/\*!

\* @file Adafruit\_SSD1306.h

\*

\* This is part of for Adafruit's SSD1306 library for monochrome

\* OLED displays: http://www.adafruit.com/category/63\_98

\*

\* These displays use I2C or SPI to communicate. I2C requires 2 pins

\* (SCL+SDA) and optionally a RESET pin. SPI requires 4 pins (MOSI, SCK,

\* select, data/command) and optionally a reset pin. Hardware SPI or

\* 'bitbang' software SPI are both supported.

\*

\* Adafruit invests time and resources providing this open source code,

\* please support Adafruit and open-source hardware by purchasing

\* products from Adafruit!

\*

\* Written by Limor Fried/Ladyada for Adafruit Industries, with

\* contributions from the open source community.

\*

\* BSD license, all text above, and the splash screen header file,

\* must be included in any redistribution.

\*

\*/

#ifndef \_Adafruit\_SSD1306\_H\_

#define \_Adafruit\_SSD1306\_H\_

// ONE of the following three lines must be #defined:

//#define SSD1306\_128\_64 ///< DEPRECTAED: old way to specify 128x64 screen

#define SSD1306\_128\_32 ///< DEPRECATED: old way to specify 128x32 screen

//#define SSD1306\_96\_16 ///< DEPRECATED: old way to specify 96x16 screen

// This establishes the screen dimensions in old Adafruit\_SSD1306 sketches

// (NEW CODE SHOULD IGNORE THIS, USE THE CONSTRUCTORS THAT ACCEPT WIDTH

// AND HEIGHT ARGUMENTS).

// Uncomment to disable Adafruit splash logo

//#define SSD1306\_NO\_SPLASH

#if defined(ARDUINO\_STM32\_FEATHER)

typedef class HardwareSPI SPIClass;

#endif

#include <Adafruit\_GFX.h>

#include <SPI.h>

#include <Wire.h>

#if defined(\_\_AVR\_\_)

typedef volatile uint8\_t PortReg;

typedef uint8\_t PortMask;

#define HAVE\_PORTREG

#elif defined(\_\_SAM3X8E\_\_)

typedef volatile RwReg PortReg;

typedef uint32\_t PortMask;

#define HAVE\_PORTREG

#elif (defined(\_\_arm\_\_) || defined(ARDUINO\_FEATHER52)) && \

!defined(ARDUINO\_ARCH\_MBED) && !defined(ARDUINO\_ARCH\_RP2040)

typedef volatile uint32\_t PortReg;

typedef uint32\_t PortMask;

#define HAVE\_PORTREG

#endif

/// The following "raw" color names are kept for backwards client compatability

/// They can be disabled by predefining this macro before including the Adafruit

/// header client code will then need to be modified to use the scoped enum

/// values directly

#ifndef NO\_ADAFRUIT\_SSD1306\_COLOR\_COMPATIBILITY

#define BLACK SSD1306\_BLACK ///< Draw 'off' pixels

#define WHITE SSD1306\_WHITE ///< Draw 'on' pixels

#define INVERSE SSD1306\_INVERSE ///< Invert pixels

#endif

/// fit into the SSD1306\_ naming scheme

#define SSD1306\_BLACK 0 ///< Draw 'off' pixels

#define SSD1306\_WHITE 1 ///< Draw 'on' pixels

#define SSD1306\_INVERSE 2 ///< Invert pixels

#define SSD1306\_MEMORYMODE 0x20 ///< See datasheet

#define SSD1306\_COLUMNADDR 0x21 ///< See datasheet

#define SSD1306\_PAGEADDR 0x22 ///< See datasheet

#define SSD1306\_SETCONTRAST 0x81 ///< See datasheet

#define SSD1306\_CHARGEPUMP 0x8D ///< See datasheet

#define SSD1306\_SEGREMAP 0xA0 ///< See datasheet

#define SSD1306\_DISPLAYALLON\_RESUME 0xA4 ///< See datasheet

#define SSD1306\_DISPLAYALLON 0xA5 ///< Not currently used

#define SSD1306\_NORMALDISPLAY 0xA6 ///< See datasheet

#define SSD1306\_INVERTDISPLAY 0xA7 ///< See datasheet

#define SSD1306\_SETMULTIPLEX 0xA8 ///< See datasheet

#define SSD1306\_DISPLAYOFF 0xAE ///< See datasheet

#define SSD1306\_DISPLAYON 0xAF ///< See datasheet

#define SSD1306\_COMSCANINC 0xC0 ///< Not currently used

#define SSD1306\_COMSCANDEC 0xC8 ///< See datasheet

#define SSD1306\_SETDISPLAYOFFSET 0xD3 ///< See datasheet

#define SSD1306\_SETDISPLAYCLOCKDIV 0xD5 ///< See datasheet

#define SSD1306\_SETPRECHARGE 0xD9 ///< See datasheet

#define SSD1306\_SETCOMPINS 0xDA ///< See datasheet

#define SSD1306\_SETVCOMDETECT 0xDB ///< See datasheet

#define SSD1306\_SETLOWCOLUMN 0x00 ///< Not currently used

#define SSD1306\_SETHIGHCOLUMN 0x10 ///< Not currently used

#define SSD1306\_SETSTARTLINE 0x40 ///< See datasheet

#define SSD1306\_EXTERNALVCC 0x01 ///< External display voltage source

#define SSD1306\_SWITCHCAPVCC 0x02 ///< Gen. display voltage from 3.3V

#define SSD1306\_RIGHT\_HORIZONTAL\_SCROLL 0x26 ///< Init rt scroll

#define SSD1306\_LEFT\_HORIZONTAL\_SCROLL 0x27 ///< Init left scroll

#define SSD1306\_VERTICAL\_AND\_RIGHT\_HORIZONTAL\_SCROLL 0x29 ///< Init diag scroll

#define SSD1306\_VERTICAL\_AND\_LEFT\_HORIZONTAL\_SCROLL 0x2A ///< Init diag scroll

#define SSD1306\_DEACTIVATE\_SCROLL 0x2E ///< Stop scroll

#define SSD1306\_ACTIVATE\_SCROLL 0x2F ///< Start scroll

#define SSD1306\_SET\_VERTICAL\_SCROLL\_AREA 0xA3 ///< Set scroll range

// Deprecated size stuff for backwards compatibility with old sketches

#if defined SSD1306\_128\_64

#define SSD1306\_LCDWIDTH 128 ///< DEPRECATED: width w/SSD1306\_128\_64 defined

#define SSD1306\_LCDHEIGHT 64 ///< DEPRECATED: height w/SSD1306\_128\_64 defined

#endif

#if defined SSD1306\_128\_32

#define SSD1306\_LCDWIDTH 128 ///< DEPRECATED: width w/SSD1306\_128\_32 defined

#define SSD1306\_LCDHEIGHT 32 ///< DEPRECATED: height w/SSD1306\_128\_32 defined

#endif

#if defined SSD1306\_96\_16

#define SSD1306\_LCDWIDTH 96 ///< DEPRECATED: width w/SSD1306\_96\_16 defined

#define SSD1306\_LCDHEIGHT 16 ///< DEPRECATED: height w/SSD1306\_96\_16 defined

#endif

/\*!

@brief Class that stores state and functions for interacting with

SSD1306 OLED displays.

\*/

class Adafruit\_SSD1306 : public Adafruit\_GFX {

public:

// NEW CONSTRUCTORS -- recommended for new projects

Adafruit\_SSD1306(uint8\_t w, uint8\_t h, TwoWire \*twi = &Wire,

int8\_t rst\_pin = -1, uint32\_t clkDuring = 400000UL,

uint32\_t clkAfter = 100000UL);

Adafruit\_SSD1306(uint8\_t w, uint8\_t h, int8\_t mosi\_pin, int8\_t sclk\_pin,

int8\_t dc\_pin, int8\_t rst\_pin, int8\_t cs\_pin);

Adafruit\_SSD1306(uint8\_t w, uint8\_t h, SPIClass \*spi, int8\_t dc\_pin,

int8\_t rst\_pin, int8\_t cs\_pin, uint32\_t bitrate = 8000000UL);

// DEPRECATED CONSTRUCTORS - for back compatibility, avoid in new projects

Adafruit\_SSD1306(int8\_t mosi\_pin, int8\_t sclk\_pin, int8\_t dc\_pin,

int8\_t rst\_pin, int8\_t cs\_pin);

Adafruit\_SSD1306(int8\_t dc\_pin, int8\_t rst\_pin, int8\_t cs\_pin);

Adafruit\_SSD1306(int8\_t rst\_pin = -1);

~Adafruit\_SSD1306(void);

bool begin(uint8\_t switchvcc = SSD1306\_SWITCHCAPVCC, uint8\_t i2caddr = 0,

bool reset = true, bool periphBegin = true);

void display(void);

void clearDisplay(void);

void invertDisplay(bool i);

void dim(bool dim);

void drawPixel(int16\_t x, int16\_t y, uint16\_t color);

virtual void drawFastHLine(int16\_t x, int16\_t y, int16\_t w, uint16\_t color);

virtual void drawFastVLine(int16\_t x, int16\_t y, int16\_t h, uint16\_t color);

void startscrollright(uint8\_t start, uint8\_t stop);

void startscrollleft(uint8\_t start, uint8\_t stop);

void startscrolldiagright(uint8\_t start, uint8\_t stop);

void startscrolldiagleft(uint8\_t start, uint8\_t stop);

void stopscroll(void);

void ssd1306\_command(uint8\_t c);

bool getPixel(int16\_t x, int16\_t y);

uint8\_t \*getBuffer(void);

protected:

inline void SPIwrite(uint8\_t d) \_\_attribute\_\_((always\_inline));

void drawFastHLineInternal(int16\_t x, int16\_t y, int16\_t w, uint16\_t color);

void drawFastVLineInternal(int16\_t x, int16\_t y, int16\_t h, uint16\_t color);

void ssd1306\_command1(uint8\_t c);

void ssd1306\_commandList(const uint8\_t \*c, uint8\_t n);

SPIClass \*spi; ///< Initialized during construction when using SPI. See

///< SPI.cpp, SPI.h

TwoWire \*wire; ///< Initialized during construction when using I2C. See

///< Wire.cpp, Wire.h

uint8\_t \*buffer; ///< Buffer data used for display buffer. Allocated when

///< begin method is called.

int8\_t i2caddr; ///< I2C address initialized when begin method is called.

int8\_t vccstate; ///< VCC selection, set by begin method.

int8\_t page\_end; ///< not used

int8\_t mosiPin; ///< (Master Out Slave In) set when using SPI set during

///< construction.

int8\_t clkPin; ///< (Clock Pin) set when using SPI set during construction.

int8\_t dcPin; ///< (Data Pin) set when using SPI set during construction.

int8\_t

csPin; ///< (Chip Select Pin) set when using SPI set during construction.

int8\_t rstPin; ///< Display reset pin assignment. Set during construction.

#ifdef HAVE\_PORTREG

PortReg \*mosiPort, \*clkPort, \*dcPort, \*csPort;

PortMask mosiPinMask, clkPinMask, dcPinMask, csPinMask;

#endif

#if ARDUINO >= 157

uint32\_t wireClk; ///< Wire speed for SSD1306 transfers

uint32\_t restoreClk; ///< Wire speed following SSD1306 transfers

#endif

uint8\_t contrast; ///< normal contrast setting for this device

#if defined(SPI\_HAS\_TRANSACTION)

protected:

// Allow sub-class to change

SPISettings spiSettings;

#endif

};

#endif // \_Adafruit\_SSD1306\_H\_