/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Arduino Text & Bitmap Display Library for multiple models of monochrome LCD display

\* Distributed under GPL v2.0

\* Copyright (c) 2013-2014 Stanley Huang <stanleyhuangyc@gmail.com>

\* All rights reserved.

\* For more information, please visit http://arduinodev.com

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <Arduino.h>

//#define MEMORY\_SAVING

typedef enum {

FONT\_SIZE\_SMALL = 0,

FONT\_SIZE\_MEDIUM,

FONT\_SIZE\_LARGE,

FONT\_SIZE\_XLARGE

} FONT\_SIZE;

#define FLAG\_PAD\_ZERO 1

#define FLAG\_PIXEL\_DOUBLE\_H 2

#define FLAG\_PIXEL\_DOUBLE\_V 4

#define FLAG\_PIXEL\_DOUBLE (FLAG\_PIXEL\_DOUBLE\_H | FLAG\_PIXEL\_DOUBLE\_V)

extern const PROGMEM unsigned char font5x8[][5];

extern const PROGMEM unsigned char digits8x8[][8] ;

extern const PROGMEM unsigned char digits16x16[][32];

extern const PROGMEM unsigned char digits16x24[][48];

extern const PROGMEM unsigned char font8x16\_doslike[][16];

extern const PROGMEM unsigned char font8x16\_terminal[][16];

class LCD\_Common

{

public:

LCD\_Common():m\_font(FONT\_SIZE\_SMALL),m\_flags(0) {}

void setFontSize(FONT\_SIZE size) { m\_font = size; }

void setFlags(byte flags) { m\_flags = flags; }

virtual void backlight(bool on) {}

virtual void draw(const PROGMEM byte\* buffer, byte width, byte height) {}

void printInt(uint16\_t value, int8\_t padding = -1);

void printLong(uint32\_t value, int8\_t padding = -1);

protected:

virtual void writeDigit(byte n) {}

byte m\_font;

byte m\_flags;

};

class LCD\_Null : public LCD\_Common, public Print

{

public:

byte getLines() { return 0; }

byte getCols() { return 0; }

void clearLine(byte line) {}

void clear() {}

void begin() {}

void setCursor(byte column, byte line) {}

size\_t write(uint8\_t c) { return 0; }

};

#include "SSD1306.h"

class LCD\_SSD1306 : public LCD\_Common, public SSD1306, public Print

{

public:

void setCursor(byte column, byte line);

void setContrast(byte Contrast);

void draw(const PROGMEM byte\* buffer, byte width, byte height);

size\_t write(uint8\_t c);

void clear(byte x = 0, byte y = 0, byte width = 128, byte height = 64);

void clearLine(byte line);

byte getLines() { return 21; }

byte getCols() { return 8; }

private:

void writeDigit(byte n);

byte m\_col;

byte m\_row;

};

class LCD\_SH1106 : public LCD\_Common, public Print

{

public:

void begin();

void setCursor(byte column, byte line);

void draw(const PROGMEM byte\* buffer, byte width, byte height);

size\_t write(uint8\_t c);

void clear(byte x = 0, byte y = 0, byte width = 128, byte height = 64);

void clearLine(byte line);

byte getLines() { return 21; }

byte getCols() { return 8; }

private:

void WriteCommand(unsigned char ins);

void WriteData(unsigned char dat);

void writeDigit(byte n);

byte m\_col;

byte m\_row;

};

#include "PCD8544.h"

class LCD\_PCD8544 : public LCD\_Common, public PCD8544

{

public:

byte getLines() { return 6; }

byte getCols() { return 14; }

void backlight(bool on)

{

pinMode(7, OUTPUT);

digitalWrite(7, on ? HIGH : LOW);

}

void clearLine(byte line)

{

setCursor(0, line);

for (byte i = 14; i > 0; i--) write(' ');

}

void draw(const PROGMEM byte\* buffer, byte width, byte height);

private:

void writeDigit(byte n);

};