

STDV710 开发板 用户手册

V1.0

深圳市博格达科技

地址：深圳市福田区赛格电子市场高科德交易中心 51868 室

联系电话：0755-33062551 13723473137

网址：www.bogodtech.com www.bogod.hqew.com

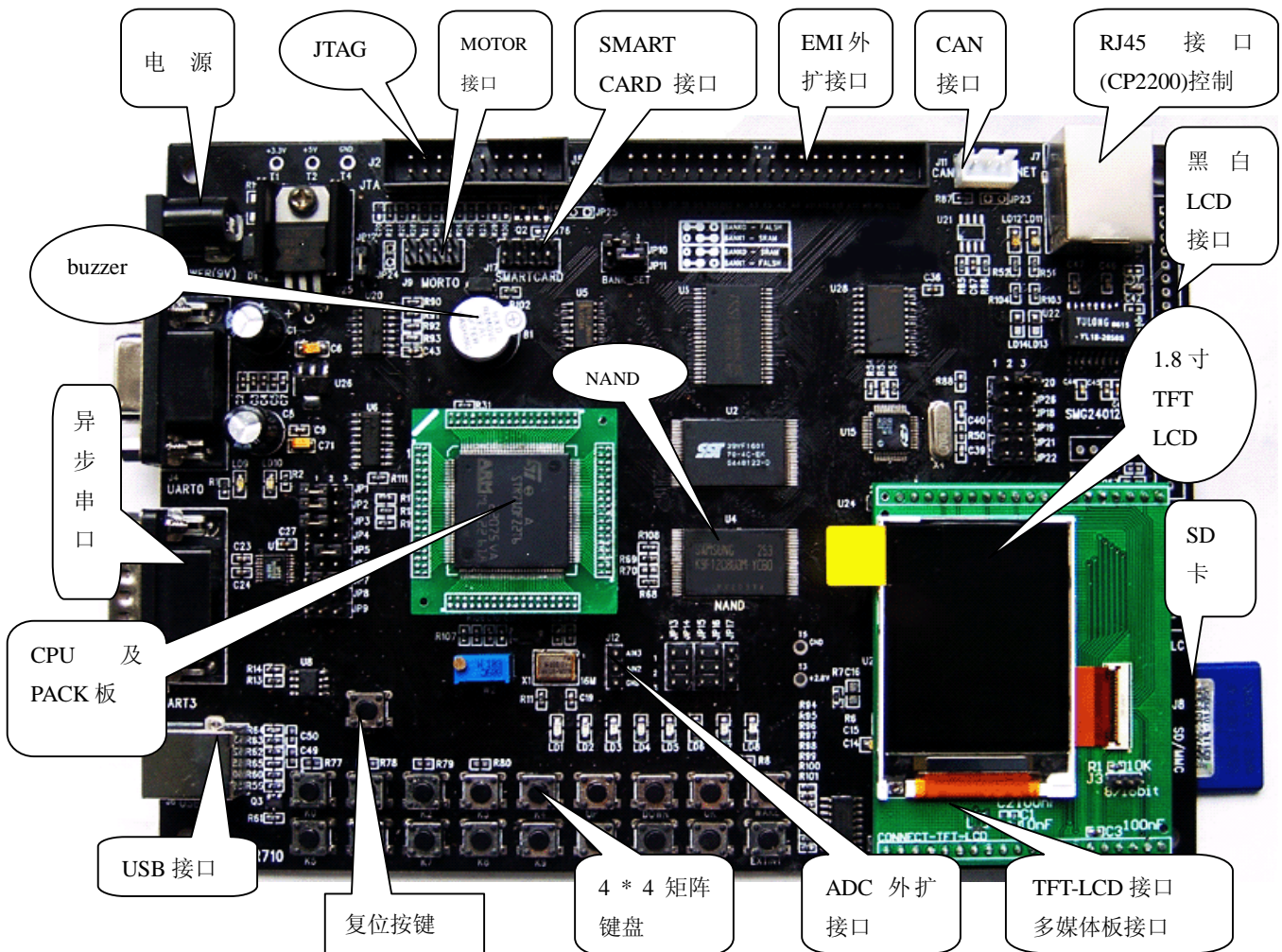
目 录

目 录.....	2
第一章 开发板硬件描述.....	4
1.1 开发板接口图.....	4
1.2 硬件规格介绍.....	4
1.3 接口一览表.....	6
1.4 跳线一览表.....	7
1.5 硬件模块介绍.....	10
1.5.1 电源电路.....	10
1.5.2 系统复位与 EEPROM 存储电路.....	10
1.5.3 系统时钟电路.....	11
1.5.4 JTAG 接口电路.....	11
1.5.5 系统主存储器电路.....	12
1.5.6 NAND FLASH 存储器电路.....	14
1.5.7 键盘电路.....	14
1.5.8 SD 卡电路.....	15
1.5.9 串口电路.....	15
1.5.10 以太网接口电路.....	16
1.5.11 蜂鸣器电路.....	16
1.5.12 LED 灯控制电路.....	17
1.5.13 ADC 电路.....	17
1.5.14 USB 接口电路.....	18
1.5.15 CAN 总线接口电路.....	18
1.5.16 点阵图形液晶接口电路.....	19
1.5.17 多媒体板与 TFT LCD 接口电路.....	19
1.5.18 TFT LCD 背光驱动电路.....	20
1.5.19 电机驱动电路接口.....	20
1.5.20 SMART CARD 接口.....	21
1.5.21 触摸屏接口.....	21
1.5.22 BOOT 模式配置.....	22
第二章 开发板软件相关描述.....	24
2.1 实例工程目录结构.....	24
2.2 实例工程程序结构与使用方法.....	25
2.3 实验程序.....	27
2.3.1 最小系统实验.....	27
2.3.2 UART 通信实验.....	28
2.3.3 行列式键盘(4×4)扫描实验.....	28
2.3.4 EEPROM 操作实验.....	28
2.3.5 定时器实验.....	28
2.3.6 外部中断实验.....	29
2.3.7 看门狗实验.....	29
2.3.8 PWM 输出控制实验.....	29
2.3.9 ADC 实验.....	29
2.3.10 低功耗模式及唤醒实验.....	29

2.3.11 SPI 控制 LED 的流水灯实验.....	30
2.3.12 USB 操作(U 盘方案)实验.....	30
2.3.13 外部 NOR FLASH 读写实验.....	30
2.3.14 NAND FLASH 读写实验.....	30
2.3.15 RTC 实验.....	30
2.3.16 SD/MMC 卡读写实验.....	31
2.3.17 以太网 (CP2200) 控制实验.....	31
2.3.18 系统时钟配置实验.....	31
2.4 带 UC/OS 实验程序存放目录结构.....	31
2.4.1 μ C/OS-II 移植实验.....	31
2.4.2 多任务控制实验.....	31
2.4.3 串口全功能实验.....	32
2.4.4 SD 卡全功能实验.....	32
2.5 为标准配置增补的实验目录(不带 OS).....	33
2.5.1 TFT LCD 驱动实验.....	33
2.5.2 LCD 与键盘实验.....	33
2.5.3 内部 FLASH 操作实验.....	33
2.5.4 LCD 与 ADC 实验.....	33
2.5.5 RTC 与 LCD 实验.....	34

第一章 开发板硬件描述

1.1 开发板接口图



1.2 硬件规格介绍

- 处理器：STR710F-Z2T6，内含 (256+16) KB FLASH、64KB SRAM；
- 电源接口 9V 电源；
- EMI 外扩：256KB x 16 SRAM (IS61LV25616)；1M x 16 NOR FLASH

(SST39VF1601);

- 10M 以太网接口 (CP2200, Silicon, 3.3V);
- USB2.0 通信接口 (DEVICE);
- CAN2.0 通信接口;
- 2 个 RS232 异步串行接口 (URAT0, UART3, 可直接连做 RS232 通信实验);
- SPI 接口 (74HC595D 驱动控制 LED 流水灯);
- SD/MMC 卡接口;
- 320 x 240 黑白图形液晶字 LCD 接口, 160 x 128 TFT - LCD 接口;
- 触摸屏接口 (采用 IIC 接口的 TSC2003 控制芯片芯片);
- 4 路 ADC 输入接口、1 个板上电位器 (ADC 实验)、2 路外扩;
- 1 个蜂鸣器;
- 1 个复位按键 (RST);
- 1 个唤醒按键 (wake);
- 4 x 4 矩阵键盘;
- 8 个 LED 灯
- SMARTCARD 接口;
- HDLC 接口
- EMI 外部扩展接口 (16 位数据总线、18 位地址总线等);
- 20 脚 JTAG 调试接口 (支持 FLASH 烧写);
- NAND FLASH 接口 (最大支持 1GBIT 的 NAND FLASH);

1.3 接口一览表

CN3, 9V 电源插头	J1
JTAG 接头, 双排 2*10	J2
UART3, 串口接口 (公)	J3
UART0, 串口接口 (母), 交叉线	J4
EXT_BUS 接口, 2*20	J5
USB 接口	J6
以太网接口	J7
KM1009461DR, SD/MMC 卡接口	J8
CON_MV, 步进电机接头, 6 针	J9
HDLC 接头, 6 针	J10
CAN 总线接头, 3 针	J11
外接 A/D 接口, 3 针	J12
黑白 LCD 接口	J16
SMARTCARD 卡接口	J17

另: U24 为 160 × 128 TFT-LCD 和多媒体板接口, 两者兼容。

1.4 跳线一览表

跳线编号	描述	设置选项	设置说明
JP1	BOOT_EN		参考BOOT模式设置
JP2	BOOT.0		
JP3	BOOT.1		
JP4	P0.10	1-2	SC_IO, SMART卡数据线
		2-3	步进电机控制线3
JP5	P0.11	1-2	SC_RST, SMART卡复位线
		2-3	用做BOOT.1 (默认)
JP6	P0.12	1-2	SCCLK, SMART卡时钟线
		2-3	步进电机控制线2
JP7	P0.9	1-2	TX_B, 串口0数据接收端
		2-3	用做BOOT.0 (默认)
JP8	P0.8	1-2	RX_B, 串口0数据发送端
		2-3	USB_DETECT, SD/MMC卡插入检测
JP9	P2.11	1-2	/RB, NAND FLASH忙检测输入引脚
		2-3	K_INT, 按键中断检测输入引脚
JP10	FLASH_CS选择	1-2	/CS0为外接FLASH的CS控制端
		2-3	/CS1为外接FLASH的CS控制端
JP11	SRAM_CS选择	1-2	/CS0为外接SRAM的CS控制端

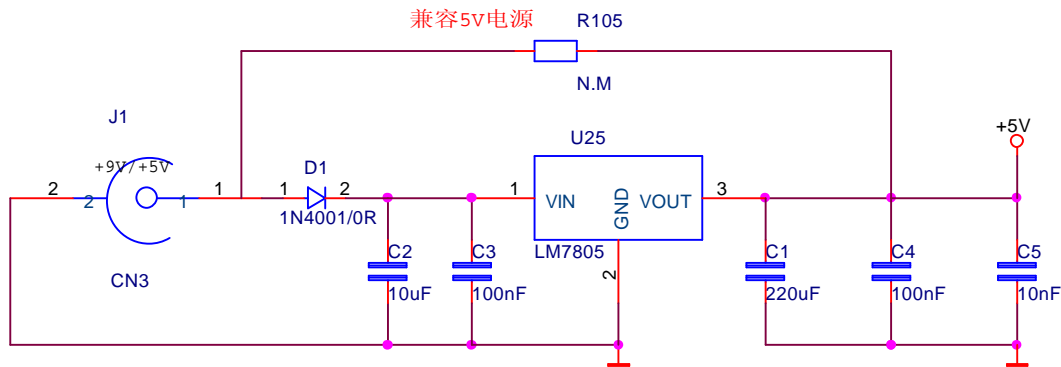
		2-3	/CS1为外接SRAM的CS控制端
JP12	P0.13	1-2	PWM_MOTOR, 步进电机控制线0
		2-3	BUZZER_EN, 蜂鸣器控制端
JP13	P2.8	1-2	INT_RTL, 以太网中断输入管脚
		2-3	SD_WP, SD/MMC卡中断输入管脚
JP14	P1.2	1-2	AIN2, 外扩ADC接口
		2-3	LCD_RST, 黑白LCD复位管脚
JP15	P1.3	1-2	AIN3, 外扩ADC接口
		2-3	MMC_INT, SD/MMC卡中断管脚
JP16	HC595电源控制	1-2	使能74HC595D电源3.3V
		2-3	关闭74HC595D电源3.3V
JP17	多媒体板CLK选择	1-2	直接使用16M有源晶振输出
		2-3	使用ARM内部输出时钟
JP18	P1.13	1-2	HCLK, HDLC接口时钟线
		2-3	NET_RST, 以太网复位管脚
JP19	P1.14	1-2	HRXD, HDLC接口数据接收端
		2-3	V938_RST, 多媒体芯片V938复位管脚
JP20	P1.15	1-2	HTXD, HDLC接口数据发送端
		2-3	V568_RST, 多媒体芯片V568复位管脚
JP21	P1.12	1-2	CANTX, CAN总线接口数据发送端
		2-3	连接独立发光二极管LD13
JP22	P1.11	1-2	CANRX, CAN总线接口数据接收端

		2-3	连接独立发光二级管LD14
JP23	CAN总线匹配	断开	总线上不匹配电阻
		短接	总线上匹配120R电阻
JP24	步进电机电源 控制	断开	切断电源
		短接	接通电源
JP25	TJAG复位选择 端	断开	使用CPU的复位信号/JTRST（默认）
		短接	使用主板上的复位信号/RST_STR710F
JP26	CAN接口模式选 择	1-2	高速模式
		2-3	低速模式

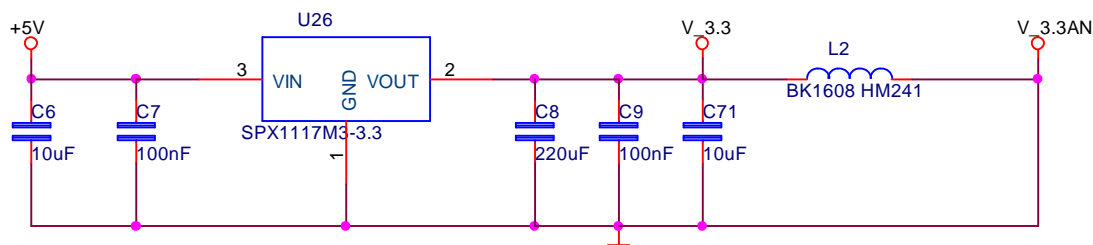
1.5 硬件接口模块介绍

1.5.1 电源电路

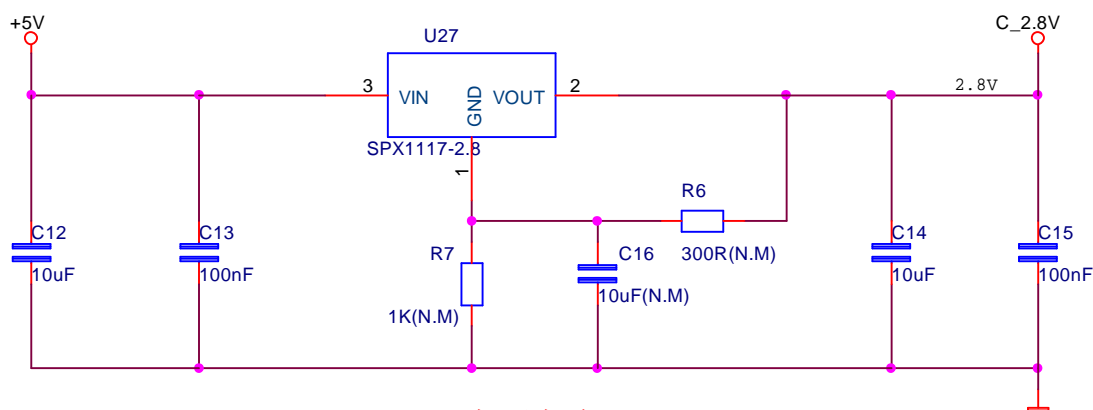
STDV710系列开发板主板使用三组电源(5.0V, 3.3V, 2.8V)具体电路如图所示。



5.0V电源电路



3.3V电源电路

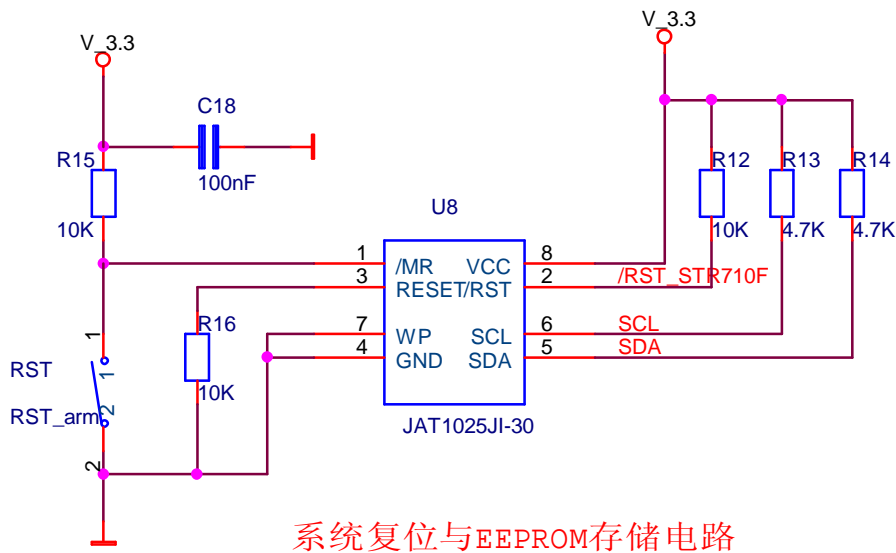


TFT LCD电源电路(2.8V)

1.5.2 系统复位与 EEPROM 存储电路

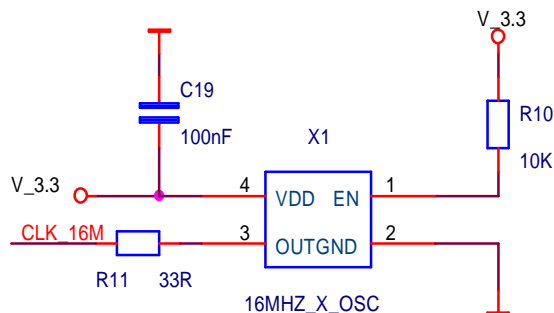
本开发板的复位电路使用了带I2C存储器的电源监控芯片CAT1025JI-30

(复位门槛电压为3.0-3.15V)，其电路原理如图所示。



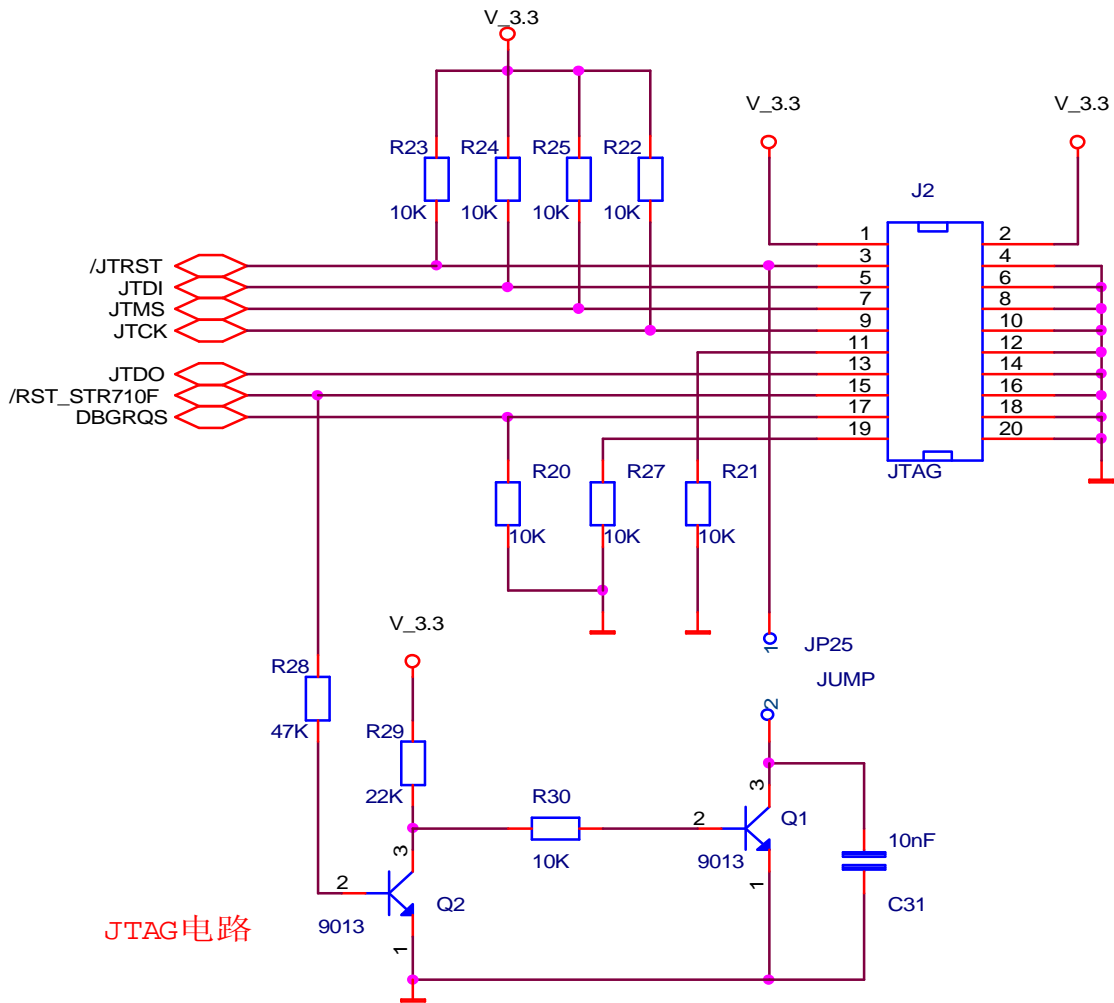
1.5.3 系统时钟电路

STDV710开发板采用的是有源晶振(16M,3.3V),其内部PLL电路可调整系统时钟,使系统运行速度更快(CPU最大操作时钟为50MHZ).电路如图。



1.5.4 JTAG 接口电路

STDV710开发板采用标准20脚JTAG仿真调试接口,支持标准的ICE在线仿真器,其JTAG信号的定义及与STDV710芯片的连接如图所示。

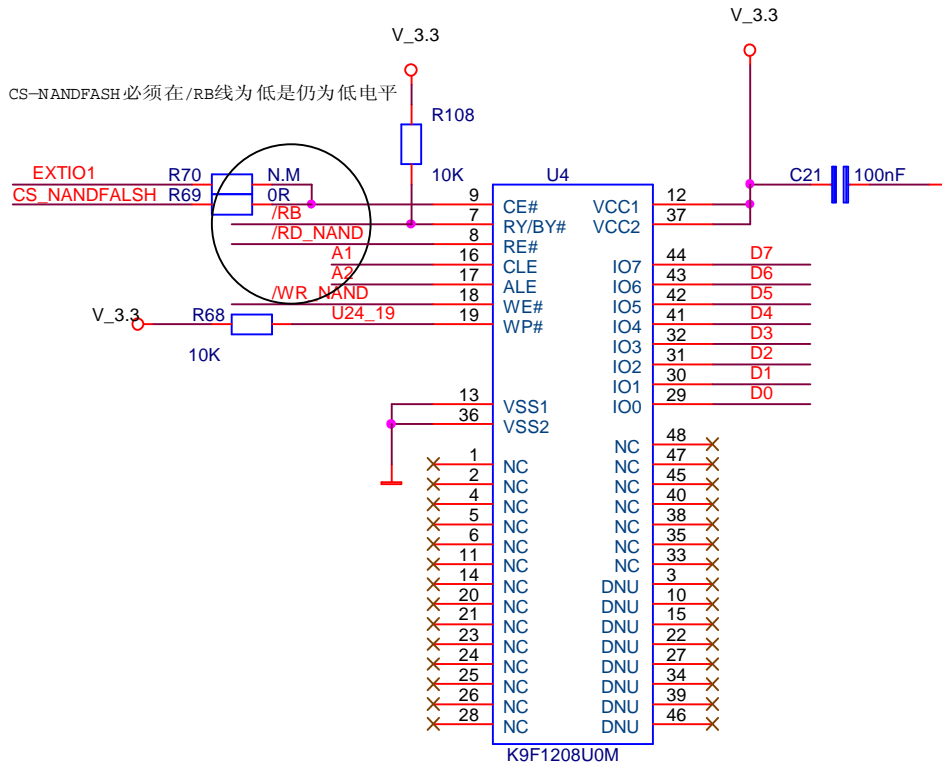


1.5.5 系统主存储器电路

STDV710开发板扩展了2MB NOR FLASH（型号SST39VF1601）和512KB SRAM（型号IS61LV25616可兼容IS61LV51216）。电路如图所示。

1.5.6 NAND FLASH 存储器电路

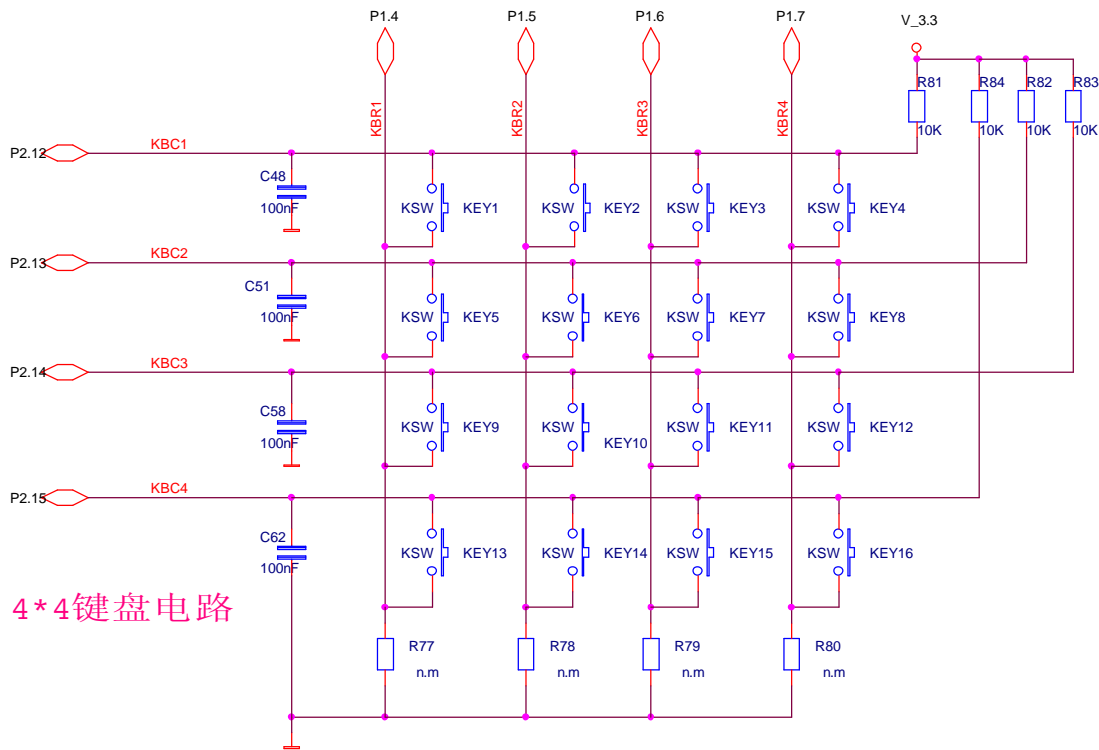
STDV710开发板扩展了64MB NAND FLASH（型号为K9F1208U0M，可最大兼容1GBIT 的NAND FLASH）。采用的是8位总线方式。电路如图所示。



64M NAND K9F1208U0M compatible 16M NAND K9F2808

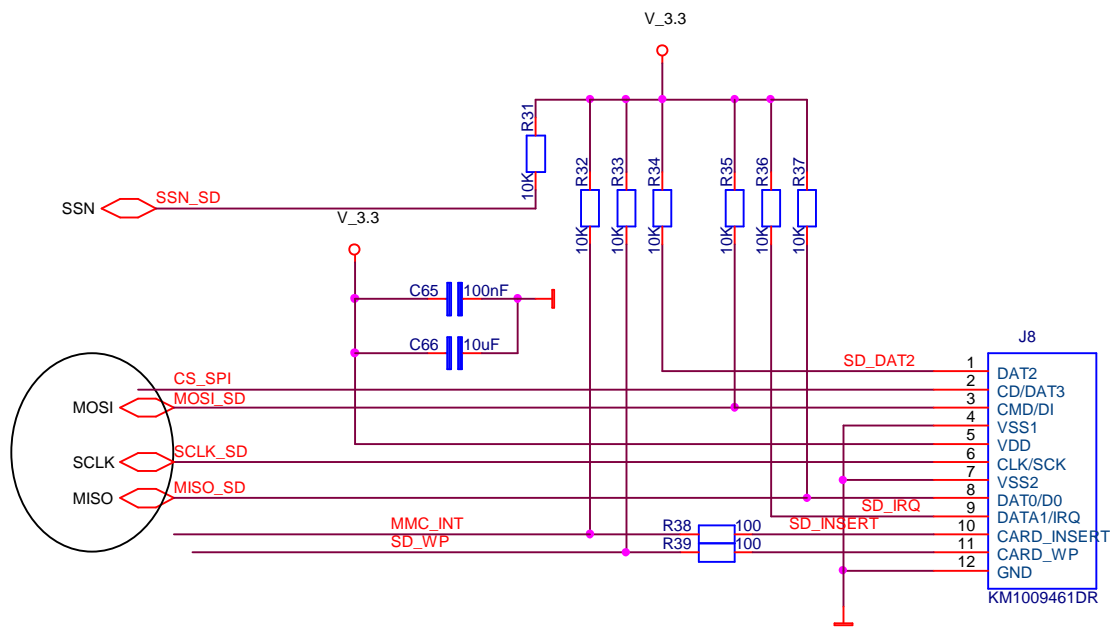
1.5.7 键盘电路

STDV710 开发板共有 16 个独立的按键，使用 4 × 4 行列矩阵键盘。电路如图所示。



1.5.8 SD 卡电路

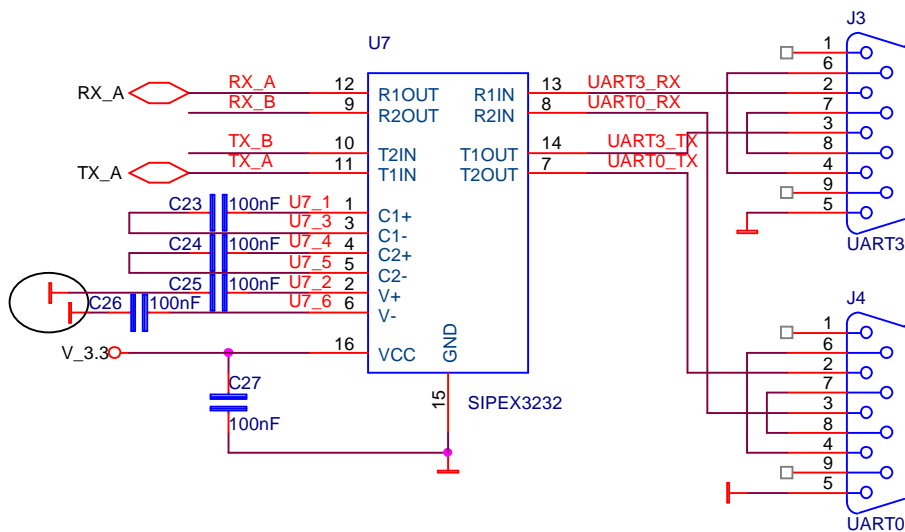
STDV710 开发板具有 SD 卡接口，支持 SD/MMC 卡的读/写。电路如图所示。



1.5.9 串口电路

STDV710 共有 4 通道的 UART, 开发板上引出了两路 UART(UART0 连接母的

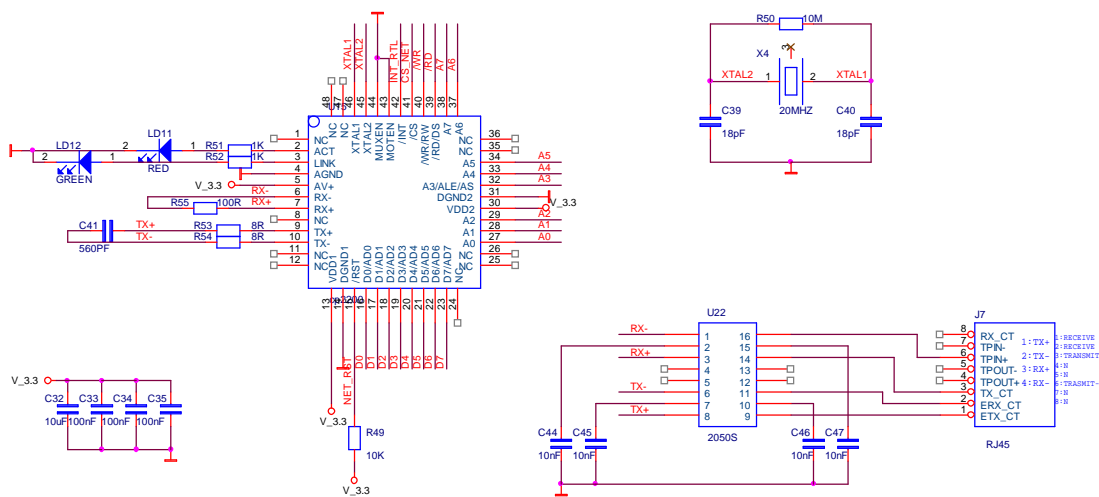
串口连接头, UART3 连接公的串口连接头), 于是可以直接在开发板上进行两个串口通信实验. 电路如图所示



UART接口电路

1.5.10 以太网接口电路

STDV710 开发板采用的是 Silicon Laboratories 公司的以太网芯片 CP2200 为核的以太网接口, 电路如图所示.

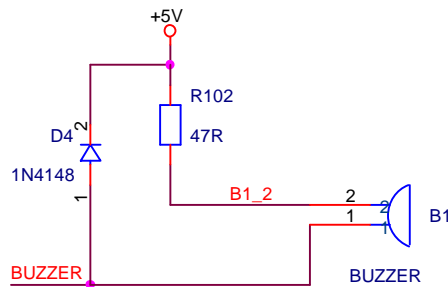


10/100M网络接口

1.5.11 蜂鸣器电路

STDV710 开发板采用的蜂鸣器使用 5V 的工作电压, 所以其控制端

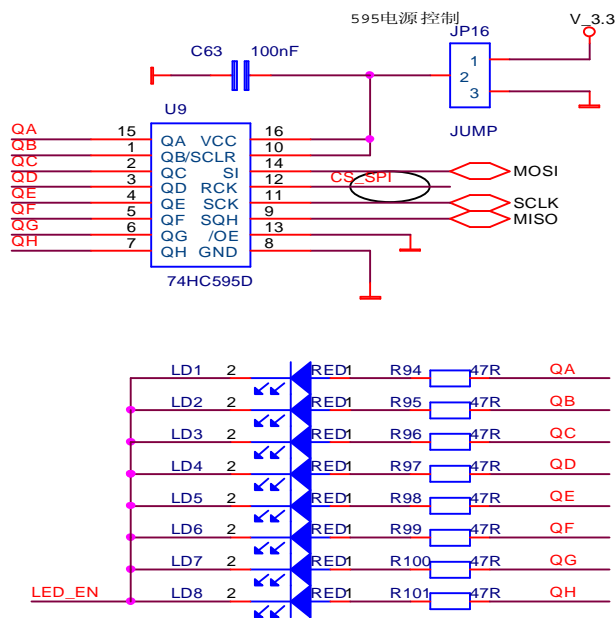
P0.13(需跳线)需要经过 ULN2003 达林顿管进行驱动,当 P0.13 控制输出为高电平时,输出为低电平蜂鸣器导通蜂鸣,反之蜂鸣器停止蜂鸣.电路如图所示.



蜂鸣器电路

1.5.12 LED 灯控制电路

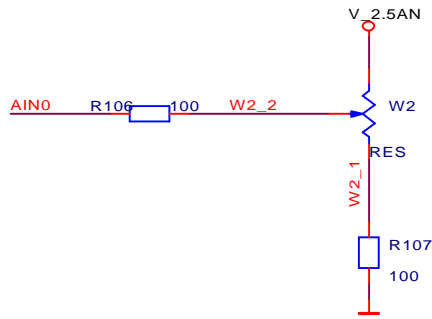
STDV710 开发板采用 ARM 芯片的 SPI 接口向 74HC595D 芯片发送的数据来控制 8 路 LED 灯的亮灭.电路如图所示.



LED灯控制电路

1.5.13 ADC 电路

STDV710 开发有 4 路 ADC 输入接口、1 个板上电位器、2 路外接 ADC 实验接口,电路如图所示.



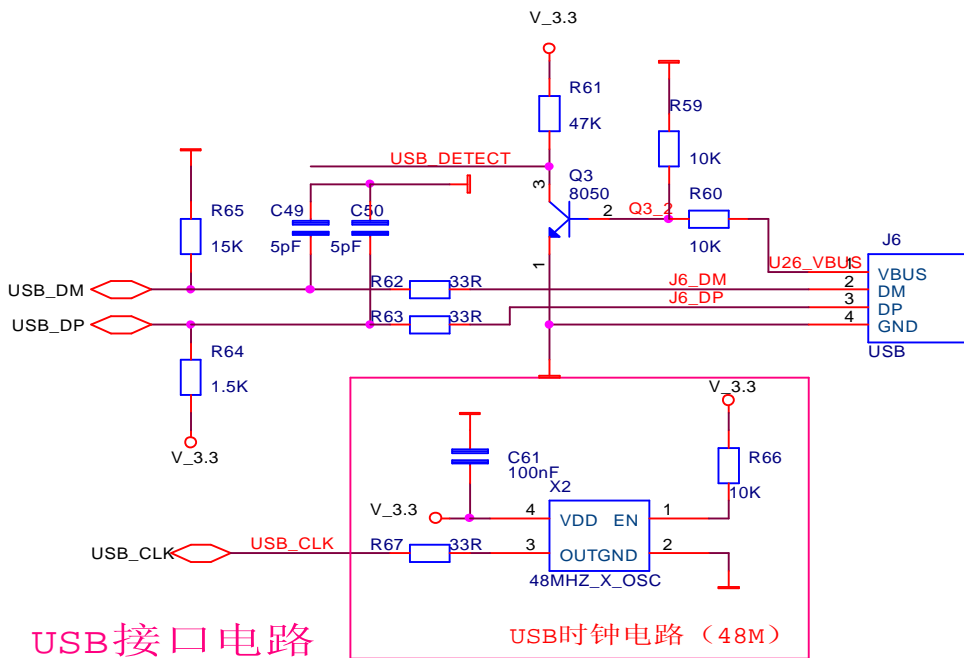
ADC 板上电位器电路



ADC 外扩接口

1.5.14 USB 接口电路

STDV710 支持 USB2.0, 开发板提供一个 USB Device 接口. 电路如图所示.

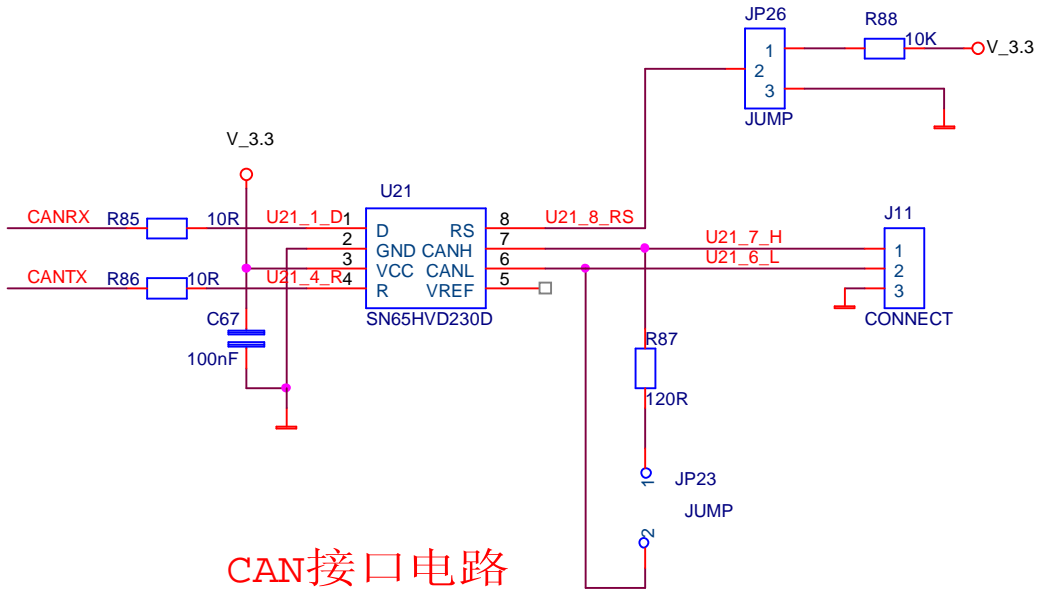


USB 接口电路

USB 时钟电路 (48M)

1.5.15 CAN 总线接口电路

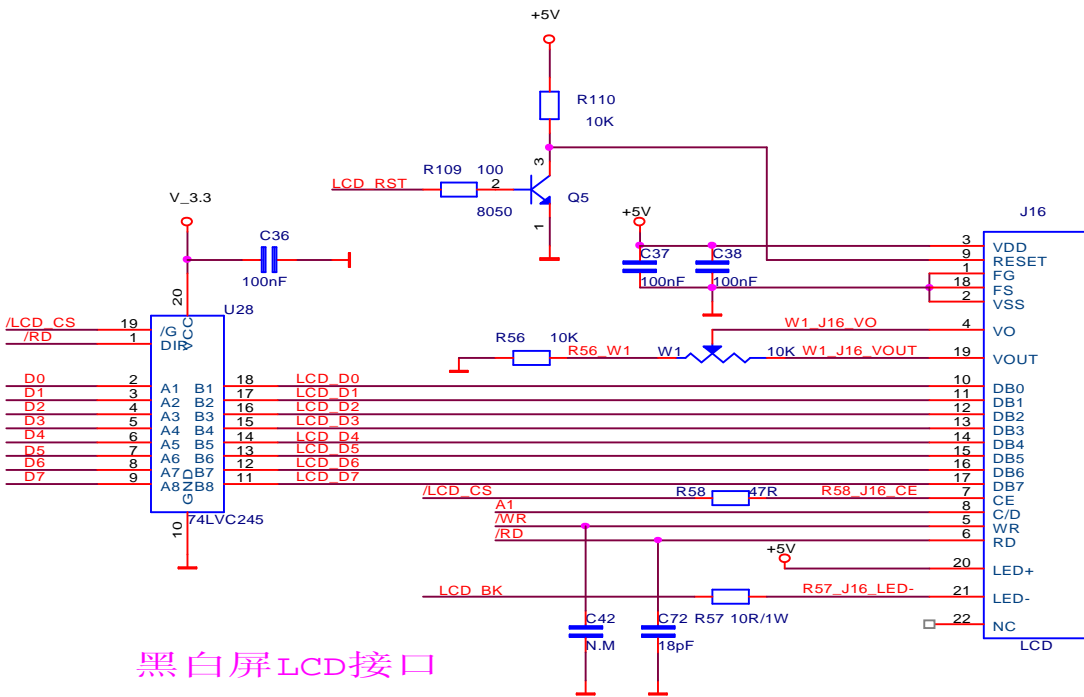
STR71X 系列芯片带有工业级的 CAN 总线接口，在 STDV710 开发板上以 TI 的 SN65HVD230D 芯片为驱动芯片，并将 CAN 总线用 3 针连接头引出。电路如图所示。



CAN接口电路

1.5.16 点阵图形液晶接口电路

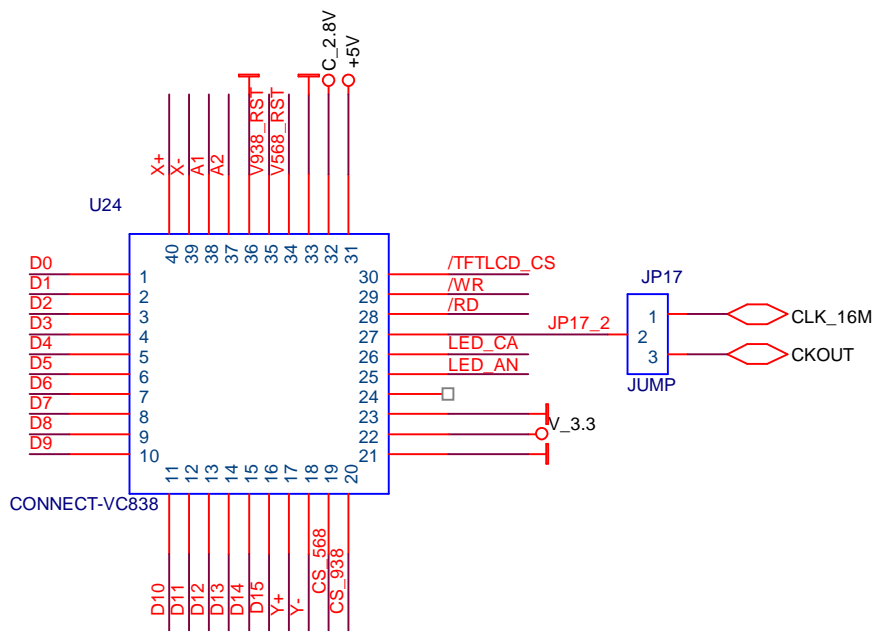
STDV710 开发板有一个 320*240 的点阵图形液晶接口, 如图所示。



黑白屏LCD接口

1.5.17 多媒体板与 TFT LCD 接口电路

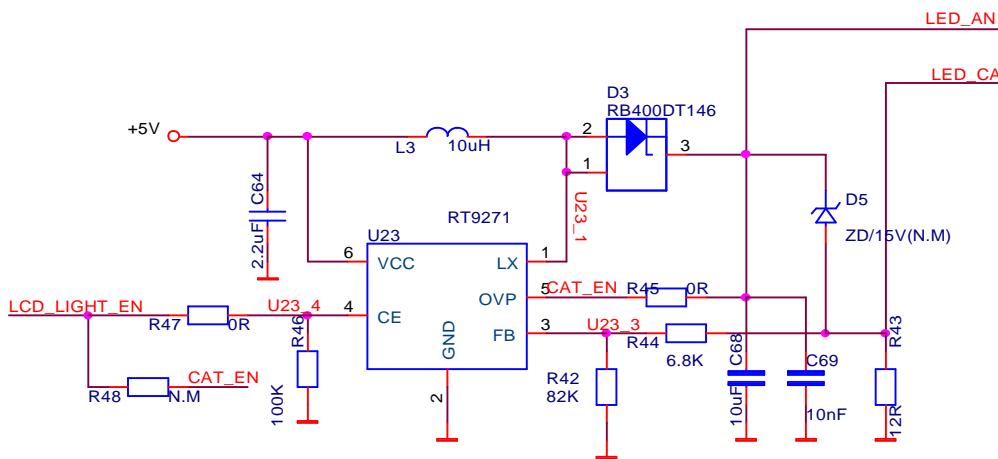
STDV710 开发板有多媒体板与单独的 TFT LCD 接口板, 两块板的接口电路如下图所示。



多媒体板与TFT LCD接口电路

1.5.18 TFT LCD 背光驱动电路

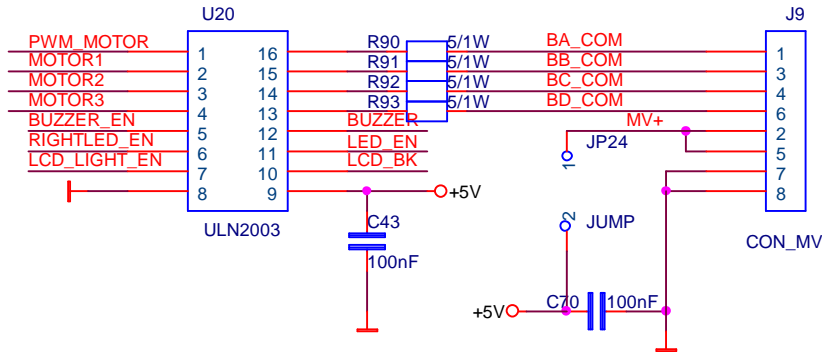
STDV710 开发板支持 TFT LCD,且通常需背光驱动电路,为了兼容性本电路可支持 3 种类型的背光驱动芯片,其电路如下图所示。



兼容CAT32EKT、RT9271、SM8121 三种芯片的背光驱动电路

1.5.19 电机驱动电路接口

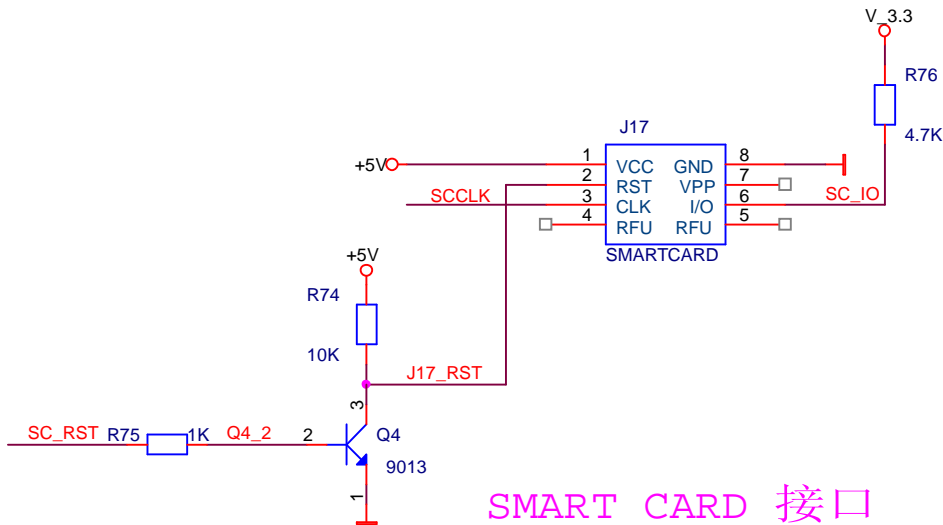
STDV710 开发板有一个直流电机、步进电机驱动接口,其电路如下图所示。



电机驱动接口电路

1.5.20 SMART CARD 接口

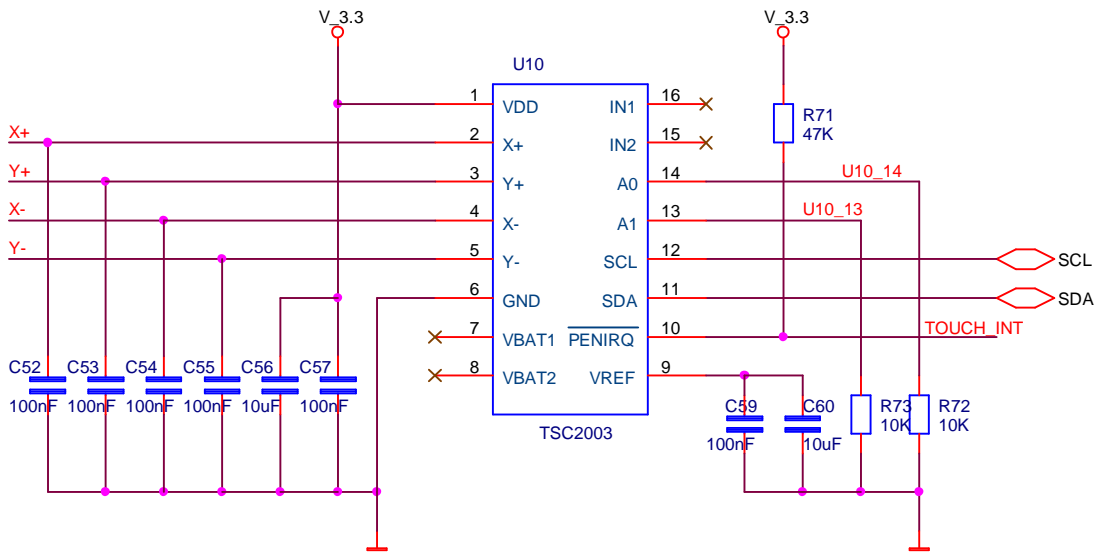
STDV710 开发板上有一个 SMART CARD 接口，其电路如下图所示。



SMART CARD 接口

1.5.21 触摸屏接口

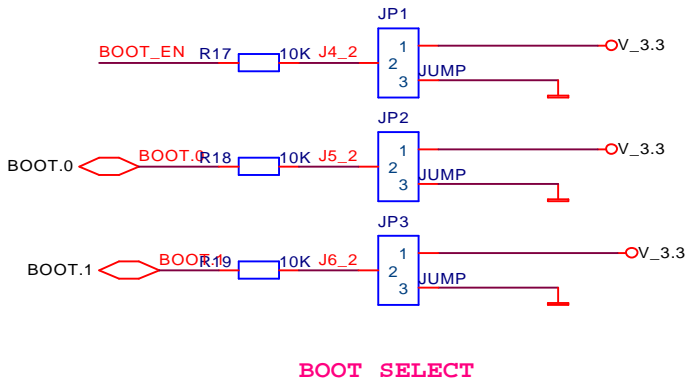
STDV710开发板上预留触摸屏接口，触摸控制器采用I2C接口方式的TSC2003芯片，其电路如下图：



触摸屏接口电路

1.5.22 BOOT 模式配置

STR71X系列ARM芯片有3种启动模式,需要用跳线来选择不同的启动模式,跳线选择图如下图:



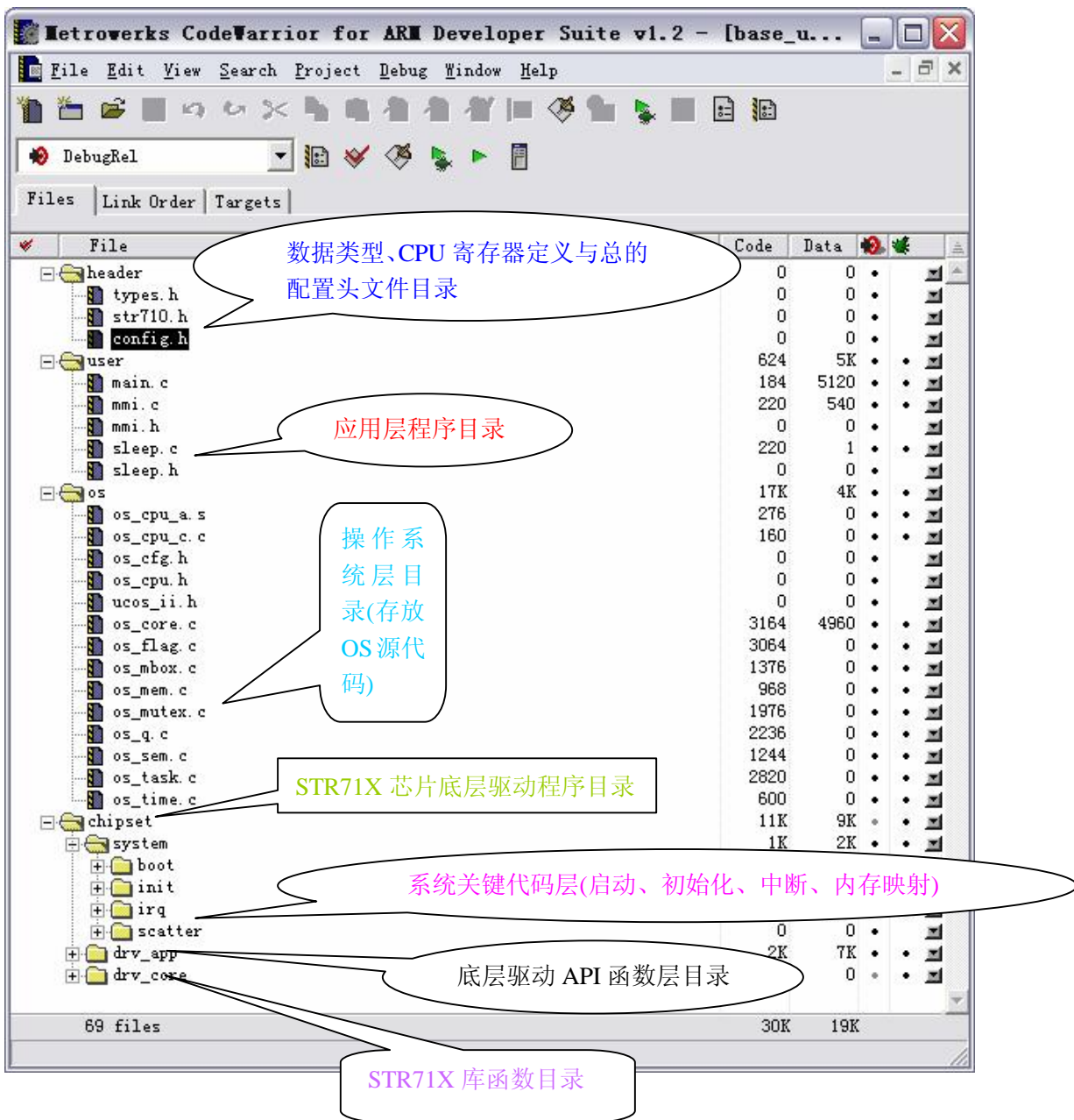
BOOT 选项具体配置如下:

BOOT EN	BOOT1 (B1)	BOOT0 (B0)	模式	启动内存映射	说明
0	X	X	用户	片内FLASH 映射到0地址	<ul style="list-style-type: none"> ■ 系统从片内FLASH开始运行
1	0	0			
1	0	1	系统存储器	系统存储器映射到0地址	<ul style="list-style-type: none"> ■ 系统从出厂前预装的引导程序开始运行 ■ 时钟被冻结
1	1	0	RAM	片内RAM映射到0地址	<ul style="list-style-type: none"> ■ 系统从片内SRAM开始运行 ■ 实验室开发用
1	1	1	外部存储器	外部存储器映射到0地址	<ul style="list-style-type: none"> ■ 系统从外部存储器开始运行

第二章 开发板软件相关描述

2.1 实例工程目录结构

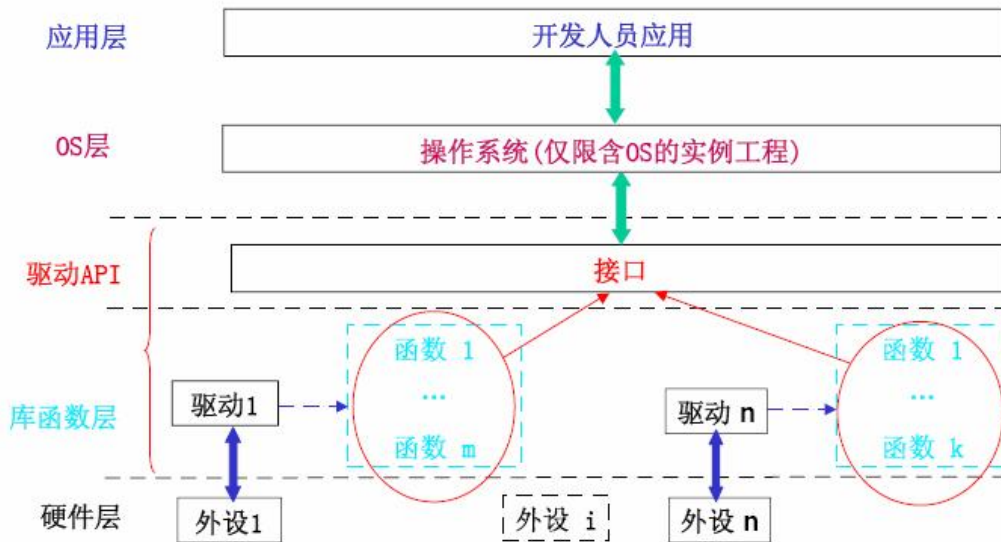
STR71X 开发板实例工程目录结构层次清晰，一目了然，具体结构如下图。



注：应用层程序：主要指与芯片操作无关的程序部分。底层驱动 API：主要指对芯片操作的库函数进一步封装，供应用层程序调用。

2.2 实例工程程序结构与使用方法

通常嵌入式系统的层次结构最底层为硬件层，之后为驱动层，在之后为操作系统层，最后才是应用层。在我们的实例工程中基本都采用这种层次结构，具体的程序结构如下图：



在驱动层 STR71X 开发板的配套软件将代码分为 3 部分：与启动、中断、初始化相关的 SYSTEM 层；驱动 API 层、STR71X 芯片函数库层。STR71X 函数库由 3 部分组成：(1)CPU 寄存器定义头文件，包含在../header 目录下其文件名为：str710.h (2) 外设驱动代码，包含在../chipset/drv_core 目录中，如 GPIO.C、GPIO.H (3) 外设驱动代码选择头文件也包含在../chipset/drv_core 中。

芯片函数库包含了对芯片大部分底层操作，在实际的工程中并不需要将函数库中的所有函数都编译连接到目标文件中，一般只是根据实际需要将要用的部分添加到工程进行编译连接，即在../chipset/drv_core 目录中有 cpu_cfg.h、drv_core.h 两个头文件，通过 cpu_cfg.h 中宏定义的值来选择需

要编译的函数库文件，`drv_core.h` 中则是根据 `cpu_cfg.h` 的宏定义值选择需要包含的函数库头文件，又 `drv_core.h` 被 `../header` 中的 `config.h` 所包含。

如需要将 `uart`，`apb`，`rccu`，`eic` 部分的库函数添加到工程进行编译，将对应的宏定义值设置为 1：

```
#ifndef __CPU_CFG_H
#define __CPU_CFG_H

#define EN_ARM_APB 1 /* 禁止(0)或允许(1)使用 ARM 的 APB 模块 */
#define EN_ARM_RCCU 1 /* 禁止(0)或允许(1)使用 ARM 的 RCCU 模块 */
#define EN_ARM_PCU 0 /* 禁止(0)或允许(1)使用 ARM 的 PCU 模块 */
#define EN_ARM_EIC 1 /* 禁止(0)或允许(1)使用 ARM 的 EIC 模块 */
#define EN_ARM_XTI 0 /* 禁止(0)或允许(1)使用 ARM 的 XTI 模块 */
#define EN_ARM_EMI 0 /* 禁止(0)或允许(1)使用 ARM 的 EMI 模块 */
#define EN_ARM_GPIO 0 /* 禁止(0)或允许(1)使用 ARM 的 GPIO 模块 */
#define EN_ARM_FLASH 0 /* 禁止(0)或允许(1)使用 ARM 的 FLASH 模块 */
#define EN_ARM_RTC 0 /* 禁止(0)或允许(1)使用 ARM 的 RTC 模块 */
#define EN_ARM_TIM 0 /* 禁止(0)或允许(1)使用 ARM 的 IIM 模块 */
#define EN_ARM_UART 1 /* 禁止(0)或允许(1)使用 ARM 的 UART 模块 */
#define EN_ARM_USB 0 /* 禁止(0)或允许(1)使用 ARM 的 USB 模块 */
#define EN_ARM_WDG 0 /* 禁止(0)或允许(1)使用 ARM 的 WDG 模块 */
#define EN_ARM_BSPI 0 /* 禁止(0)或允许(1)使用 ARM 的 BSPi 模块 */
#define EN_ARM_CAN 0 /* 禁止(0)或允许(1)使用 ARM 的 CAN 模块 */
#define EN_ARM_ADC12 0 /* 禁止(0)或允许(1)使用 ARM 的 ADC12 模块 */
#define EN_ARM_I2C 0 /* 禁止(0)或允许(1)使用 ARM 的 I2C 模块 */

#endif /* __CPU_CFG_H */
```

通过上面的设置后工程编译时会自动编译被选择的模块函数库头文件，以及自动在 `config.h` 中包含所选择函数库模块的头文件。启动代码、中断处理、系统高级初始化相关的文件放在 `../chipset/system` 目录，启动代码部分一般不用修改，初始化部分根据实际需要添加目标板高级初始化代码，如果实验工程用到中断，那么中断处理部分定义在 `irq.c` 中，若工程需要添加中断例程时只需要在 `irq.c` 中中断服务例程表中添加对应的中断函数名即可。在 `../chipset/drv_app` 中主要存放对芯片函数库封装后供用户层调用的

API。../user 目录则包含所有的上层应用代码。../header/config.h 包含了工程中的大部分头文件，在除芯片函数库外的所有 C 源文件只需包含这个头文件(config.h)即可。

2.3 基础实验程序

在使用仿真器调试程序前，先确定程序将在 STR71X 的内部还是外部 RAM 中运行。如果是在内部 RAM 调试程序，请先将开发板的启动模式配置为内部 RAM 模式，否则配置为外部存储器模式，且还须合理配置 JP10、JP11 跳线，以保证外部 RAM 的起始地址为 0x60000000。

不带 OS 实验程序存放目录结构

目录名	内容简介
minsyst	验证最小系统正常工作工程，主要包括 GPIO 的操作
uart	UART 实验
kpd	4 × 4 行列式键盘扫描实验
i2c_eeeprom	EEPROM 读写实验
timer	定时器实验
xti	外部中断实验
watchdog	看门狗实验
pwm	PWM 输出实验
adc	ADC 实验程序
wake_up	低功耗唤醒实验
spl_led	SPI 控制 LED 的流水灯实验
usb	U 盘方案实验
ext_flash	外部 NOR FLASH 读写实验
nand	NAND FLASH 读写实验
rtc	实时钟实验
sd_mmc	SD 卡读、写、擦出基本实验
net_cp2200	以太网 (CP2200) 控制实验
clkcfg	系统时钟配置实验
sd_reader	SD/MMC 读卡器实验

2.3.1 最小系统实验

最小系统实验主要用来验证目标板是否能正常工作，主要包括 GPIO 的操

作，即对一个连接到蜂鸣器和两个独立的发光二级管的 GPIO 进行操作。在正常的情况下，蜂鸣器会间隔发声，且每次发声时开发板上的 LD13 与 LD14 交替点亮。

2.3.2 UART 通信实验

UART 通信实验通过 STR71X 串口以查询的方式发送数据，首先实现 UART 的基本发送功能，即字节发送。最终实现如同标准 C 语言中 `printf(char * format, ...)` 函数类似的功能。

2.3.3 行列式键盘(4×4)扫描实验

4×4 行列式键盘扫描实验通过软件对键盘上的某一行(列)送低电平，其他行(列)为高电平，然后读取列(行)值。若列(行)值中有一位是低，则表明该行(列)与低电平对应列(行)的键被按下；否则扫描下一行(列)的方法，即扫描法对键盘进行扫描。然后根据读到的行列值的组合查表可获取按下键的码。本程序运行时，当有按键按下时程序将扫描的键值通过 UART3 发送到 PC 机，同时蜂鸣器蜂鸣一次。

2.3.4 EEPROM 操作实验

EEPORM 操作实验实现对 EEPROM 的读写操作。EEPORM 器件 CAT1025 与 ARM 的 IIC 接口相连接，通过调用 ARM STR71X I2C 部分的库函数实现对 CAT1025 器件指定的地址进行读取和存储数据操作。即通过对 EEPROM 器件指定地址写入一页数据，随后将先前写入的数据读出，在通过 UART3 发送到 PC 机，利用 PC 机上的串口工具查看结果。

2.3.5 定时器实验

定时器实验利用 TIMERO 的比较输出 A 方式进行定时，每 100MS 中断一次，在主程序中检测中断的次数，如果中断次数被 20 整除，则 Buzzer 蜂鸣一次，

同时通过 UART3 打印检测到的中断次数。

2.3.6 外部中断实验

外部中断实验利用开发板上的 EXINT 按键触发外部中断。当按键按下时响应外部中断，在中断例程中置位标志，且在主循环中不断检测标志，如果检测到标志 BUZZER 蜂鸣，提示有中断产生并且已经响应。

2.3.7 看门狗实验

看门狗实验首先启动看门狗定时器并禁止看门狗功能，当看门狗定时器中断 30 次，大约 3 秒钟后，停止看门狗定时器并允许看门狗功能，接着等待系统复位。实验结果和过程信息可以通过 UART3 发送到 PC 机通过串口工具显示相关信息。注：该实验需要生成目前文件下载到内部或者外部 FLASH 运行，本实验缺省设置为外部 FLASH 运行设置。

2.3.8 PWM 输出控制实验

该实验通过 PWM 输出不同频率的波形使得 BUZZER 交替发出两种不同的声音，或者控制直流电机的转速度。

2.3.9 ADC 实验

ADC 实验对 ARM STR710FZ 12 位 ADC 的通道 0 进行采样，采用单通道模式，并读取采样的原始值，其值通过 UART3 发送到 PC 机显示。

2.3.10 低功耗模式及唤醒实验

该实验中，程序运行时 LD13,LD14 轮流闪烁，可通过按键使 CPU 进入不同的模式。

按键	功能
KEY1	从 SLOW 模式恢复到正常模式
KEY2	进入 SLOW 模式

KEY3	进入 WFI 模式
KEY4	进入 STOP 模式
KEY5	进入 STANDBY 模式
KEY6	从 WFI、STOP、STANDBY 模式唤醒

2.3.11 SPI 控制 LED 的流水灯实验

LED 流水灯实验通过 SPI 接口向 74HC595D 发送数据来控制 8 路 LED 灯的亮灭，在实验的过程中需要将 P1.8 置成高电平，即 LED 公共端使能。

2.3.12 USB 操作(U 盘方案)实验

该实验实现 U 盘的部分功能，程序运行时用 USB 线与 PC 的 USB 接口连接，几秒钟左右后，PC 机的设备管理器的通用串行控制器栏会标识有 USB Mass Storage Device 设备插入，在磁盘驱动器栏标识检测到磁盘设备，与此同时在资源管理器中会出现可移动盘的标识及盘符。

2.3.13 外部 NOR FLASH 读写实验

该实验实现对开发板上的 NOR flash 进行读、写、擦除操作。程序运行时先读取 ID，并将 ID 值通过 UART3 发送到 PC，然后将指定的数据写入 NOR FLASH，之后将先前写入的数据读出通过 UART3 发送到 PC 显示。

2.3.14 NAND FLASH 读写实验

该实验实现对开发板上的 NAND flash 进行读、写、擦除操作。程序运行时先读取 ID，并将 ID 值通过 UART3 发送到 PC，然后将指定的数据写入 NAND FLASH，之后将先前写入的数据读出通过 UART3 发送到 PC 显示。

2.3.15 RTC 实验

RTC 实验实现对 STR710FZ 的内部集成 RTC 模块的时间日期进行设置和读取，并将读取时间日期值在主循环中通过 UART3 发送到 PC 利用串口调试助手

等工具显示读取结果。

2.3.16 SD/MMC 卡读写实验

该实验实现对 SD/MMC 卡的读，写，擦除操作。程序运行时，先获取卡的相关信息，通过 UART3 将卡信息发送到 PC 显示，之后擦除指定的扇区，并向这个扇区写指定的数据，随后读取所写的的数据，还通过 UART3 将读取的数据发送到 PC 显示。

2.3.17 以太网（CP2200）控制实验

该实验实现对 CP2200 以太网控制器的基本操作。程序运行时通过双绞线将开发板上的 RJ45 头与 PC 相连，并通过 PING 命令验证网络模块是否正常工作。

2.3.18 系统时钟配置实验

本实验练习如何对系统时钟进行配置，体验不同时钟频率对程序运行速度的影响。

带 UC/OS 实验程序存放目录结构

目录名	内容简介
plant	μC/OS-II 移植实验
multi_task	多任务控制实验
smart_uart	串口全功能实验
sd_reader	SD/MMC 读卡器实验

2.3.19 μC/OS-II 移植实验

该实验完成对 μC/OS-II 的移植,该实验建立了 2 个任务, INIT_TASK 用于初始化应用、OS 相关的组件, START_TASK 任务控制 LD13、LD14 有规律的量灭。达到验证 μC/OS-II 移植效果的目的。

2.3.20 多任务控制实验

本程序建立了 4 个任务 INIT_TASK 用于初始化应用、OS 相关的组件,

KPD_TASK 作为键盘扫描专用，LED_TASK 控制 74HC595 驱动的 8 个 LED 显示，MMI_TASK 任务为人机界面控制任务。当程序运行后 74HC595 控制的 8 个 LED 亮灭会循环变化，此时触发 0~5 按键，会出现不同的现象：按 K0:Buzzer 短叫一声，K1:Buzzer 长叫一声，K2:LD13 亮，LD14 灭，K3: LD14 亮，LD13 灭，K4: LD13 亮，LD14 亮，K5: LD13 灭，LD14 灭。即体验多任务与前后台系统控制程序的差别。



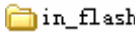


2.3.21 串口全功能实验

本实验采用串口的中断方式对数据收发，程序建立了 4 个任务 INIT_TASK 用于初始化应用，OS 相关的组件，KPD_TASK 作为键盘扫描专用，UART_RECV_TASK，MMI_TASK 用于人机界面管理。本程序的功能主要实现 UART3 接收、发送方式都采用中断驱动的方式，同时接收发送数据都使用了环状缓冲区，即要发送的数据先添加到发送缓冲区，同时允许发送中断，数据发送完成后禁止发送中断，直到有新的数据添加到发送缓冲区；当从 UART 收到一个字节数据时，将其添加到接收缓冲区由 UART_RECV_TASK 从接收缓冲区中读取数据，并将接收的数据以消息的方式发送到 MMI 任务。

2.3.22 SD 卡全功能实验

本实验通过 PC 软件“SD/MMC 卡读卡器实验软件”控制 STDV710 开发板上 SD 卡的初始化、擦除、读、写操作。程序建立了 3 个任务，INIT_TASK 用于初始化应用、OS 相关的组件，UART_RECV_TASK 专门处理数据接收，MMI_TASK 用于人机界面管理，本例通过与 PC 的串口进行通信，通过 PC 的软件控制 STDV710 开发板上 SD 卡实现对 SD 卡的初始化、擦除、读、写操作。

2.4 标准实验程序

目录名	内容简介
 TFT_LCD	TFT LCD 驱动实验
 kpd	LCD 与键盘实验
 in_flash	内部 FLASH 操作实验
 adc	LCD 与 ADC 实验
 rtc	RTC 与 LCD 实验

标准实验程序的运行结果多数通过 TFT-LCD 显示，同时也将程序运行过程和结果通过开发板上的 UART3 发送到 PC 通过诸如串口调试助手之类的工具软件显示。

不带 OS 实验程序存放目录结构

2.4.1 TFT LCD 驱动实验

本实验实现 TFT - LCD 驱动的基本功能：刷图、画线、画园、画弧线等操作。

2.4.2 LCD 与键盘实验

本实验实现 KPD 扫描，并将键盘扫描值在 TFT LCD 上显示。

2.4.3 内部 FLASH 操作实验

本实验对 STR710FZ 内部 FLASH 的 BLANK1 的读、写、擦出操作,即将随机写入的数据读出然后在 LCD 上循环显示。



2.4.4 LCD 与 ADC 实验

该实验对 ADC 通道 0 进行采样，将采样的原始值在 TFT LCD 上显示。

2.4.5 RTC 与 LCD 实验

该实验对 STR710FZ 的内部集成 RTC 模块进行操作，对时间日期进行设置与读取并将结果在 TFT LCD 上显示。

带 OS 实验程序存放目录结构

目录名	内容简介
 rtc_led_lcd_task	多任务控制实验二,基于 $\mu\text{C}/\text{OS-II}$
 kpd_lcd_task	多任务控制实验三,基于 Nucleus plus

2.4.6 多任务控制实验二

本实验主要为熟悉多任务的协同控制，该程序建立了 4 个任务，INIT_TASK 任务初始化专用，RTC_TASK 专门管理时钟刷新，MMI_TASK 界面显示任务 LED_TASK 控制由 SPI 控制得 8 个 LED 得亮灭。程序运行后首先显示一幅图片，两秒钟后画同心圆，接着显示另外一幅图片延时几秒后显示当前时间，随后重复上面得运行过程。

3.5.2 多任务控制实验三

该实验进一步熟悉优秀的嵌入式实时操作系统 Nucleus plus 多任务的控制方式及其使用。

注 2：本章所描述软件的开发环境均为 MDK

第三章 开发工具描述

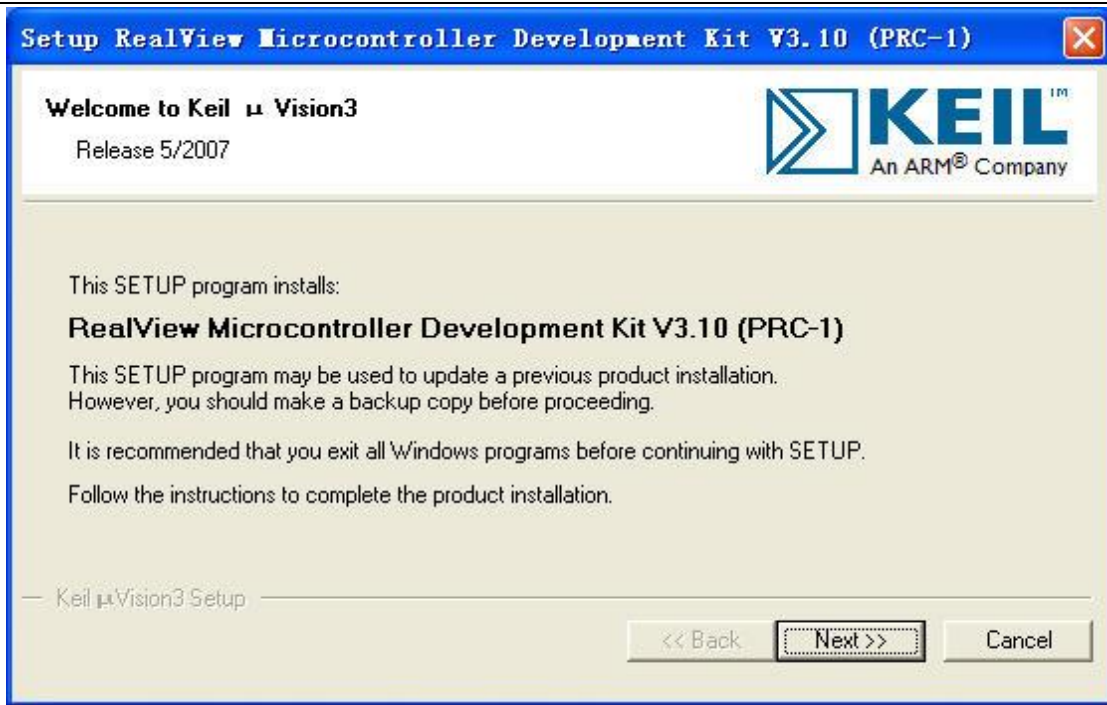
1. ARM 集成开发环境 ADS1.2

有关 ARM 集成开发环境 ADS1.2 的使用方法请参见 ARM STR71X 开发套件光盘或者我们网站上的<<ADS 使用简介>>PDF 文档。

2. ARM 集成开发环境 MDK 介绍

RealView MDK 开发套件是 ARM 公司目前最新推出的针对各种嵌入式处理器的软件开发工具。RealView MDK 集成了业内最领先的技术,包括 μ Vision3 集成开发环境与 RealView 编译器。支持 ARM7、ARM9 和最新的 Cortex-M3 核处理器,自动配置启动代码,集成 Flash 烧写模块,强大的 Simulation 设备模拟,性能分析等功能。

MDK 软件可以从 STM32 评估板配套光盘获取,或者从 <http://www.realview.com.cn> 下载中心下载最新版本。双击安装文件 setup.exe,出现如下的安装界面,根据界面安装向导的提示,完成 Keil u Vision3 的安装。

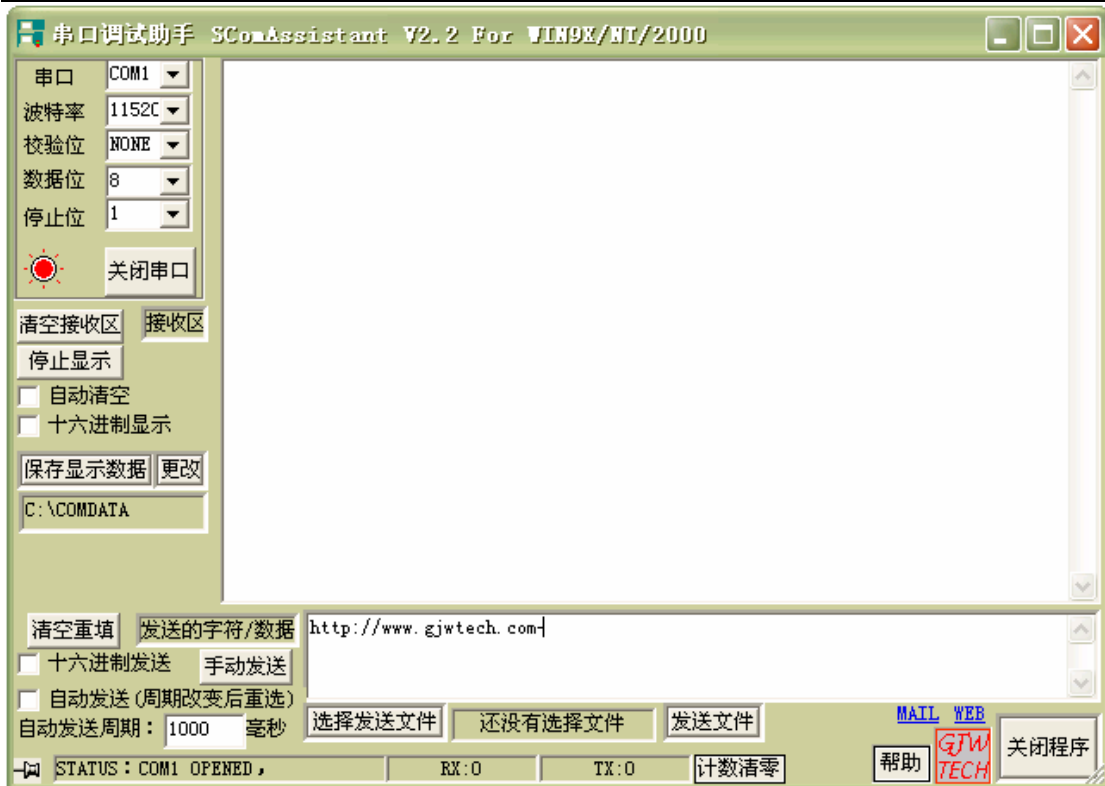


3. PC 串口工具

在我们大多数的实验工程中都有 TRACE 语句，即通过串口跟踪程序的运行过程和结果，因此需要一个 PC 串口工具显示这些 TRACE 信息，我们推荐使用免费工具“串口调试助”。

1. 查看 TRACE 信息

启动串口调试助，正确选择使用的串口端口号，设置波特率为 115200，其他串口参数默认设置，自动清空选项可以去掉。仅仅查看 TRACE 信息不需要十六进制显示。设置界面如下图：



2. PC 发送数据到 ARM STR71X 开发板

如果需要将数据从 PC 发送到开发板，可使用串口调试助手的发送功能，比如需要从 PC 以十六进制的方式发送 8 个数据到开发板，开发板将收到的数据回到 PC，串口调试助手的设置如下图：

