T6963\_240X128(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t a0, uint8\_t wr, uint8\_t rd, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_t6963\_240x128\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd, reset)

{ }

};

class U8GLIB\_T6963\_128X128 : public U8GLIB

{

public:

U8GLIB\_T6963\_128X128(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t a0, uint8\_t wr, uint8\_t rd, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_t6963\_128x128\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd, reset)

{ }

};

class U8GLIB\_T6963\_240X64 : public U8GLIB

{

public:

U8GLIB\_T6963\_240X64(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t a0, uint8\_t wr, uint8\_t rd, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_t6963\_240x64\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd, reset)

{ }

};

class U8GLIB\_T6963\_128X64 : public U8GLIB

{

public:

U8GLIB\_T6963\_128X64(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t cs, uint8\_t a0, uint8\_t wr, uint8\_t rd, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_t6963\_128x64\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, cs, a0, wr, rd, reset)

{ }

};

class U8GLIB\_ST7687\_C144MVGD: public U8GLIB

{

public:

U8GLIB\_ST7687\_C144MVGD(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_st7687\_c144mvgd\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_ST7687\_C144MVGD(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs, uint8\_t a0, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ks0108\_128x64\_fast, d0, d1, d2, d3, d4, d5, d6, d7, en, cs, U8G\_PIN\_NONE, a0, rw, reset)

{ }

};

class U8GLIB\_ILI9325D\_320x240 : public U8GLIB

{

public:

/\*

U8GLIB\_ILI9325D\_320x240(uint8\_t d0, uint8\_t d1, uint8\_t d2, uint8\_t d3, uint8\_t d4, uint8\_t d5, uint8\_t d6, uint8\_t d7,

uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ili9325d\_320x240\_8bit, d0, d1, d2, d3, d4, d5, d6, d7, en, cs1, U8G\_PIN\_NONE, di, rw, reset)

{ }

\*/

U8GLIB\_ILI9325D\_320x240( uint8\_t en, uint8\_t cs1, uint8\_t di, uint8\_t rw = U8G\_PIN\_NONE, uint8\_t reset = U8G\_PIN\_NONE)

{ init8BitFixedPort(&u8g\_dev\_ili9325d\_320x240\_8bit, en, cs1, di, rw, reset); }

};

class U8GLIB\_SSD1351\_128X128\_332 : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128\_332(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_332\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128\_332(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_332\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128\_4X\_332 : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128\_4X\_332(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_4x\_332\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128\_4X\_332(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_4x\_332\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128GH\_332 : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128GH\_332(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_332\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128GH\_332(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_332\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128GH\_4X\_332 : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128GH\_4X\_332(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_4x\_332\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128GH\_4X\_332(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_4x\_332\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128\_IDX : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128\_IDX(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_idx\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128\_IDX(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_idx\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128\_HICOLOR : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128\_HICOLOR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_hicolor\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128\_HICOLOR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_hicolor\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128\_4X\_HICOLOR : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128\_4X\_HICOLOR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_4x\_hicolor\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128\_4X\_HICOLOR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128\_4x\_hicolor\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128GH\_HICOLOR : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128GH\_HICOLOR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_hicolor\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128GH\_HICOLOR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_hicolor\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_SSD1351\_128X128GH\_4X\_HICOLOR : public U8GLIB

{

public:

U8GLIB\_SSD1351\_128X128GH\_4X\_HICOLOR(uint8\_t sck, uint8\_t mosi, uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_4x\_hicolor\_sw\_spi, sck, mosi, cs, a0, reset)

{ }

U8GLIB\_SSD1351\_128X128GH\_4X\_HICOLOR(uint8\_t cs, uint8\_t a0, uint8\_t reset = U8G\_PIN\_NONE)

: U8GLIB(&u8g\_dev\_ssd1351\_128x128gh\_4x\_hicolor\_hw\_spi, cs, a0, reset)

{ }

};

class U8GLIB\_FLIPDISC\_2X7 : public U8GLIB

{

public:

U8GLIB\_FLIPDISC\_2X7(void) : U8GLIB(&u8g\_dev\_flipdisc\_2x7)

{ }

};

class U8GLIB\_VS : public U8GLIB

{

public:

U8GLIB\_VS(void) : U8GLIB(&u8g\_dev\_vs)

{ }

};

#endif /\* \_CPP\_U8GLIB \*/