前 言

2812开发板基于TI公司的处理器TMS320F2812,采用双面电路板设计,TMS320F2812 采用TI公司的2000系列的处理器核,内部带有ROM、FLASH、大容量的RAM和丰富的外 部接口,具有体积小、速度快、使用简单、功能强大、功耗低、性能稳定、编程简单的特 点,非常适合在工业控制场合使用。

硬件上外扩了很多通用接口如(串口、网路接口、RS485、和电机输出接口),为了 方便用户用最快的时间掌握DSP2000系列的使用技巧,在硬件上使用了多种总线接口和多 种外部通讯方式的并配备了,LED、走马灯、工业用LCD显示屏,这样就使用户以最快的 速度,试验、学习、设计,并把这些处理技巧熟练的嵌入到产品中,使用户迅速的成为硬 件的设计高手,我们在硬件设计中以工业现场的硬件要求来提供方案,除电源部分在工业 现场需要加入保护措施外,对于板级信号,我们使用了信号完整性分析技术、电磁兼容技 术,在关键位置使用了电感、电容等电磁兼容器件,这样就能更容易把我们的经验使用到 每一个用户的产品中,如果在用户使用条件的允许下,能降低频率、减小电路设计的复杂 性、用多层板外加PCB仿真布线、使用已经通过电磁兼容测试和抗干扰测试的电源,在工 业场合的应用会带来更好的效果。

在软件上,提供了一个通用的软件的模块,并在此基础上编写了各种接口的软件试验 程序,并且给出了开机可以使用的烧路程序,并指导用户一步一步的把编制的软件脱机运 行。

目 录

1	产品	出技术参数	1
	1.1	基本参数	1
	1.2	产品软件特性	2
	1.3	产品清单	2
2	TM	S320F2812 的基本系统	3
	2.1	时钟电路	3
	2.2	TMS320F2812 存储空间的配置	3
	2.3	2812 外部存储器接口	4
	2.4	外部扩展的控制/状态寄存器	4
	2.5	SPI 同步串口	4
	2.6	SCI 异步串口	5
	2.7	增强型的 CAN 总线控制器	5
	2.8	2812 的中断	5
	2.9	外扩端口寄存器的定义	5
	2.10	F2812 工作方式的配置	5
	2.11	JTAG	5
	2.12	ETHERNET 接口	6
	2.13	LCD 接口	6
3	外剖	B接插件管脚定义:	6
	3.1	地址、数据总线扩展接口 J5(焊盘形状为正方形的管脚为第一脚	6
	3.2	A/D 输入扩展接口 J1(焊盘形状为正方形的管脚为第一脚)	6
	3.3	电机扩展接口 J2	7
	3.4	电机扩展接口 J4	7
4	2812	2 软件调试	7
	4.1	2812 开发学习板的测试程序包括下面几个方面:	7
	4.2	DSP 编程的基本步骤:	7
	4.3	程序测试	.12
	4.3.	1 TIMER0 测试	. 12
	4.3.2	2 TIMER2 测试	. 13
	4.3.3	3 ExRam 测试	. 14
	4.3.4	4 Key 测试	. 15
	4.3.5	5 LCD 测试	. 16
	4.3.0	6 SCI 测试	. 17
	4.3.7	7 SPI 测试	. 18
	4.3.8	8 39VF400 测试	. 19
	4.3.9	9 RTL8019AS 测试	. 21
5	烧写	FLASH	. 24



- 1 产品技术参数
- 1.1 基本参数
 - ◆ TMS320F2812: 16 位 DSP2000 系列内核,标称工作频率 150M。
 - ♦ 片上存储器:

FLASH 128K X 16位

SRAM 18K X 16位

BOOT ROM 4K X 16 位

OPT ROM 1K X 16位

♦ 片上外设:

PWM 12路

QEP 6通道

ADC 2 X 8通道, 12位, 80ns转换时间, 0--3V输入量程

SCI异步串口 2通道

MCBPS同步串口 1通道

第1页共28页

CAN 1通道

SPI同步串口 1通道

- ◆ 外扩 SRAM, 最大容量 512K X 16 位, 基本配置 64K X 16 位
- ◆ 外扩 FLASH, 最大容量 512 X 16 位, 基本配置 256 X 16 位
- ◆ 外扩以太网 10M 接口, 兼容 NE2000
- ◆ 外扩图形 LCD 接口
- ◆ 外扩微动按键
- ◆ 外扩1路 RS232 接口,1路 RS485 接口
- ◆ 提供上电复位、手动复位、电源检测、看门狗,系统可靠,稳定
- ◆ 标准的 JTAG 接口,方便调试
- ◆ 标准化的扩展总线

1.2 产品软件特性

- ◆ 所有测试程序在 CCS2000 下运行
- ◆ 测试程序有: FLASHI、CAN、CUPTIMER、EXRAM、KEY、MCBSP、RTL8019AS、 SCI、SPI、GPIO、LCD、EXTINT。
- 1.3 产品清单

产品硬件清单

- ◆ 2812 学习型开发板一个
- ◆ DB9 串口线一条
- ◆ 交叉网线一条
- ◆ 5V 稳压电源一个
- ◆ USB ICE 仿真器一个
- ◆ 开发资料光盘一套

开发系统软件清单

- ◆ DSP 烧写插件软件
- ◆ DSP2000 开发系统
- ◆ 测试程序源码

开发系统文档清单

- ◆ 开发系统使用手册
- ◆ 开发系统原理图

◆ 开发系统元器件手册

2 TMS320F2812 的基本系统

2.1 时钟电路

2812 用 20M 外部晶体给 DSP 提供时钟,并使能 F2812 片上 PLL 电路。PLL 倍频系数由 PLL 控制寄存器 PLLCR 的低 4 位控制,可由软件动态的修改。外部复位信号(RS)可将此 4 位清零(CCS中的复位命令将不能对这 4 位清零)。TMS320F2812 的 CPU 最高可工作在 150M 的主频下,也即是对 30M 输入频率进行 5 倍频。通过 IF8 双口 RAM 通信协议完成 SIG-MI1 电路板与处理机的双向信息交互。

2.2 TMS320F2812 存储空间的配置

TMS320F2812为哈佛结构的DSP,在逻辑上有4M X 16位的程序空间和4M X 16位数据 空间,但在物理上已将程序空间和数据空间统一成一个4M X 16位的空间。

TMS320F2812片上有128K X 16位的FLASH, 18K X 16位的SRAM, 4K X 16位的BOOT ROM, 1K X 16位的OTP ROM。

在2812的摸板上还外扩了64K X 16位SRAM(基本配置为64K X 16位,最大配置为 512K X 16位),片上还外扩了256K X 16位的FLASH(基本配置为256K X 16位,最大配 置为512K X 16位)。扩展总线(扩展总线有4个独立的存储空间CE0—CE3,每个CEX空 间为1M X 16位,只支持异步读写访问)和若干个控制状态接口。

地址范围	数据空间	程序空间	备注
0X002000 0X005FFF	16K X 16 位 外 扩控制/状态寄存 器	16K X 16 位 外扩 控制/状态寄存器	占 Zone 0 和 Zone 1 具体 定义见后面所示
0X080000	512K X 16 位外	512K X 16 位外扩	占 Zone2(基本配置 256K)
0X0FFFFF	扩 EL A SH	EL ASH	具体完义见后面所示
0X100000	512K X 16 位 外	512K X 16 位 外扩	占 Zone6,(基本配置 64K)
0X17FFFF	扩 SRAM	SRAM	具体定义见后面所示
0X3FC000	16K X 16 位 外	16K X 16 位 外扩	占 Zone7,当 MP/MC=1,
0X3FFFFF	扩 SRAM	SRAM	占 SRAM 的高 16 K

这些存储器的访问速度如下所示。

地址范围	存储体	等待周期数	备注
0X002000	外扩控制/状态	可编程最少1个等	XTIMING和XTIMING1寄存
0X005FFF	寄存器	待周期	器编程

0X080000	512K X 16 位	可编程最少1个等	由 XTIMING2 寄存器编程 具体等待周期应根据芯片决定
0X0FFFFF	外扩 FLASH	待周期	
0X100000	512K X 16 位	10ns 可编程最少1	由 XTIMING6 寄存器编程
0X17FFFF	外扩 SRAM	个等待周期	
0X3FC000	16K X 16 位	10ns 可编程最少1	当 MP/MC=1 由 XTIMING7
0X3FFFFF	外扩 SRAM	个等待周期	寄存器编程

2.3 2812 外部存储器接口

TMS320F2812 的外部存储器接口包括: 19 位地址线,16 位数据线,3 个片选控制线 及读写控制线。这3 个片选线映射到5 个存储区域,Zone0,Zone1,Zone2,Zone6 和 Zone7。 其中,Zone0 和 Zone1 共用1 个片选线 XCS0AND1,Zone6 和 Zone7 共用1 个片选线 XCS6AND7。这5 个存储区域可分别设置不同的等待周期。

Zone0	存储区域:	0X002000—0X003FFF,	8K X 16 位
Zone1	存储区域:	0X004000—0X005FFF,	8K X 16 位
Zone2	存储区域:	0X0800000X0FFFFF,	512K X 16 位
Zone6	存储区域:	0X100000—0X17FFFF,	512K X 16 位
Zone7	存储区域:	0X3FC000—0X3FFFFF,	16K X 16 位

2812 的外部存储器接口 XINTF 的详细说明和编程操作请参考《TMS320F28X External Interface (XINTF) Reference Guid》。

2.4 外部扩展的控制/状态寄存器

2812 上配置有 Ethernet 接口, LCD 接口, 输入输出等控制寄存器和状态寄存器等, 它映射在 F2812 的 Zone0 和 Zone1 存储空间中, 具体的定义如下:

功能/名称	寄存器 名称	地址/区域	操作
输出接口控制寄存器	LedRegs	0X0023FF, Zone0	8 位只写
输入接口控制寄存器	KeyReg	0X0033FF, Zone0	8 位只读
LCD 命令寄存器 1	Lcd_com1	0X0039FE, Zone0	8 位只写
LCD 数据寄存器 1	Lcd_DAT1	0X0039FF, Zone0	8 位可读写
LCD 命令寄存器 2	Lcd_Com2	0X00B8FE, Zone0	8 位只写
LCD 数据寄存器 2	Lcd_dat2	0X00B8FF, Zone0	8位可读写
以太网基地址寄存器	BaseAddr	0X0037FE, Zone0	8位可读写

2.5 SPI 同步串口

2812 上有一个 SPI 同步串口,在 2812 板上与 74HC595 相连接。

2.6 SCI 异步串口

2812 上有两个 SCI 异步串口, 经 485 和 232 电平转换芯片连接到 DB1 和 DB2。提供 外部使用。

2.7 增强型的 CAN 总线控制器

2812 上有一个增强型的 CAN 总线控制器,符合 CAN2.0 接口规范,经 CAN 收发器 驱动后连接到 CON1 上,提供外部使用。CON1 连接器定义见原理图。

2.8 2812 的中断

2812 的中断有 3 个中断引脚: XINT1, XINT2 和 XNMI_INT13, 每个中断可配置为 上升沿或下降沿触发,也可以被使能或禁止。 2812 板上 EXINT5 作为外部中断按键 K4 的输入,还有一个终端 Xint2 作为网片的中断输入。

2.9 外扩端口寄存器的定义

输出接口控制寄存器:

D7	D6	D5	D4	D3	D2	D1	D0
R8019RST	R485CS	SPICS	FMQ	LED3	LED2	LED1	NC

输入接口控制寄存器:

D7	D6	D5	D4	D3	D2	D1	D0
NC	NC	NC	K3	K2	K1	NC	IRQ

FMQ: 2812板上蜂鸣器控制信号,高电平有效(即蜂鸣器响)

4850E: 485电平转换芯片发送、接收控制信号

8019RST: RTL8019AS芯片复位信号

SPICS: 74HC595 RCLK控制信号

2.10 F2812 工作方式的配置

2812直接将MP/MC接地,即F2812工作在微计算机MC方式。2812将SCITXDA上拉为 "1",即F2812工作在"Jump to Flash"上电自举方式。

2.11 **JTAG**

F2812的仿真接口为JTAG形式,它能与各种形式的JTAG仿真器连接进行仿真与调试, 2812板上的JTAG接口为J7。

2.12 Ethernet 接口

2812板上有网卡接口芯片RTL8019AS,通讯速率为10M,兼容NE2000。在2812板上, RTL8019AS的基地址为: 0X0037FE

2.13 LCD 接口

2812板上有图形LCD接口,图形LCD接口的命令寄存器和数据寄存器为:

- LCD_COM: 0X39FF LCD命令寄存器(可读可写)
- LCD_DAT: 0X39FE LCD数据寄存器(可读可写)
- LCD_COM: 0xb8FF LCD 命令寄存器 1 (可读可写)
- LCD_DAT: 0Xb8FE LCD 数据寄存器1(可读可写)
- 3 外部接插件管脚定义:

3.1 地址、数据总线扩展接口 J5(焊盘形状为正方形的管脚为第一脚

A0	1	2	A1
A2	3	4	A3
A4	5	6	A5
A6	7	8	A7
A8	9	10	A9
A10	11	12	A11
A12	13	14	A13
A14	15	16	A15
A16	17	18	A17
A18	19	20	A19
D1	21	22	D2
D3	23	24	D4
D5	25	26	D6
D7	27	28	D8
D9	29	30	D10
D11	31	32	D12
D13	33	34	D14
D15	35	36	RD
CS1	37	38	WE
VCC	39	40	GD

3.2 A/D 输入扩展接口 J1(焊盘形状为正方形的管脚为第一脚)

SIGNAL GND	1	2	3.3VA
ADCINA7	3	4	ADCINA6
ADCINA5	5	6	ADCINA4
ADCINA3	7	8	ADCINA2
ADCINA1	9	10	ADCINA0

ADCLO	11	12	ADCLO
ADCINB7	13	14	ADCINB6
ADCINB5	15	16	ADCINB4
ADCINB3	17	18	ADCINB2
ADCINB1	19	20	ADCINB1

3.3 电机扩展接口 J2

PWM1	1	2	PWM2
PWM3	3	4	PWM4
PWM5	5	6	PWM6
T1CMP	7	8	T2CMP
GND	9	10	QEP1
OEP2	11	12	QEP11
T1CTRIP	13	14	VCC
T2CTRIP	15	16	TDIRA
TCLKINA	17	18	C1TRIP
C2TRIP	19	20	C3TRIP

3.4 电机扩展接口 J4

PWM7	1	2	PWM8
PWM9	3	4	PWM10
PWM11	5	6	PWM12
T3CMP	7	8	T4CMP
QEP3	9	10	QEP4
QEP12	11	12	C4TRIP
C5TRIP	13	14	C6TRIP
TDIRB	15	16	TCLKINB
GND	17	18	T3CTRIP
VCC	19	20	T4CTRIP

4 2812 软件调试

4.1 2812 开发学习板的测试程序包括下面几个方面:

- 测试2812的存储系统,包括SRAM和外扩FLASH
- 测试BOOT过程
- 测试LCD功能
- 测试所有F2812的外设
- 测试以太网接收和发送

4.2 DSP 编程的基本步骤:

第一步: 安装CCS,可参考其他书籍,主要提醒一点,如果不使用仿真器,CCS的运行 环境要设置成一个模拟仿真器,如果使用傅立叶公司的仿真器参见仿真器说明书。 第二步: 运行CCS,则进入CCS开发环境,如下图所示:



第三步: 新建一个工程, 如下图



第四步: 选择路径, 输入工程项目名, 并确定, 则如下图所示



第五步: 添加源文件,添加CMD文件,添加库文件,如下图所示:



添加CMD文件和库文件跟添加源文件的方法相似,在这里不在赘述。添加完成后整

个工程如下所示:



第六步: 设置BUILDING OPTIONS

/F2812 Device Simulator/CPU - 28xx (Simulator) - Code Composer Studio File Edit View Project Debug Profiler GEL Option Tools DSP/BIOS Window Help 🗒 🖷 🕺 健 🕫 わらせる| - A A % 强 🗠 № (물 수물 씨을 없을 💌 🖉 📇 📥 🕲 🗶 💌 ff. pjt ▼ Debug F2 60 🖞 🗏 🔜 🔤 🖻 P ? × Build Options for ff.pjt 2) 👰 Files 主 🧰 GEL files P General Compiler Linker Link Order 🖻 🔄 Projects 1 🖻 📩 ff. pjt (Debug) -q -c -m".\Debug\ff.map" -o".\Debug\ff.out" -🚞 Dependent Projects 2 DSP/BIOS Config **O** 🚞 Generated Files 🚞 Include Basic Category: →{} Basic 📄 Libraries 🔽 Suppress Banner (-q) Advanced 🗄 🔄 Source Exhaustively Read Libraries (-x) {+} 計 39vf400. c 計 39VF400TEST. c Output Module: ž • 🗄 DSP28_CpuTimers. c . \Debug\ff. out Output Filename (-o): 浴 DSP28_DefaultIsr.c Map Filename (-m): . \Debug\ff.map 3 DSP28_GlobalVariableDefs.c DSP28_Gpio.c Autoinit Model: Run-time Autoinitializ 🔻 DSP28_InitPeripherals. c Heap Size (-heap): οx DSP28_PieCtrl.c Stack Size (-stack): DSP28_PieVect.c DSP28_SysCtrl.c Fill Value (-f): 函 📩 DSP28_Xintf. c Code Entry Point (-e): 5 Library Search Path Include Libraries (-1): 确定 取消 帮助

2812 开发板使用说明书

主要是Linker选项:

Output Module: absolute executable

Autoinit Mode: run-time Autoinitialize

Stack Size: 200h (可以设置,也可以不设置,可在CMD中包含)

Code Entry Point: main 这是默认设置。

第七步: 编译, 生成 *.out文件, 如图下图所示: 如果程序有错误, 必须修改正确。直 到编译没错误。



2812 开发板使用说明书

第八步: LOAD程序,如下图所示(注意: *.out文件在工程目录中的DEBUG文件夹中)

Edit View Project Debug Profiler GEL Option Tools DSP/BIOS Mindow Help New Image: Constraint of the second sec	
New Image: Constraint of the state of th	
Save Ctrl+S Save As Saye All	4 6
Load Program	
Reload Program	
Load Program ? ▼ 查找范围(1): Debug ▼ 全 企 管 頭▼ ③9F400. out ●	
文件名 (M): 文件类型 (T): *. out ▼ 取消	

第九步: 单步运行程序,把光标摆在 main()函数中第一条语句,按"执行到光标处"图标,如下图所示:



第十步: 单步跟踪或全速运行,观看结果,具体操作根据具体情况,在此不在赘述。

4.3 程序测试

注意:所有演示程序必须拷到硬盘,并且去掉其"只读属性"。

4.3.1 TIMER0 测试

F2812上有3个32位的CPU定时器,本程序主要对CPU定时器0进行操作,100MS产生1次中断,在中断中让2812开发板上的L1—L3发光二极管闪烁,实现跑马灯的效果。

将CpuTime0目录COPY到CCS开发环境中的myproject目录下;

在CCS中用project/Open.....命令,加载CpuTime0目录下的CpuTime0.pjt;

在CCS中用File/Load Program...... 命令,加载CpuTime0目录中debug 目录下的

CpuTime0.out.

在CCS中用Debug

Go Main执行程序到main()函数处。

可单步,也可全速。直接在板卡上观察结果:发光二极管从右到左依次点亮,熄灭。

注意:如果发现程序不能正确运行或程序跑飞。可点击CCS中Debug /Reset CPU,然后重新Load程序。如果还是不能正确运行或程序跑飞,可给板卡重新上电。

🦸 /F2812 XDS510 Emulator/CPU_1 - 28xx - Code Composer Studio - [CpuTimer0.c] 😻 File Edit View Project Debug Profiler GEL Option Tools DSP/BIOS Window Help 🖫 📑 💾 🦹 🖆 📴 ကြင်းပလဲ က 😚 🔏 🔏 🐐 🐄 / 🗁 🎀 💽 🕸 🛗 👗 🕘 🛞 🦻 CpuTimeO.pjt Debug 🗅 🖩 📑 🗑 🔤 🙇 🗎 Ŷ 题 66 ConfigCpuTimer(&CpuTimer0, 10, 100000 ጉ 🍑 Files StartCpuTimerO(); 🗄 🚞 GEL files 큔 🗄 🔂 Projects /*开中断*/ P 🗄 📩 CpuTimeO. pjt IER |= M INT1; 🚞 Dependent Pr Ъ PieCtrl.PIEIER1.bit.INTx7=1; 🚞 DSP/BIOS Con 큔 🚞 Generated Fi 🗄 🔄 Include EINT: // Enable Global interrupt IN →{} 🖹 DSP28_Add ERTM: // Enable Global realtime int DSP28_Cpt 🖹 DSP28_Def //*LedReg = 0x00; ノ/ 美 LED Ż DSP28_Dev //*Led8Lock = 0x00; // 灭数码管 📄 DSP28_Dev 浴 SendData(0xFF); 📄 DSP28_EC∈ for(;;); ② 🗐 DSP28_Ev. DSP28_G1

2812 开发板使用说明书

4.3.2 TIMER2 测试

F2812上有3个32位的CPU定时器,本程序主要对CPU定时器2进行操作,100MS产生1次中断,在中断中让2812开发板上的L1—L3发光二极管闪烁,实现跑马灯的效果。



将CpuTime0目录COPY到CCS开发环境中的myproject目录下:

在CCS中用project/Open.....命令,加载CpuTime0目录下的CpuTime2.pjt;

在CCS中用File/Load Program...... 命令,加载CpuTime2目录中debug 目录下的CpuTime2.out。

在CCS中用Debug

Go Main执行程序到main()函数处。

可单步,也可全速。直接在板卡上观察结果:发光二极管从右到左依次点亮,熄灭。

注意:如果发现程序不能正确运行或程序跑飞。可点击CCS中Debug /Reset CPU,然后重新Load程序。如果还是不能正确运行或程序跑飞,可给板卡重新上电。

4.3.3 ExRam 测试

在2812开发套件上外扩了一片256K * 16位SRAM,其地址映射在ZONE6或ZONE7上, SRAM的测试过程主要是对外部SRAM的 0X100000---0X104000进行写操作,即向 0X100000—0X104000中写入0—0X3FFF,然后读出到0X104000---0X107FFF空间,用户可 以观察这两个地址空间。



将ExRam目录COPY到CCS开发环境中的myproject目录下;

在CCS中用project/Open.....命令,加载ExRam目录下的ExRam.pjt;

在CCS中用File/Load Program...... 命令,加载ExRam目录中debug 目录下的

ExRam.out。在CCS中用Debug/Go Main执行程序到main()函数处。 按下图设置断点,按F5运行。

点View Memory......输入地址,如下图所示:

lenory	Vindow Options	X
<u>T</u> itl <u>A</u> ddress Q-Value	Memory 0x100000	
<u>F</u> ormat	Hex - C Style Use IEEE Float	•
Page: Enabl Start	Data e <u>R</u> eference Buffe	_
End Updat	0x00000000 e Reference <u>B</u> uffer Aut	omat
<u>0</u> :	K <u>C</u> ancel <u>H</u> el	р

-														
ſ	Ox00100000:	0x0000	0x0001	0x0002	0x0003	0x0004	0x0005	0x0006	0x0007	0x0008	0x0009	0x000A	0x000B	~
I	Ox0010000C:	0x000C	OxOOD	OxOOOE	OxOOOF	0x0010	0x0011	0x0012	0x0013	0x0014	0x0015	0x0016	0x0017	
I	Ox00100018:	0x0018	Ox0019	0x001A	OxOO1B	0x001C	OxOO1D	0x001E	Ox001F	0x0020	0x0021	0x0022	0x0023	
I	Ox00100024:	0x0024	0x0025	0x0026	0x0027	0x0028	0x0029	0x002A	0x002B	0x002C	0x002D	0x002E	0x002F	
I	Ox00100030:	0x0030	0x0031	0x0032	0x0033	0x0034	0x0035	0x0036	0x0037	0x0038	0x0039	0x003A	0x003B	
I	0x0010003C:	0x003C	OxOO3D	0x003E	OxOO3F	0x0040	0x0041	0x0042	0x0043	0x0044	0x0045	0x0046	0x0047	
I	0x00100048:	0x0048	0x0049	0x004A	0x004B	0x004C	OxOO4D	0x004E	OxOO4F	0x0050	0x0051	0x0052	0x0053	
l	0x00100054:	0x0054	0x0055	0x0056	0x0057	0x0058	0x0059	0x005A	0x005B	0x005C	0x005D	0x005E	0x005F	
l	Ox00100060:	0x0060	0x0061	0x0062	0x0063	0x0064	0x0065	0x0066	0x0067	0x0068	0x0069	0x006A	0x006B	
l	0x0010006C:	0x006C	0x006D	0x006E	0x006F	0x0070	0x0071	0x0072	0x0073	0x0074	0x0075	0x0076	0x0077	
l	0x00100078:	0x0078	0x0079	0x007A	0x007B	0x007C	0x007D	0x007E	0x007F	0x0080	0x0081	0x0082	0x0083	
l	0x00100084:	0x0084	0x0085	0x0086	0x0087	0x0088	0x0089	0x008A	0x008B	0x008C	0x008D	0x008E	0x008F	
I	Ox00100090:	0x0090	0x0091	0x0092	0x0093	0x0094	0x0095	0x0096	0x0097	0x0098	0x0099	0x009A	0x009B	
l	0x0010009C:	0x009C	0x009D	0x009E	0x009F	OxOOAO	Ox00A1	0x00A2	0x00A3	0x00A4	0x00A5	0x00A6	OxOOA7	
l	0x001000A8:	0x00A8	OxOOA9	OxOOAA	OxOOAB	OxOOAC	OxOOAD	OxOOAE	OxOOAF	OxOOBO	Ox00B1	0x00B2	0x00B3	
l	0x001000B4:	0x00B4	0x00B5	OxOOB6	0x00B7	0x00B8	OxOOB9	OxOOBA	OxOOBB	OxOOBC	OxOOBD	OxOOBE	OxOOBF	
I	Ox001000C0:	OxOOCO	Ox00C1	0x00C2	0x00C3	OxOOC4	0x00C5	0x00C6	0x00C7	0x00C8	0x00C9	OxOOCA	OxOOCB	
I	Ox001000CC:	OxOOCC	OxOOCD	OxOOCE	OxOOCF	OxOODO	Ox00D1	OxOOD2	OxOOD3	OxOOD4	0x00D5	OxOOD6	OxOOD7	_
l	0x001000D8:	0x00D8	OxOOD9	OxOODA	OxOODB	OxOODC	OxOODD	OxOODE	OxOODF	OxOOEO	OxOOE1	0x00E2	0x00E3	
l	Ox001000E4:	0x00E4	0x00E5	OxOOE6	0x00E7	0x00E8	OxOOE9	OxOOEA	OxOOEB	OxOOEC	OxOOED	OxOOEE	OxOOEF	
l	Ox001000F0:	OxOOFO	OxOOF1	OxOOF2	0x00F3	OxOOF4	OxOOF5	OxOOF6	OxOOF7	0x00F8	OxOOF9	OxOOFA	OxOOFB	
I	0x001000FC:	OxOOFC	OxOOFD	OxOOFE	OxOOFF	0x0100	0x0101	0x0102	0x0103	0x0104	0x0105	0x0106	0x0107	
I	0x00100108:	0x0108	Ox0109	Ox010A	0x010B	Ox010C	Ox010D	Ox010E	Ox010F	0x0110	0x0111	0x0112	0x0113	
I	0x00100114:	0x0114	0x0115	0x0116	0x0117	0x0118	Ox0119	Ox011A	Ox011B	0x011C	Ox011D	Ox011E	Ox011F	
I	0x00100120:	0x0120	0x0121	0x0122	0x0123	0x0124	0x0125	0x0126	0x0127	0x0128	0x0129	0x012A	0x012B	
I	0x0010012C:	0x012C	Ox012D	Ox012E	Ox012F	0x0130	Ox0131	0x0132	0x0133	0x0134	0x0135	0x0136	0x0137	
I	0x00100138:	0x0138	Ox0139	Ox013A	0x013B	Ox013C	Ox013D	Ox013E	Ox013F	0x0140	0x0141	0x0142	0x0143	
I	0x00100144:	0x0144	0x0145	Ox0146	0x0147	0x0148	Ox0149	OxO14A	Ox014B	Ox014C	OxO14D	Ox014E	OxO14F	
I	0x00100150:	0x0150	0x0151	0x0152	0x0153	0x0154	0x0155	0x0156	0x0157	0x0158	0x0159	0x015A	0x015B	
I	0x0010015C:	0x015C	Ox015D	Ox015E	0x015F	0x0160	0x0161	0x0162	0x0163	0x0164	0x0165	0x0166	0x0167	
I	Ox00100168:	0x0168	Ox0169	Ox016A	OxO16B	Ox016C	OxO16D	Ox016E	Ox016F	0x0170	0x0171	0x0172	0x0173	
I	Ox00100174:	0x0174	0x0175	0x0176	0x0177	0x0178	Ox0179	0x017A	Ox017B	0x017C	Ox017D	Ox017E	Ox017F	
I	0x00100180:	0x0180	0x0181	0x0182	0x0183	0x0184	0x0185	0x0186	0x0187	0x0188	0x0189	Ox018A	0x018B	~
J	∩⊷∩∩1∩∩19″∙	0.0180	0.0190	0.0181	0.0181	∩⊕∩10∩	∩⊕∩101	0⊕0102	0.0103	∩⊕∩10/	∩⊕∩10⊑	∩⊕∩106	0₩0107	

按OK后显示如下画面,如果数值从0X0000依次递增到0X3FFF,则表示外部RAM没问题。 否则说明外部RAM有问题。

注意:如果发现程序不能正确运行或程序跑飞。可点击CCS中Debug /Reset CPU,然后 重新Load程序。如果还是不能正确运行或程序跑飞,可给板卡重新上电。

4.3.4 Key 测试

2812板上有6个微动按键, 其测试过程如下:

将key目录COPY到CCS开发环境中的myproject目录下。在CCS中用project/Open.....命

令,加载Key目录下的Key.pjt 在CCS中用File/Load Program...... 命令,加载key目录中 debug目录下的 key.out在CCS中用Debug/Go Main执行程序到main()函数处。

第 15 页 共 28 页

按下图设置断点,按F5运行。

🦉 /F2	812 XDS510 Emulator/CPU_	1 - 28xx - Code Composer Studio - [Disassembly]
🔄 <u>F</u> il	e <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>D</u> ebug P <u>r</u>	cofiler <u>GEL Option T</u> ools DSP/BIOS <u>W</u> indow <u>H</u> elp
1 🖆 🖆	·	🖓 💽 👬 🏭 🗁 🕅
Key. pj	t 💽 Debug	- ※ ※ 🎂 🚢 🍝 🕐 🔊
5 6	8° 📋 🗏 🗏 🔤 🗖 🗖	8
で で で で で で で で で で で で で で	Files Files GEL files Projects Dependent Proje DSP/BIOS Config Generated Files Include Include Include DSP28_CpuTin DSP28_Global DSP28_Global DSP28_FieCta DSP28_FieCta DSP28_FieCta DSP28_FieVec DSP28_Vintf. key. c Icd_mo. c	3F88FA 1A34 OR @52,#0x0001 3F88FC FE83 SUBB SP,#3 3F88FD 0006 .word 0x0006 3F88FD 0006 .word 0x0006 3F88FE _c_int00:

单步,可扫描到键值,用户可通过查看Temp得到键值,或单步跟踪到KeyFunction() 函数中查看键值。

注意:如果发现程序不能正确运行或程序跑飞。可点击CCS中Debug Reset CPU,然 后重新Load程序。如果还是不能正确运行或程序跑飞,可给板卡重新上电。

4.3.5 LCD 测试

2812板上有LCD接口,其测试过程如下:将LCD目录COPY到CCS开发环境中

的myproject目录下,在CCS中用project/Open.....命令,加载LCD目录下的LCD.pjt;

在CCS中用File/Load Program......命令,加载LCD目录中debug 目录下的 LCD.out; 在CCS中用Debug/Go Main执行程序到main()函数处。

可全速或单步运行

🦉 /F2812 XDS510 Emul	ator/CPU_1 -	28xx - Code	e Composer	Studio - [Di	sassembly]
🔄 <u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roje	ct <u>D</u> ebug P <u>r</u> ofil	er <u>G</u> EL <u>O</u> ption	. <u>T</u> ools D <u>S</u> P/H	3IOS <u>W</u> indow <u>H</u> el	.p
: 웥 🚅 🔛 X 🖻 💼	က်က် ကို ကြို			• 8 % %	🐕 🎒 😽
LCD.pjt 💌 D	ebug	- 🖉 🖄	i 🛗 👗 🕚	2 2 2	
· 🔊 🗠 📋 📰 🖽	🔤 🖻 🧟 🖇				
Piles Piles	3F Jebug) 3F Jebug) 3F Sconfig 3F Sc	85C6 1A34 85C8 FE83 85C9 0006 85CA 28AD 85CA 28AD 85CC FF69 85CC 561F 85CE 5616 85CF 561A 85D0 7622 85D1 B9C0 85D2 2840 85D4 761A 85D5 2940 85D6 761F 85D8 2902 85D9 761B 85D4 7622 85D5 2940 85D6 761F 85D7 28A9 85D8 28A9 85D9 28A8 85D7 0901 85E0 611D 85E1 76FF	OR @ SUBB S .word C _c_int00: MOV @ SPM C SETC C CLRC A EALLOW MOV @ EDIS CLRC F MOV @ CLRC F MOV 0 CLRC C ASP EALLOW MOV @ ADDB A SB I MOV 2 SB I MOV 2 SB I	252,#0x0001 27,#0x0001 27,#0x0400 27,#0x0400 27,#0x0400 28,M0DE 40M1MAP 441,#0x0068 29,#0x0000 29,#0x0000 20,#1	CC << 2

注意:如果发现程序不能正确运行或程序跑飞。可点击CCS中Debug/Reset CPU,然后 重新Load程序。如果还是不能正确运行或程序跑飞,可给板卡重新上电。

4.3.6 SCI 测试

F2812上有两个SCI接口,SCIA和SCIB,其中SCIA接UART,SCIB接RS485,本程序 对UART进行操作,首先,打开调试工具中的串口调试软件,设置波特率为9600BPS,8 位数据位,无校验,1位停止位,设置完成后应如下图所示,然后用标准串口连接线连接 PC和2812,开始运行程序,先由PC机上的串口调试软件给2812发送数据,2812接收到数 据后把收到的数据发给PC机,在串口调试软件的接收窗口会看见收到的数据,看发送和 接收的数据是否一致,来判断UART是否正常。

UART有两种模式,中断模式和查询模式,程序通过设定变量SCIA_INT来决定采用那一种方式。SCIA_INT=1,中断方式;SCIA_INT=0,查询方式。SCI的测试过程如下:

用标准串口连接线连接PC机的COM1和2812上的DB9串口;

打开调试工具下的"串口调试软件",设置串口参数,如下图所示;

将SCI目录COPY到CCS开发环境中的myproject目录下

在CCS中用project/Open.....命令,加载SCI目录下的SCI.pjt在CCS中用File/Load

Program...... 命令,加载SCI目录中debug 目录下的SCI.out

在CCS中用Debug/Go Main执行程序到main()函数处。

全速运行,则串口调试器中会显示如下内容。

🦻 串口调试器 2002		
串口设置	发送字符: 🔽 自动发送	发送周期 1000 毫秒
选择串口: COM1 ▼ 波特率: 9600 ▼ 数据位: 8 ▼ 校验位: None ▼ 停止位: 1 ▼	12	
关闭串口	▶ 十六进制	洁空显示 手工发送
通讯状态	接收字符:	
 串口状态: 打 开 发送字节: 0 接收字节: 0 	12	
关于 退出	☑ 十六进制 🔲 自动分行	清空显示 保存数据

关于串口的更多操作,请参看测试程序的源代码和其他相关文档。

注意:如果发现程序不能正确运行或程序跑飞。可点击CCS中Debug/Reset CPU,然后 重新Load程序。如果还是不能正确运行或程序跑飞,可给板卡重新上电。

4.3.7 SPI 测试

F2812片上有1个SPI串口,在2812开发板上SPI和74HC595相连来驱动数码管,可通过 SPI串口送出数据直接在数码管上显示,可以让用户直接明了的看到结果。 SPI的测试过 程如下:

将SPI目录COPY到CCS开发环境中的myproject目录下,在CCS中用project/Open.....命令,加载SPI目录下的SPI.pjt

在CCS中用File/Load Program......命令,加载SPI目录中debug 目录下的 SPI.out; 在CCS中用Debug/Go Main执行程序到main()函数处。 可全速或单步运行。

4.3.8 39VF400 测试

2812板卡上有4MB的FLASH,其地址范围映射在F2812的Zone2区,其地址范围从 0X80000开始。 39VF400的测试过程如下:

将39VF400目录COPY到CCS开发环境中的myproject目录下;

在CCS中用project/Open.....命令,加载39VF400目录下的39VF400.pjt;

在CCS中用File/Load Program...... 命令,加载AD目录中debug 目录下的39VF400.out;

在CCS中用Debug/Go Main执行程序到main()函数处。

按下图设置断点,然后按F5运行。

😻 /F2812 XDS510 Emulator/	CPU_1 = 28xx = Code Composer Studio = [Disassembly]
🔄 File Edit View Project Deb	ug Profiler <u>G</u> EL Option Tools DSP/BIOS <u>W</u> indow <u>H</u> elp
🍟 🖻 📕 🖇 🖻 💼 🗠 🌣	
39F400.pjt 🔽 Debug	💌 🖉 📇 👗 🕛 🆓 📎
💭 60° 📋 🚍 🖽 🖂 🗖	4 8
 Files GEL files GEL files GP GEL files Dependent Projects DSP/BIOS Config Generated Files Generated Files Include Include Source 39vf400.c 39vf400.c 39vf400.c 39vf400.c 39vF400TEST. Source DSP28_CpuTin DSP28_Gpio.c DSP28_Gpio.c DSP28_FieVed DSP28_FieVed DSP28_SysCtu DSP28_SysCtu 	3F84BE 1A2C OR @44,#0x0180 3F84C0 182D AND @45,#0xFFBF 3F84C2 0006 .word 0x0006 3F84C3 _c_int00: 3F84C3 28AD MOV @SP,#0x0400 3F84C5 FF69 SPM 0 3F84C5 FF69 SPM 0 3F84C6 561F SETC OBJMODE 3F84C7 5616 CLRC AMODE 3F84C8 561A SETC MOM1MAP 3F84C9 7622 EALLOW 3F84C9 7622 3F84C1 761A EDIS 3F84C2 2940 CLRC PAGE0 3F84D1 2902 CLRC OVM 3F84D1 2902 CLRC OVM 3F84D2 761B ASP 3F84D3 7622 EALLOW 3F84D4

点View

Lenory	Vindov Options 🛛 🔀
<u>T</u> itl	Memory
<u>A</u> ddress	0x00100000
Q-Value	0
<u>F</u> ormat	Hex - C Style 💌
	🔲 <u>V</u> se IEEE Float
Page:	Data 💌
🔲 Enabl	e <u>R</u> eference Buffe
<u>S</u> tart	0x0000000
<u>E</u> nd	0x0000000
🔲 Updat	e Reference <u>B</u> uffer Automat
<u>0</u>)	K <u>C</u> ancel <u>H</u> elp

Memory......输入地址,如下图所示:

_													
ſ	Ox00100000:	0x0000	0x0001	0x0002	0x0003	0x0004	0x0005	0x0006	0x0007	0x0008	0x0009	0x000A	Ox
l	Ox0010000C:	0x000C	OxOODD	OxOOOE	OxOOOF	0x0010	0x0011	0x0012	0x0013	0x0014	0x0015	0x0016	OxI
l	Ox00100018:	0x0018	0x0019	0x001A	OxOO1B	0x001C	OxOO1D	Ox001E	OxOO1F	0x0020	0x0021	0x0022	OxI
l	Ox00100024:	0x0024	0x0025	0x0026	0x0027	0x0028	0x0029	0x002A	0x002B	0x002C	0x002D	0x002E	OxI
l	Ox00100030:	0x0030	0x0031	0x0032	0x0033	0x0034	0x0035	0x0036	0x0037	0x0038	0x0039	0x003A	OxI
l	0x0010003C:	0x003C	OxOO3D	OxOO3E	OxOO3F	0x0040	0x0041	0x0042	0x0043	0x0044	0x0045	0x0046	OxI
l	Ox00100048:	0x0048	0x0049	0x004A	OxOO4B	0x004C	OxOO4D	0x004E	OxOO4F	0x0050	0x0051	0x0052	OxI
l	Ox00100054:	0x0054	0x0055	0x0056	0x0057	0x0058	0x0059	0x005A	0x005B	0x005C	0x005D	0x005E	OxI
l	Ox00100060:	0x0060	0x0061	0x0062	0x0063	0x0064	0x0065	0x0066	0x0067	0x0068	0x0069	0x006A	OxI
l	Ox0010006C:	0x006C	0x006D	0x006E	0x006F	0x0070	0x0071	0x0072	0x0073	0x0074	0x0075	0x0076	OxI
l	Ox00100078:	0x0078	0x0079	0x007A	0x007B	0x007C	0x007D	0x007E	0x007F	0x0080	0x0081	0x0082	OxI
l	Ox00100084:	0x0084	0x0085	0x0086	0x0087	0x0088	0x0089	0x008A	0x008B	0x008C	0x008D	0x008E	OxI
l	Ox00100090:	0x0090	0x0091	0x0092	0x0093	0x0094	0x0095	0x0096	0x0097	0x0098	0x0099	0x009A	OxI
l	0x0010009C:	0x009C	0x009D	0x009E	0x009F	OxOOAO	OxOOA1	OxOOA2	0x00A3	0x00A4	0x00A5	0x00A6	OxI
l	Ox001000A8:	0x00A8	0x00A9	OxOOAA	OxOOAB	OxOOAC	OxOOAD	OxOOAE	OxOOAF	OxOOBO	OxOOB1	0x00B2	OxI
l	Ox001000B4:	0x00B4	0x00B5	0x00B6	OxOOB7	0x00B8	OxOOB9	OxOOBA	OxOOBB	OxOOBC	OxOOBD	OxOOBE	OxI
l	Ox001000CO:	OxOOCO	OxOOC1	0x00C2	Ox00C3	OxOOC4	0x00C5	0x00C6	0x00C7	0x00C8	0x00C9	OxOOCA	OxI
l	Ox001000CC:	OXOOCC	OxOOCD	OxOOCE	OxOOCF	OxOODO	OxOOD1	OxOOD2	0x00D3	OxOOD4	OxOOD5	OxOOD6	OxI
l	Ox001000D8:	0x00D8	OxOOD9	OxOODA	OxOODB	OxOODC	OxOODD	OxOODE	OxOODF	OxOOEO	OxOOE1	0x00E2	OxI
l	Ox001000E4:	0x00E4	0x00E5	OxOOE6	OxOOE7	OxOOE8	OxOOE9	OxOOEA	OxOOEB	OxOOEC	OxOOED	OxOOEE	OxI
l	Ox001000F0:	OxOOFO	OxOOF1	OxOOF2	OxOOF3	OxOOF4	OxOOF5	OxOOF6	OxOOF7	OxOOF8	OxOOF9	OxOOFA	OxI
l	Ox001000FC:	OxOOFC	OxOOFD	OxOOFE	OxOOFF	0x0100	0x0101	0x0102	0x0103	0x0104	0x0105	0x0106	OxI
l	Ox00100108:	0x0108	0x0109	Ox010A	Ox010B	0x010C	Ox010D	Ox010E	Ox010F	0x0110	0x0111	0x0112	OxI
l	Ox00100114:	0x0114	0x0115	Ox0116	Ox0117	Ox0118	Ox0119	Ox011A	Ox011B	Ox011C	Ox011D	Ox011E	OxI
l	Ox00100120:	0x0120	0x0121	0x0122	0x0123	0x0124	0x0125	0x0126	0x0127	0x0128	0x0129	0x012A	OxI
l	Ox0010012C:	Ox012C	Ox012D	Ox012E	Ox012F	0x0130	0x0131	0x0132	0x0133	0x0134	0x0135	0x0136	OxI
l	Ox00100138:	0x0138	0x0139	Ox013A	Ox013B	0x013C	Ox013D	Ox013E	Ox013F	0x0140	0x0141	0x0142	OxI
l	Ox00100144:	0x0144	0x0145	0x0146	0x0147	0x0148	Ox0149	OxO14A	OxO14B	Ox014C	OxO14D	Ox014E	OxI
l	Ox00100150:	0x0150	0x0151	0x0152	0x0153	0x0154	0x0155	0x0156	0x0157	0x0158	0x0159	0x015A	OxI
l	0x0010015C:	Ox015C	0x015D	0x015E	Ox015F	0x0160	Ox0161	0x0162	0x0163	0x0164	0x0165	0x0166	OxI
I	Ox00100168:	0x0168	0x0169	0x016A	OxO16B	0x016C	OxO16D	OxO16E	OxO16F	0x0170	0x0171	0x0172	Ox
I	Ox00100174:	0x0174	0x0175	0x0176	0x0177	0x0178	0x0179	0x017A	0x017B	0x017C	0x017D	0x017E	OxI
I	Ox00100180:	0x0180	0x0181	0x0182	0x0183	0x0184	0x0185	0x0186	0x0187	0x0188	0x0189	Ox018A	OxI
J	n∵nn1nn180.	0.0180	n∵019D	N∵019F	N₩019₽	n⇔∩1Q∩	⊓⇔∩101	n⊕0102	n⇔∩102	⊓⇔∩10/	∩⇔∩10⊑	⊓⊕∩106	0.571

按OK后显示如下画面,如果数值从0X0000依次递增到0X3FFF,则表示外部FLASH读写

成功。否则说明外部FLASH读写有问题。

注意:如果发现程序不能正确运行或程序跑飞。可点击CCS中Debug/Reset CPU,然后 重新Load程序。如果还是不能正确运行或程序跑飞,可给板卡重新上电。

注意:如果发现FLASH没法写入,可能FLASH中有数据,必须檫除后才能写入,另 外写FLASH速度比较慢,频率不能太高。

4.3.9 RTL8019AS 测试

将2812开发附件中的网线一头接PC机网络接口,另一头接2812开发板网络插 孔。 将PC机的IP地址设置为: 192.168.1.40

将PC机的掩码地址设置为: 255.255.255.0

Internet 协议(ICP/IP)属性	i ? 🔀
常规	
如果网络支持此功能,则可以获取日 您需要从网络系统管理员处获得适当	司动指派的 IP 设置。否则, 当的 IP 设置。
○ 自动获得 IP 地址 (2)	
●使用下面的 IP 地址(S):	
IP 地址(L):	192 .168 . 1 . 40
子网掩码(U):	255 . 255 . 255 . 0
默认网关 @):	· · ·
○ 自动获得 DNS 服务器地址(B)	
──④使用下面的 DWS 服务器地址(函):
首选 DNS 服务器(P):	
备用 DNS 服务器(A):	· · ·
	高级(Y)
	确定 取消

安装网络抓包工具,运行网络抓包软件。

将RTL8019AS目录COPY到CCS开发环境中的myproject目录下,

在CCS中用project / Open.....命令,加载RTL8019AS目录下的RTL8019AS.pjt,

在CCS中用File/Load Program...... 命令,加载RTL8019AS目录中debug 目录下的 RTL8019AS.out,

在CCS中用Debug/Go Main执行程序到main()函数处。

按下图设置断点,然后按F5运行。



当程序运行到最后一个断点时,网络抓包软件显示如下内容:

🗹 无标题 - CaptureWet 🛛 v3.	12																×
<u>F</u> ile <u>V</u> iew <u>C</u> apture <u>H</u> elp																	
🗙 🗃 🔚 💡 🛛 Adapter : Real	tek P	ackets	in	2		8	א [₽] פ	epNe 1	t								
	No.	Time	(h:m:s:	ms)		MAC s	ouro	e ad	ldr		Fram	e]	Prot	ocol			
	0	0:9:49	9:265		1	02-EO	-4C-	A0-7	E-7A	L	ARP	L.	ARP-1	Requ	iest		
17 address 192.168.1.40	1	0:9:49	9:265		I	00-04	-61-	7E-B	D-65	;	ARP	J	ARP-)	Repl	y		
-Capture																	
7																	
Gz ^{Z²} Start	<u> </u>																2
Capture State	0000:	FF	FF FF	FF	FF	FF	02	EO	4 C	A0	7E	7 A	08	06	00	01	
▼ Display captu	0010:	08	00 06	5 04	00	01	02	EO	4C	A0	7E	7 A	C0	A 8	01	01	
	0020:	00	00 00	00 (00	00	C0	A 8	01	28	ЗF	00	C2	83	ЗF	00	•
-Hardware Filter	0030:	C2	83 3E	r 00	C2	83	ЗF	00	C2	83	ЗF	00					÷
🔽 Promiscuous																	
🗌 Directed 🛛 🛐																	

点View

Memory......输入地址0X3F9440,则显示如下:

Ox003F9440:	RxNetBu	ıf									
Ox003F9440:	Ox9EC9	Ox69DB	OxFEDO	Ox13BF	Ox7CA8	0xBD90	Ox71FO	0x08A8	OxCEA4	0x1D98	0x1712
Ox003F944C:	0x72E8	0x0408	0xCFF8	OxC1B7	0x746F	0x3065	OxEADB	0x8300	0x6A37	0x38A3	0x60A3
Ox003F9458:	0x460E	0x3483	0x77B2	0xB30B	OxC6A6	Ox11D7	0x28B5	0x1527	Ox7EAD	0x6AA8	OxE9A4
Ox003F9464:	0xAD3C	OxB3E6	0xE366	Ox6661	0x965D	OxD947	0x87F7	0x3836	0x3BA5	OxA1CD	OxF31F
Ox003F9470:	Ox8F4D	0x75A1	0xAEC3	OxO46F	OxOD93	Ox4D7B	Ox6EED	Ox1CA3	OxBF37	0x604C	OxF7F6
Ox003F947C:	OxA3F4	0x862A	0xFD6B	Ox2691	Ox4DD9	Ox6883	0xF565	OxCO6C	0xA8A9	0x491B	0x52C1
Ox003F9488:	0x8EDD	0x660D	0xFE76	0x4D75	0x3614	0x2808	0x2327	0xAC84	0xB67B	0xA2B4	Ox5E6D
Ox003F9494:	0x5805	0x5A27	0x17A7	Ox91FB	0x9E65	Ox167B	Ox6EEA	OxD43D	0xDA41	0x8F55	Ox9D1B
0x003F94A0:	0xE05F	OxFCOC	0x834F	0x39A2	OxC732	0xF899	Ox3F77	Ox1AB3	0x9D3F	0x46F7	0x9E2F
OxOO3F94AC:	0x8FAE	0x0310	0xCOO6	0x1024	Ox71FC	Ox6417	0xD971	Ox6813	0x861C	0xBD90	0x5A4D
Ox003F94B8:	0x25F6	0x1087	OxE4D7	Ox3688	Ox7AE4	0x17C0	0x952D	Ox4CE1	0x8DCF	Ox 1ACE	0xF30B
Ox003F94C4:	OxEFDA	0x7818	0xE263	0x1840	OxEB6F	OxD8D0	Ox3E2C	0x5038	OxE79A	OxOC19	Ox2C7A
OxOO3F94DO:	0x7674	0x050B	0xB8E6	0x0269	0x9737	0xA99D	0xC2AD	Ox4B3B	OxAEOF	0x41C3	0x12D3
OxOO3F94DC:	0x742E	0x2DAO	OxF4DE	OxBEBE	0xD5F9	0x8D96	OxDOBD	OxC69C	OxAFEF	0x430C	OxFCBE
Ox003F94E8:	Ox6EFC	0x1035	0x1F33	OxOBAA	Ox06CE	Ox6FF1	0x5A6F	0x7833	OxD7A7	0xA6A4	Ox9E7D
Ox003F94F4:	0x6E97	0x4623	OxC734	0x236F	Ox2EF7	Ox1DOD	0xE69B	0x6C20	OxCO7C	Ox4EOA	OxFF71
Ox003F9500:	0x3A56	0x8347	OxD6CD	0xB98A	0x2F65	Ox9B3E	OxC3BE	0x8B30	0x5AA5	OxOB16	0x8D66
Ox003F950C:	OxD76D	OxB4DC	OxC610	0x8C52	OxDE17	0x815E	Ox 1ADC	0x5C61	OxEDBB	0x5D62	Ox43FD
Ox003F9518:	0x865B	OxB1F3	OxCEF7	0xB989	OxDBOA	0x854E	0x7A39	0x4CA7	OxD3B6	0x0329	0x96A4
Ox003F9524:	0x1F43	0x8C5C	0x0A44	0xFED8	0x8F79	0xE611	0x849B	Ox72CB	0x31B4	0x905D	0x58BD
Ox003F9530:	Ox8BEC	Ox43DA	0xD526	OxCA9D	0xD46F	0x4C77	0x7186	0x0315	OxEDFA	0x0A2E	Ox3C3B
0x003F953C:	0xD80E	0x63E3	0x7689	Ox1906	0xFB2C	Ox4FC3	0x5735	0x9008	0x867E	0xB86E	0x9B1D
Ox003F9548:	0x1E28	0x092F	0x1292	0x30B1	0x889F	OxDBE4	0xDE9F	0x46F4	OxCCFE	0x5106	Ox7EFO
OxOO3F9554:	OxCED5	0xFC33	OxDO81	0x98E3	0x8EEF	0x2A25	0x9230	0x9014	OxE41D	Ox91CE	OxE3DB
Ox003F9560:	OxBE11	0x21A5	Ox9FCC	0x4024	0x9F73	OxOB97	OxEF15	0x0123	0x78A6	0x2CEF	0xCOC9
Ox003F956C:	OxBD1A	Ox1216	0x669D	0xB2CO	0xD4B5	0x9820	OxDE64	0x1223	Ox6D19	OxOB3D	0xE131
Ox003F9578:	OxEEAD	0xA7E7	0x7483	Ox1458	OxDE2F	Ox 1AA9	0xF7DD	0x9D33	Ox69AE	0xFA27	0xD50D
OxOO3F9584:	0x2097	OxD27D	0x65A1	OxBCE7	Ox4BF1	OxFD1D	0xE28F	0x09F6	0xF85C	0x2BD7	0x9503
Ox003F9590:	0x425A	0xD55C	0x5F14	0xDC3A	0x206C	Ox4D8C	0xC937	0x01C6	0x570D	0x605A	0x48D3
Ox003F959C:	0x2D93	0x5F72	0x63D7	0x6F28	0x36B5	0x390E	Ox9F0C	0x0850	0xDD98	0x36D1	0xFB39

地址0X3F9440处开始为8019接收到PC发回的ARP应答。同时也可以打开网络抓包软件SpyNet,对比8019接收到数据正确与否。本程序主要是通过8019向PC机发送ARP请求, 当PC机收到2812发出的ARP请求时,会发一个应答包给2812,用户可查看接收缓冲区的 内容查看应答包的内容。

♂无标题 - CaptureNet v3.	12											X
<u>F</u> ile <u>V</u> iew <u>C</u> apture <u>H</u> elp												
🗙 🗃 🔚 🦹 🗛 Adapter : Real	.tek P	ackets in	2	9	PeepNe	t						
	No.	Time (h:m	(sims)	MAC	source a	ldr	Fram	e Pro	ocol			
	0	0:9:49:26	5	02-E	D-4C-AO-1	E-7A	ARP	ARP-	·>Requ	iest		
If address 192.168.1.40	1	0:9:49:26	5	00-0	4-61-7E-B	3D-65	ARP	ARP-	·>Repl	.y		
Capture												
Can ZZZ Start	<											>
Capture	0000:	02 E0	4C A0	7E 7A	00 04	61 7	E BD	65 08	06	00	01	·
✓ Display captu	0010:	08 00	06 04	00 02	00 04	61 7	E BD	65 <mark>C</mark>	A 8	01	28	•
	0020:	02 E0	4C A0	7E 7A	CO A8	01 0	1					·
-Hardware Filter												
🔽 Promiscuous												
Directed States												

5 烧写 FLASH

第一步: 正确安装CCS和F2812的烧写插件(请参考其他资料)

第二步: 运行CCS,则进入CCS开发环境,如下图所示



第三步: 将SPI目录COPY到CCS开发环境中的myproject目录下在CCS中用 project/Open.....命令,加载SPI目录下的SPI.pjt,编译,生成.out文件。 第四步: 点击CCS中的Tools —>F28XX On-Chip Flash Programer

On-Chip Flash Programmer
Clock Configuration Frase Sector Selection OSCCLK 30.000 PLLCR 10 ▼ StSCLKOUT (MHz): 150.0000 STSCLKOUT (MHz): 150.0000 Code Security Password ✓ Sector I (3E000-3EFFFF) Key 7 FFFF Key 6 FFFF Key 5 FFFF Key 4 FFFF Key 2 FFFF Key 1 FFFF Key 0 FFFF Valock Lock Program Password Calculate Checksum Flash Prog

第五步: 点击"Execute Operation"则显示如下:

然后就会自动完成烧写。

lock Configuration —	Erase Sector Selection
CCLK 30.000	🔽 Sector A (3F6000-3F7FFF 🔽 Sector I (3E4000-3E7FFF
·	Sector B (3F4000-3F5FFF 🔽 Sector ((3E0000-3E3FFF
LLCR 10 💌	Sector C (3F0000-3F3FFF 🔽 Sector 1 (3DC000-3DFFFF)
weervour (uv.), 150,0000	Sector D (3EC000-3EFFFF, V Sector I (3DA000-3DBFFF)
150LR001 (MAZ). 150.0000	
Code Security Password—	Operation Place musify the COPP file to
Key 7 FFFF	Trease specify the Corr file to
Key 6 FFFF	E:\dsp2812test5_22\spi\Debug\Spi.out 3rowse
Kev 5	💿 Erase, Program, Vei 🔿 Load RAM Only
Key 4	C Erase Only C Frequency Test
Kerr 2	C Program, Veri Register GPAMux
Key 5 FFFF	C Program Only Pin PWM1 (0)
Key Z	C Verify On
Key 1 FFFF	Flack Render Weit 15 Calculate Checksum
Key O FFFF	Flash
	Flash Page Wart 15 MOTP:
URLOCK LOCK	OTP Wait 31 V Flash+OTP:
Program Password	
	Execute Uperation Help