#include "Arduino.h"

#define SSD1306\_I2C\_ADDRESS 0x3C // 011110+SA0+RW - 0x3C or 0x3D

// Address for 128x32 is 0x3C

// Address for 128x32 is 0x3D (default) or 0x3C (if SA0 is grounded)

/\*=========================================================================

SSD1306 Displays

-----------------------------------------------------------------------

The driver is used in multiple displays (128x64, 128x32, etc.).

Select the appropriate display below to create an appropriately

sized framebuffer, etc.

SSD1306\_128\_64 128x64 pixel display

SSD1306\_128\_32 128x32 pixel display

You also need to set the LCDWIDTH and LCDHEIGHT defines to an

appropriate size

-----------------------------------------------------------------------\*/

#define SSD1306\_128\_64

// #define SSD1306\_128\_32

/\*=========================================================================\*/

#if defined SSD1306\_128\_64 && defined SSD1306\_128\_32

#error "Only one SSD1306 display can be specified at once in SSD1306.h"

#endif

#if !defined SSD1306\_128\_64 && !defined SSD1306\_128\_32

#error "At least one SSD1306 display must be specified in SSD1306.h"

#endif

#if defined SSD1306\_128\_64

#define SSD1306\_LCDWIDTH 128

#define SSD1306\_LCDHEIGHT 64

#endif

#if defined SSD1306\_128\_32

#define SSD1306\_LCDWIDTH 128

#define SSD1306\_LCDHEIGHT 32

#endif

#define SSD1306\_SETCONTRAST 0x81

#define SSD1306\_DISPLAYALLON\_RESUME 0xA4

#define SSD1306\_DISPLAYALLON 0xA5

#define SSD1306\_NORMALDISPLAY 0xA6

#define SSD1306\_INVERTDISPLAY 0xA7

#define SSD1306\_DISPLAYOFF 0xAE

#define SSD1306\_DISPLAYON 0xAF

#define SSD1306\_SETDISPLAYOFFSET 0xD3

#define SSD1306\_SETCOMPINS 0xDA

#define SSD1306\_SETVCOMDETECT 0xDB

#define SSD1306\_SETDISPLAYCLOCKDIV 0xD5

#define SSD1306\_SETPRECHARGE 0xD9

#define SSD1306\_SETMULTIPLEX 0xA8

#define SSD1306\_SETLOWCOLUMN 0x00

#define SSD1306\_SETHIGHCOLUMN 0x10

#define SSD1306\_SETSTARTLINE 0x40

#define SSD1306\_MEMORYMODE 0x20

#define SSD1306\_COMSCANINC 0xC0

#define SSD1306\_COMSCANDEC 0xC8

#define SSD1306\_SEGREMAP 0xA0

#define SSD1306\_CHARGEPUMP 0x8D

#define SSD1306\_EXTERNALVCC 0x1

#define SSD1306\_SWITCHCAPVCC 0x2

// Scrolling #defines

#define SSD1306\_ACTIVATE\_SCROLL 0x2F

#define SSD1306\_DEACTIVATE\_SCROLL 0x2E

#define SSD1306\_SET\_VERTICAL\_SCROLL\_AREA 0xA3

#define SSD1306\_RIGHT\_HORIZONTAL\_SCROLL 0x26

#define SSD1306\_LEFT\_HORIZONTAL\_SCROLL 0x27

#define SSD1306\_VERTICAL\_AND\_RIGHT\_HORIZONTAL\_SCROLL 0x29

#define SSD1306\_VERTICAL\_AND\_LEFT\_HORIZONTAL\_SCROLL 0x2A

class SSD1306 {

public:

SSD1306(int8\_t SCLK, int8\_t DC, int8\_t RST, int8\_t CS);

SSD1306(int8\_t RST = 4);

void begin(uint8\_t switchvcc = SSD1306\_SWITCHCAPVCC, uint8\_t i2caddr = SSD1306\_I2C\_ADDRESS);

void ssd1306\_command(uint8\_t c);

void ssd1306\_data(uint8\_t c);

void invertDisplay(uint8\_t i);

void draw8x8(byte\* buffer, byte x, byte y);

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 fill(unsigned char dat);

void clearBuffer();

protected:

uint8\_t \_i2caddr;

private:

int8\_t sclk, dc, rst, cs;

};