



S698-T 芯片开发系统 S698T-Dkit

使用说明书

(版本: V1.0)

珠海欧比特控制工程股份有限公司

地址: 广东珠海唐家东岸白沙路 1 号欧比特科技园 邮编: 519080
电话: 0756-3391979 传真: 0756-3391980 网址: www.myorbita.net



修改记录

日期	修改内容	版本变更	备注
2011-06-23	初次创建	V1.0	

前 言

感谢您使用我公司的 S698-T 芯片开发系统 S698T-DKit 产品。

为了使您能尽快熟练地操作本产品，我们随机配备了内容详细的使用说明书，在您第一次使用时，请务必仔细阅读所有随机资料。

基于提高部件及系统性能和可靠性的需要，我们有时会对系统（包括硬件和软件）做一些改动，届时，我们会尽量修改或增加资料，但仍可能在某些描述上与实际改动后的不一致，敬请谅解。本使用说明书中如有错误和疏漏之处，热切欢迎您的指正。

厂家相关信息

- 生产企业名称: **珠海欧比特控制工程股份有限公司**
- 生产企业地址: **广东省珠海市唐家东岸白沙路 1 号**
- 产品标准编号: **S698T-Dkit**
- 售后服务单位: **珠海欧比特控制工程股份有限公司**
- 联系方式:

地 址: **广东省珠海市唐家东岸白沙路 1 号**

电 话: **0756-3391979**

传 真: **0756-3391980**

邮 编: **519080**

目 录

1	产品简介	1
1.1	系统资源	2
1.2	技术指标	2
1.3	系统框图	3
1.4	系统布局	4
2	系统安装及连线	5
2.1	安装步骤	5
2.2	外部信号接线	6
2.2.1	硬件调试支持接口信号说明	6
2.2.2	RS232 接口说明	7
2.2.3	UART16550 接口说明	7
2.2.4	CAN 接口说明	8
2.2.5	ARINC429 接口说明	9
2.2.6	高速 (10Mbps) 1553B 接口说明	10
2.2.7	低速 (1Mbps) 1553B 接口说明	12
2.2.8	ADC、DAC 接口说明	12
3	操作说明	13
3.1	高速 1553B 总线模块	13
3.2	低速 1553B 总线模块	14
3.3	ARINC429 总线模块	14
3.4	CAN 总线模块	15
3.5	ADC 及 DAC 模块	15
3.6	UART16550 模块	16
3.7	DSU 及 UART 接口模块	17
3.8	数码管	17
3.9	存储器模块	18
3.10	外部中断模块	18
3.11	时钟的倍频及分频	20
3.12	LED 灯	23
3.13	测试点	24
3.14	其他说明	25
4	集成开发环境	26
4.1	安装 ORION 4.0 集成开发环境	26
4.2	ORION 4.0 的启动	27
4.3	ORION 4.0 使用说明	29
4.3.1	主窗口	29
4.3.2	选择工作间	30
4.3.3	工程的新建	30
4.3.4	编译调试环境的配置	34
4.3.5	调试运行环境的设置	36



4.3.6	工程的编译、链接.....	41
4.3.7	程序的调试.....	41
4.4	DSU命令行下的调试.....	45
4.4.1	DSU 调试步骤.....	45
4.4.2	FLASH程序固化.....	46
5	维护和故障处理.....	49
5.1	日常维护.....	49
5.2	故障处理.....	49
附录A: 订货信息.....		50
附录B: 装箱单.....		51

图目录

图 1-1 S698T-DKIT实物图.....	1
图 1-2 S698T-DKIT系统框图布局说明	3
图 1-3 S698T-DKIT布局图.....	4
图 2-1 S698T-DKIT接线示意图.....	5
图 2-2 DSU接头信号布局	6
图 2-3 RS232 接头信号布局.....	7
图 2-4 UART16550 接头信号布局	8
图 2-5 CAN接头信号布局	8
图 2-6 ARINC429 接头信号布局	9
图 2-7 高速 1553B接头信号布局.....	10
图 3-1 DAC模块的原理图	16
图 3-2 外部中断输入电路原理.....	19
图 3-3 时钟倍频及分频设置跳线帽.....	21
图 4-1 ORION 4.0 安装界面.....	27
图 4-2 ORION 4.0 启动界面.....	27
图 4-3 软件开发流程图	28
图 4-4 ORION 4.0 主界面.....	29
图 4-5 创作工作间	30
图 4-6 选择MANAGE C MAKE PROJECT	31
图 4-7 输入工程名	31
图 4-8 选择PROJECT TYPE.....	32
图 4-9 新建源文件 1	33
图 4-10 新建源文件 2	33
图 4-11 在工程中添加源文件.....	34
图 4-12 编译选项设置	34
图 4-13 选择EXTERNAL TOOLS 1	35
图 4-14 选择EXTERNAL TOOLS 2	35
图 4-15 CONFIGURATIONS	36
图 4-16 新建调试\运行环境 1	37
图 4-17 配置MAIN设置页	38
图 4-18 配置GRMON设置页	39
图 4-19 输入HOST和PORT值	39
图 4-20 配置DEBUGGER设置页	40
图 4-21 DEBUG视图	41
图 4-22 设置断点	42
图 4-23 观察内存	43
图 4-24 观察寄存器.....	44
图 4-25 删除调试.....	44

表目录

表 2-1 硬件调试接口 DSU 接口信号说明	6
表 2-2 RS232 接口信号说明	7
表 2-3 UART16550 接口信号说明	8
表 2-4 CAN 接口信号说明	8
表 2-5 ARINC429 接口信号说明	9
表 2-6 高速 1553B 接口信号说明	11
表 2-7 低速 1553B 接口说明	12
表 3-1 高速 1553B 和低速 1553B 信号跳线选择说明	13
表 3-2 MFIO 对应 MAX7219 的 SPI 接口信号说明	18
表 3-3 外部中断输入操作说明	19
表 3-4 倍频设置定义说明	22
表 3-5 分频设置定义说明	22
表 3-6 系统板所有的 LED 灯说明	23
表 3-7 系统板上测试点说明	24
表 3-8 其它跳线说明	25
表 5-1 故障处理方法	49

1 产品简介

S698-T 芯片开发系统 S698T-Dkit（以下简称 S698T-DKit）是一款专门针对欧比特公司的 S698-T SOC 芯片而研制的应用开发系统，随机还提供集成开发环境及开发例程等软件，可以让用户迅速的掌握 S698-T 芯片的应用开发。此外，S698T-DKit 还可以供 SPARC V8 嵌入式处理器用户学习和开发之用。



图 1-1 S698T-Dkit 实物图

S698-T 芯片以高性能的 SPARC V8 (IEEE-1754) 架构标准的 32 位 RISC 整数单元 IU 为主控内核，配以 IEEE-754 标准的 64 位双精度浮点处理单元 FPU。此外，S698-T 内部还集成了 1553B 总线控制器、ARINC429 总线控制器、CAN 总线控制器、多功能 IO 接口、UART 接口、在线硬件调试支持单元 DSU、DAC 模块、ADC 模块等多种功能模块。关于 S698-T 芯片的更多信息请参考《S698-T 芯片用户手册》，关于 S698-T 芯片应用开发的更多信息请参考《S698-T 芯片应用开发手册》。

1.1 系统资源

S698T-Dkit 具有资源丰富、操作简单、方便实用等特点，其包含的主要的系统资源如下：

- 1) S698-T 芯片；
- 2) 2 路的双冗余高速（10Mbps）1553B 总线；
- 3) 2 路的双冗余低速（1Mbps）1553B 总线；
- 4) 4 路接收和 2 路发送的 ARINC429 总线；
- 5) 2 路的 CAN 通讯总线，支持 BasicCAN 和 PeliCAN 模式；
- 6) 4 通道 10bit 的 DAC 模块，转换速率可达 100Msps；
- 7) 8 通道 10bit 的 ADC 模块，转换速率可达 1Msps；
- 8) 2 个 32bit 的定时器模块；
- 9) 2 个 24bit 的定时器模块；
- 10) 4 路外部中断输入；
- 11) 2 路 16550 兼容的 UART 接口；
- 12) 2 路 RS232 标准的 UART 接口；
- 13) 8 位 8 段数码管；
- 14) 4 个 LED 灯，由多功能 IO 控制；
- 15) 4M Byte SRAM；
- 16) 4M Byte FLASH；
- 17) 2x64Mbyte 的 SDRAM；
- 18) 硬件调试 DSU 接口，无需外置仿真器即可支持硬件在线调试；
- 19) 相关软件：
 - 应用例程；
 - v8mon 调试软件；
 - Orion 4.0 集成开发环境；

1.2 技术指标

- 供电：电源适配器供电，电源适配器输入为 100V~240VAC，输出为 5VDC/1A；
- 通讯接口：
 - ◇ 2 路双冗余 1553B 总线接口，通讯速率：10Mbps；
 - ◇ 2 路双冗余 1553B 总线接口，通讯速率：1Mbps；
 - ◇ 1 个具有 4 收 2 发通道的 ARINC429 总线接口，通讯速率：12.5Kbps、100Kbps、

48Kbps、50Kbps 或其它值；

- ◇ RS232 接口，波特率:1200~115200bps 可选；
- ◇ 16550 接口，波特率:1200~6.5Mbps 可选；
- ◇ 2 个独立的 CAN 通讯接口；
- 工作温度：-20℃~+75℃；
- 相对湿度：10%~80%；
- 外形尺寸：420*330*100mm³；

1.3 系统框图

S698T-Dkit 以 S698-T 芯片作为核心，集成 1553B 总线模块、ARINC429 总线模块、CAN 总线模块、在线硬件调试支持单元 DSU、DAC 模块、ADC 模块、多功能 IO 接口模块、UART16550 接口模块、RS232 接口模块等多种外围接口模块。S698T-Dkit 的系统框图如图 1-2 所示。

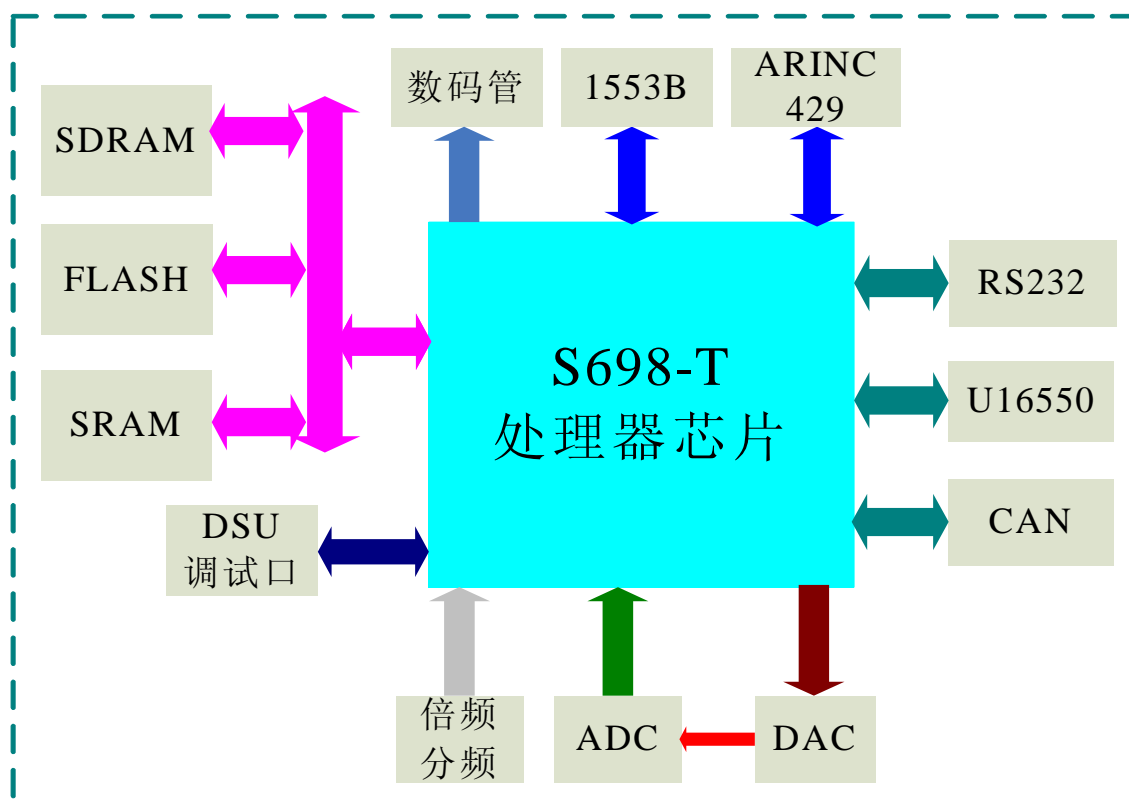


图 1-2 S698T-Dkit 系统框图布局说明

1.4 系统布局

S698T-Dkit 的系统布局如图 1-3 所示，通过布局示意图，用户可以快速的找到系统的各个功能模块。

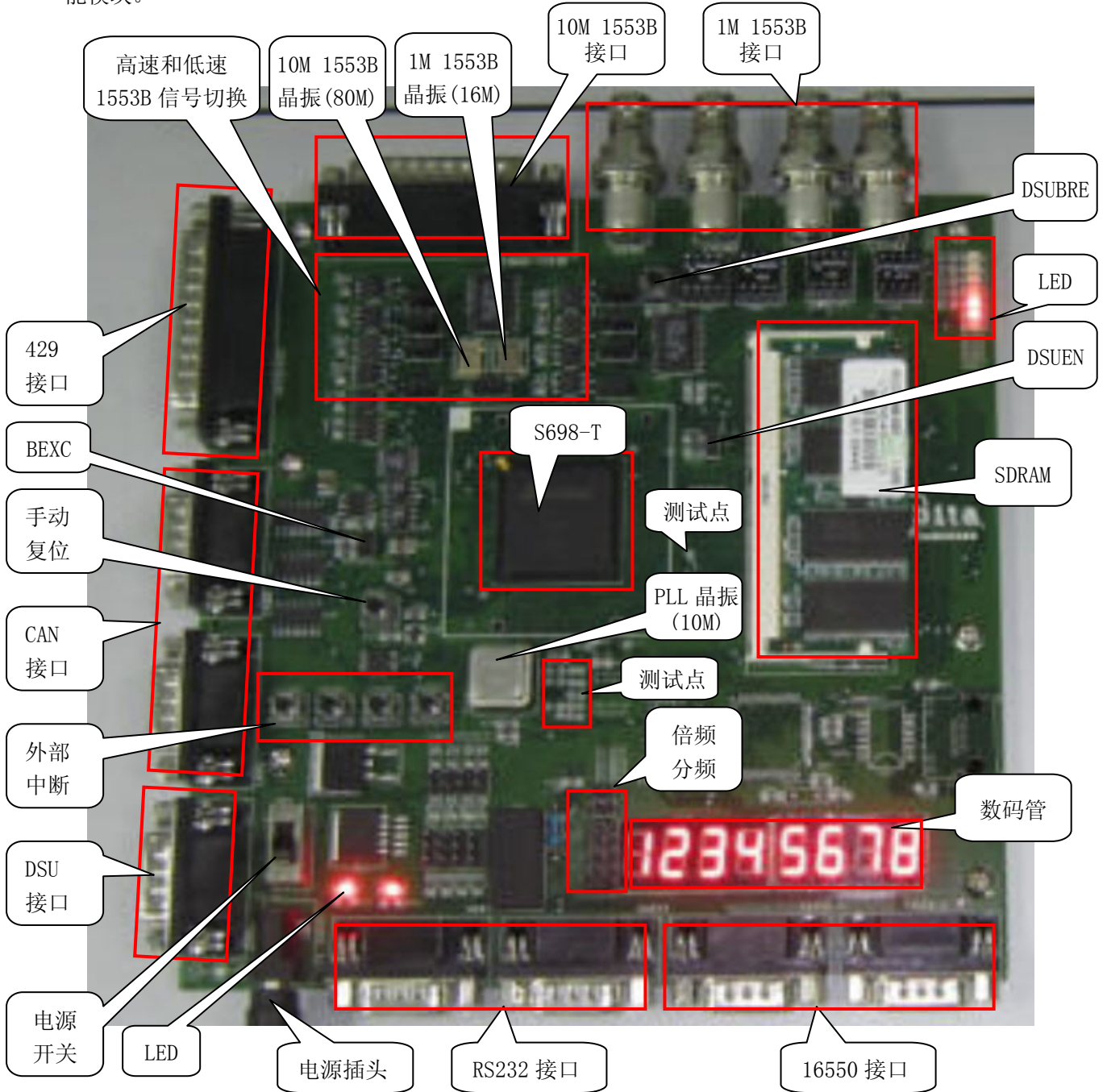


图 1-3 S698T-Dkit 布局图

2 系统安装及连线

在对 S698T-Dkit 系统进行信号接头或连线插拔时，务必请要先将电源断开，否则可能引起硬件的损毁。

2.1 安装步骤

第 1 步：检查产品包装箱

在您打开本产品的包装箱后应做以下检查：

- 检查包装箱内的设备及配件是否和装箱单上所列的产品型号、数量一致；
- 检查设备及配件是否完好，有没有在运输过程中损坏；

第 2 步：安装 Orion 4.0 开发环境

详见本文档的第 4 章或《ORION4.0 用户手册》。

第 3 步：根据使用需要进行信号接线

根据用户自身的需要，进行信号连线，如 DSU 调试线、1553B 通讯线、CAN 通讯线、CAN 通讯线等。图 2-1 为本产品的接线示意图。使用时，除了电源线是必须要连接的，其它的连线根据具体的需要，选择性进行连接，用不到的功能时，没必要连接相应的信号线。

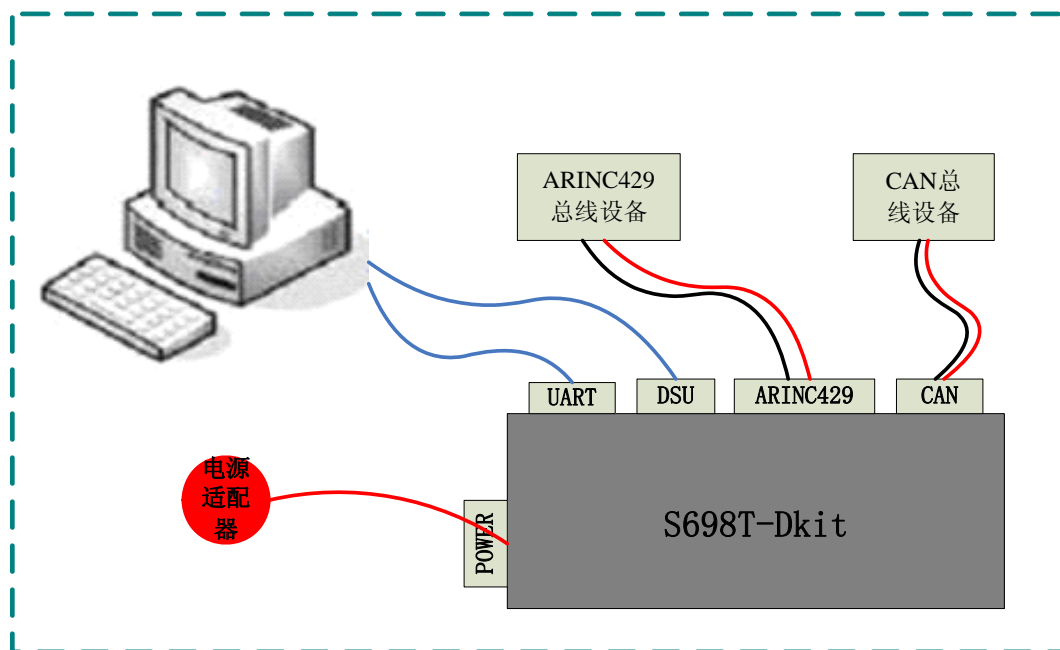


图 2-1 S698T-Dkit 接线示意图

第 4 步：给系统上电

信号接线完毕后，认真检查接线是否正确，确认无误后，将随机配发的电源适配器接通 220V 市电，给本产品上电。

第 5 步：进行系统操作

通电后，在集成开发环境等软件的支持下，可以通过调试接口，对系统进行进行调试、开发等操作。

2.2 外部信号接线

如图 1-3 所示，本产品外部接口包括：

- 一个 DSU 调试接口；
- 两个 RS232 串行接口；
- 两个 UART16550 接口
- 两个 CAN 总线接口；
- 4 收 2 发的 ARINC429 接口；
- 2 个高速（10Mbps）1553B 接口；
- 2 个低速（1Mbps）1553B 接口；

2.2.1 硬件调试支持接口信号说明

S698T-DKit 包含了 1 个硬件调试支持接口 DSU，其由 DB9 公座引出，对应 PCB 板上丝印标识为 DSU1，其具体的信号说明如图 2-2 以及表 2-1 所示。通过交叉串口线，把 DSU 接口跟 PC 机连接起来，配以相应的软件即可进行通讯。

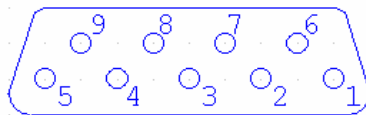


图 2-2 DSU 接头信号布局

表 2-1 硬件调试接口 DSU 接口信号说明

管脚号	信号说明
-----	------

2	数据接收 DSURX
3	数据发送 DSUTX
5	地线

注：其他未说明管脚悬空。

2.2.2 RS232 接口说明

S698T-DKit 包含两个 RS232 接口（对应 S698-T 芯片的普通通用异步串口 UART0 和 UART1），其信号都是由 DB9 公座引出，对应 PCB 板上的丝印为 UART0 和 UART1，具体的接口信号说明如如图 2-3 以及表 2-2 所示。通过交叉串口线，把 DSU 接口跟目标设备连接起来，配以相应的软件即可进行通讯。

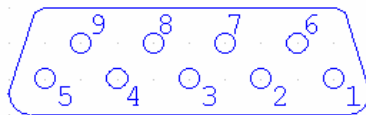


图 2-3 RS232 接头信号布局

表 2-2 RS232 接口信号说明

管脚号	信号说明
2	数据接收 UART_RXD
3	数据发送 UART_TXD
5	地线

注：其他未说明管脚悬空。

2.2.3 UART16550 接口说明

S698T-DKit 包含两个 UART16550 接口（对应 S698-T 芯片的 UART16550-0 和 UART16550-1），信号由 DB9 公座引出，对应 PCB 板上丝印为 16550_0、16550_1，具体的接口信号说明如图 2-4 以及表 2-3 所示。通过交叉串口线，把系统的 UART16550 接口跟目标设备连接起来，配以相应的软件即可进行通讯。

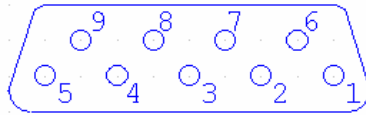


图 2-4 UART16550 接头信号布局

表 2-3 UART16550 接口信号说明

管脚号	信号说明
2	数据接收 UART16550_RXD
3	数据发送 UART16550_TXD
5	地线

注：其他未说明管脚悬空。

2.2.4 CAN 接口说明

S698T-DKit 包含两个 CAN 接口（对应 S698-T 芯片的 CAN-0 和 CAN-1），其信号都是由 DB9 公座引出，对应 PCB 板上接头丝印为 CAN_0、CAN_1，具体的接口信号说明如图 2-5 以及表 2-4 所示。通过 CAN 总线电缆（可用交叉串口线代替），把系统的 CAN 总线接口跟外部 CAN 总线设备连接起来，配以相应的软件即可进行通讯。

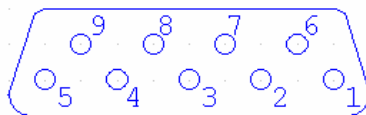


图 2-5 CAN 接头信号布局

表 2-4 CAN 接口信号说明

管脚号	信号说明
2	CAN 数据信号负 (CANL)
5	地线
7	CAN 数据信号正 (CANH)

注：其他未说明管脚悬空。

2.2.5 ARINC429 接口说明

S698T-DKit 包含 4 个 ARINC429 接收通道和 2 个 ARINC429 发送通道，信号由 DB25 公座引出，对应 PCB 板上的丝印为 ARINC429，具体的接口信号说明如图 2-6 以及表 2-5 所示。通过 ARINC429 总线电缆（屏蔽双绞线），把系统的 ARINC429 接口跟外部 ARINC429 总线设备连接起来，配以相应的软件即可进行通讯。

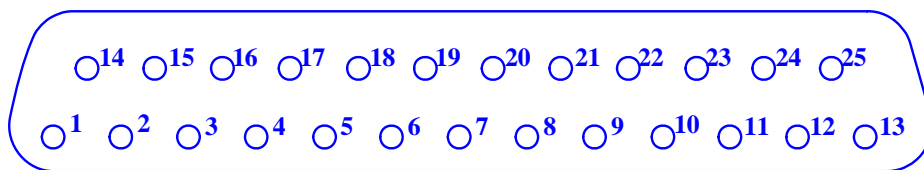


图 2-6 ARINC429 接头信号布局

表 2-5 ARINC429 接口信号说明

管脚号	信号说明
1	ARINC429 接收通道 1 输入信号 HI
2	接地 GND
3	ARINC429 接收通道 2 输入信号 0
4	ARINC429 接收通道 3 输入信号 L0
5	ARINC429 接收通道 4 输入信号 HI
6	悬空 NC
7	接地 GND
8	悬空 NC
9	悬空 NC
10	悬空 NC
11	ARINC429 发送通道 2 输出信号 HI

12	接地 GND
13	ARINC429 发送通道 1 输出信号 L0
14	ARINC429 接收通道 1 输入信号 L0
15	ARINC429 接收通道 2 输入信号 HI
16	ARINC429 接收通道 3 输入信号 HI
17	接地 GND
18	ARINC429 接收通道 4 输入信号 L0
19	接地 GND
20	接地 GND
21	接地 GND
22	接地 GND
23	悬空 NC
24	ARINC429 发送通道 2 输出信号 L0
25	ARINC429 发送通道 1 输出信号 HI

2.2.6 高速（10Mbps）1553B 接口说明

S698T-DKit 包含 2 路双冗余的高速（10Mbps）1553B 接口，高速 1553B 物理层采用 RS485 总线，接口驱动芯片选用的是 MAX3485E。高速 1553B 的信号由 DB25 公座引出，对应 PCB 板上丝印为 10M-1553B，具体的接口信号说明如图 2-7 以及表 2-6 所示。通过 RS485 总线电缆，把系统的高速 1553B 接口跟目标设备连接起来，配以相应的软件即可进行通讯。

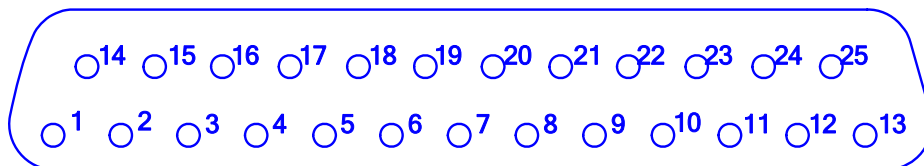


图 2-7 高速 1553B 接头信号布局

表 2-6 高速 1553B 接口信号说明

管脚号	信号说明
1	第 2 路高速 1553B 总线通道 B 的数据信号的 RS485 差分信号正
2	接地 GND
3	第 2 路高速 1553B 总线通道 B 的数据信号反的 RS485 差分信号负
4	第 2 路高速 1553B 总线通道 A 的数据信号的 RS485 差分信号负
5	第 2 路高速 1553B 总线通道 A 的数据信号反的 RS485 差分信号正
6	NC
7	接地 GND
8	NC
9	第 1 路高速 1553B 总线通道 B 的数据信号反的 RS485 差分信号负
10	第 1 路高速 1553B 总线通道 A 的数据信号反的 RS485 差分信号正
11	第 1 路高速 1553B 总线通道 B 的数据信号的 RS485 差分信号正
12	接地 GND
13	第 1 路高速 1553B 总线通道 A 的数据信号的 RS485 差分信号正
14	第 2 路高速 1553B 总线通道 B 的数据信号的 RS485 差分信号负
15	第 2 路高速 1553B 总线通道 B 的数据信号反的 RS485 差分信号正
16	第 2 路高速 1553B 总线通道 A 的数据信号的 RS485 差分信号正
17	接地 GND
18	第 2 路高速 1553B 总线通道 A 的数据信号反的 RS485 差分信号负
19	NC
20	NC
21	第 1 路高速 1553B 总线通道 B 的数据信号反的 RS485 差分信号正

22	接地 GND
23	第 1 路高速 1553B 总线通道 A 的数据信号反的 RS485 差分信号负
24	第 1 路高速 1553B 总线通道 B 的数据信号的 RS485 差分信号负
25	第 1 路高速 1553B 总线通道 A 的数据信号的 RS485 差分信号正

2.2.7 低速（1Mbps）1553B 接口说明

698T-DKit 包含 2 路双冗余低速（1Mbps）1553B 接口，信号由 1553B 接头 CBBJR79 引出，对应 PCB 板上的丝印为 1553B1、1553B2、1553B3、1553B4。CBBJR79 接头的中心芯点为信号正，中间芯点为信号负，最外层为屏蔽。通过 1553B 总线电缆，把系统的低速 1553B 接口跟外部低速 1553B 总线设备连接起来，配以相应的软件即可进行通讯。具体的每个接口通道说明如表 2-7 所示。

表 2-7 低速 1553B 接口说明

接口接头丝印	接口说明
1553B1	第 1 路高速 1553B 总线的通道 A
1553B2	第 1 路高速 1553B 总线的通道 B
1553B3	第 2 路高速 1553B 总线的通道 B
1553B4	第 2 路高速 1553B 总线的通道 A

2.2.8 ADC、DAC 接口说明

S698-T 芯片集成了 10bit 的 DAC 模块和 10bit 的 ADC 模块。其中 DAC 模块支持 4 通道输出，转换速率可高达 100MSPS，ADC 模块支持 8 通道输入，转换速率可高达 1MSPS。

在设计 S698T-DKit 时，把 S698-T 芯片的 DAC 的 4 个输出分别经过射随放大电路后直接连接到 S698-T 芯片的 ADC 的 4 个输入（ADC0~ADC3），这样无须进行外部接线即可进行 S698-T 芯片的 ADC 以及 DAC 模块的设计开发。

3 操作说明




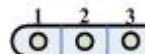
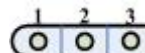


3.1 高速 1553B 总线模块

S698T-DKit 包含 2 路双冗余高速(10Mbps)1553B 总线接口,10Mbps 1553B 信号经过 MAX3485E 驱动输出,通过 1553B 总线电缆网络实现跟外部高速 1553B 总线设备之间进行数据通讯。高速 1553B 总线模块的操作步骤如下:

- 1、依照 2.2.6 节的高速 1553B 总线接头的信号说明,采用 485 电缆把本系统与目标系统设备连接起来;
- 2、采用交叉串口线,把本系统的 DSU 接口跟调试计算机的串口连接起来;
- 3、依照表 3-1 中“高速 1553B 信号设置”,使用跳线帽,连接 JMP1、JMP5、JMP6、JMP10、JMP11、JMP12、JMP13、JMP17、JMP18,连通高速 1553B 信号;
- 4、在集成开发环境中进行高速 1553B 模块的控制程序编写,并编译生成可执行文件;
- 5、给系统正常上电,把编译好的可执行文件下载到系统中,然后运行,查看数据通讯是否正常。

注:高速 1553B 总线模块的参考例程位于配套光盘中的“\参考程序\1553B\1553B-10M”目录。

表 3-1 高速 1553B 和低速 1553B 信号跳线选择说明

跳线选择	高速 1553B 信号设置	低速速 1553B 信号设置
JMP1 	短路 2、3 脚	短路 1、2 脚
JMP5 	短路 2、3 脚	短路 1、2 脚
JMP6 	短路 2、3 脚	短路 1、2 脚
JMP10 	短路 2、3 脚	短路 1、2 脚
JMP11 	短路 2、3 脚	短路 1、2 脚
JMP12 	短路 2、3 脚	短路 1、2 脚
JMP13 	短路 2、3 脚	短路 1、2 脚

JMP17 	短路 2、3 脚	短路 1、2 脚
JMP18 	短路 2、3 脚	短路 1、2 脚

3.2 低速 1553B 总线模块

S698T-DKit 包含 2 路双冗余低速 (1Mbps) 1553B 总线接口, 实现了低速 (1Mbps) 1553B 总线设备的功能。1553B 数据信号经过 1553B 信号专用驱动芯片 HI-1573 和变压器驱动输出, 通过 1553B 总线电缆网络实现跟外部低速 1553B 总线设备之间进行数据通讯。低速 1553B 总线模块的操作步骤如下:

- 1、依照 2.2.7 节的低速 1553B 总线接口说明, 采用 1553B 总线电缆把本系统的低速 1553B 接口 (1553B1/1553B2/1553B3/1553B4) 跟目标设备连接起来;
- 2、采用交叉串口线, 把系统的 DSU 接口跟调试计算机的串口连接起来;
- 3、依照表 3-1 中“低速 1553B 信号设置”, 使用跳线帽, 连接 JMP1、JMP5、JMP6、JMP10、JMP11、JMP12、JMP13、JMP17、JMP18, 连通低速 1553B 信号;
- 4、在集成开发环境中进行低速 1553B 模块的控制程序编写, 并编译生成可执行文件;
- 5、给系统正常上电, 把编译好的可执行文件下载到系统中, 然后运行, 查看数据通讯是否正常。

注: 低速 1553B 总线模块的参考例程位于配套光盘中的“\参考程序\1553B\1553B-1M”目录。

3.3 ARINC429 总线模块

S698T-DKit 包含 4 个 ARINC429 数据接收通道和 2 个 ARINC429 数据发送通道, 每个通道之间互相独立, 可独立实现跟外部 ARINC429 总线设备进行数据通讯。该模块的操作步骤如下:

- 1、依照 2.2.5 节的 ARINC429 总线接口信号说明, 采用 ARINC429 总线电缆 (屏蔽双绞线) 把本系统的 ARINC429 总线接口跟目标设备连接起来;
- 2、采用交叉串口线, 把系统的 DSU 接口跟调试计算机的串口连接起来;
- 3、在集成开发环境中进行 ARINC429 总线模块的控制程序编写, 并编译生成可执行文件;
- 4、给系统正常上电, 把编译好的可执行文件下载到系统中, 然后运行, 查看数据通讯是否

正常。

注：ARINC429 总线模块的参考例程位于配套光盘中的“\参考程序\ARINC429”目录。

3.4 CAN 总线模块

S698T-DKit 包含了 2 路 CAN 总线接口，支持支持 BasicCAN 和 PeliCAN 模式，可实现系统跟外部 CAN9 总线设备进行数据通讯。该模块的操作步骤如下：

- 1、依照 2.2.4 章的 CAN 总线接口信号说明，采用屏蔽 CAN 总线电缆（蔽双绞线）把本系统的 CAN 总线接口 CAN0 或 CAN1 跟外部的目标 CAN 总线设备连接起来；
- 2、采用交叉串口线，把系统的 DSU 接口跟调试计算机的串口连接起来；
- 3、在集成开发环境中进行 CAN 总线模块的控制程序编写，并编译生成可执行文件；
- 4、给系统正常上电，把编译好的可执行文件下载到系统中，然后运行，查看数据通讯是否正常。

注：CAN 总线模块模块的参考例程位于配套光盘中的“\参考程序\CAN”目录。

3.5 ADC 及 DAC 模块

S698-T 芯片集成了 10bit 的 DAC 模块和 10bit 的 ADC 模块。其中 DAC 模块支持 4 通道输出，转换速率可高达 100MSPS，ADC 模块支持 8 通道输入，转换速率可高达 1MSPS。

在设计 S698T-DKit 时，把 S698-T 芯片的 DAC 的 4 个输出分别经过射随放大电路后直接连接到 S698-T 芯片的 ADC 的 4 个输入（ADC0~ADC3），这样无须进行外部接线即可进行 S698-T 芯片的 ADC 以及 DAC 模块的设计开发。

ADC 以及 DAC 模块的操作步骤如下：

- 1、采用交叉串口线，把系统的 DSU 接口跟调试计算机的串口连接起来；
- 2、在集成开发环境中进行 DAC 以及 ADC 模块的控制程序编写，并编译生成可执行文件；
- 3、给系统正常上电，把编译好的可执行文件下载到系统中，然后运行，输出结果。

注：ADC 以及 DAC 模块的参考例程位于配套光盘中的“\参考程序\ADC+DAC”目录。

特别说明

- (1) S698-T 的 DAC 模块是电流型驱动数模转换器，每个转换输出引脚最大的驱动能力为 34.1mA。
- (2) 若驱动电流型的负载，则可直接将 DAC 的输出引脚连接至被驱动对象；
- (3) 若驱动电压型的负载，则需在输出引脚输出连接一变换电阻 R_L （图中为 37.5 为欧姆）到地，使得输出的电流流经电阻后转换成电压，然后经过一个射极跟随器进行输出阻抗变换，再接到负载，射极跟随器输出端的输出电压值计算方法如下（单位 mV）：

$$V_{out} = I_{out} \times R_L = 34.1 \times \frac{D}{1023} \times R_L$$

其中： D 为该通道控制寄存器 out 域换算成十进制的值；

R_L 为变换电阻的值，单位为欧姆；

V_{out} 为射极跟随器输出端的电压；

从上式中可以看出，虽然 V_{out} 与 $D \cdot L$ 成正比，但 V_{out} 的值不可超过 3.3V。

推荐的 DAC 模块的原理图如下：

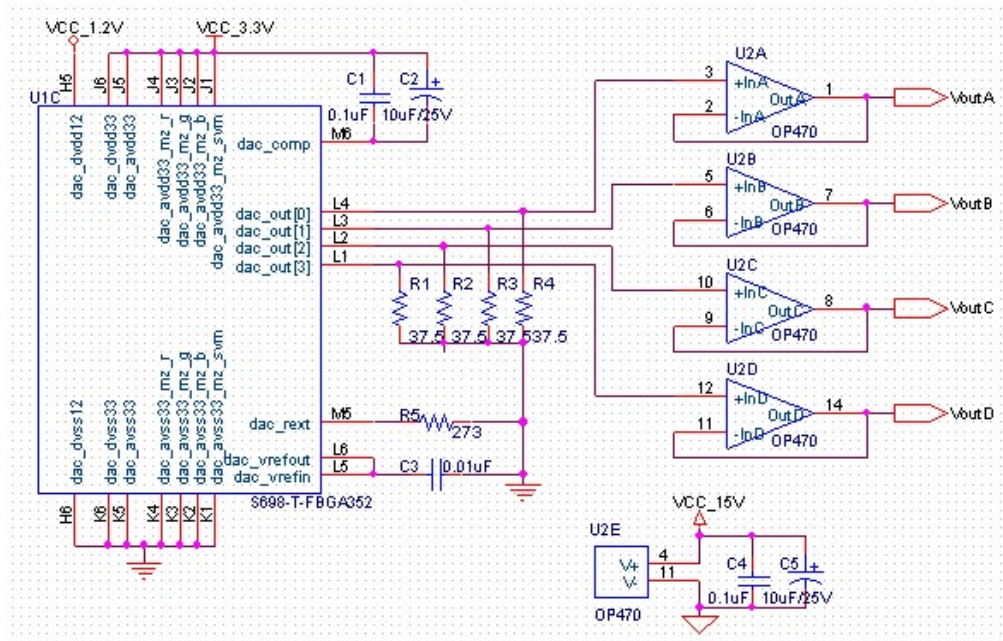


图 3-1 DAC 模块的原理图

3.6 UART16550 模块

本系统集成 2 个独立的 UART16550 总线接口，实现了 UART16550 总线设备的功能，可独立实现跟外部 UART16550 总线设备进行数据通讯。该模块的操作步骤如下：

- 1、依照 2.2.3 节的 UART16550 总线接口信号说明，采用交叉串口线把本系统的 UART16550 总线接口跟外部 UART16550 总线设备连接起来；
- 2、采用交叉串口线，把系统的 DSU 接口跟调试计算机的串口连接起来；
- 3、在集成开发环境中进行 UART16550 总线模块的控制程序编写，并编译生成可执行文件；
- 4、给系统正常上电，把编译好的可执行文件下载到系统中，然后运行，查看数据通讯是否正常。

注：UART16550 模块的参考例程位于配套光盘中的“\参考程序\UART16550”目录。

3.7 DSU 及 UART 接口模块

S698T-DKit 包含 1 个 DSU 接口和 2 个独立的 RS232 标准的 UART 接口，DSU 接口用于系统调试用，2 个 RS232 接口可以实现系统跟外部 RS232 总线设备之间进行数据通讯。DSU 模块的操作在各个模块的操作中已经全部涉及到，在此不作详述。RS232 接口模块的操作步骤如下：

- 1、依照 2.2.1 节以及 2.2.2 节中的接口信号说明，采用交叉串口线把本系统的 RS232 总线接口跟外部 RS232 总线设备连接起来；
- 2、采用交叉串口线，把系统的 DSU 接口跟调试计算机的串口连接起来；
- 3、在集成开发环境中进行 RS232 总线模块的控制程序编写，并编译生成可执行文件；
- 4、给系统正常上电，把编译好的可执行文件下载到系统中，然后运行，查看数据通讯是否正常。

注：RS232 模块的参考例程位于配套光盘中的“\参考程序\UART”目录。

3.8 数码管

为了测试/开发 S698-T 的多功能 I/O 口 MFIO，S698T-DKit 设计了 8 位数码管，用于系统信息显示，选用了数码管驱动芯片 MAX7219，MAX7219 支持 SPI 接口，S698T-DKit 系统采用多功能 I/O 脚来模拟 SPI 总线接口，以控制和操作数码管。

MAX7219 是一种高集成化的串行输入/输出的共阴极 LED 显示驱动器。每片可驱动 8 位 7 段加小数点的共阴极数码管，可以数片级联，而与微处理器的连接只需 3 根线。MAX7219 内部设有扫描电路，除了更新显示数据时从单片机接收数据外，平时独立工作，极大地节省了 MCU 有限的运

行时间和程序资源。MAX7219 芯片上包括 BCD 译码器、多位扫描电路、段驱动器、位驱动器和用于存放每个数据位的 8×8 静态 RAM 以及数个工作寄存器。通过指令设置这些工作寄存器，可以使 MAX7219 进入不同的工作状态。

S698T-DKit 系统采用 S698-T 芯片的多功能 I/O 脚来模拟 SPI 总线接口，具体的信号定义如表 3-2 所示。

表 3-2 mfio 对应 Max7219 的 SPI 接口信号说明

S698-T 的 mfio	MAX7219 的 SPI 接口信号
mfio[15]	LOAD
mfio[14]	CLK
mfio[13]	DIN

注：数码管控制（即 MFIO）的参考例程位于配套光盘中的“\参考程序\LED”目录。

3.9 存储器模块

S698T-DKit 系统集成了 4M Byte SRAM 和 4M Byte FLASH，用于存储数据和程序，SRAM 和 FLASH 芯片焊接在系统板的背面，SRAM 的位宽为 32 位，FLASH 的位宽为 16 位。

S698T-DKit 系统集成了 SDRAM 接口，支持 2 个 BANK 的 PC100/PC133 型的 SDRAM 存储器，支持 8~12 位的列地址（Column-address）、最大支持 13 位行地址（Row-address），数据总线的宽度固定为 32 位模式。SDRAM 接口为标准的 SDRAM 座，用户可以根据系统需要选择更换不同容量的 SDRAM。

3.10 外部中断模块

S698-T 处理器支持 8 路外部中断输入（extint0~7），每个外部中断的触发条件可以软件配置成：高电平、低电平、上升沿、下降沿触发。

S698T-DKit 系统为外部中断 0、1、2、3（由于原理雷同，故本系统中没有为外部中断 4~7 设计触发电路）设计了能产生高电平、低电平、上升沿、下降沿四种触发条件的产生电路，如图

3-2 所示。

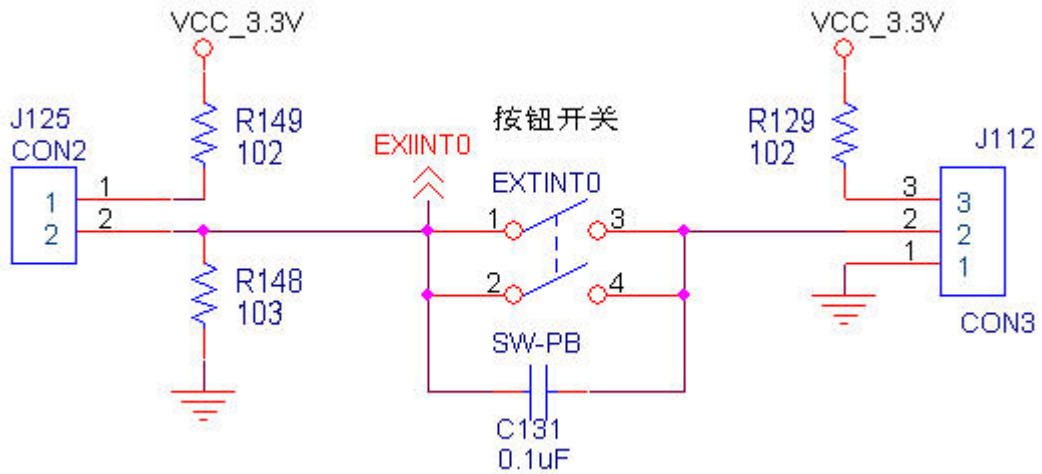
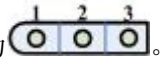


图 3-2 外部中断输入电路原理

如图 3-2 所示，通过 J125、J112 和按钮开关 EXTINT0 可以选择高电平、低电平、上升沿、下降沿输入。S698T-DKit 系统为外部中断 0、1、2、3 分别设计了同样的输入电路，通过跳线以及按钮可以实现为 4 个外部中断输入不同的触发条件的中断信号，具体操作如表。

表 3-3 外部中断输入操作说明

外部中断源	中断触发条件	操作说明
外部中断 0	高电平触发	用跳线帽短路 J125，J112 保持断开；
	低电平触发	J125、J112 都保持断开；
	上升沿触发	用跳线帽短路 J112 的 2、3 脚，J125 保持断开，按下按钮开关时将产生上升沿输入；
	下降沿触发	用跳线帽短路 J125，同时短路 J112 的 1、2 脚，按下按钮开关时将产生下降沿输入；
外部中断 1	高电平触发	用跳线帽短路 J113，J114 保持断开；
	低电平触发	J113、J114 都保持断开；
	上升沿触发	用跳线帽短路 J114 的 2、3 脚，J113 保持断开，按下按钮开关时将产生上升沿输入；
	下降沿触发	用跳线帽短路 J113，同时短路 J114 的 1、2 脚，

		按下按钮开关时将产生下降沿输入；
外部中断 2	高电平触发	用跳线帽短路 J115, J116 保持断开；
	低电平触发	J115、J116 都保持断开；
	上升沿触发	用跳线帽短路 J116 的 2、3 脚, J115 保持断开, 按下按钮开关时将产生上升沿输入；
	下降沿触发	用跳线帽短路 J115, 同时短路 J116 的 1、2 脚, 按下按钮开关时将产生下降沿输入；
外部中断 3	高电平触发	用跳线帽短路 J117, J118 保持断开；
	低电平触发	J117、J118 都保持断开；
	上升沿触发	用跳线帽短路 J118 的 2、3 脚, J117 保持断开, 按下按钮开关时将产生上升沿输入；
	下降沿触发	用跳线帽短路 J117, 同时短路 J118 的 1、2 脚, 按下按钮开关时将产生下降沿输入；
备注：		
其中 J112、J114、J116、J118 为三脚 2.54mm 间距的插针，它的管脚分布为  。		

3.11 时钟的倍频及分频

S698-T 芯片内部有五个时钟域，

- (1) **IU 时钟**：供 IU、FPU、cache 和 Register File 模块使用；
- (2) **IUN 时钟**：供 Register File 模块使用；
- (3) **SYSCLK 时钟**：供 AHB、APB 和片内外设模块使用；
- (4) **1553 时钟**：供 1553 模块使用；
- (5) **ADC 时钟**：供 ADC 模块使用；

其中 1553 时钟和 ADC 时钟由相关引脚从片外直接输入，而 IU 时钟、IUN 时钟、SYSCLK 时钟则由芯片内部的时钟管理模块提供的时钟信号驱动。

S698-T 时钟管理模块内部集成了锁相环（PLL）模块，锁相环有 2 种工作模式：NORMAL 模式和 BYPASS 模式，模式的选择由图 3-3 中的 PLL_BYPS1 跳线帽来选择：

- 锁相环 NORMAL 模式：PLL_BYPS1 连接到 GND（即摘除跳线帽）；
- 锁相环 BYPASS 模式：PLL_BYPS1 连接到 3.3V（即接上跳线帽）；

锁相环(PLL)模块的倍频数为 M, M 取值范围为 0~25, 由图 3-3 中的 PLL_MLT0~PLL_MLT3 跳线帽来设定, 具体如表 3-4 所示。

锁相环 (PLL) 模块的分频数为 D, D 取值范围为 1、2、4、8, 由图 3-3 中的 PLL_DIV0~PLL_DIV1 跳线帽来设定, 具体如表 3-5 所示。

时钟频率计算方法:

- 锁相环 NORMAL 模式下:
 - $F_{IUCLK} = F_{IUCLKN} = F_{CLOCK} * M$
 - $F_{SYSCLK} = F_{IUCLK} / D$
- 锁相环 BYPASS 模式下:
 - $F_{IUCLK} = F_{IUCLKN} = F_{CLOCK}$
 - $F_{SYSCLK} = F_{IUCLK} / D$

其中, F_{IUCLK} : IUCLK 时钟频率;
 F_{IUCLKN} : IUCLKN 时钟频率;
 F_{SYSCLK} : SYSCLK 时钟频率;
 F_{CLOCK} : 外部参考时钟的频率;
 M : 倍频值;
 D : 分频值;

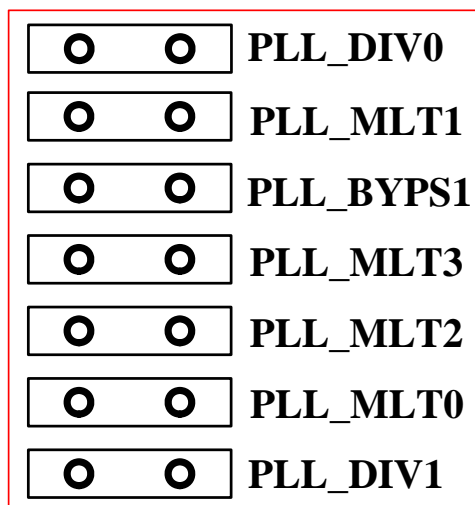


图 3-3 时钟倍频及分频设置跳线帽

表 3-4 倍频设置定义说明

PLL_MLT3	PLL_MLT2	PLL_MLT1	PLL_MLT0	倍频倍数 M
0	0	0	0	10
0	0	0	1	11
0	0	1	0	12
0	0	1	1	13
0	1	0	0	14
0	1	0	1	15
0	1	1	0	16
0	1	1	1	17
1	0	0	0	18
1	0	0	1	19
1	0	1	0	20
1	0	1	1	21
1	1	0	0	22
1	1	0	1	23
1	1	1	0	24
1	1	1	1	25

备注：

- 1、使用跳线帽短接表示“1”，不短接表示“0”；
- 2、使能 PLL 功能需要使用跳线帽短接 PLL_BYPS1 连接器；
- 3、倍频公式：主频=10+PLLMUL[3:0]。

表 3-5 分频设置定义说明

PLL_DIV1	PLL_DIV0	D
0	0	1 分频

0	1	2 分频
1	0	4 分频
1	1	8 分频
备注：使用跳线帽短接表示“1”，不短接表示“0”；		

3.12 LED 灯

S698T-DKit 系统设计了一些 LED 灯，它们具体代表的含义进行详细说明，具体如表 3-6 所示。

表 3-6 系统板所有的 LED 灯说明

LED 丝印标号	含义说明
D911	系统电源 3.3V 指示灯： <ul style="list-style-type: none"> 灯亮：表示系统电源 3.3V 正常； 灯灭：表示系统电源 3.3V 异常。
D912	系统电源 5V 指示灯： <ul style="list-style-type: none"> 灯亮：表示系统电源 5V 正常； 灯灭：表示系统电源 5V 异常。
MFIO13	第 2 路多功能 IO 输出指示灯： <ul style="list-style-type: none"> 灯亮：表示第 2 路 IO 输出低电平； 灯灭：表示第 2 路 IO 输出低电平； 灯闪：表示第 2 路 IO 输出脉冲信号。
MFIO14	第 3 路多功能 IO 输出指示灯： <ul style="list-style-type: none"> 灯亮：表示第 3 路 IO 输出低电平； 灯灭：表示第 3 路 IO 输出低电平； 灯闪：表示第 3 路 IO 输出脉冲信号。
MFIO15	第 4 路多功能 IO 输出指示灯： <ul style="list-style-type: none"> 灯亮：表示第 4 路 IO 输出低电平； 灯灭：表示第 4 路 IO 输出低电平； 灯闪：表示第 4 路 IO 输出脉冲信号。
MFIO16	第 1 路多功能 IO 输出指示灯： <ul style="list-style-type: none"> 灯亮：表示第 1 路 IO 输出低电平； 灯灭：表示第 1 路 IO 输出低电平； 灯闪：表示第 1 路 IO 输出脉冲信号。

ERROR1	S698-T 处理器错误指示信号灯： <ul style="list-style-type: none"> • 灯亮：处理器处于错误模式； • 灯灭：处理器处于正常模式时灯灭。
DSUACT1	DSU 激活状态指示信号灯： <ul style="list-style-type: none"> • 灯灭：表示 DSU 被激活，此时处理器处于 Debug 模式，被 DSU 控制； • 灯亮：表示 DSU 未被激活。
WDOG1	看门狗输出指示信号灯： <ul style="list-style-type: none"> • 灯灭：表示看门狗输出低电平； • 灯亮：表示看门狗输出高电平，处于上拉状态。

3.13 测试点

为了方便故障查找定位以及部分关键信号的测量，S698T-DKit 系统板上设置了部分测试点，其具体含义如表 3-7 所示。

表 3-7 系统板上测试点说明

测试点丝印标号	测试点的信号说明
T0	系统时钟信号。
T1	外部存储器数据总线方向指示信号 DDIR： <ul style="list-style-type: none"> • 高电平：表示数据总线的数据流方向由 S698-T 流向存储器（即写操作）； • 低电平：表示数据总线的数据流方向存储器由流向 S698-T（即写操作）。
T2	S698-T 复位输出信号。
T3	外部存储器数据总线方向指示信号 READ： <ul style="list-style-type: none"> • 低电平：表示数据总线的数据流方向由 S698-T 流向存储器（即写操作）； • 高电平：表示数据总线的数据流方向存储器由流向 S698-T（即写操作）。
T4	romsn[0]
T5	rwen[0]，SRAM BANK0 写使能信号，低电平有效。
T6	rwen[1]，SRAM BANK1 写使能信号，低电平有效。
T7	rwen[2]，SRAM BANK2 写使能信号，低电平有效。
T8	rwen[3]，SRAM BANK3 写使能信号，低电平有效。

T9	ramoen[0], SRAM BANK0 读使能信号, 低电平有效。
T10	ramoen[1], SRAM BANK1 读使能信号, 低电平有效。
T11	ramoen[2], SRAM BANK2 读使能信号, 低电平有效。
T12	ramoen[3], SRAM BANK3 读使能信号, 低电平有效。
T13	ramoen[4], SRAM BANK4 读使能信号, 低电平有效。
T14	ramsn[1], SRAM BANK1 片选信号, 低电平有效。
T15	ramsn[2], SRAM BANK2 片选信号, 低电平有效。
T16	ramsn[3], SRAM BANK3 片选信号, 低电平有效。
T17	ramsn[0], SRAM BANK0 片选信号, 低电平有效。
A0	address[0]
A25	address[25]
A27	address[24]

3.14 其他说明

表 3-8 其它跳线说明

跳线连接器丝印	说明
跳线 DSUEN1	DSU 调试模式使能控制信号： <ul style="list-style-type: none"> • 1: DSU 被使能, 处理器处于调试模式; • 0: DSU 被屏蔽, 处理器处于正常模式; 注: 使用跳线帽短接表示“1”, 不短接表示“0”。
跳线 BEXC1	外部存储器总线异常指示信号： <ul style="list-style-type: none"> • 1: BEXCN 信号为高电平; • 0: BEXCN 信号为低电平;; 如果 S698-T 的 MCFG1 寄存器的 bit25=1, 则 S698-T 在读取 ROM、SRAM 或 IO 时, 将会同时采样 BEXCN 信号与数据总线信号, 若此时 BEXCN 信号有效 (为低电平), 这片内将产生一个 AHB 总线错误中断 (一级中断控制器的 1 号中断)。 注: 使用跳线帽短接表示“1”, 不短接表示“0”。

4 集成开发环境

对于嵌入式系统的开发，其应用软件的开发属于跨平台的开发，所以需要有一个集成开发环境。其开发模式是，在一台通用计算机运行软件的编辑编译等操作，然后下载到嵌入式设备中进行运行和调试。用来开发的通用计算机可以选择比较常见的 PC 机、工作站等，运行 Windows 或 Linux 操作系统，开发计算机一般被称为宿主机，而嵌入式设备被称为目标机。在宿主机上编写应用程序，然后通过交叉编译程序编译，下载到目标机上运行，交叉开发环境提供调试工具对目标机上运行的程序进行调试。

S698T-Dkit 有三种调试方式。其分别是：

- Linux 操作系统下的集成开发环境的交叉调试方式。
- Windows 操作系统下的集成开发环境的交叉调试方式。
- DSU 命令行下的调试方式。

这里主要举例介绍 Windows 操作系统下的集成开发环境 Orion 4.0 的交叉调试方法，然后再介绍 DSU 命令行下的调试方式，Windows 操作系统下的集成开发环境 V8mon 的调试方法在此不作详述，用户可参考《v8mon 使用手册》进行调试。

请用户在使用前请安装 Orion 4.0 集成开发环境，下面介绍 Orion 4.0 集成开发环境的安装与使用。

4.1 安装 Orion 4.0 集成开发环境

直接双击附带光盘中的 Orion4.0 的安装文件 setup.exe，系统将引导您安装 Orion 4.0，安装过程相对简单，这里不做过多介绍。



图 4-1 Orion 4.0 安装界面

4.2 Orion 4.0 的启动

在 Orion4.0 软件安装完毕，即可使用该软件，双击 Orion4.0 应用程序图标，软件启动界面如图 4-2 所示。软件启动后即可进行软件开发，基于 Orion 4.0 的软件开发流程如图 4-3 所示。



图 4-2 Orion 4.0 启动界面

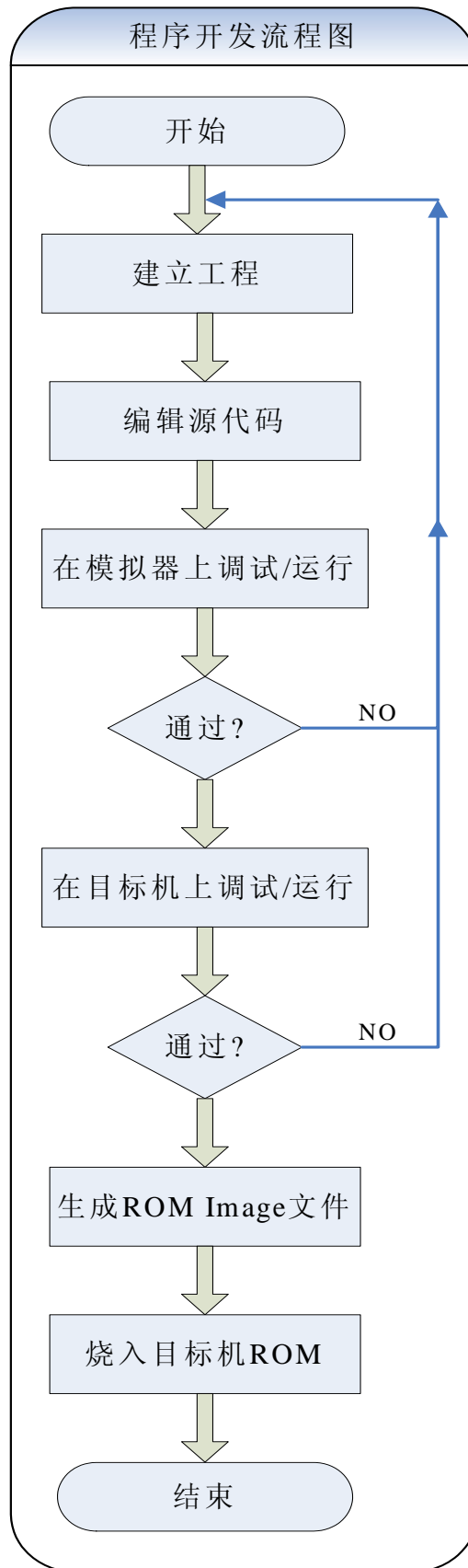


图 4-3 软件开发流程图

4.3 Orion 4.0 使用说明

4.3.1 主窗口

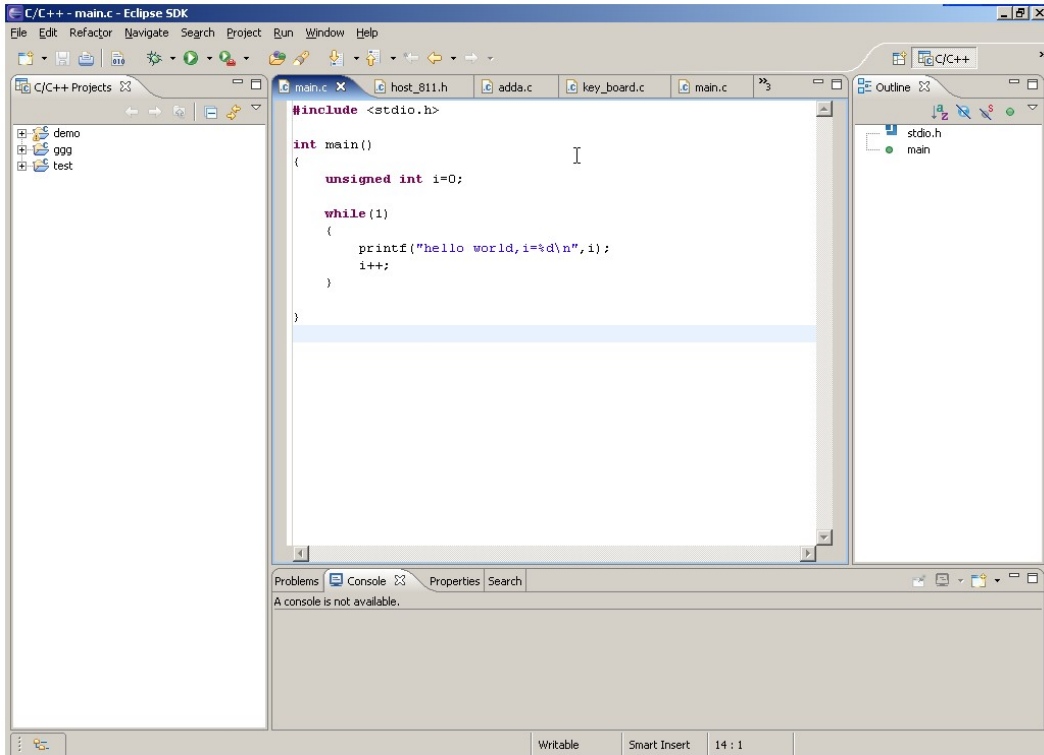


图 4-4 Orion 4.0 主界面

这里先介绍 Orion 4.0 开发环境的主界面，其界面是典型的 windows 风格，与 Visual c++ 等集成开发工具类似；主界面左边显示工程的相关信息和层次关系，右边是代码编辑窗口和编译、调试的返回信息的窗口，一般用户根据其向导很容易创建自己的工程。下面就工程的创建、程序的编译、调试等几个重要部分给出用 Orion 4.0 开发 SPARC 程序的一些简要说明，不详之处请参看 Orion 4.0 的帮助文档。

4.3.2 选择工作间



图 4-5 创作工作间

工作间是工程的集合，一个工作间里可以有多个工程；ORION 4.0 登陆时首先要求选择工作间，输入工作间路径后点 OK 进入主界面；此时可以新建工程，打开、关闭已有的工程，但不可以打开其它工作间里的工程；可以点菜单 File->Switch Workspace 切换工作间。注意：工作间的路径名中不允许包含空格等特殊字符，否则调试下载时将产生错误。

4.3.3 工程的新建

新建工程的步骤如下：

1. 在 ORION 4.0 的主界面上，点击 file/new/project 选择 Manage C Make Project，单击 next 按钮。

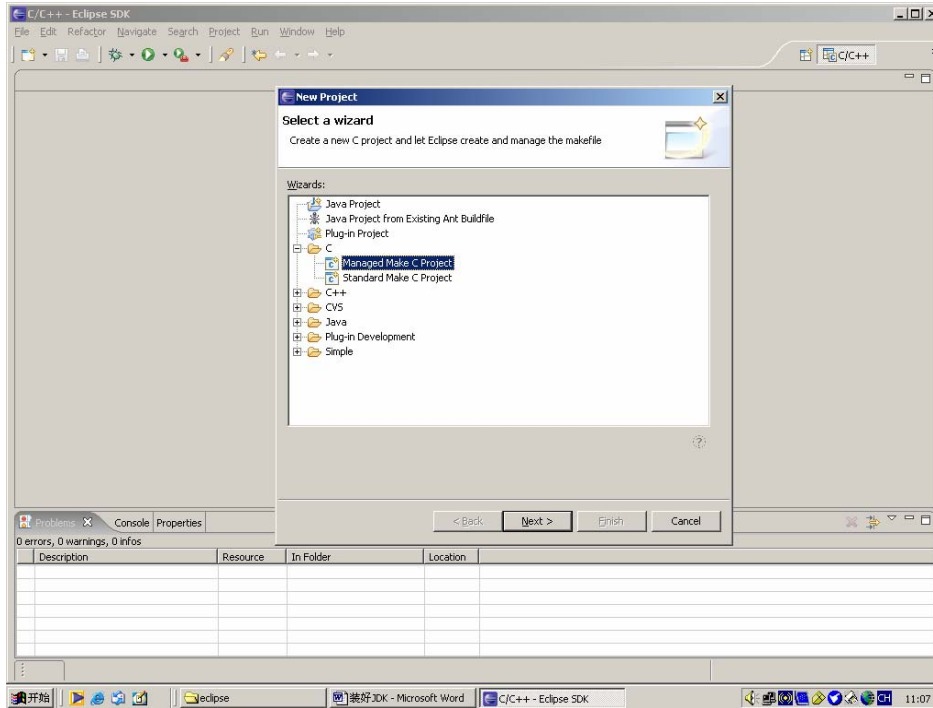


图 4-6 选择 Manage C Make Project

2. 在工程名输入框中输入工程名（例图中为 TestRtems），单击 next 按钮。

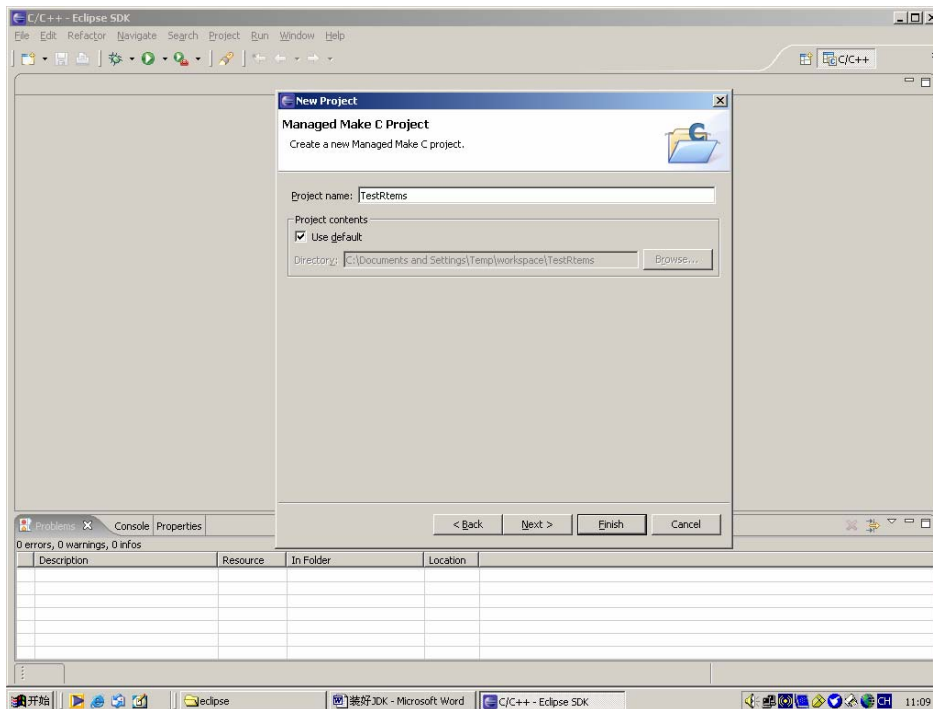


图 4-7 输入工程名

3. 在 Project Type 中选择 Executable(sparc-rtems toolchain, link against rtems-OS)，单

击 next 按钮。

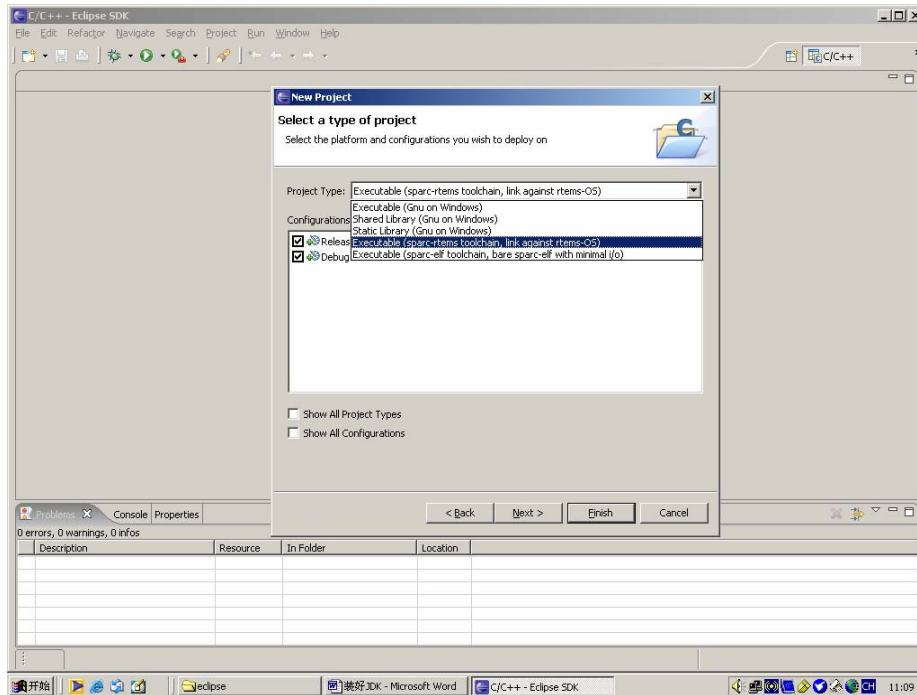


图 4-8 选择 Project Type

4. 点击“Finish”按钮，建立工程完毕。
5. 新建源文件：在工程导航栏中点右键，选择 New/Source File, 输入文件名（例图中为 test.c），单击 Finish 按钮；如果要新建多个源文件，可重复以上步骤依次添加；

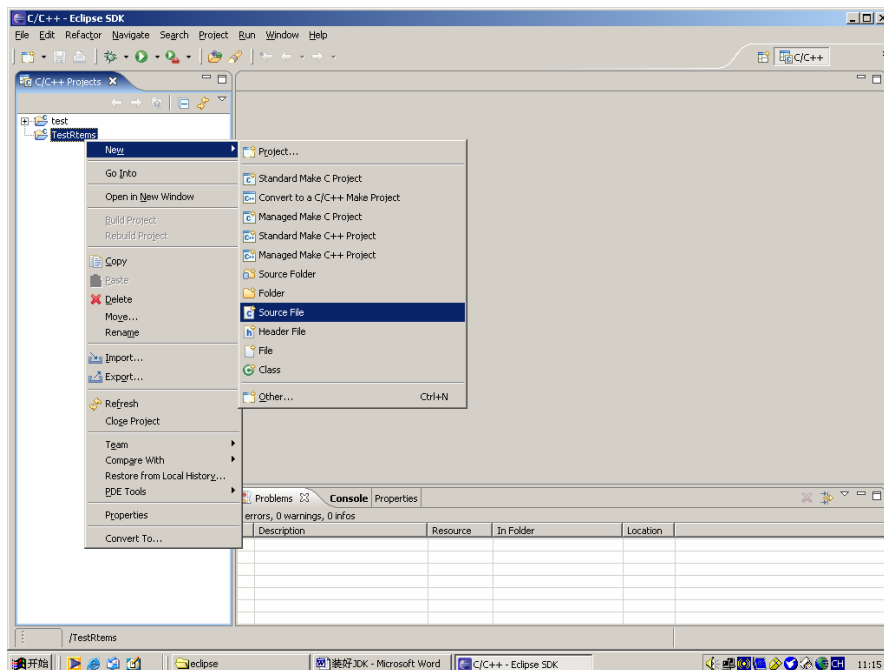


图 4-9 新建源文件 1

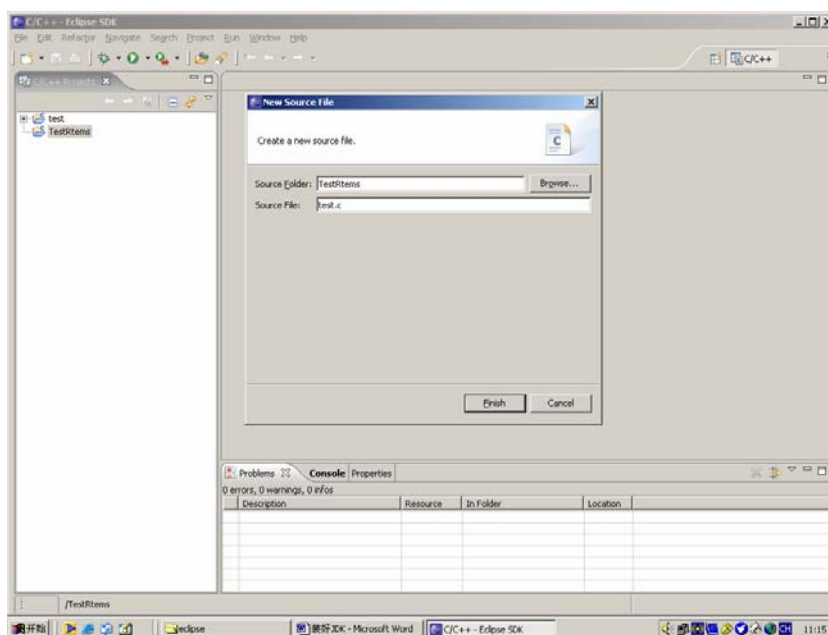


图 4-10 新建源文件 2

6. 向工程中添加源文件：在工程导航栏选中工程，点右键，浮动菜单中选择 Import 项，弹出文件选择对话框，如图 4-11 所示：

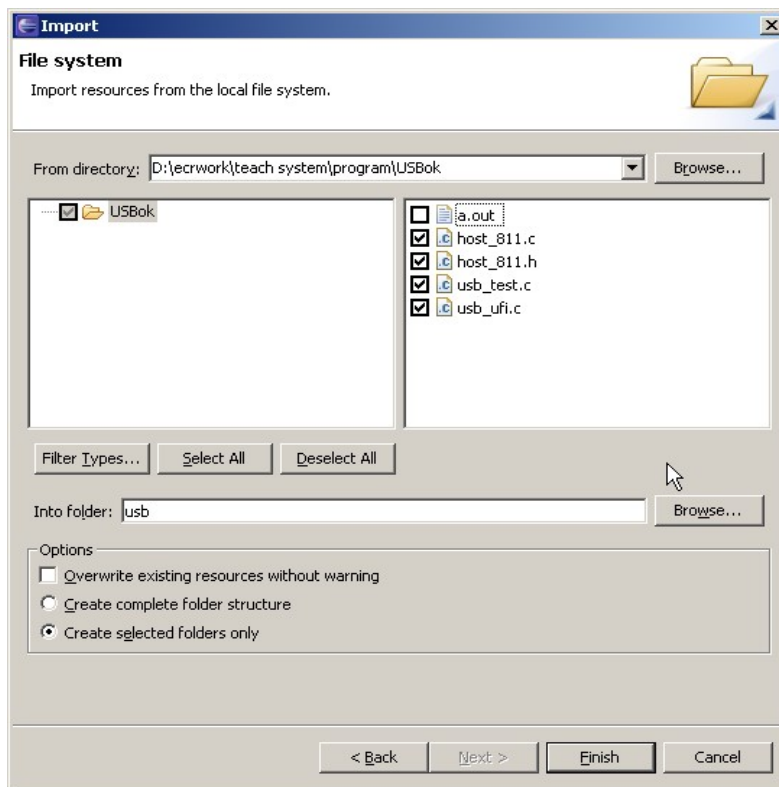


图 4-11 在工程中添加源文件

单击 Browse 按钮选择文件路径，在右边的文件选择栏中勾选要添加到工程中的文件，单击 Finish 按钮。至此，完成了工程的新建。

4.3.4 编译调试环境的配置

1. 编译环境的配置：在工程导航栏选中工程，右击选择 Properties,再选中 C/C++Build, 编译选项设置如图 4-12 所示：

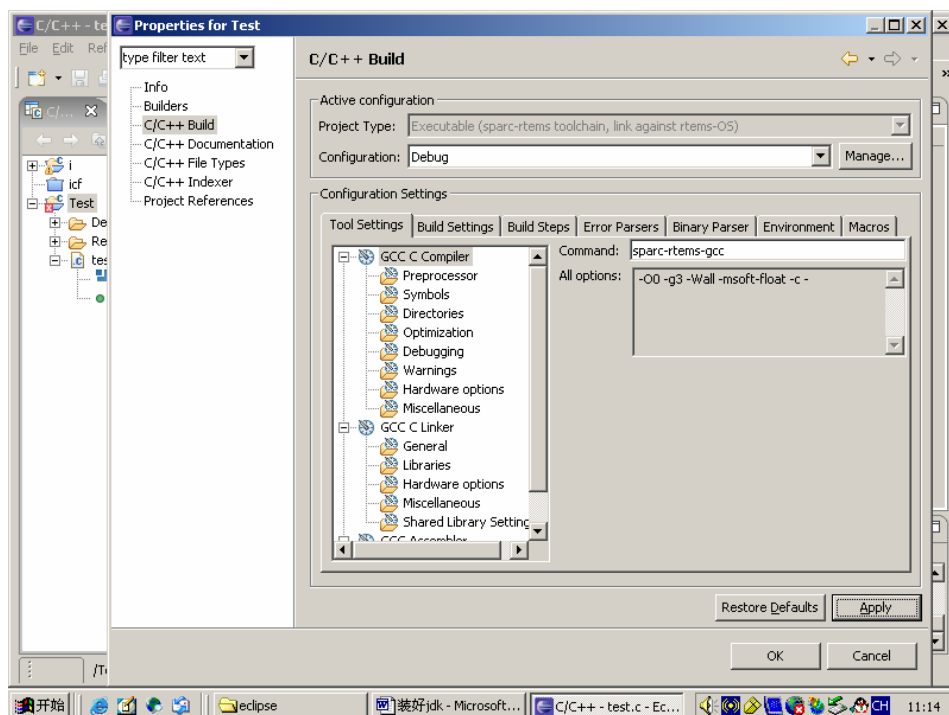


图 4-12 编译选项设置

新建的工程系统都给出了默认值，默认值一般不需要改动；如有更改要单击 Apply 按钮，再单击 OK 按钮完成更改；

2. 软件模拟器 tsim-ORBITA 的内部启动设置：

- a) 点 run/External Tools/External Tools, 如图 4-13 和图 4-14 所示：

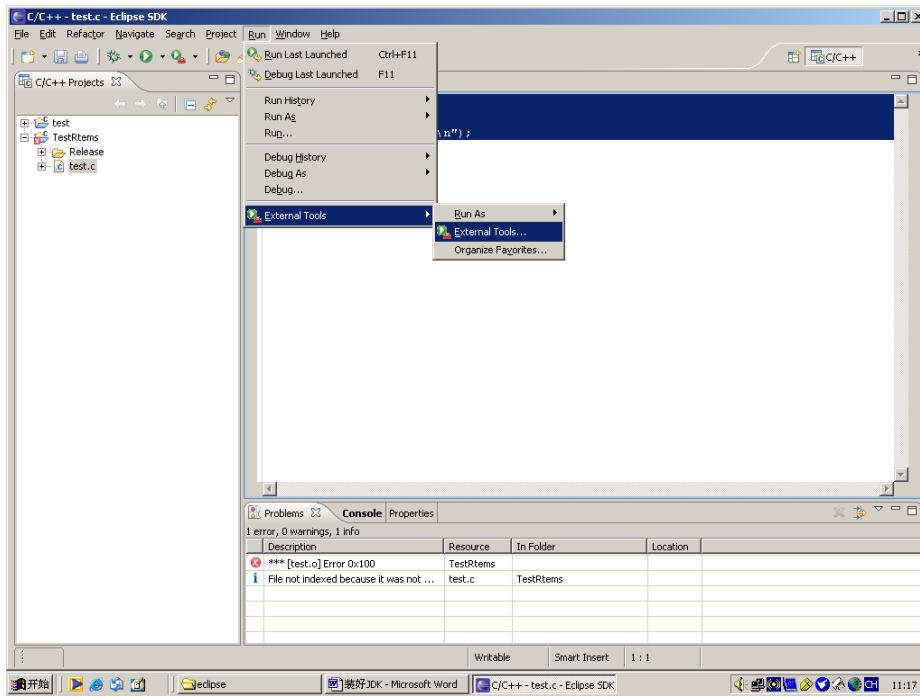


图 4-13 选择 External Tools 1

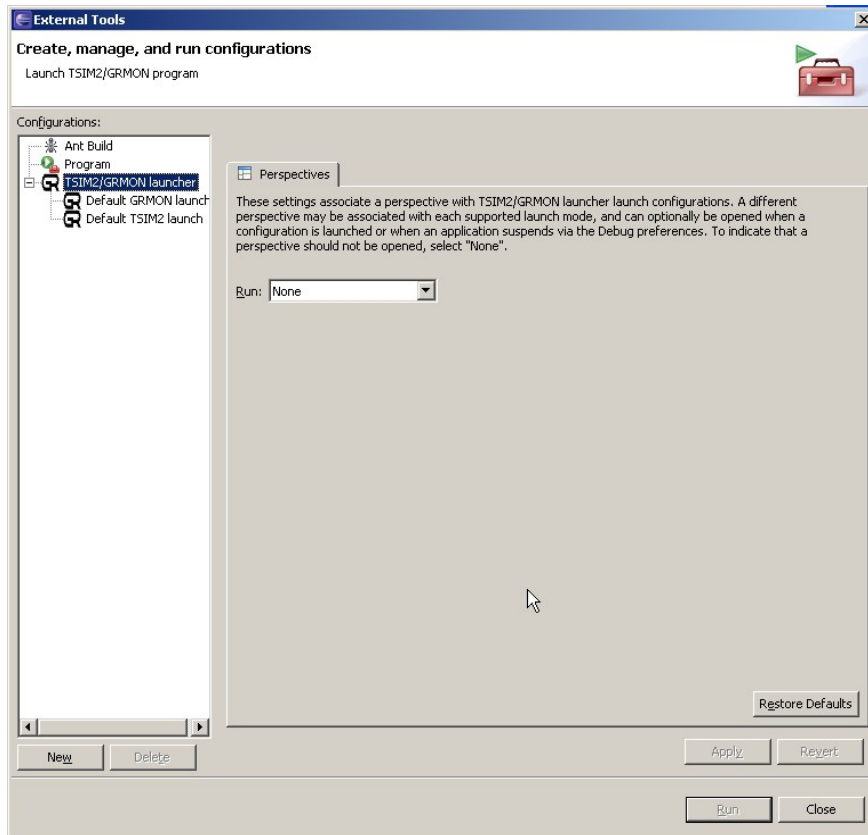


图 4-14 选择 External Tools 2

- b) 在左边的 Configurations 导航栏中, 选择 Default TSIM2 launch, 在右边的 Main 页 Executable 栏中输入可执行文件名: tsim-ORBITA-mm.exe, 在 Arguments 栏中输入 -gdb ; 其它项保持默认值。

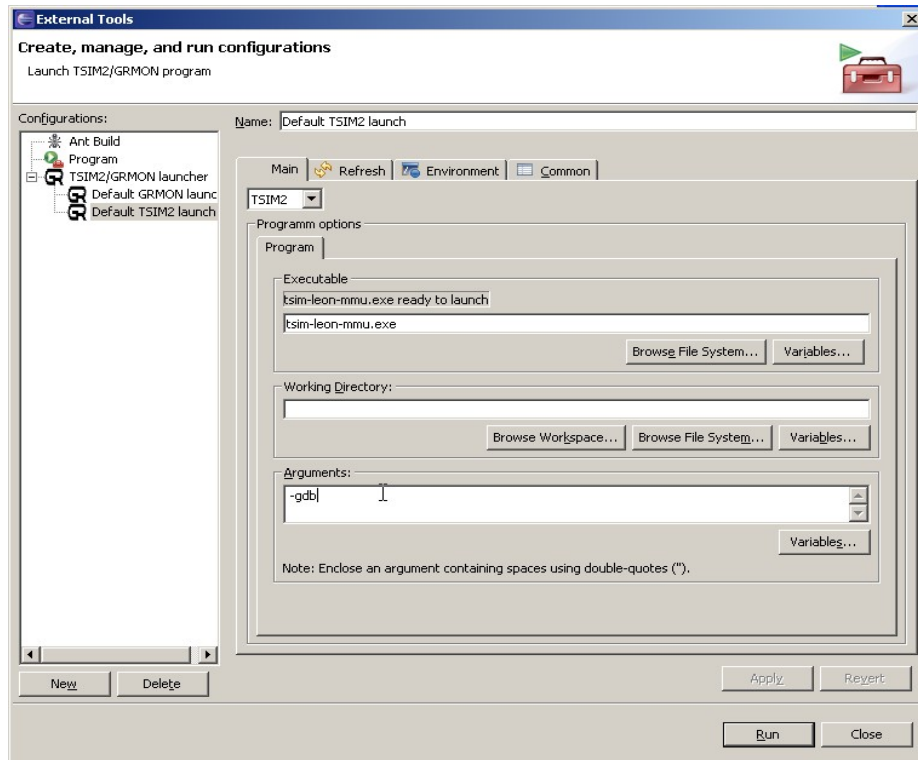


图 4-15 Configurations

3. 硬件调试器 GRMON 的内部启动设置:

- 点 run/External Tools/External Tools;
- 左边的 Configurations 选择框中选中 Default GRMON launch, 在右边的 Main 页 Executable 栏中输入可执行文件名: grmon-eval.exe, 在 Arguments 栏中输入 - ORBITA2 - i - u - gdb ; 其它项保持默认值;
- 点 Close 按钮结束设置。

4.3.5 调试运行环境的设置

1、新建调试\运行环境:

程序调试\运行之前, 在工程生成的 EXE 文件中点右键, 选择 Debug As, 选择 Debug, 在弹出视图中的 C/C++ Application running on Tsim/Grmon 项中右键选择 New (或双击) 以新建调

试\运行环境，如图 4-16 所示：

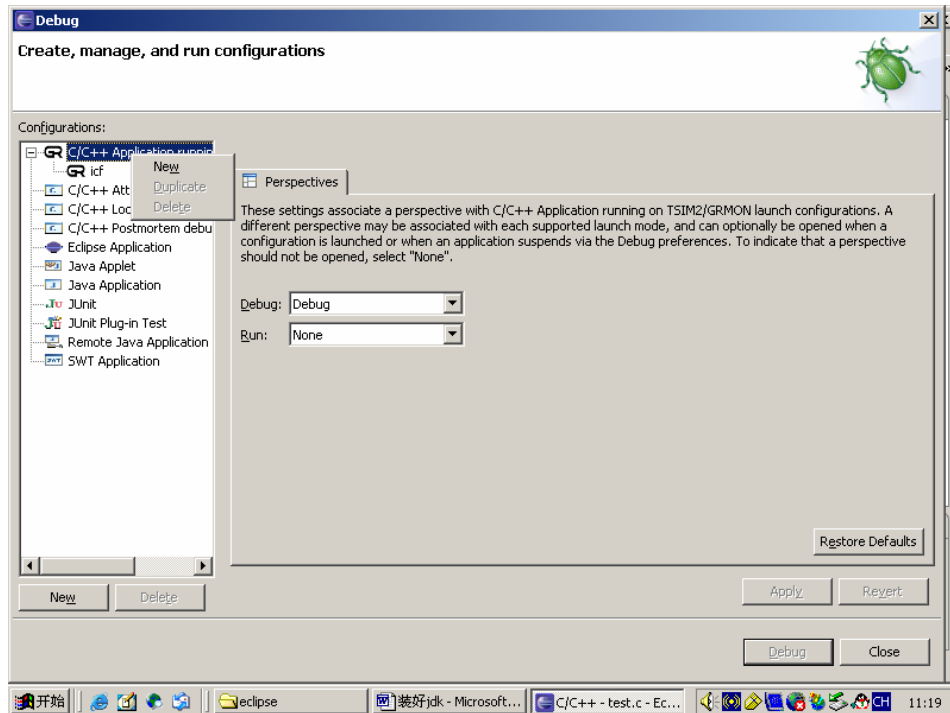


图 4-16 新建调试\运行环境 1

2. 定义运行环境的名称：

Name 输入框中输入新建调试运行环境的名称，Orion 4.0 允许建立多个调试\运行环境，不同的环境之间用用户自定义的名称区分；

3. 配置 Main 设置页：

Main 设置页中指定要调试\运行的工程和可执行文件：Project 输入框中输入工程名，一般用默认值；C/C++ Application 输入框中输入可执行文件名，可点击后面的 Search Project 按钮选择文件；

注意：

同时有 Debug 和 Release 两个可执行文件时请注意选择，Release 目录下的可执行文件不包含调试信息，运行时不能进行设断点、单步跟踪等操作；

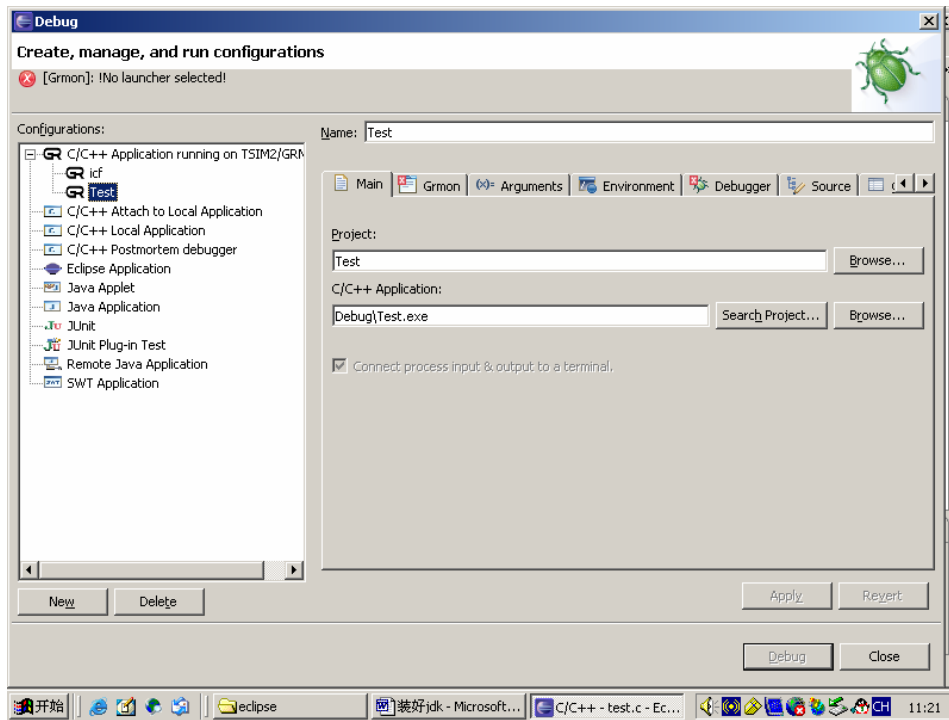


图 4-17 配置 main 设置页

4. 配置 Grmon 设置页:

Grmon 设置页指定了调试\运行时是用软件模拟器 TSIM 还是用硬件调试器 GRMON, 以及调试\运行之前是在 Orion 4.0 内部启动还是在外部启动他们; 一般情况下, Start Mode 应选择 Start TSIM2/GRMON inside Orion 4.0, 也就是内部启动; 然后根据具体情况选择启动 TSIM2 还是 GRMON, 选择框中的选择项为上文 run/External Tools/External Tools 中对应的设置名, 也可以单击选择框后面的 Edit 和 new 按钮来编辑, 如图 4-18 所示。

如果选择外部启动 TSIM2/GRMON, 则此处要求输入 HOST 和 PORT 值, HOST 中输入值一般为 localhost; 如果用 TSIM 调试 PORT 中输入值为 1234, 用 GRMON 调试 PORT 中输入值为 2222; 外部启动通常用在调试前需要对系统板进行配置的情况。如图 4-19 所示。

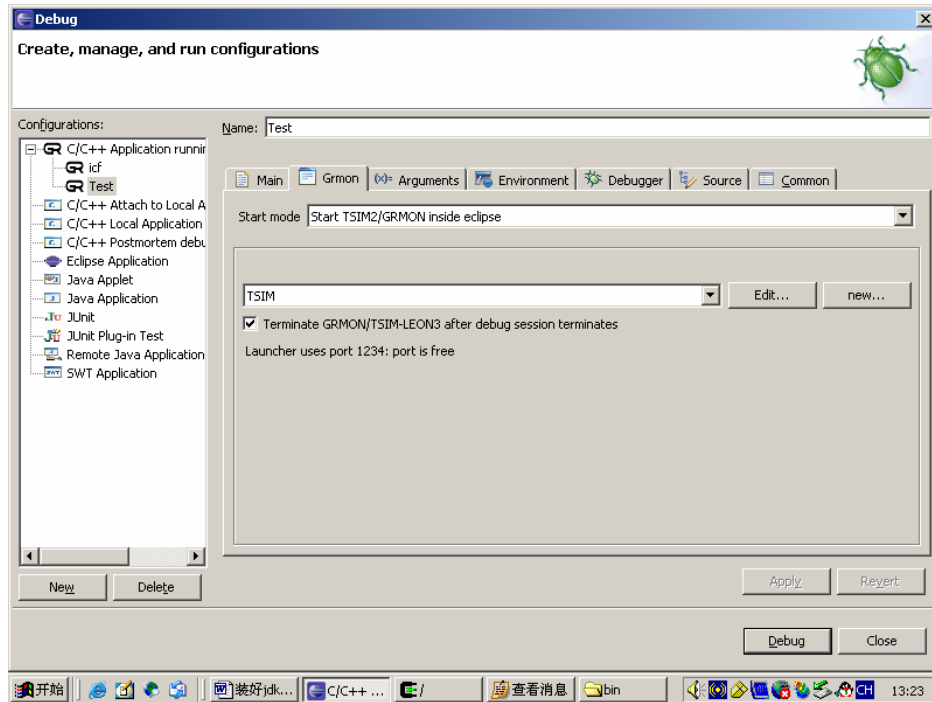


图 4-18 配置 Grmon 设置页

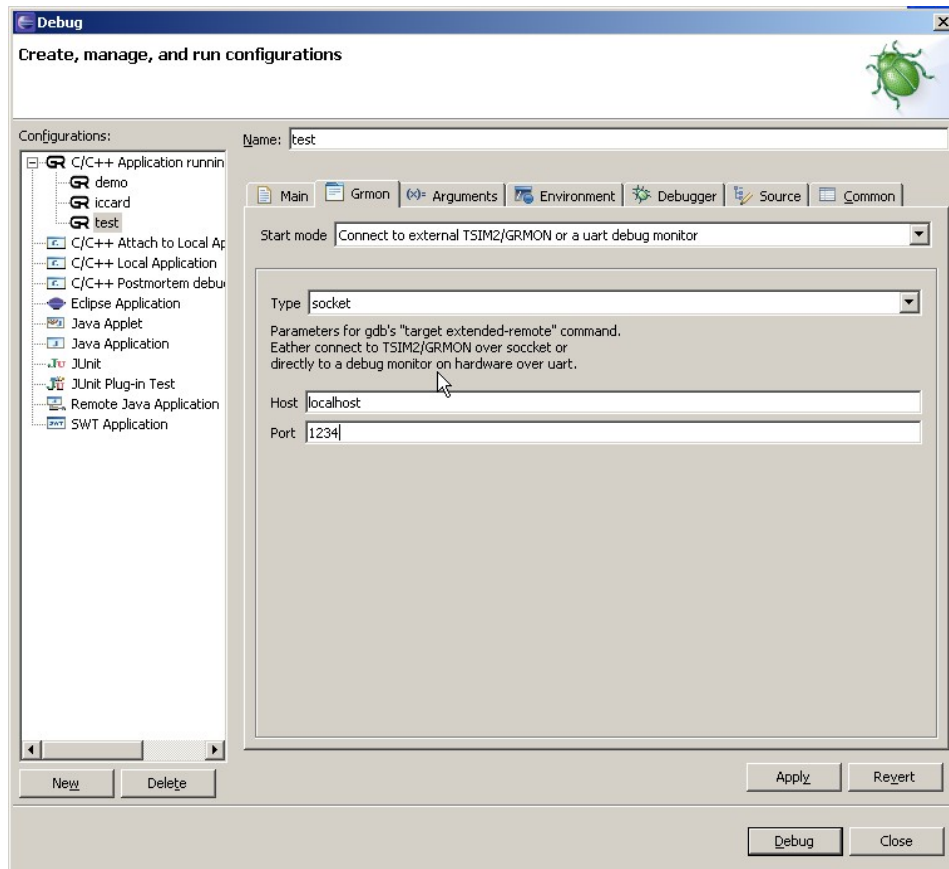


图 4-19 输入 HOST 和 PORT 值

注意：如果选择外部启动调试器，则在调试之前必须通过 Cygwin 先启动 TSIM 或者 GRMON, 命令参数如下：

```
$ tsim-ORBITA-mmu.exe -gdb
```

```
$ grmon-eval.exe -ORBITA2 -i -u -gdb
```

5. 配置 Debugger 设置页：

Debugger 设置页中指定了 gdb 调试器的一些参数，具体设置如下：

- Debugger 选择框中选 sparc-rtems GDB Debugger；
- Main 子页中 GDB Debugger 输入框中输入 sparc-rtems-gdb；
- GRMON/tsim-ORBITA3 hostname or IP address 中输入 localhost ，既本地调试；
- GRMON/tsim-ORBITA3 port name 中输入 gdb 通讯时对应的 Socket 端口号，软件模拟器 TSIM 的端口号为 1234，硬件调试器 GRMON 的端口号为 2222；
- 其它文中未提到的设置项请用默认值。

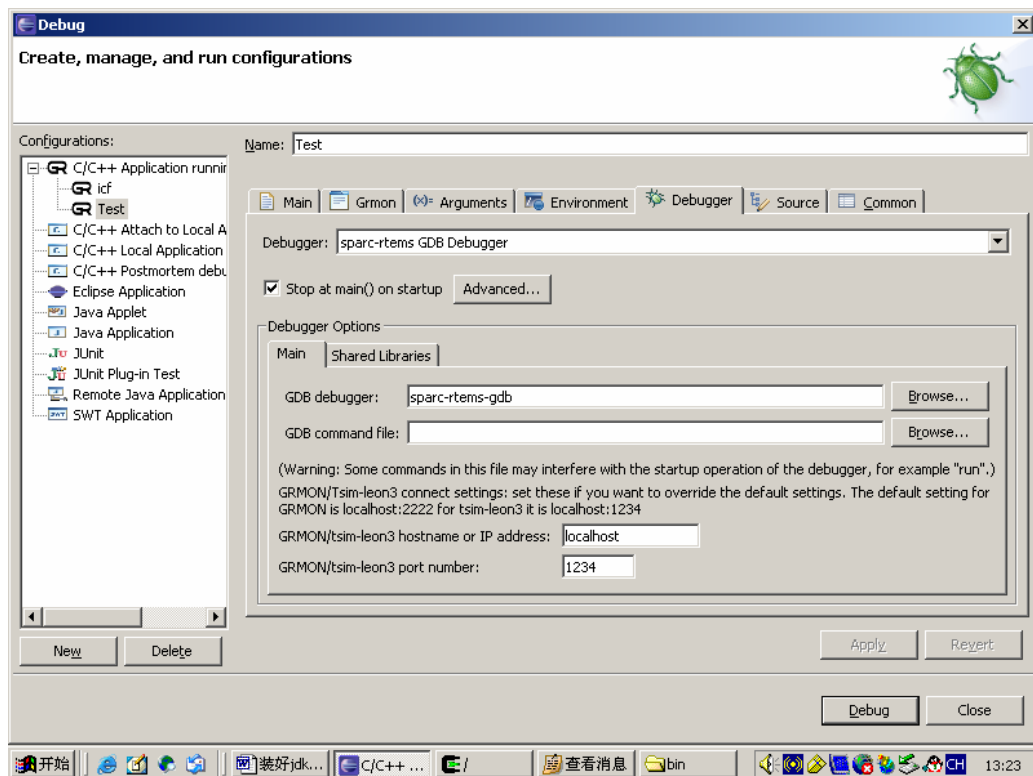



图 4-20 配置 Debugger 设置页

4.3.6 工程的编译、链接

编写完程序后就可以开始编译了，编译成功的前提是编译环境的正确配置。

默认情况下的 Orion 4.0 是“自动编译”，就是一保存原文件就会自动编译，不喜欢的可以取消 project 菜单下的 Build Automatically 项，待需要编译时点击工具栏中的 Build All 按钮来编译工程。

4.3.7 程序的调试

程序编译成功后就剩下的工作就是调试，Orion 4.0 具有强大的调试功能，调试步骤和主要辅助功能简单介绍如下：

1. 启动调试

调试环境设置好后单击 Debug 按钮，启动调试（如果设置了多个调试环境，单击 Debug 按钮默认启用第一个调试环境），此时主界面切换到 Debug 视图（如图 4-21 所示，如果没有进入到 Debug 视图的话可在菜单 Window/Open Perspective/Other/Debug 打开 Debug 视图，主界面下方的 Console 输出信息框中提示调试连接的状况，如果连接失败请检查设置选项；注意：硬件调试前请正确连接硬件系统。

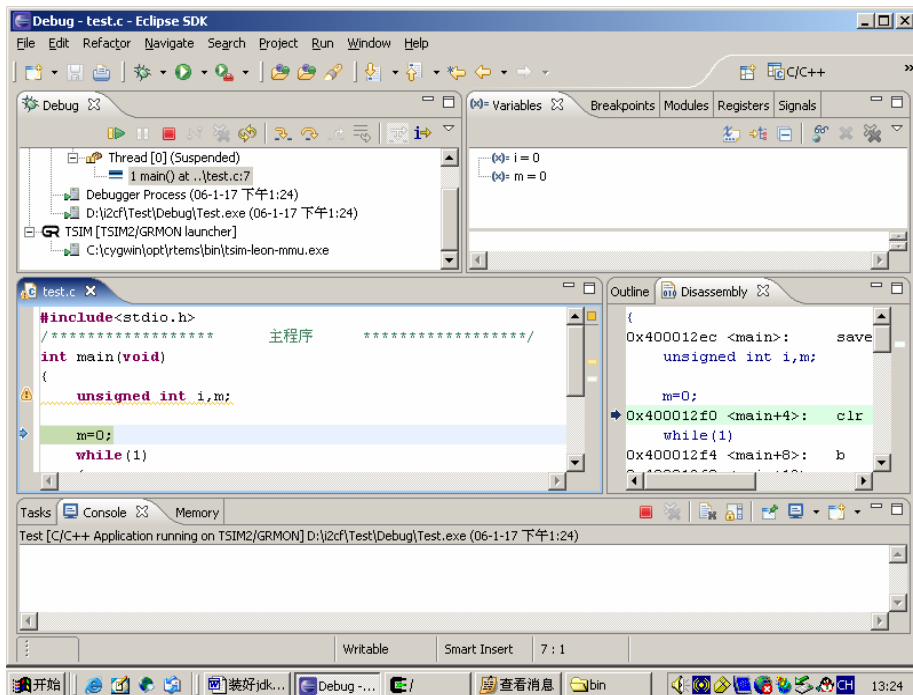


图 4-21 Debug 视图

2. 设置断点

设置断点通常有以下几种方法：

- 1) 将鼠标指向源程序行左边需设定断点处，点右键，在浮动菜单中选择 Toggle Breakpoint；
- 2) 将光标移动到需要设置断点的代码行处，选择 Run/Toggle Line Breakpoint；
- 3) 将光标移动到需要设置断点的代码行处，按快捷键 Ctrl+shift+B；

删除断点的方法跟设置断点的方法类似。当程序运行到断点处时，会停止在有效断点处，源程序左边出现指示箭头；建议调试开始之前先设置好断点，可防止一启动调试后程序就立刻运行完毕。

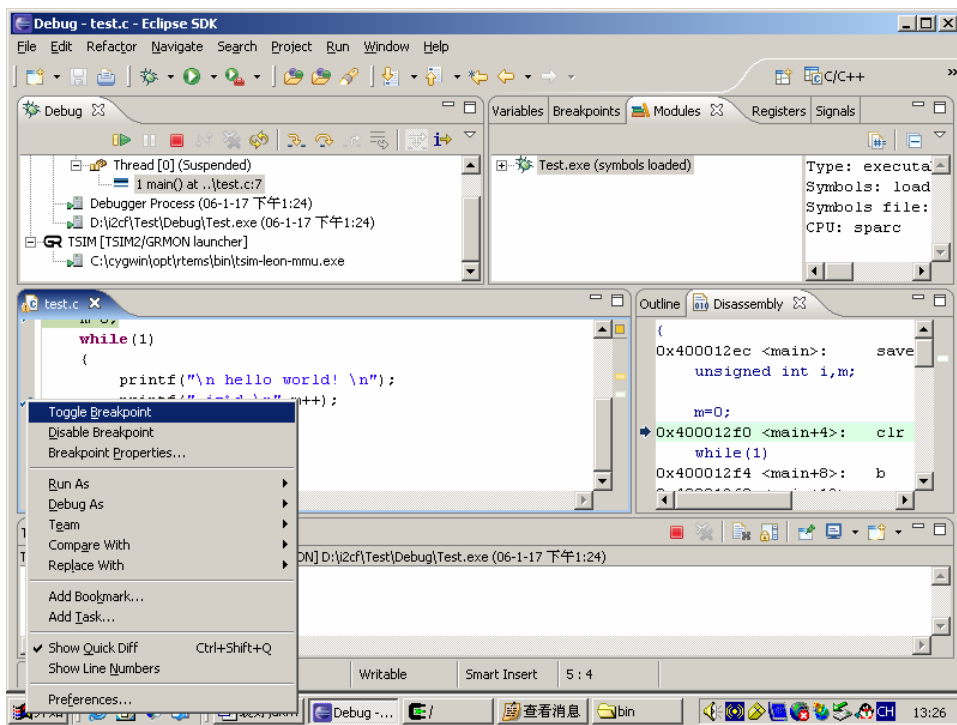




图 4-22 设置断点

3. 单步运行

程序运行到断点处后，可单击 Step Into 按钮  或 Step Over 按钮  进行单步运行，其中 Step Into 是运行到函数处会进入函数内部，Step Over 则会跳过函数运行函数后的程序行；源程序左边的箭头指示当前正待运行的程序行。

4. 观察变量

选择 Windows/Show View/Variables 项, 可打开变量观察窗, 在其中点击右键可进行添加删除观察变量的操作, Format 项用来选择数据显示的方式 (10 进制、16 进制); 如果所观察的变量值改变, 则系统以红色显示该变量。

5. 观察内存

选择 Windows/Show View/Memory 项, 可打开内存观察窗, 单击 Add Memory Monitor 按钮输入要观察的内存起始地址 (建议输入十六进制如: 0x40000000), 点 OK 按钮后, 右边界面将显示当前内存地址的值, 数据以 16 个字节为一行依次显示, 拖动滚动条系统将按照对应地址提取数据显示, 内存显示如图 4-23 所示:

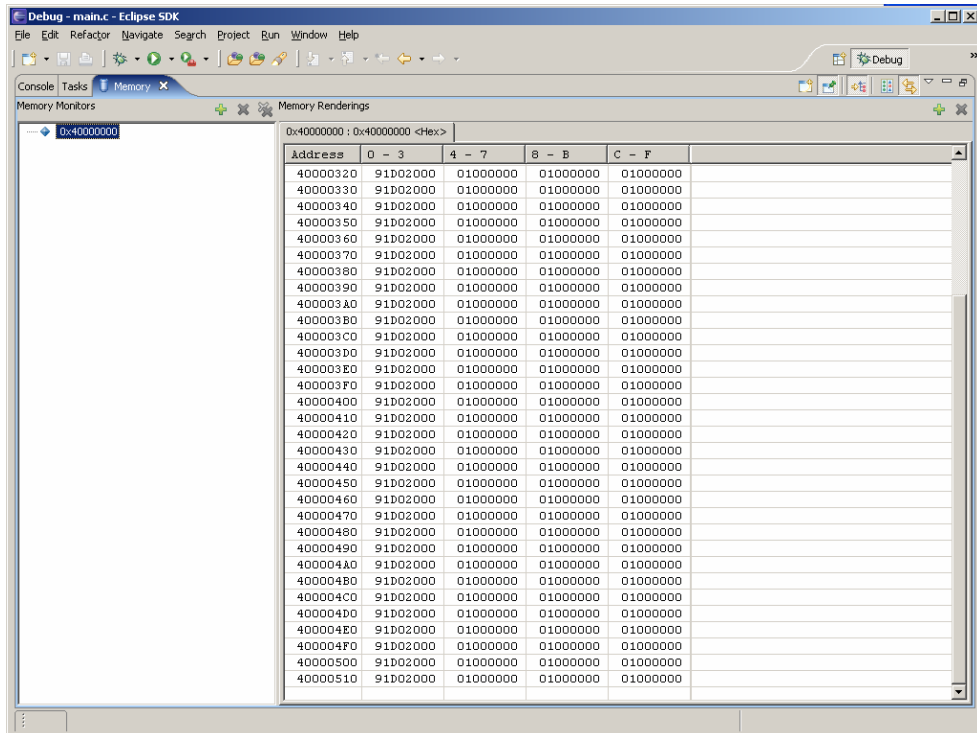


图 4-23 观察内存

6. 观察寄存器

选择 Windows/Show View/Registers 项, 可打开寄存器观察窗, 如图 4-24 所示; 随着程序运行到不同的程序行, 可以观察到各寄存器所产生的变化; 如果寄存器值改变, 则系统以红色显示该寄存器。

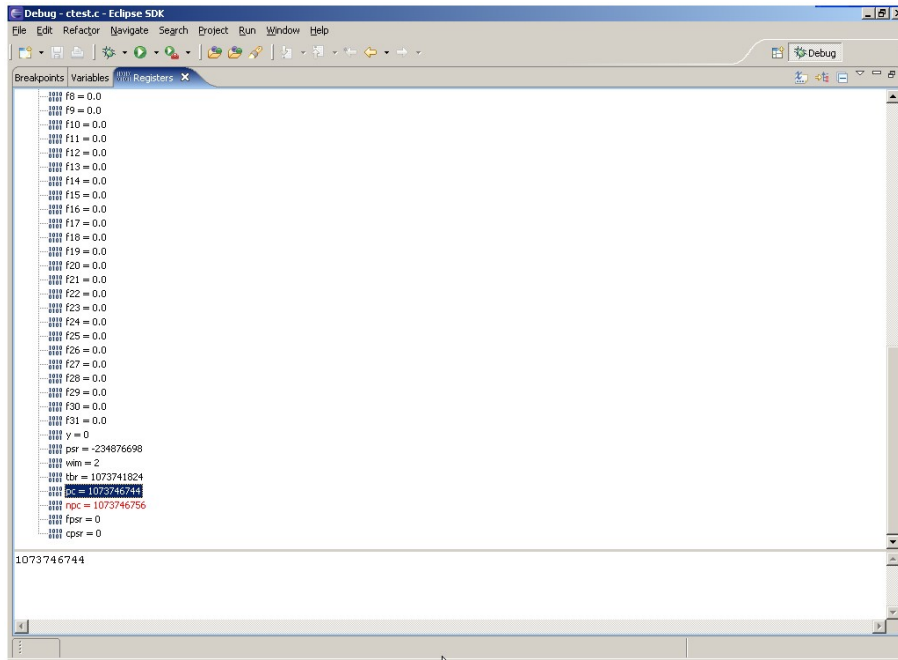




图 4-24 观察寄存器

7. 删除调试

调试完成或者中途需要重新开始调试时要将当前的调试进程关闭；单击 Debug 窗口或者 Console 窗口右上方的 Remove All Terminated Launches 按钮  可完成删除操作，如果调试进程仍在运行需要先单击 terminate 按钮  终止调试后再进行删除。

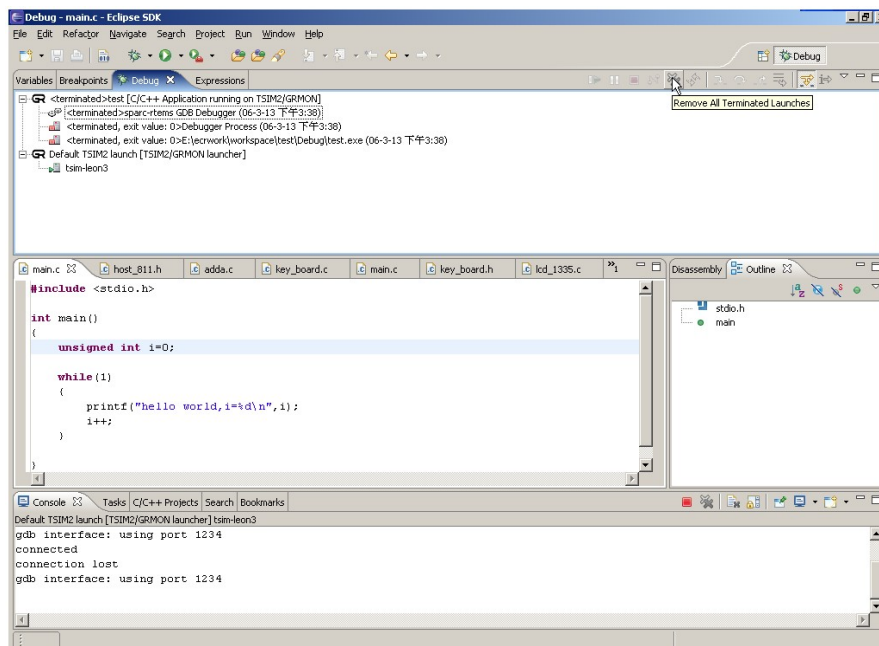


图 4-25 删除调试


4.4 DSU 命令行下的调试

S698-T 芯片内部集成了专门用来调试系统软、硬件的 DSU (Debug Support Unit) 调试接口。DSU 是 SPARC V8 架构所特有的硬件调试接口, 通过 DSU, 用户可以对 CPU 内部所有的寄存器和系统所有的存储设备进行读、写操作, 并且可以设置断点, 从而使系统软、硬件的调试变的极为方便。

DSU 的详细资料请参阅《DSU Monitor 用户手册》使用手册。

4.4.1 DSU 调试步骤

DSU 命令行下的调试步骤如下:

1. 关闭电源;
2. 采用交叉串口线把系统板上 DSU 调试接口连接到主机的 COM1 口;
3. 双击桌面上 DSU 调试软件 cygwin 的快捷方式 , 打开 cygwin 软件;
4. 编写开发程序;
5. 在 cygwin 窗口的命令行下, 通过 `cd [文件路径]` 命令来进入准备要编译的程序所在目录, 例如编写好的程序文件 LED.c 存放在 K:\LED 目录下, 即可通过命令 “`cd k: /LED`” 来进行 LED 文件夹;
6. 通过命令 `sparc-rtms-gcc [用户程序的文件名] -o LED.Out`, 把编写好的目标文件 LED.c 文件编译生成可执行文件 LED.out, 该文件存放在当前目录下;
7. 给系统正常上电;
8. 在 cygwin 的窗口中, 输入 `dsumon -i -u` 命令后, 主机通过串口 1 与系统板的 DSU 调试串口 (dsurx, dsutx) 建立通信连接 (此时, 用户可通过主机对目标板的 CPU 进行调试)。dsumon 命令连接成功后会提示如下信息:

```
processor frequency   : 49.77 MHz
register windows      : 8
v8 hardware mul/div   : yes
floating-point unit   : meiko
instruction cache     : 2 * 16 kbytes, 32 bytes/line (32 kbytes total)
data cache            : 2 * 8 kbytes, 32 bytes/line (16 kbytes total)
hardware breakpoints  : 2
```

```
trace buffer      : 512 lines, mixed cpu/ahb tracing
sram width       : 32 bits
sram banks       : 1
sram bank size   : 4096 kbytes
sdram            : 2 * 64 Mbyte @ 0x60000000
sdram parameters : column bits: 9, cas delay: 2, refresh 15.6 us
stack pointer    : 0x403ffff0
UART 1 in DSU mode
dsu>
```

9. 在 DSU>提示符后, 使用 lo 命令将编译好的可执行文件 (LED.out) 下载到目标板的 SRAM 区。

信息如下:

```
dsu> lo LED.out
section: .text at 0x40000000, size 32224 bytes
section: .data at 0x40007de0, size 1936 bytes
total size: 34160 bytes (90.5 kbit/s)
dsu>
```

10. 在 DSU>提示符后, 使用 run 命令使目标板 SRAM 中的程序开始运行。在运行过程中, 用户可通过 ctrl+c 使程序停止运行。

4.4.2 FLASH 程序固化

在 DSU 命令行下, 可以编译程序, 并可以把程序烧写到系统板上的 FLASH 中。下面以应用程序 LED.c 为例, 说明在 cygwin 的窗口中如何将程序烧录到 FLASH 中。

首先根据系统资源编写好 mkefile 文件, 例如系统 APB 总线时钟为 50MHz, 系统板上 SRAM 容量为 4M Byte, 位宽为 32 位, FLASH 容量为 4M Byte, 位宽为 16 位宽, FLASH 品牌为 AMD, 那么 mkefile 文件编写如下:

```
prom_options= -freq 50 -baud 38400 -ramsize 4096 -ramcs 1 -ramws 15 -ramwidth
32 -romsize 4096 -rmw
all:
##### Executable file #####
    sparc-rtems-gcc LED.c -o LED.exe
    mkprom $(prom_options) LED.exe -o a.prom
```

```
sparc-rtems-objcopy -O binary a.prom a.bin
mkflash -amd -width 16 -erasews 64 a.bin -o flash.out
##### clean #####
clean:
-rm *.out *.prom *.bin *.exe *.s
```

编写好 makefile 文件后，把它存放在目标程序 (LED.c) 所在的目录，并在 cygwin 窗口下通过命令进入到该目录下，输入 “make” 命令，进行文件编译并生成 FLASH 烧写文件 flash.out。具体如下：

```
Administrator@dongsheng /cygdrive/k/flash
$ make
sparc-rtems-gcc LED.c -o LED.exe
mkprom -freq 50 -baud 38400 -ramsize 4096 -ramcs 1 -ramws 15 -ramwidth 32 -romsize 4096 -rmw LED.exe -o a.prom

section: .text at 0x40000000, size 32224 bytes
section: .data at 0x40007de0, size 1936 bytes

sparc-rtems-objcopy -O binary a.prom a.bin
mkflash -amd -width 16 -erasews 64 a.bin -o flash.out

loading: a.bin.
file stream length:21040 byte.

creating S698 flash download: flash.out
```

编译完程序后，通过 “dsumon.exe -i -u” 命令连接到目标板，再通过 “lo flash.out” 命令把编译好可执行文件 flash.out 烧写到 flash 中。具体如下：

```
processor frequency   : 49.77 MHz
register windows      : 8
v8 hardware mul/div   : yes
floating-point unit   : meiko
```

```
instruction cache      : 2 * 16 kbytes, 32 bytes/line (32 kbytes total)
data cache            : 2 * 8 kbytes, 32 bytes/line (16 kbytes total)
hardware breakpoints  : 2
trace buffer         : 512 lines, mixed cpu/ahb tracing
sram width           : 32 bits
sram banks           : 1
sram bank size       : 4096 kbytes
sdram                : 2 * 64 Mbyte @ 0x60000000
sdram parameters     : column bits: 9, cas delay: 2, refresh 15.6 us
stack pointer        : 0x403ffff0

UART 1 in DSU mode
dsu> lo flash.out
section: .text at 0x40000000, size 54432 bytes
section: .data at 0x4000d4a0, size 1904 bytes
total size: 56336 bytes (90.5 kbit/s)
dsu> run
erasing the flash,Please wait.
erase finished.

begin write to flash,wait.
.....
end write to flash.
write total 21040 bytes data to flash.

Program exited normally.
dsu>
```

执行以上步骤后，程序已固化到 FLASH 中，关闭电源，重新上电，则程序将自动运行。光盘
中带有系统默认烧写文件，如果用户不小心修改了 FLASH 信息，可重新烧写。

5 维护和故障处理

5.1 日常维护

为了使系统长期良好运行，在日常维护中应做到以下几点：

- 应尽可能使用在条件良好的环境，周围没有强烈震动、热辐射、强烈光照、灰尘较多以及高温度、高湿度的场合；
- 定期检查系统的运行情况；
- 保持系统的卫生，避免上面堆积灰尘。

5.2 故障处理

下面列出了一些可能出现的故障现象以及处理方法，如果系统出现了其它故障，请速与厂家联系。

表 5-1 故障处理方法

故障现象	处理方法
开机无显示，电源指示灯灭	检查输入 220V AC 是否在设备允许的范围内，如果输入不正常，应立即关闭输入电源，以免损坏设备。如果输入正常，设备有故障，请和厂家联系。
DSU 连接错误，调试软件提示“IU ERROR”	检查用于连接 DSU 接口和调试计算机串口之间的交叉串口线是否完好，接口是否连接好。尝试进行系统复位后，重新进行 DSU 连接。如问题仍然存在，设备有故障，请和厂家联系。
DSU 连接时提示“Cannot read registers, processor not in debug mode”，错误，调试软件提示“IU ERROR”	检查跳线选择连接器 DSUEN1 是否已经用跳线帽短接，保证 DSUEN1 已经短接后，重新进行调试。如问题仍然存在，设备有故障，请和厂家联系。
DSU 无法连接，或连接成功后无法运行程序，提示“IU ERROR”	检查系统处理器工作外频是否小于 90MHZ，通过倍频、分频模块设置后，重新进行调试。如问题仍然存在，设备有故障，请和厂家联系。

附录 A：订货信息

产品型号	说明
S698T-Dkit	S698-T 芯片开发系统

附录 B：装箱单

珠海欧比特控制工程股份有限公司

地址：广东省珠海市唐家东岸白沙路 1 号欧比特科技园

邮编：519080

电话：+86 (756) 3391979

传真：+86 (756) 3391980

网站：www.myorbita.net

装 箱 单

产品名称：S698-T 芯片开发系统

产品型号：S698T-Dkit

序号	物品名称/型号	单位	数量	备注
1	S698T-Dkit	台	1	
2	电源适配器(220VAC/5VDC)	个	1	
3	RS232 交叉串口线	条	2	
4	CAN 转接线	条	1	
5	ARINC429 转接线	条	1	
6	10M 1553B 转接线	条	1	
7	配套资料光盘	张	1	
8	保修卡	张	1	
9	合格证	张	1	