

rgb_backlight.c

```
1 // RGB Backlight PWM Example
2 // Jason Losh
3
4 //-----
5 // Hardware Target
6 //-----
7
8 // Target Platform: EK-TM4C123GXL Evaluation Board
9 // Target uC:      TM4C123GH6PM
10 // System Clock:   40 MHz
11
12 // Hardware configuration:
13 // Red Backlight LED:
14 //   MOPWM3 (PB5) drives an NPN transistor that powers the red LED
15 // Green Backlight LED:
16 //   MOPWM5 (PE5) drives an NPN transistor that powers the green LED
17 // Blue Backlight LED:
18 //   MOPWM4 (PE4) drives an NPN transistor that powers the blue LED
19 // ST7565R Graphics LCD Display Interface:
20 //   MOSI (SSI2Tx) on PB7
21 //   MISO (SSI2Rx) is not used by the LCD display but the pin is used for GPIO for A0
22 //   SCLK (SSI2Clk) on PB4
23 //   A0 connected to PB6
24 //   ~CS connected to PB1
25
26 //-----
27 // Device includes, defines, and assembler directives
28 //-----
29
30 #include <stdint.h>
31 #include <stdbool.h>
32 #include <string.h>
33 #include "tm4c123gh6pm.h"
34 #include "wait.h"
35 #include "graphics_lcd.h"
36
37 //-----
38 // Subroutines
39 //-----
40
41 // Blocking function that returns only when SW1 is pressed
42 // Initialize Hardware
43 void initHw()
44 {
45     // Configure HW to work with 16 MHz XTAL, PLL enabled, system clock of 40 MHz
46     // PWM is system clock / 2
47     SYSCTL_RCC_R = SYSCTL_RCC_XTAL_16MHZ | SYSCTL_RCC_OSCSRC_MAIN | SYSCTL_RCC_USESYSDIV | (4
48     << SYSCTL_RCC_SYSDIV_S)
49     | SYSCTL_RCC_USEPWMDIV | SYSCTL_RCC_PWMDIV_2;
50
51     // Set GPIO ports to use APB (not needed since default configuration -- for clarity)
52     SYSCTL_GPIOHBCTL_R = 0;
53
54     // Enable GPIO port B and E peripherals
55     SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOB | SYSCTL_RCGC2_GPIOE;
```

rgb_backlight.c

```

56 // Configure three backlight LEDs
57 GPIO_PORTB_DIR_R |= 0x20; // make bit5 an output
58 GPIO_PORTB_DR2R_R |= 0x20; // set drive strength to 2mA
59 GPIO_PORTB_DEN_R |= 0x20; // enable bit5 for digital
60 GPIO_PORTB_AFSEL_R |= 0x20; // select auxilary function for bit 5
61 GPIO_PORTB_PCTL_R = GPIO_PCTL_PB5_MOPWM3; // enable PWM on bit 5
62 GPIO_PORTB_PCTL_R |= 0x30; // make bits 4 and 5 outputs
63 GPIO_PORTB_DR2R_R |= 0x30; // set drive strength to 2mA
64 GPIO_PORTB_DEN_R |= 0x30; // enable bits 4 and 5 for digital
65 GPIO_PORTB_AFSEL_R |= 0x30; // select auxilary function for bits 4 and 5
66 GPIO_PORTB_PCTL_R = GPIO_PCTL_PE4_MOPWM4 | GPIO_PCTL_PE5_MOPWM5; // enable PWM on bits 4
and 5
67
68 // Configure PWM module0 to drive RGB backlight
69 // RED on MOPWM3 (PB5), MOPWM1b
70 // BLUE on MOPWM4 (PE4), MOPWM2a
71 // GREEN on MOPWM5 (PE5), MOPWM2b
72 SYSTCTL_RCGCO_R |= SYSTCTL_RCGCO_PWM0; // turn-on PWM0 module
73 __asm(" NOP"); // wait 3 clocks
74 __asm(" NOP");
75 __asm(" NOP");
76 SYSTCTL_SRPWM_R = SYSTCTL_SRPWM_R0; // reset PWM0 module
77 SYSTCTL_SRPWM_R = 0; // leave reset state
78 PWM0_1_CTL_R = 0; // turn-off PWM0 generator 1
79 PWM0_2_CTL_R = 0; // turn-off PWM0 generator 2
80 PWM0_1_GENB_R = PWM0_GENB_ACTCMPBD_ZERO | PWM0_GENB_ACTLOAD_ONE;
81 // output 3 on PWM0, gen 1b, cmpb
82 PWM0_2_GENA_R = PWM0_GENA_ACTCMPAD_ZERO | PWM0_GENA_ACTLOAD_ONE;
83 // output 4 on PWM0, gen 2a, cmpa
84 PWM0_2_GENB_R = PWM0_GENB_ACTCMPBD_ZERO | PWM0_GENB_ACTLOAD_ONE;
85 // output 5 on PWM0, gen 2b, cmpb
86 PWM0_1_LOAD_R = 1024; // set period to 40 MHz sys clock / 2 /
1024 = 19.53125 kHz
87 PWM0_2_LOAD_R = 1024;
88 PWM0_INVERT_R = PWM0_INVERT_PWM3INV | PWM0_INVERT_PWM4INV | PWM0_INVERT_PWM5INV;
89 // invert outputs for duty cycle
increases with increasing compare values
90 PWM0_1_CMPB_R = 0; // red off (0=always low, 1023=always
high)
91 PWM0_2_CMPB_R = 0; // green off
92 PWM0_2_CMPA_R = 0; // blue off
93
94 PWM0_1_CTL_R = PWM0_CTL_ENABLE; // turn-on PWM0 generator 1
95 PWM0_2_CTL_R = PWM0_CTL_ENABLE; // turn-on PWM0 generator 2
96 PWM0_ENABLE_R = PWM0_ENABLE_PWM3EN | PWM0_ENABLE_PWM4EN | PWM0_ENABLE_PWM5EN;
97 // enable outputs
98
99 // Configure A0 and ~CS for graphics LCD
100 GPIO_PORTB_DIR_R |= 0x42; // make bits 1 and 6 outputs
101 GPIO_PORTB_DR2R_R |= 0x42; // set drive strength to 2mA
102 GPIO_PORTB_DEN_R |= 0x42; // enable bits 1 and 6 for digital
103
104 // Configure SS12 pins for SPI configuration
105 SYSTCTL_RCGCSSI_R |= SYSTCTL_RCGCSSI_R2; // turn-on SS12 clocking
106 GPIO_PORTB_DIR_R |= 0x90; // make bits 4 and 7 outputs
107 GPIO_PORTB_DR2R_R |= 0x90; // set drive strength to 2mA

```

rgb_backlight.c

```

108     GPIO_PORTB_AFSEL_R |= 0x90;           // select alternative functions for MOSI,
        SCLK pins
109     GPIO_PORTB_PCTL_R |= GPIO_PCTL_PB7_SSI2TX | GPIO_PCTL_PB4_SSI2CLK; // map alt fns to SSI2
110     GPIO_PORTB_DEN_R |= 0x90;           // enable digital operation on TX, CLK
        pins
111
112     // Configure the SSI2 as a SPI master, mode 3, 8bit operation, 1 MHz bit rate
113     SSI2_CR1_R &= ~SSI_CR1_SSE;         // turn off SSI2 to allow
        re-configuration
114     SSI2_CR1_R = 0;                     // select master mode
115     SSI2_CC_R = 0;                     // select system clock as the clock
        source
116     SSI2_CPSR_R = 40;                   // set bit rate to 1 MHz (if SR=0 in CRO)
117     SSI2_CRO_R = SSI_CRO_SPH | SSI_CRO_SPO | SSI_CRO_FRF_MOTO | SSI_CRO_DSS_8; // set SR=0,
        mode 3 (SPH=1, SPO=1), 8-bit
118     SSI2_CR1_R |= SSI_CR1_SSE;         // turn on SSI2
119 }
120
121 setRgbColor(uint16_t red, uint16_t green, uint16_t blue)
122 {
123     PWMO_1_CMPB_R = red;
124     PWMO_2_CMPA_R = blue;
125     PWMO_2_CMPB_R = green;
126 }
127
128 //-----
129 // Main
130 //-----
131
132 int main(void)
133 {
134     // Initialize hardware
135     initHw();
136
137     // Initialize graphics LCD
138     initGraphicsLcd();
139
140     // Turn on all pixels for maximum light transmission
141     drawGraphicsLcdRectangle(0, 0, 128, 64, SET);
142
143     // Cycle through colors
144     int16_t i = 0;
145     while(1)
146     {
147         // Backlight off
148         setRgbColor(0, 0, 0);
149         waitMicrosecond(1000000);
150         // Ramp from off to bright white
151         for (i = 0; i < 1024; i++)
152         {
153             setRgbColor(i, i, i);
154             waitMicrosecond(10000);
155         }
156         // Red
157         setRgbColor(1023, 0, 0);
158         waitMicrosecond(1000000);

```

rgb_backlight.c

```
159 // Orange
160 setRgbColor(1023, 384, 0);
161 waitMicrosecond(1000000);
162 // Yellow
163 setRgbColor(1023, 1023, 8);
164 waitMicrosecond(1000000);
165 // Green
166 setRgbColor(0, 1023, 0);
167 waitMicrosecond(1000000);
168 // Cyan
169 setRgbColor(0, 1023, 1023);
170 waitMicrosecond(1000000);
171 // Blue
172 setRgbColor(0, 0, 1023);
173 waitMicrosecond(1000000);
174 // Magenta
175 setRgbColor(1023, 0, 1023);
176 waitMicrosecond(1000000);
177 }
178 }
179
```