



北京大学
PEKING UNIVERSITY

嵌入式系统编程与实践课
程讨论

K210 Liquid Crystal Display Image

例程编号：2.6

汇报学生：范世昌

1

```
25  static void io_set_power(void)
26  {
27      sysctl_set_power_mode(SYSCTL_POWER_BANK6, SYSCTL_POWER_V18);
28      sysctl_set_power_mode(SYSCTL_POWER_BANK7, SYSCTL_POWER_V18);
29  }
```

- 功能：设置指定电源的电压，供给LCD显示屏和相关接口
- 具体操作：使用 `sysctl_set_power_mode` 函数将 `SYSCTL_POWER_BANK6` 和 `SYSCTL_POWER_BANK7` 的电压设置为 1.8V

52	LCD_D7	1.8V	LCD数据专用引脚D7
53	LCD_D6	1.8V	LCD数据专用引脚D6
54	LCD_D5	1.8V	LCD数据专用引脚D5
55	LCD_D4	1.8V	LCD数据专用引脚D4
56	LCD_D3	1.8V	LCD数据专用引脚D3
57	LCD_D2	1.8V	LCD数据专用引脚D2
58	LCD_D1	1.8V	LCD数据专用引脚D1
59	LCD_D0	1.8V	LCD数据专用引脚D0

Change



```
(SYSCTL_POWER_BANK0, SYSCTL_POWER_V18);
(SYSCTL_POWER_BANK1, SYSCTL_POWER_V18);
```

2

```
31 static void io_init(void)
32 {
33     fpioa_set_function(LCD_DC_PIN, FUNC_GPIOHS0 + LCD_DC_IO);
34     fpioa_set_function(LCD_CS_PIN, FUNC_SPI0_SS3);
35     fpioa_set_function(LCD_RW_PIN, FUNC_SPI0_SCLK);
36     fpioa_set_function(LCD_RST_PIN, FUNC_GPIOHS0 + LCD_RST_IO);
37
38     sysctl_set_spi0_dvp_data(1);
39
40     // LCD Backlight
41     fpioa_set_function(LCD_BLIGHT_PIN, FUNC_GPIOHS0 + LCD_BLIGHT_IO);
42     gpiohs_set_drive_mode(LCD_BLIGHT_IO, GPIO_DM_OUTPUT);
43     gpiohs_set_pin(LCD_BLIGHT_IO, GPIO_PV_LOW);
44 }
```

- 功能：设置 LCD 屏幕控制所需的 GPIO 和 SPI 引脚，以及背光控制
- 具体操作：
 - (1) fpioa_set_function 函数用于将功能引脚映射到具体的GPIO引脚上。例如，LCD_DC_PIN 被映射到 FUNC_GPIOHS0 + LCD_DC_IO，是 LCD 的数据控制引脚。并设置 LCD 的芯片选择引脚 LCD_CS_PIN、读写引脚LCD_RW_PIN和复位引脚LCD_RST_PIN
 - (2) 调用 sysctl_set_spi0_dvp_data(1) 设置 SPI 接口的工作模式，以便与DVP接口传输数据
 - (3)设置背光引脚的驱动模式为输出模式，表示该引脚可以向外输出电压信号，从而可以控制 LCD 背光的开关。其将背光电平设置为低，即关闭背光

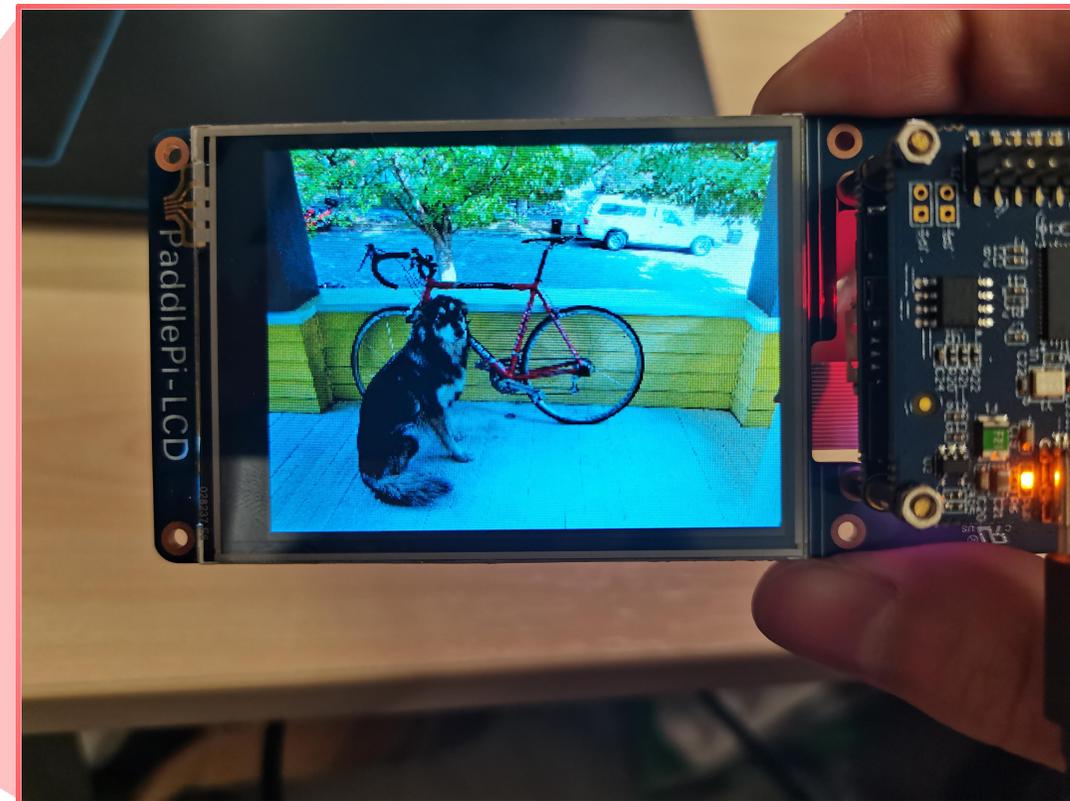
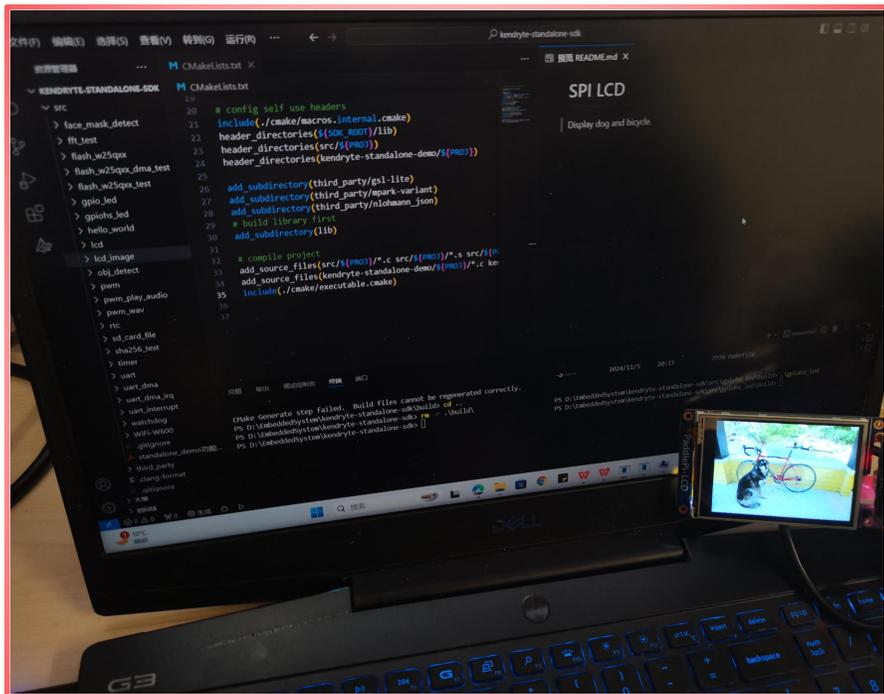
1

```
6  v uint32_t rgb_image[] __attribute__((aligned(64))) = {
7      0x39c639e6, 0x39e639c6, 0x39e639c6, 0x39e639e6, 0x39e639e6, 0x3a063a06, 0x3a064207, 0x3a277bcc,
8      0x950b9d4d, 0x84689d4c, 0x9d0ba60e, 0x8dcd9e10, 0x9dae8ccc, 0xa5d1848c, 0xc653846c, 0x8d2c7cea,
9      0x6be85ac7, 0x9d11950e, 0x8c4d8529, 0x8d2e8c2a, 0x94ed8c6c, 0xd6d83b24, 0x52c884ca, 0x94eebd92,
10     0xd615cdf2, 0xde94a58d, 0xc6518cca, 0x8d4b84eb, 0x7c8a52e4, 0x840b844b, 0xc67294ee, 0xa52facce,
11     0xad0d9d6d, 0x9e2d7d6b, 0x8d0ebe91, 0x8da974a7, 0xcf0d7b1, 0xb5ee8586, 0xbeececf4, 0xeff8cf72,
12     0xd7d3bf30, 0xbeb0ae8d, 0xae2bd714, 0xc72fbe71, 0xae2d958d, 0x956ddf34, 0x84aab6d0, 0x9e0b95eb,
13     0x5ca8a54f, 0x53044a84, 0x9d6da40e, 0x94eca58d, 0x9ccc7bc7, 0xcdb2ce50, 0xa5ec7ca8, 0xa68ca6e9,
14     0x64e38e87, 0x8dca962a, 0x3ae0ae2f, 0x962c8dcb, 0x7c699dca, 0x6c85c753, 0xdff4bf2f, 0xb76f8daa,
15     0xa5918ceb, 0x9d6b8444, 0xb630d6f6, 0xd6d5e7b9, 0xc6f1d7b2, 0xe7f74bc1, 0xc752a6ed, 0x9eccccf92,
```

...

- `uint32_t rgb_image[]`: 定义一个 `rgb_image` 数组，用于存储图像的 RGB 数据，每个像素为 32 位的无符号整数，4（16进制，每个数代表4个2进制）*8（8个数），用于存储 24 位颜色（8 位红色、8 位绿色、8 位蓝色和 8 位透明度）。在这种情况下，RGBA 色彩空间通常是这样的编码：8 位红色通道、8 位绿色通道、8 位蓝色通道和 8 位透明度通道，共有 38400 个数据，存储的是压缩过的数据
- `__attribute__((aligned(64)))`: 是一个编译器指令，表示将数组的内存对齐到 64 字节，可能是为了优化内存访问效率

显示效果





北京大学
PEKING UNIVERSITY

嵌入式系统编程与实践课
程讨论

敬请讨论与指正！
谢谢大家！

例程编号：2.6

汇报学生：范世昌