



立宇泰电子

Liyutai Elec. CO., Ltd.

专注于做最好的嵌入式计算机系统供应商

LCD接口转VGA输出视频卡产品说明书

Rev 1.2

2010年8月24日

杭州立宇泰电子有限公司
HangZhou LiYuTai Elec.Co.,Ltd

目 录

目 录	1
一、产品概述	2
二、硬件说明	2
三、安装与使用	4
四、电气特性	5
五、LCD接口时序图	6
六、外形尺寸	7
七、配套驱动程序的修改	7
一，在ARMSYS2440 的WINCE5.0 BSP中，要修改如下 5 个文件	8
二，在ARMSYS6410 的WINCE6.0 BSP中，需要修改以下文件	9
八、LCD控制器输出场频计算	15
九、VCLK(位时钟)输出频率推荐值(LCD 控制器输出)	16
十、适用平台及测试情况	16
1 网站	18
2 保修	18
2.1 一年有限硬件保修	18
2.2 保修限制	18
3 技术支持	18

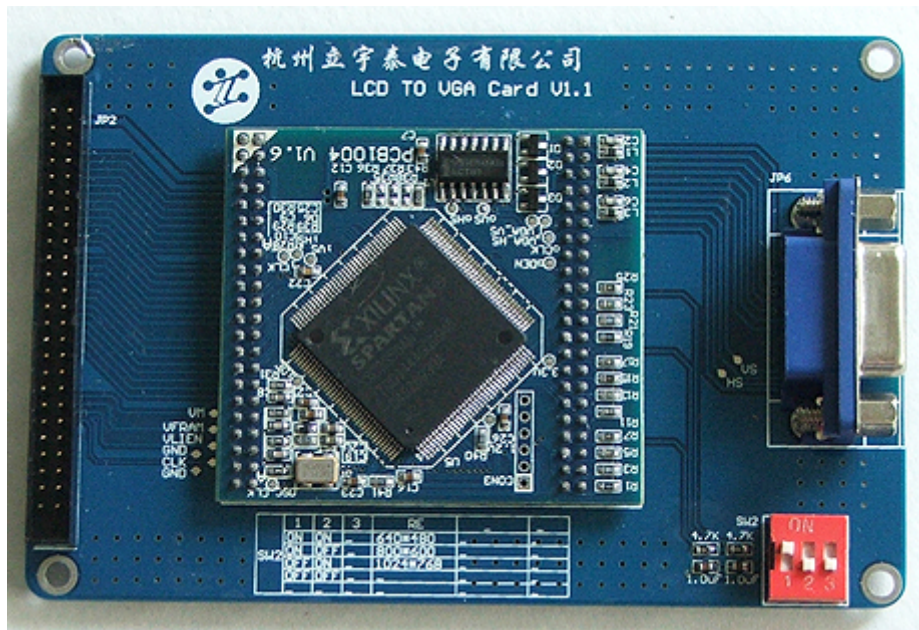
一、产品概述

LCD 接口转 VGA 输出视频卡(简称 LCD2VGA)是一款以 FPGA 做为桥接芯片,集成高效显存技术的视频转换卡。不同于简易的 D/A 转换模式, LCD2VGA 上的 FPGA 能够自动将 LCD 接口提供的刷新频率提高到 CRT 显示器所需要的高刷新频率,解决了简易的 D/A 转换模式下屏幕闪动的问题。

带有 LCD 控制器的 CPU 通过此视频卡,可以驱动带 VGA 接口的液晶显示器或 CRT 显示器。LCD2VGA 能够支持 16bppTFT(真彩)型数字 LCD 接口向 VGA 接口的转换,兼容 640×480、800×600、1024×768 三种分辨率,支持输出 60Hz 刷新率;其工作方式和工作时序与 TFT(真彩)LCD 完全一样。

LCD2VGA 信号输入接口全部采用抗干扰逻辑设计和防尖峰电压电路设计,视频卡最突出的优点在于它能同时兼容并灵活切换支持多种分辨率和刷新率,只需在断电或开机状态下切换 SW2 编码开关即可。

该产品适合用于人机接口、工控仪表、查询机、数据采集终端等应用领域。

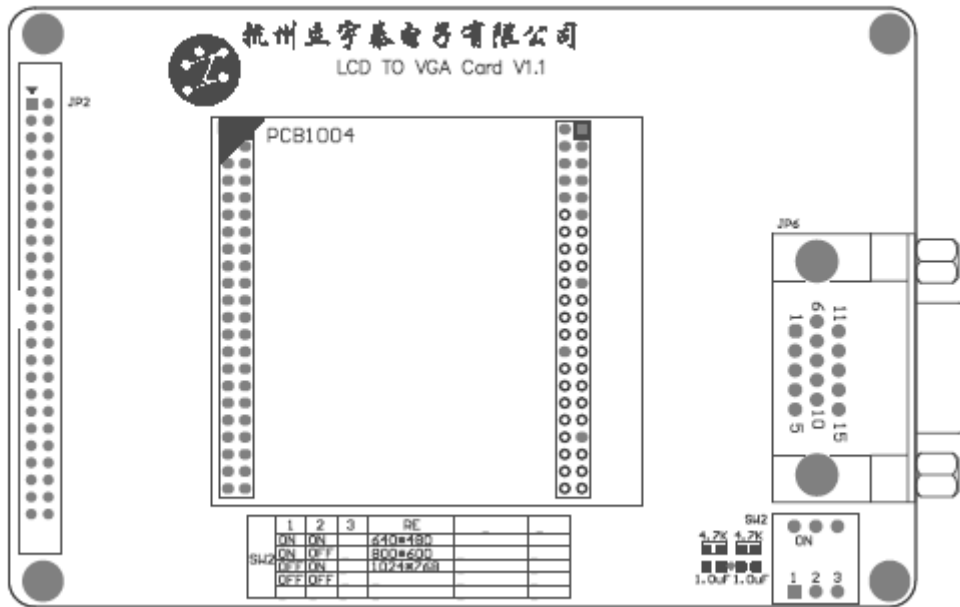


二、硬件说明

↙	1↗	2↗	3↗	LCD_MODE↔	RE↔	VRR↔	↖
↙	on↗	on↗	_↗	TFT_16BIT↔	640*480↔	60MHz↔	↖
↙	off↗	on↗	_↗	TFT_16BIT↔	800*600↔	60MHz↔	↖
↙	on↗	off↗	_↗	TFT_16BIT↔	1024*768↔	60MHz↔	↖

640*480
800*600
1024*768

UD0
UD1
UD2
UD3
UD4
UD5
UD6
UD7
UD8
UD9
UD10
UD11
UD12
UD13
UD14
UD15
UFRAME
ULINE
UCLK
UM



JP2: LCD 信号输入接口，要求数字 16bppTFT 型 LCD 信号接入，同时为 LCD2VGA 板提供直流 +5V 电源。该接口采用 2mm 间距 50pin 插座，引脚排布与 ARMSYS2410/2440/6410 开发板的 TFT 型 LCD 接口完全兼容。引脚定义如下表：

表 1: LCD2VGA 模块 JP2 引脚定义表格

1	VCC(+5.0V)	2	VCC(+5.0V)	17	G1	18	GND	33	空	34	GND
3	VCC(+5.0V)	4	GND	19	G2	20	G3	35	空	36	VM(DEN)
5	空	6	空	21	G4	22	G5	37	VFRAME (Vsync)	38	VLIN (Hsync)
7	空	8	空	23	空	24	空	39	VCLK	40	空
9	B1	10	B2	25	R0	26	R1	41	空	42	空
11	B3	12	B4	27	R2	28	R3	43	空	44	空
13	B5	14	空	29	R4	30	R5	45	空	46	空
15	空	16	G0	31	GND	32	空	47	空	48	空
49	空	50	GND								

JP6: VGA 信号输出接口，连接到标准 15 针 VGA 接口，与电脑液晶显示器或 CRT 显示器接口完全兼容。

SW2: 用来设置 LCD2VGA 板的工作模式。通过该编码开关来切换 LCD2VGA 板工作在什么分辨率、什么刷新率模式下。注意，LCD2VGA 板的工作分辨率与 LCD 接口输出的分辨率必须一致，才能输出正确的 VGA 图像。

SW2 的编码位置如下表 2 所示。

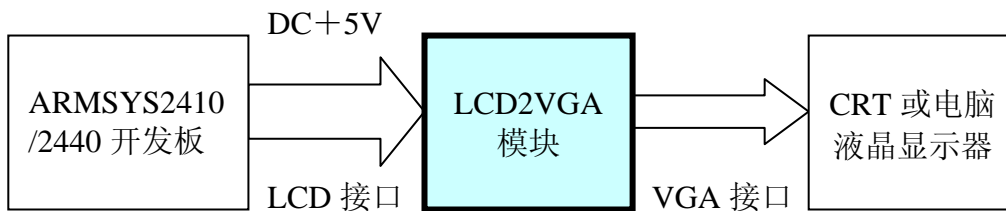
以下用红色文字标明的是修改或添加内容！

表 2: VGA 输出 SW2 配置表

1	2	3	VGA 分辨率	VGA 刷新率
ON	ON	-	640×480	60Hz
OFF	ON	-	800×600	60Hz
ON	OFF	-	1024×768	60Hz

三、安装与使用

LCD2VGA 模块在系统连接中的位置如下图所示：



安装 LCD2VGA 时，

- 采用 50pin 排线将 JP2 连接到开发板的 LCD 接口；
- 采用 VGA 电缆线将 JP6 连接到 VGA 接口；
- SW2 拨动到需要的配置。

对开发板和显示器分别连接好电源，开启开发板，系统界面就会显示在显示器上。例如，将 LCD2VGA 连接到运行 WINCE5.0 的 ARMSYS2440 开发板，再连接到 17 英寸电脑液晶作为显示屏，同时在 SW2 上将分辨率设置到 1024x768，显示效果如下图：



四、电气特性

表 4: LCD2VGA 模块电气特性最大值

Item	Symbol	Min.	Max.	Unit	Note
Supply Voltage	V _{CC}	-0.3	6.0	V	
Input Signal Voltage	V _I	-0.5	4.1	V	
Storage Temperature	T _{STG}	-65	+150	° C	

表 5: LCD2VGA 模块电气特性表

Item	Symbol	Min.	Typ.	Max.	Unit	
Supply Voltage	V_{CC}	4.5	5.0	6.0	V	
High Level Input Voltage	V_{IH}	2.0	----	4.1	V	
Low Level Input Voltage	V_{IL}	-0.5	----	0.8	V	
Current Consumption	I_{CC}	----	----	160	mA	
Power Consumption	P_{CC}	----	----	0.8	W	
Operating Temperature	Commercial	T_{Ic}	0	----	+85	° C
	Industrial	T_{Ii}	-40	----	+100	° C

五、LCD 接口时序图

图：TFT 模式时序图（LCD 控制器输出，LCD2VGA 模块输入）

TFT Mode Timing Diagrams (LCD Controller Output)

TFT模式时序图（LCD控制器输出）

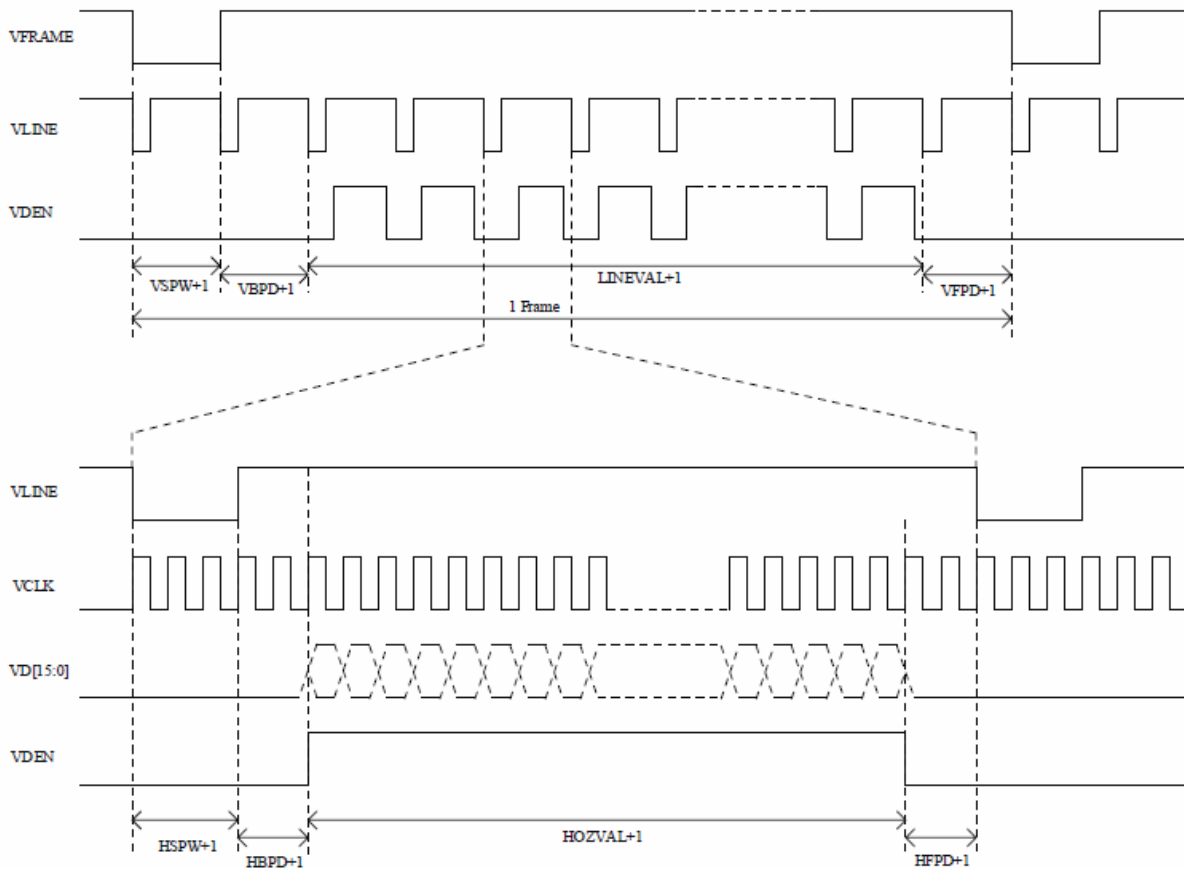


表 6：时序特性图

Timing Characteristics(TFT Mode)

Item	Symbol	Refresh Rate	Min.	Typ.	Max.	Unit
Clock Frequency	VCLK	--	--	--	37.50	MHz
Vertical back porch	VBPD+1	--	--	29	--	Line
Vertical Front Porch	VFPD+1	--	2	13	--	Line
Vertical sync pulse width	VSPW+1	--	2	3	--	Line
Horizontal back porch	HBPD+1	--	--	40	--	VCLK
Horizontal front porch	HFPD+1	640*480	20	40	--	VCLK
		800*600	20	40	--	VCLK
		1024*768	20	40	--	VCLK
Horizontal sync pulse width	HSPW+1	--	20	48	--	VCLK

六、外形尺寸

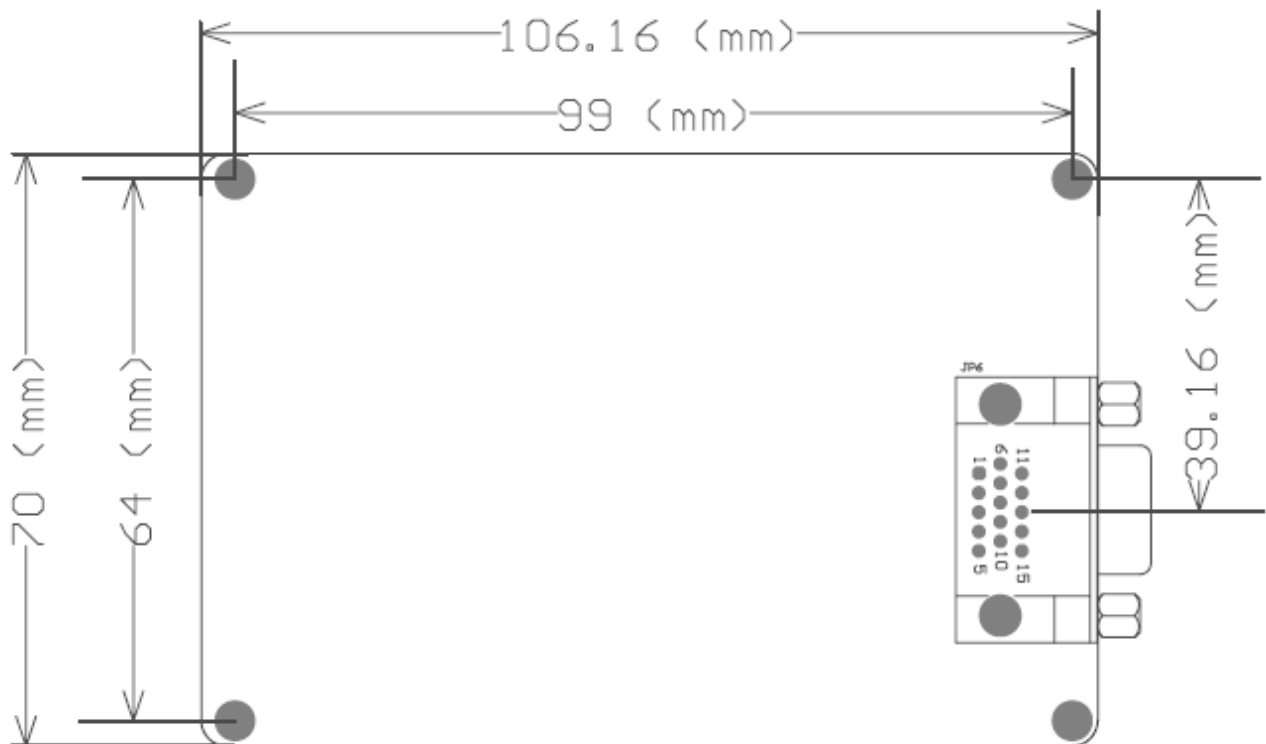


图 5: LCD2VGA 模块尺寸图

七、配套驱动程序的修改

LCD2VGA 可以与任何能够输出所需 LCD 接口信号的 LCD 控制器连接使用，例如三星 S3C2410/2440/6410 这样内置 TFT LCD 控制器的处理器。要输出正确的图像，要求 LCD 控制器输出的分辨率和 LCD2VGA 上设置的分辨率一致。这里我们分别以立宇泰的 ARMSYS2440

开发板（操作系统 WINCE5.0）和立宇泰的 ARMSYS6410 开发板（操作系统 WINCE6.0）为例，说明如何通过修改驱动程序，使开发板能够通过 LCD2VGA 来驱动类似 1024×768 这样的大分辨率的显示屏幕。

一，在 ARMSYS2440 的 WINCE5.0 BSP 中，要修改如下 5 个文件

- ..\WINCE500\PLATFORM\SMDK2440A\Src\Inc\ s3c2440a_lcd.h
- ..\WINCE500\PLATFORM\SMDK2440A\Src\Drivers\Display\s3c2440disp.cpp
- ..\WINCE500\PLATFORM\SMDK2440A\Src\Kernel\Oal\init.c
- ..\WINCE500\PLATFORM\SMDK2440A\Files\config.bib
- ..\WINCE500\PLATFORM\SMDK2440A\Src\Inc\ image_cfg.h

以下是以 1024×768 分辨率为例，需要修改的部分用红色字体表示。

◆ s3c2440a_lcd.h 文件的修改：

```
#define LCD_SCR_XSIZE (1024) // 按照所需分辨率修改
#define LCD_SCR_YSIZE (768) // 按照所需分辨率修改
#define LCD_SCR_XSIZE_TFT (1024) // 按照所需分辨率修改
#define LCD_SCR_YSIZE_TFT (768) // 按照所需分辨率修改
#define LCD_XSIZE_TFT 1024 // 按照所需分辨率修改
#define LCD_YSIZE_TFT 768 // 按照所需分辨率修改

//时序参数—START（以下参数，按照第 5 章表 6 进行修改）
#define LCD_VBPD ((29-1)&0xff)
#define LCD_VFPD ((13-1)&0xff)
#define LCD_VSPW ((3-1) &0x3f)
#define LCD_HBPD ((40-1)&0x7f)
#define LCD_HFPD ((40-1)&0xff)
#define LCD_HSPW ((48-1)&0xff)

//时序参数--END
```

◆ s3c2440disp.cpp 文件：

```
m_nScreenWidth =1024; // 按照所需分辨率修改
m_nScreenHeight =768; // 按照所需分辨率修改
m_cxPhysicalScreen =1024; // 按照所需分辨率修改
m_cyPhysicalScreen =768; // 按照所需分辨率修改
m_pvFlatFrameBuffer = 0x30060000; //调整的缓存地址，与 config.bib 文件修改一致
fbSize = 0x1a0000; //缓存大小
```

◆ init.c 文件：//寄存器配置，直接改为以下设置即可

```
s2440LCD->LCDCON1=(2<<8)|(0<<7)|(3<<5)|(12<<1)|(0<<0);//
s2440LCD->LCDCON5=(0<<12)|
```

```
(1 << 11) |
(0 << 10) |
(0 << 9) |
(0 << 8) |
(0 << 7) |
(0 << 6) |
(0 << 5) |
(0 << 4) |
(0 << 3) |
(0 << 2) |
(0 << 1) |
(1 << 0);
```

◆ config.bib 文件：//内存分配，与 s3c2440disp.cpp 文件的修改一致。

```
DISPLAY          80060000 001a0000 RESERVED
```

◆ image_cfg.h 文件：//现存的物理地址与虚拟地址，与 s3c2440disp.cpp 文件、config.bib 文件一致。

```
#define IMAGE_FRAMEBUFFER_UA_BASE      0xA0060000
#define IMAGE_FRAMEBUFFER_DMA_BASE    0x30060000
```

修改好后，重新编译 BSP 部分，产生 nk 文件。下载到 ARMSYS2440 开发板，并按照第二节说明连接好系统，进行测试即可。

二，在 ARMSYS6410 的 WINCE6.0 BSP 中，需要修改以下文件

- ...\\WINCE600\\PLATFORM\\SMDK6410\\SRC\\INC\\bsp_cfg.h
- ...\\WINCE600\\PLATFORM\\SMDK6410\\DRIVERS\\DISPLAY\\s3c6410_ldi_lib\\s3c6410_ldi.h
- ...\\WINCE600\\PLATFORM\\SMDK6410\\DRIVERS\\DISPLAY\\s3c6410_ldi_lib\\s3c6410_ldi.c
- ...\\WINCE600\\PLATFORM\\SMDK6410\\FILES\\config.bib
- ...\\WINCE600\\PLATFORM\\SMDK6410\\SRC\\INC\\image_cfg.h
- ...\\WINCE600\\PLATFORM\\SMDK6410\\SRC\\BOOTLOADER\\EBOOT\\main.c
- ...\\WINCE600\\PLATFORM\\SMDK6410\\SRC\\BOOTLOADER\\EBOOT\\bootpart1.cpp
- ...\\WINCE600\\PLATFORM\\SMDK6410\\SRC\\DRIVERS\\DISPLAY\\s3c6410_disp_drv\\s3c6410_disp.cpp

下面用红色文字标明的是修改或添加内容！

◆bsp_cfg.h文件修改

```
#define LCD_MODULE_HUA350      (0) // 3.5" QVGA RGB16 (320x240)
#define LCD_MODULE_SAM350     (1) // 3.5" QVGA RGB16 (320x240)
```

```
#define LCD_MODULE_QIM350      (2) // 3.5" QVGA RGB16 (320x240)
#define LCD_MODULE_SHP430     (3) // 4.3" WVGA RGB16 (480x272)
#define LCD_MODULE_QUN700     (4) // 7.0" WVGA RGB16 (800x480)
#define LCD_MODULE_VGA6448    (5) // VGA RGB16 (640x480)
#define LCD_MODULE_VGA8060    (6) // VGA RGB16 (800x600)
#define LCD_MODULE_QUN560     (7) // 5.6" VGA RGB16 (640x480)
#define LCD_MODULE_QUN800     (8) // 8.0" RGB16 (800x600)
#define LCD_MODULE_1024       (9) //宏定义

//#define SMDK6410_LCD_MODULE (LCD_MODULE_HUA350)//Hualw3.5"320x240
//#define SMDK6410_LCD_MODULE (LCD_MODULE_SAM350)//Samsung 3.5"320x240
//#define SMDK6410_LCD_MODULE (LCD_MODULE_QIM350)//Qimei 3.5"320x240
//#define SMDK6410_LCD_MODULE (LCD_MODULE_SHP430)//Sharp 4.3"480x272
//#define SMDK6410_LCD_MODULE (LCD_MODULE_QUN700)//Qunchuang 7"800x480
//#define SMDK6410_LCD_MODULE (LCD_MODULE_VGA6448)//VGA640x480
//#define SMDK6410_LCD_MODULE (LCD_MODULE_VGA8060)//VGA800x600
//#define SMDK6410_LCD_MODULE (LCD_MODULE_QUN560)//Qun 5.6"640x480
//#define SMDK6410_LCD_MODULE (LCD_MODULE_QUN800)//Qunchuang 8"800x600
#define SMDK6410_LCD_MODULE(LCD_MODULE_1024) //定义模块类型

#if (SMDK6410_LCD_MODULE == LCD_MODULE_HUA350)
#define LCD_WIDTH      320
#define LCD_HEIGHT     240
#define LCD_BPP        16
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_SAM350)
#define LCD_WIDTH      320
#define LCD_HEIGHT     240
#define LCD_BPP        16
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QIM350)
#define LCD_WIDTH      320
#define LCD_HEIGHT     240
#define LCD_BPP        16
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_SHP430)
#define LCD_WIDTH      480
#define LCD_HEIGHT     272
#define LCD_BPP        16
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN700)
#define LCD_WIDTH      800
#define LCD_HEIGHT     480
#define LCD_BPP        16
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_VGA6448)
#define LCD_WIDTH      640
#define LCD_HEIGHT     480
#define LCD_BPP        16
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_VGA8060)
#define LCD_WIDTH      800
#define LCD_HEIGHT     600
#define LCD_BPP        16
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN560)
#define LCD_WIDTH      640
#define LCD_HEIGHT     480
#define LCD_BPP        16

#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN800)
```

```
#define LCD_WIDTH          800
#define LCD_HEIGHT        600
#define LCD_BPP            16

#elif (SMDK6410_LCD_MODULE == LCD_MODULE_1024) //定义分辨率
#define LCD_WIDTH          1024
#define LCD_HEIGHT        768
#define LCD_BPP            16
#else
#error LCD_MODULE_UNDEFINED_ERROR
#endif
```

◆ s3c6410_ldi.h文件修改

```
typedef enum
{
    LDI_HUA350_RGB,          // 3.5" PQVGA 240x320
    LDI_SAM350_RGB,          // 3.5" QVGA 320x240
    LDI_QIM350_RGB,          // 3.5" QVGA 320x240
    LDI_SHP430_RGB,          // 4.3" WVGA 480x272
    LDI_QUN700_RGB,          // 7.0" WVGA 800x480
    LDI_VGA6448_RGB,         // VGA 640x480
    LDI_VGA8060_RGB,         // VGA 800x600
    LDI_QUN560_RGB,
    LDI_QUN800_RGB,
    LDI_1024_RGB,            //定义枚举成员
    LDI_MODULE_TYPE_MAX
} LDI_LCD_MODULE_TYPE;
```

```
static void LDI_1024_RGB_port_initialize(void); //函数声明,
static void LDI_1024_RGB_power_on(void);       //可能并没用到。
static void LDI_1024_RGB_power_off(void);
static LDI_ERROR LDI_1024_RGB_initialize(void);
```

◆ s3c6410_ldi.c文件修改

在 `LDI_ERROR LDI_fill_output_device_information(void *pDevInfo)` 函数下的 switch 语句里加入:

```
case LDI_1024_RGB:
    LDI_INF((_T("[LDI: INF] Output Device Type [%d] = LDI_1024_RGB\n\r"), g_ModuleType));
    pDeviceInfo->RGBOutMode = DISP_16BIT_RGB565_P;
    pDeviceInfo->uiWidth = 1024;
    pDeviceInfo->uiHeight = 768;
    pDeviceInfo->VBPD_Value = 2;
    pDeviceInfo->VFPD_Value = 2;
    pDeviceInfo->VSPW_Value = 2;
    pDeviceInfo->HBPD_Value = 16;
    pDeviceInfo->HFPD_Value = 240;
    pDeviceInfo->HSPW_Value = 16;
    pDeviceInfo->VCLK_Polarity = IVCLK_FALL_EDGE;
    pDeviceInfo->HSYNC_Polarity = IHSYNC_HIGH_ACTIVE;
    pDeviceInfo->VSYNC_Polarity = IVSYNC_HIGH_ACTIVE;
    pDeviceInfo->VDEN_Polarity = IVDEN_HIGH_ACTIVE;
    pDeviceInfo->PNR_Mode = PNRMODE_RGB_P;
    pDeviceInfo->VCLK_Source = CLKSEL_F_HCLK;
    pDeviceInfo->VCLK_Direction = CLKDIR_DIVIDED;
```

```
pDeviceInfo->Frame_Rate = 22;  
break;
```

以下信息可能并无实际作用，但以防编译出错需要添加

```
static void LDI_1024_RGB_port_initialize(void)  
{  
    LDI_MSG((_T("[LDI]++LDI_1024_RGB_port_initialize()\n\r")));  
    // nReset(GREST, STBY) : GPF[14]  
    // LCD_PANNEL_ON      : GPF[13]  
/*  
    // set GPIO Initial Value to Low  
    g_pGPIOReg->GPFDAT &= ~(0x3<<13); // nReset, LCD_PANNEL_ON  
                                     // Pull Up/Down Disable  
    g_pGPIOReg->GPFPU &= ~(0xf<<26); // nReset, LCD_PANNEL_ON  
                                     // Set GPIO direction to output  
    g_pGPIOReg->GPFCON = (g_pGPIOReg->GPFCON & ~(0xf<<26)) | (5<<26);  
    // nReset, LCD_PANNEL_ON  
*/  
    LDI_MSG((_T("[LDI]--LDI_1024_RGB_port_initialize()\n\r")));  
}  
static void LDI_1024_RGB_power_on(void)  
{  
    LDI_MSG((_T("[LDI]++LDI_1024_RGB_power_on()\n\r")));  
    // nReset(GREST, STBY) : GPF[14]  
    // LCD_PANNEL_ON      : GPF[13]  
/*  
    // LCD Pannel Power On and nReset  
    g_pGPIOReg->GPFDAT &= ~(1<<14); // nReset Low  
    g_pGPIOReg->GPFDAT |= (1<<13); // LCD_PANNEL_ON High  
    DelayLoop_1ms(10); // 10 ms  
                                     // Release nReset  
    g_pGPIOReg->GPFDAT |= (1<<14); // nReset High  
    DelayLoop_1ms(5); // 5 ms  
*/  
    LDI_MSG((_T("[LDI]--LDI_1024_RGB_power_on()\n\r")));  
}  
static void LDI_1024_RGB_power_off(void)  
{  
    LDI_MSG((_T("[LDI]++LDI_1024_RGB_power_off()\n\r")));  
    // nReset(GREST, STBY) : GPF[14]  
    // LCD_PANNEL_ON      : GPF[13]  
/*  
    // LCD Pannel Power Off  
    g_pGPIOReg->GPFDAT &= ~(1<<13); // LCD_PANNEL_ON Low  
    g_pGPIOReg->GPFDAT &= ~(1<<14); // nReset Low  
    DelayLoop_1ms(5); // 5 ms  
*/  
    LDI_MSG((_T("[LDI]--LDI_1024_RGB_power_off()\n\r")));  
}  
static LDI_ERROR LDI_1024_RGB_initialize(void)  
{  
    LDI_ERROR error = LDI_SUCCESS;  
    LDI_MSG((_T("[LDI]++LDI_1024_RGB_initialize()\n\r")));  
    // TODO: Initialize Pannel with Serial Interface  
    LDI_MSG((_T("[LDI]--LDI_1024_RGB_initialize() : %d\n\r"), error));  
    return error;  
}
```

}

在这个函数LDI_ERROR LDI_initialize_LCD_module(void)加入:

```
//1024*768
    case LDI_1024_RGB:
        LDI_INF((_T("[LDI:INF] LDI_initialize_LCD_module() : Type [%d]
LDI_1024_RGB\n\r"), g_ModuleType));
        LDI_1024_RGB_port_initialize();
        LDI_1024_RGB_power_on();
        LDI_1024_RGB_initialize();
        break;
```

在这个函数LDI_ERROR LDI_deinitialize_LCD_module(void)加入:

```
//1024*768
    case LDI_1024_RGB:
        LDI_INF((_T("[LDI:INF] LDI_deinitialize_LCD_module() : Type [%d]
LDI_1024_RGB\n\r"), g_ModuleType));
        LDI_1024_RGB_power_off();
        break;
```

◆ config.bib文件修改

; Common RAM areas

AUD_DMA	80002000	00002000	RESERVED
TEMPS	80010000	00010000	RESERVED
DBGSER_DMA	80022000	00002000	RESERVED
ARGS	80020800	00000800	RESERVED
SER_DMA	80024000	00002000	RESERVED
IR_DMA	80026000	00002000	RESERVED
SLEEP	80028000	00002000	RESERVED
EDBG	80030000	00020000	RESERVED
CMM	85D00000	00300000	RESERVED
FIMG_BUF	86000000	00800000	RESERVED
JPEG	86800000	00800000	RESERVED
DISPLAY	87000000	00800000	RESERVED//显示缓冲大小
MFC	87800000	00800000	RESERVED
; DISPLAY	86800000	00C00000	RESERVED
; MFC_JPEG	87400000	00C00000	RESERVED
ETHNET	94000000	00100000	RESERVED

◆ image_cfg.h文件修改

// Eboot Display Frame Buffer

// 2MB

```
#define EBOOT_FRAMEBUFFER_OFFSET (0x07E00000) //显示缓冲起始地址
#define EBOOT_FRAMEBUFFER_PA_START
(DRAM_BASE_PA_START+EBOOT_FRAMEBUFFER_OFFSET)
#define EBOOT_FRAMEBUFFER_UA_START
(DRAM_BASE_UA_START+EBOOT_FRAMEBUFFER_OFFSET)
#define EBOOT_FRAMEBUFFER_SIZE (0x00200000) //显示缓冲大小
```

// Display Frame Buffer

// 8MB

```
#define IMAGE_FRAMEBUFFER_OFFSET 0x07000000 //显示缓冲起始地址
#define IMAGE_FRAMEBUFFER_PA_START
(DRAM_BASE_PA_START+IMAGE_FRAMEBUFFER_OFFSET)
#define IMAGE_FRAMEBUFFER_UA_START
```

```
(DRAM_BASE_UA_START+IMAGE_FRAMEBUFFER_OFFSET)  
#define IMAGE_FRAMEBUFFER_SIZE    0x00800000 //显示缓冲大小
```

◆ main.c文件修改

在这个函数static void InitializeDisplay(void)里加入:

```
// Set LCD Module Type  
#if (SMDK6410_LCD_MODULE == LCD_MODULE_HUA350)  
    LDI_set_LCD_module_type(LDI_HUA350_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_SAM350)  
    LDI_set_LCD_module_type(LDI_SAM350_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QIM350)  
    LDI_set_LCD_module_type(LDI_QIM350_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_SHP430)  
    LDI_set_LCD_module_type(LDI_SHP430_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN700)  
    LDI_set_LCD_module_type(LDI_QUN700_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_VGA6448)  
    LDI_set_LCD_module_type(LDI_VGA6448_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_VGA8060)  
    LDI_set_LCD_module_type(LDI_VGA8060_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN560)  
    LDI_set_LCD_module_type(LDI_QUN560_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN800)  
    LDI_set_LCD_module_type(LDI_QUN800_RGB);  
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_1024)  
    LDI_set_LCD_module_type(LDI_1024_RGB);
```

在这个函数static void DisprograssBar(void)里加入:

```
#elif(SMDK6410_LCD_MODULE == LCD_MODULE_1024)  
{  
    SC_SIZE = FB_SIZE / 10  
    pd = (unsigned short *) (FB_START_ADDR+SC_SIZE*6);  
    fbp_cp = pd;  
    #define LEFT    620  
    #define RIGHT   824  
    #define HEIGHT  15  
}
```

◆ bootpart1.cpp文件修改

在这个函数BOOL BP_ReadData(HANDLE hPartition, LPBYTE pbBuffer, DWORD, dwLength)加入:

```
#elif(SMDK6410_LCD_MODULE == LCD_MODULE_1024)  
{  
    SC_SIZE = FB_SIZE / 10;  
    pd = (unsigned short *) (FB_START_ADDR+SC_SIZE*6);  
    pFB = pd;  
    #define LEFT    620  
    #define RIGHT   824  
}
```

◆ s3c6410_disp.cpp文件修改

在这个函数BOOL S3C6410Disp::DevInitialize(void)中加入:

```
// Set LCD Module Type
#if (SMDK6410_LCD_MODULE == LCD_MODULE_HUA350)
    LDI_set_LCD_module_type(LDI_HUA350_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_SAM350)
    LDI_set_LCD_module_type(LDI_SAM350_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QIM350)
    LDI_set_LCD_module_type(LDI_QIM350_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_SHP430)
    LDI_set_LCD_module_type(LDI_SHP430_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN700)
    LDI_set_LCD_module_type(LDI_QUN700_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_VGA6448)
    LDI_set_LCD_module_type(LDI_VGA6448_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_VGA8060)
    LDI_set_LCD_module_type(LDI_VGA8060_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN560)
    LDI_set_LCD_module_type(LDI_QUN560_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_QUN800)
    LDI_set_LCD_module_type(LDI_QUN800_RGB);
#elif (SMDK6410_LCD_MODULE == LCD_MODULE_1024)
    LDI_set_LCD_module_type(LDI_1024_RGB);
#endif
```

修改好后，重新编译BSP部分，产生nk文件。下载到ARMSYS6410开发板，并按照第二节说明连接好系统，进行测试即可

◆ s3c6410_disp.cpp文件修改

八、LCD 控制器输出场频计算

CSTN LCD Model:

$$VRR = 1 / (((WDLY + LINEBLANK + WLH) \times (1 / (64 \times 10^6)) + HOZVAL \times (1 / VCLK)) \times LINEVAL)$$

其中VRR是场频(刷新率)，VCLK是LCD控制器的位时钟输出频率， $1/(64 \times 10^6)$ 是64MHz 系统时钟的一个时钟周期。HOZVAL是可视区的行象素数，LINEVAL是列象素数(不分虚拟区和可视区)。

TFT LCD Model:

$$VRR = 1 / (((HBPD + 1) + (HFPD + 1) + (HSPW + 1) + (HOZVAL + 1)) \times ((VBPD + 1) + (VFPD + 1) + (VSPW + 1) + (LINEVAL + 1)) \times (1 / VCLK))$$

其中 VRR 是场频(刷新率)，VCLK 是 LCD 控制器的位时钟输出频率，(HOZVAL+1)是可视区的行象素数，(LINEVAL+1)是可视区的列象素数。|

九、VCLK(位时钟)输出频率**推荐值**(LCD 控制器输出)

分辨率	VCLK 频率		LCD 刷新率(LCD 控制器输出)	
	Min(MHz)	Max(MHz)	Min(Hz)	Max(Hz)
640*480	10.6	25.4	25	60
800*600	15.6	35.0	25	55.9
1024*768	24.3	66.5	25	70

十、适用平台及测试情况

1. 目前已测试的 CPU:

- (1) SAMSUNG 公司的 ARM9 系列微处理器 S3C2410。
- (2) SAMSUNG 公司的 ARM9 系列微处理器 S3C2440。
- (3) SAMSUNG 公司的 ARM11 系列微处理器 S3C6410。

2. 目前已测试的操作系统:

- (1) Windows CE .NET 4.20;
- (2) Windows CE .NET 5.00;
- (3) Windows CE .NET 6.00;
- (4) Linux2.4;

3. LCD2VGA 性能测试情况:

测试一, 基于 ARMSYS2410

硬件配置:

- 平台: ARMSYS2410
- CPU: S3C2410A
- 主频: 200MHz
- 内存: 64M
- LCD 控制器: S3C2410A 集成 LCD 控制器

测试条件:

- 操作系统: Windows CE 5.00
- LCD 控制器工作模式: TFT 16 位色
- 分辨率: 1024×768
- VCLK 频率: 25MHz
- 场频: 25.78Hz (帧/秒)

测试结果:

WINCE 视窗界面显示稳定、无闪动、无颜色失真; 窗体切换流畅, 鼠标拖动无拖尾情况。

测试二, 基于 ARMSYS2440**硬件配置:**

- 平台: ARMSYS2440
- CPU: S3C2440A
- 主频: 400MHz
- 内存: 64M
- LCD 控制器: S3C2440A 集成 LCD 控制器

测试条件:

- 操作系统: Windows CE .NET 4.20
- LCD 控制器工作模式: TFT 16 位色
- 分辨率: 1024×768
- VCLK 频率: 33MHz
- 场频: 25.78Hz (帧/秒)

测试结果:

WINCE 视窗界面显示稳定、无闪动、无颜色失真; 窗体切换流畅, 鼠标拖动无拖尾情况。

测试三, 基于 ARMSYS6410**硬件配置:**

- 平台: ARMSYS6410
- CPU: S3C6410X
- 主频: 667MHz
- 内存: MDDR 128M
- LCD 控制器: S3C6410 集成 LCD 控制器

测试条件:

- 操作系统: Windows CE 6.0
- LCD 控制器工作模式: TFT 16 位色
- 分辨率: 1024×768
- VCLK 频率: 22MHz
- 输入场频: 22Hz (帧/秒)

测试结果:

WINCE 视窗界面显示稳定、无闪动、无颜色失真; 窗体切换流畅, 鼠标拖动无拖尾情况。
采用 MediaPlayer 播放 640x480, MPEG4 格式视频文件, 帧率为 30fps 左右; 播放 720x432, MPEG4 视频文件, 帧率为 20fps 左右。

1 网站

欢迎进入立宇泰电子网站www.hzlitai.com.cn获得该产品的最新信息。

2 保修

以下为本产品保修申明，请仔细阅读。

2.1 一年有限硬件保修

用户购买本产品一年时间内，立宇泰为该产品提供质量担保，如果产品存在质量缺陷可以免费维修和更换。

2.2 保修限制

上述保修不适用于下列原因造成的缺陷：

- 出于任何原因擅自拆装产品；
- 在产品环境规范之外操作或存储产品；
- 使用不当、跌落、受潮；
- 使用非本产品配套附件；
- 人为损坏；

3 技术支持

支持方式：

- (1)电话, 0571-56763523 56763525 56763526
- (2)邮件, lyt_tech@yahoo.com.cn（工作日24小时内回复）
- (3)立宇泰嵌入式技术论坛:<http://www.hzlitai.com.cn/bbs>

支持时间：周一至周五 9:00—17:30

支持范围：

- 1、了解产品提供了哪些软、硬件资源；
- 2、如何安装随产品提供的工具软件；
- 3、如何使用手持设备；
- 4、判定是否存在故障。