/\*

\* LedControl.h - A library for controling Leds with a MAX7219/MAX7221

\* Copyright (c) 2007 Eberhard Fahle

\*

\* 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:

\*

\* 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 LedControl\_h

#define LedControl\_h

#include <avr/pgmspace.h>

#if (ARDUINO >= 100)

#include <Arduino.h>

#else

#include <WProgram.h>

#endif

/\*

\* Segments to be switched on for characters and digits on

\* 7-Segment Displays

\*/

const static byte charTable [] PROGMEM = {

B01111110,B00110000,B01101101,B01111001,B00110011,B01011011,B01011111,B01110000,

B01111111,B01111011,B01110111,B00011111,B00001101,B00111101,B01001111,B01000111,

B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,

B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,

B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,

B00000000,B00000000,B00000000,B00000000,B10000000,B00000001,B10000000,B00000000,

B01111110,B00110000,B01101101,B01111001,B00110011,B01011011,B01011111,B01110000,

B01111111,B01111011,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,

B00000000,B01110111,B00011111,B00001101,B00111101,B01001111,B01000111,B00000000,

B00110111,B00000000,B00000000,B00000000,B00001110,B00000000,B00000000,B00000000,

B01100111,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,

B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00001000,

B00000000,B01110111,B00011111,B00001101,B00111101,B01001111,B01000111,B00000000,

B00110111,B00000000,B00000000,B00000000,B00001110,B00000000,B00010101,B00011101,

B01100111,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,

B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000

};

class LedControl {

private :

/\* The array for shifting the data to the devices \*/

byte spidata[16];

/\* Send out a single command to the device \*/

void spiTransfer(int addr, byte opcode, byte data);

/\* We keep track of the led-status for all 8 devices in this array \*/

byte status[64];

/\* Data is shifted out of this pin\*/

int SPI\_MOSI;

/\* The clock is signaled on this pin \*/

int SPI\_CLK;

/\* This one is driven LOW for chip selectzion \*/

int SPI\_CS;

/\* The maximum number of devices we use \*/

int maxDevices;

public:

/\*

\* Create a new controler

\* Params :

\* dataPin pin on the Arduino where data gets shifted out

\* clockPin pin for the clock

\* csPin pin for selecting the device

\* numDevices maximum number of devices that can be controled

\*/

LedControl(int dataPin, int clkPin, int csPin, int numDevices=1);

/\*

\* Gets the number of devices attached to this LedControl.

\* Returns :

\* int the number of devices on this LedControl

\*/

int getDeviceCount();

/\*

\* Set the shutdown (power saving) mode for the device

\* Params :

\* addr The address of the display to control

\* status If true the device goes into power-down mode. Set to false

\* for normal operation.

\*/

void shutdown(int addr, bool status);

/\*

\* Set the number of digits (or rows) to be displayed.

\* See datasheet for sideeffects of the scanlimit on the brightness

\* of the display.

\* Params :

\* addr address of the display to control

\* limit number of digits to be displayed (1..8)

\*/

void setScanLimit(int addr, int limit);

/\*

\* Set the brightness of the display.

\* Params:

\* addr the address of the display to control

\* intensity the brightness of the display. (0..15)

\*/

void setIntensity(int addr, int intensity);

/\*

\* Switch all Leds on the display off.

\* Params:

\* addr address of the display to control

\*/

void clearDisplay(int addr);

/\*

\* Set the status of a single Led.

\* Params :

\* addr address of the display

\* row the row of the Led (0..7)

\* col the column of the Led (0..7)

\* state If true the led is switched on,

\* if false it is switched off

\*/

void setLed(int addr, int row, int col, boolean state);

/\*

\* Set all 8 Led's in a row to a new state

\* Params:

\* addr address of the display

\* row row which is to be set (0..7)

\* value each bit set to 1 will light up the

\* corresponding Led.

\*/

void setRow(int addr, int row, byte value);

/\*

\* Set all 8 Led's in a column to a new state

\* Params:

\* addr address of the display

\* col column which is to be set (0..7)

\* value each bit set to 1 will light up the

\* corresponding Led.

\*/

void setColumn(int addr, int col, byte value);

/\*

\* Display a hexadecimal digit on a 7-Segment Display

\* Params:

\* addr address of the display

\* digit the position of the digit on the display (0..7)

\* value the value to be displayed. (0x00..0x0F)

\* dp sets the decimal point.

\*/

void setDigit(int addr, int digit, byte value, boolean dp);

/\*

\* Display a character on a 7-Segment display.

\* There are only a few characters that make sense here :

\* '0','1','2','3','4','5','6','7','8','9','0',

\* 'A','b','c','d','E','F','H','L','P',

\* '.','-','\_',' '

\* Params:

\* addr address of the display

\* digit the position of the character on the display (0..7)

\* value the character to be displayed.

\* dp sets the decimal point.

\*/

void setChar(int addr, int digit, char value, boolean dp);

};

#endif //LedControl.h