

```

/*
 * lcd-jan.c
 *   Programmet modifierat 2014-05-31 av Jan Pihlgren
 *   Kompilera med:
 *   make lcd-jan
 *   Starta med:
 *   lcd-jan 1
 *   för att få full ljusstyrka från början.
 *
 *   Text-based LCD driver test code
 *   This is designed to drive the Adafruit RGB LCD Plate
 *   with the additional 5 buttons for the Raspberry Pi
 *
 *   Copyright (c) 2012-2013 Gordon Henderson.
 *   *****
 *   This file is part of wiringPi:
 *   https://projects.drogon.net/raspberry-pi/wiringpi/
 *
 *   wiringPi is free software: you can redistribute it and/or modify
 *   it under the terms of the GNU Lesser General Public License as published by
 *   the Free Software Foundation, either version 3 of the License, or
 *   (at your option) any later version.
 *
 *   wiringPi is distributed in the hope that it will be useful,
 *   but WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Lesser General Public License for more details.
 *
 *   You should have received a copy of the GNU Lesser General Public License
 *   along with wiringPi. If not, see <http://www.gnu.org/licenses/>.
 *   *****
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>

#include <wiringPi.h>
#include <mcp23017.h>
#include <lcd.h>

#ifdef TRUE
#define TRUE (1==1)
#define FALSE (1==2)
#endif

// Defines for the Adafruit Pi LCD interface board

#define AF_BASE          100
#define AF_RED           (AF_BASE + 6)
#define AF_GREEN         (AF_BASE + 7)
#define AF_BLUE          (AF_BASE + 8)

#define AF_E             (AF_BASE + 13)
#define AF_RW            (AF_BASE + 14)
#define AF_RS            (AF_BASE + 15)

#define AF_DB4           (AF_BASE + 12)
#define AF_DB5           (AF_BASE + 11)
#define AF_DB6           (AF_BASE + 10)
#define AF_DB7           (AF_BASE + 9)

#define AF_SELECT        (AF_BASE + 0)
#define AF_RIGHT         (AF_BASE + 1)
#define AF_DOWN          (AF_BASE + 2)
#define AF_UP            (AF_BASE + 3)

```

```

#define      AF_LEFT                (AF_BASE + 4)

#define      SIZE 256

// Global lcd handle:

static int lcdHandle ;

/*
 * usage:
 * *****
 */

int usage (const char *progName)
{
    fprintf (stderr, "Usage: %s colour\n", progName) ;
    return EXIT_FAILURE ;
}

/*
 * scrollMessage:
 * *****
 */

static const char *message =
    "
    "Modifierad av Jan Pihlgren/"
    "
    " ;

void scrollMessage (int line, int width)
{
    char buf [32] ;
    static int position = 0 ;
    static int timer = 0 ;

    if (millis () < timer)
        return ;

    timer = millis () + 200 ;

    strncpy (buf, &message [position], width) ;
    buf [width] = 0 ;
    lcdPosition (lcdHandle, 0, line) ;
    lcdPuts (lcdHandle, buf) ;

    if (++position == (strlen (message) - width))
        position = 0 ;
}

/*
 * setBacklightColour:
 * The colour outputs are inverted.
 * *****
 */

static void setBacklightColour (int colour)
{
    //colour &= 7 ;
    colour &= 1 ;

    digitalWrite (AF_RED, !(colour & 1)) ;
    digitalWrite (AF_GREEN, !(colour & 2)) ;
    digitalWrite (AF_BLUE, !(colour & 4)) ;
}

/*

```

```

* adafruitLCDSetup:
*   Setup the Adafruit board by making sure the additional pins are
*   set to the correct modes, etc.
*****
*/

static void adafruitLCDSetup (int colour)
{
    int i ;

//    Backlight LEDs

    pinMode (AF_RED,    OUTPUT) ;
    pinMode (AF_GREEN,  OUTPUT) ;
    pinMode (AF_BLUE,   OUTPUT) ;
    setBacklightColour (colour) ;

//    Input buttons

    for (i = 0 ; i <= 4 ; ++i)
    {
        pinMode (AF_BASE + i, INPUT) ;
        pullUpDnControl (AF_BASE + i, PUD_UP) ; // Enable pull-ups, switches close to 0v
    }

// Control signals

    pinMode (AF_RW, OUTPUT) ; digitalWrite (AF_RW, LOW) ; // Not used with wiringPi - always in
write mode

// The other control pins are initialised with lcdInit ()

    lcdHandle = lcdInit (2, 16, 4, AF_RS, AF_E, AF_DB4,AF_DB5,AF_DB6,AF_DB7, 0,0,0,0) ;

    if (lcdHandle < 0)
    {
        fprintf (stderr, "lcdInit failed\n") ;
        exit (EXIT_FAILURE) ;
    }
}

int main (int argc, char *argv[])
{
    int colour ;
    int cols = 16 ;
    int waitForRelease = FALSE ;
    char buffer[SIZE];

    struct tm *t ;
    time_t tim ;
    time_t curtime;

    struct tm *loctime;
    char buf [32] ;

    if (argc != 2)
        return usage (argv [0]) ;

    printf ("\nRaspberry Pi Adafruit LCD test\n") ;
    printf ("=====\n") ;

    colour = atoi (argv [1]) ;

    wiringPiSetupSys () ;
    mcp23017Setup (AF_BASE, 0x20) ;

    adafruitLCDSetup (colour) ;

    for (;;)

```

```

{
    // Datum
    curtime = time( NULL ) ;
    localtime = localtime(&curtime);
    strftime (buffer,SIZE," %Y-%m-%d", localtime);
    lcdPosition (lcdHandle, 2, 0);
    lcdPuts(lcdHandle,buffer);
    // Klockslog
    tim = time (NULL) ;
    t = localtime (&tim) ;

    sprintf (buf, "%02d:%02d:%02d", t->tm_hour, t->tm_min, t->tm_sec) ;

    lcdPosition (lcdHandle, (cols - 8) / 2, 1) ;
    lcdPuts (lcdHandle, buf) ;

// Check buttons to cycle colour
// If Up or Down are still pushed, then skip

    if (waitForRelease)
    {
        if ((digitalRead (AF_UP) == LOW) || (digitalRead (AF_DOWN) == LOW))
            continue ;
        else
            waitForRelease = FALSE ;
    }

    if (digitalRead (AF_UP) == LOW) // Pushed
    {
        colour = colour + 1 ;
        if (colour == 8)
            colour = 0 ;
        setBacklightColour (colour) ;
        waitForRelease = TRUE ;
    }

    if (digitalRead (AF_DOWN) == LOW) // Pushed
    {
        colour = colour - 1 ;
        if (colour == -1)
            colour = 7 ;
        setBacklightColour (colour) ;
        waitForRelease = TRUE ;
    }

    if (digitalRead (AF_LEFT) == LOW) // Stäng programmet
    {
        lcdClear (lcdHandle) ;
        adafruitLCDSetup (4) ;
        exit(0);
    }
}

return 0 ;
}

```