/\*

\* PCD8544 - Interface with Philips PCD8544 (or compatible) LCDs.

\*

\* Copyright (c) 2010 Carlos Rodrigues <cefrodrigues@gmail.com>

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy

\* of this software and associated documentation files (the "Software"), to deal

\* in the Software without restriction, including without limitation the rights

\* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

\* copies of the Software, and to permit persons to whom the Software is

\* furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in

\* all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

\* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

\* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

\* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

\* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

\* THE SOFTWARE.

\*/

#ifndef PCD8544\_H

#define PCD8544\_H

#if ARDUINO < 100

#include <WProgram.h>

#else

#include <Arduino.h>

#endif

// Chip variants supported...

#define CHIP\_PCD8544 0

#define CHIP\_ST7576 1

#define PCD8544\_WIDTH 84

#define PCD8544\_HEIGHT 48

#define PCD8544\_CMD LOW

#define PCD8544\_DATA HIGH

class PCD8544: public Print {

public:

// All the pins can be changed from the default values...

PCD8544(unsigned char sclk = 2, /\* clock (display pin 2) \*/

unsigned char sdin = 3, /\* data-in (display pin 3) \*/

unsigned char dc = 4, /\* data select (display pin 4) \*/

unsigned char reset = 6, /\* reset (display pin 8) \*/

unsigned char sce = 5); /\* enable (display pin 5) \*/

// Display initialization (dimensions in pixels)...

void begin(unsigned char model=CHIP\_PCD8544);

void stop();

// Erase everything on the display...

void clear();

void clearLine(); // ...or just the current line

// Control the display's power state...

void setPower(bool on);

// For compatibility with the LiquidCrystal library...

void display();

void noDisplay();

// Activate white-on-black mode (whole display)...

void setInverse(bool inverse);

// Place the cursor at the start of the current line...

void home();

// Place the cursor at position (column, line)...

void setCursor(unsigned char column, unsigned char line);

// Assign a user-defined glyph (5x8) to an ASCII character (0-31)...

void createChar(unsigned char chr, const unsigned char \*glyph);

// Write an ASCII character at the current cursor position (7-bit)...

#if ARDUINO < 100

virtual void write(uint8\_t chr);

#else

virtual size\_t write(uint8\_t chr);

#endif

// Draw a chart element at the current cursor position...

void drawColumn(unsigned char lines, unsigned char value);

void draw8x8(const unsigned char \*data);

void draw16x16(const unsigned char \*data);

protected:

// Current cursor position...

unsigned char column;

unsigned char line;

// Send a command or data to the display...

void send(unsigned char type, unsigned char data);

private:

unsigned char pin\_sclk;

unsigned char pin\_sdin;

unsigned char pin\_dc;

unsigned char pin\_reset;

unsigned char pin\_sce;

// User-defined glyphs (below the ASCII space character)...

const unsigned char \*custom[' '];

};

#endif /\* PCD8544\_H \*/

/\* vim: set expandtab ts=4 sw=4: \*/