



C8051F 系列 单片机开发与 C语言编程



童长飞 编著



北京航空航天大学出版社



C8051F 系列单片机开发与 C 语言编程

童长飞 编著

北京航空航天大学出版社

内 容 简 介

在介绍 C8051Fxxx 单片机工作原理的基础上,对单片机外设 ADC、DAC、比较器、定时器、PWM、方波输出和定时器捕捉等均给出调试代码;在总线方面,对 CAN、SMBUS I²C、SPI 和 UART 均给出双机调试代码和一些高层应用,以帮助读者更好地理解单片机工作原理;还增加了 LCD、红外编码解码、步进电机、直流电机、舵机、超声测距、PROM 及串行时钟等方面的应用,以提高实用性。

本书实例较多,而且采用 C 语言编程,程序代码可读性和移植性强,大部分代码稍加修改即可成为其他型号单片机的子程序;含光盘一张,内含各章节调试代码,以方便读者学习。

本书可作为使用 C8051F 系列单片机进行产品开发的工程技术人员的硬件和软件设计参考书,部分内容对其他类型单片机的开发也具有一定的参考价值。

图书在版编目(CIP)数据

C8051F 系列单片机开发与 C 语言编程/童长飞编著.

北京:北京航空航天大学出版社,2005.1

ISBN 7-81077-544-8

I. C… II. 童… III. ①单片微型计算机,C8051F 系列—程序设计②C 语言—程序设计 IV. ①TP368.1
②TP312

中国版本图书馆 CIP 数据核字(2005)第 001961 号

C8051F 系列单片机开发与 C 语言编程

童长飞 编著

责任编辑 梁厚蕴

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1 092 1/16 印张:36 字数:922 千字

2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-544-8 定价:54.00 元(含光盘 1 张)

前 言

Silicon Laboratories 公司出品的 C8051Fxxx 单片机*是完全集成的混合信号系统级芯片(SOC),具有与 MCS-51 完全兼容的指令内核。该系列单片机采用流水线处理(pipe line)技术,不再区分时钟周期和机器周期,能在执行指令期间预处理下一条指令,提高了指令执行效率。而且大部分 C8051F 单片机具备控制系统所需的模拟和数字外设,包括看门狗、ADC、DAC、电压比较器、电压基准输出、定时器、PWM、定时器捕捉和方波输出等,并具备多种总线接口,包括 UART、SPI、SMBUS(与 I²C 兼容)总线以及 CAN 总线。C8051F 系列单片机采用 Flash ROM 技术,集成 JTAG,支持在线编程。C8051F 系列单片机诸多特点和优越性,使其广受单片机系统设计工程师青睐,成为很多测控系统设计的首选机型。

本书以 C8051F040 为讲述对象,是因为该单片机为 C8051F 系列单片机中功能最全,最具有代表性的一款。熟悉了 C8051F040 单片机工作原理和编程方法后,可较快地对 C8051F 系列其他单片机上手编程。事实上模拟和数字外设以及各种总线具有共性,因而本书也可作为其他厂商单片机的参考资料,很多调试程序稍加修改,即可应用于其他单片机。而且本书全部代码采用 C 语言编程,增强了程序的可读性和可移植性。

作者对 C8051F040 绝大部分外设均开发了调试代码,并给出运行结果,以及一些改进措施,以帮助读者更好地理解单片机的工作原理,并确保程序的可靠性和实用性。本书大部分代码可作为用户开发代码的子程序,缩短了用户调试开发周期。为了增强实用性,作者在 C8051F040 基础上增加了两章:LCD 输入/输出子程序和单片机应用。此两章内容不局限于 C8051F 系列单片机,同样适用于其他系列单片机和 DSP。

本书含光盘一张,内含各章节调试代码以及相关资料的网络资源,以方便读者学习。

作者纠正了 Silicon Laboratories 公司 C8051F040 技术白皮书中的某些错误,以及 C8051F040.h 文件中的一处错误。

参与本书整理编写工作的其他人员还包括:陈安宁、葛颂、汤卿、张亮、陈浩、

* C8051Fxxx 单片机是原 Cygnal 公司的产品。该公司于 2003 年并入 Silicon Laboratories 公司。

桂本恒、唐亮、张远辉、陈青青、靳碧、王立、张子发、陈哲明、刘进锋、虞旦、祝群卫、付安新和刘志明。

作者导师孙优贤院士、熊蓉老师、章辉老师以及沈阳新华龙电子有限公司对本书的出版给予了很多的帮助,在此表示衷心的感谢。

由于时间仓促,作者水平有限,书中错误和不妥之处,恳请读者批评指正。作者联系方式:cftong@iipc.zju.edu.cn。

童长飞

2004年5月

于浙江大学

目 录

第 1 章 Keil C 编程

1.1 Keil C 快速入门	1
1.1.1 文件载入	1
1.1.2 端口硬件操作	5
1.1.3 中断操作	7
1.1.4 存储空间分配	11
1.1.5 指 针	13
1.2 Keil C 集成软件设置	20
1.2.1 Medwin 设置和 C51 芯片烧写	20
1.2.2 Silabs 集成软件设置	22
1.2.3 在 μ Vision51 中调试 Silabs 单片机	24
1.2.4 其他深入学习资料	24
1.3 优化程序	24
1.4 Keil C 库函数	25
1.4.1 CTYPE. H 字符函数	25
1.4.2 STDIO. H 标准 I/O 函数	27
1.4.3 STRING. H 字符串函数	31
1.4.4 STDLIB. H 标准库函数	33
1.4.5 MATH. H 数学函数	34
1.4.6 ABSACC. H 绝对地址访问函数	35
1.4.7 INTRINS. H 内部函数	35
1.4.8 STDARG. H 变量参数表	36
1.4.9 SETJMP. H 全程跳转函数	37

第 2 章 CIP51 处理器

2.1 CIP51 处理器特性	38
2.2 存储组织结构	41
2.3 中断系统	43
2.4 电源管理模式	45
2.5 CIP51 寄存器	46
2.5.1 SFRPGCN(SFR Page Control Register)	46
2.5.2 SFRPAGE(SFR Page Register)	47
2.5.3 SFRNEXT(SFR Next Register)	47
2.5.4 SFRLAST(SFR Last Register)	47
2.5.5 SP(Stack Pointer)	48

2.5.6	DPL(Data Pointer Low)	48
2.5.7	DPH(Data Pointer High)	48
2.5.8	PSW(Program Status Word)	49
2.5.9	ACC(Accumulator)	50
2.5.10	B(B Register)	50
2.5.11	IE(Interrupt Enable)	50
2.5.12	IP(Interrupt Priority)	51
2.5.13	EIE1(Extended Interrupt Enable 1)	52
2.5.14	EIE2(Extended Interrupt Enable 2)	53
2.5.15	EIP1(Extended Interrupt Priority 1)	54
2.5.16	EIP2(Extended Interrupt Priority 2)	54
2.5.17	PCON(Power Control)	55

第3章 12位模/数转换器 ADC0

3.1	模拟通道选择器和 PGA 可调增益	56
3.1.1	AMUX 模拟输入配置	57
3.1.2	HVDA 高压差动放大器	58
3.2	ADC0 工作模式	60
3.2.1	ADC0 启动方式	60
3.2.2	ADC0 跟踪模式	61
3.2.3	ADC0 信号建立时间	62
3.3	内部温度传感器	63
3.4	ADC0 可编程监控模块	63
3.5	ADC0 电气参数	67
3.6	参考电压	68
3.6.1	参考电压工作原理	68
3.6.2	参考电压设置寄存器	70
3.7	ADC0 寄存器	71
3.7.1	AMX0CF(AMUX0 Configuration Register)	71
3.7.2	AMX0SL(AMUX0 Channel Select Register)	71
3.7.3	HVA0CN(High Voltage Difference Amplifier Control Register)	72
3.7.4	ADC0CF(ADC0 Configuration Register)	73
3.7.5	ADC0CN(ADC0 Control Register)	74
3.7.6	ADC0H(ADC0 Data Word MSB Register)	75
3.7.7	ADC0L(ADC0 Data Word LSB Register)	75
3.7.8	ADC0GTH(ADC0 Greater Than Data High Byte Register)	76
3.7.9	ADC0GTL(ADC0 Greater Than Data Low Byte Register)	76
3.7.10	ADC0LTH(ADC0 Less Than Data High Byte Register)	76
3.7.11	ADC0LTL(ADC0 Less Than Data Low Byte Register)	77
3.8	ADC0 调试例程	77

例程 3-1 内部温度传感器	77
例程 3-2 利用过采样实现 16 位 ADC 转换	80
例程 3-3 ADC0 单端输入模式	86
例程 3-4 ADC0 差动输入模式	90
例程 3-5 高差压输入 ADC0 转换	94
例程 3-6 ADC0 监控中断	101

第 4 章 8 位模/数转换器 ADC2

4.1 ADC2 操作模式	106
4.1.1 ADC2 启动模式	107
4.1.2 ADC2 跟踪模式	107
4.1.3 ADC2 信号建立时间	107
4.2 ADC2 可编程监控模块	108
4.3 ADC2 电气参数	109
4.4 ADC2 寄存器	111
4.4.1 AMX2CF(AMUX2 Configuration Register)	111
4.4.2 AMX2SL(AMUX2 Channel Select Register)	111
4.4.3 ADC2CF(ADC2 Configuration Register)	112
4.4.4 ADC2CN(ADC2 Control Register)	113
4.4.5 ADC2(ADC2 Data Word Register)	114
4.4.6 ADC2GT(ADC2 Greater Than Data Register)	115
4.4.7 ADC2LT(ADC2 Less Than Data Register)	116
4.5 ADC2 调试例程	116

第 5 章 12 位数/模转换器 DAC

5.1 DAC 输出调度	121
5.2 DAC 操作寄存器	122
5.2.1 DAC0H(DAC0 High Byte Register)	122
5.2.2 DAC0L(DAC0 Low Byte Register)	123
5.2.3 DAC0CN(DAC0 Control Register)	123
5.2.4 DAC1H(DAC1 High Byte Register)	124
5.2.5 DAC1L(DAC1 Low Byte Register)	124
5.2.6 DAC1CN(DAC1 Control Register)	125
5.3 DAC 电气参数	126
5.4 DAC 调试例程	127

第 6 章 比较器

6.1 比较器工作寄存器	132
6.1.1 CPTnCN(Comparator 0, 1 and 2 Control Register)	132
6.1.2 CPTnMD(Comparator Mode Selection Register)	133

6.2	比较器电气参数	133
6.3	比较器调试例程	134
第 7 章 RESET 复位源		
7.1	上电复位(Power-on Reset)	140
7.2	失电复位(Power-fail Reset)	141
7.3	外部复位(External Reset)	141
7.4	时钟失步监测复位(Missing Clock Detector Reset)	141
7.5	比较器 0 复位(Comparator0 Reset)	142
7.6	外部 CNVSTR0 复位(External CNVSTR0 Pin Reset)	142
7.7	看门狗复位(Watchdog Timer Reset)	142
7.8	看门狗与复位源寄存器	143
7.8.1	WDTCN(Watchdog Timer Control Register)	143
7.8.2	RSTSRC(Reset Source Register)	144
7.9	复位电气参数	145
第 8 章 晶 振		
8.1	可编程内部晶振	147
8.2	内部晶振电气参数	149
8.3	外部晶振驱动电路	149
第 9 章 闪 存		
9.1	闪存编程	153
9.2	闪存安全技术	154
9.3	闪存操作寄存器	154
9.3.1	FLSCL(Flash Memory Control)	154
9.3.2	PSCTL(Program Store Read/Write Control)	155
9.3.3	FLACL(Flash Access Limit)	155
9.4	闪存调试例程	156
第 10 章 内 存		
10.1	内存访问方式	159
10.2	XRAM 接口选择与配置	160
10.3	XRAM 数据地址复用模式与非复用模式	161
10.3.1	数据地址复用模式	161
10.3.2	数据地址非复用模式	162
10.4	XRAM 访问控制时序	162
10.4.1	数据地址非复用模式时序	162
10.4.2	数据地址复用模式时序	165
10.5	内存控制寄存器	168

10.5.1	EMI0CN(External Memory Interface Control)	168
10.5.2	EMI0CF(External Memory Configuration)	169
10.5.3	EMI0TC(External Memory Timing Control)	170
10.6	外部内存接口时序参数	171

第 11 章 输入/输出端口

11.1	P0~P3 交叉开关配置	173
11.2	I/O 端口配置	176
11.2.1	I/O 端口输出模式配置	178
11.2.2	I/O 端口输入模式配置	179
11.3	I/O 端口寄存器	179
11.3.1	XBR0(Port I/O Crossbar Register 0)	179
11.3.2	XBR1(Port I/O Crossbar Register 1)	180
11.3.3	XBR2(Port I/O Crossbar Register 2)	181
11.3.4	XBR3(Port I/O Crossbar Register 3)	182
11.3.5	P0(Port0 Data Register)	183
11.3.6	P0MDOUT(Port0 Output Mode Register)	183
11.3.7	P1(Port1 Data Register)	184
11.3.8	P1MDIN(Port1 Input Mode Register)	184
11.3.9	P1MDOUT(Port1 Output Mode Register)	185
11.3.10	P2(Port2 Data Register)	185
11.3.11	P2MDIN(Port2 Input Mode Register)	185
11.3.12	P2MDOUT(Port2 Output Mode Register)	186
11.3.13	P3(Port3 Data Register)	186
11.3.14	P3MDIN(Port3 Input Mode Register)	187
11.3.15	P3MDOUT(Port3 Output Mode Register)	187
11.3.16	P4(Port4 Data Register)	187
11.3.17	P4MDOUT(Port4 Output Mode Register)	188
11.3.18	P5(Port5 Data Register)	188
11.3.19	P5MDOUT(Port5 Output Mode Register)	189
11.3.20	P6(Port6 Data Register)	189
11.3.21	P6MDOUT(Port6 Output Mode Register)	189
11.3.22	P7(Port7 Data Register)	190
11.3.23	P7MDOUT(Port7 Output Mode Register)	190

第 12 章 CAN

12.1	CAN 简介	191
12.2	CAN 协议	192
12.2.1	数据帧	194
12.2.2	远程帧	197

12.2.3	错误帧	197
12.2.4	过载帧	198
12.2.5	帧间空间	199
12.2.6	CAN 总线其他相关定义和要求	200
12.3	CAN 控制器寄存器	204
12.3.1	CAN Control Register (addresses 0x01 & 0x00)	205
12.3.2	Status Register (addresses 0x03 & 0x02)	206
12.3.3	Error Counter (addresses 0x05 & 0x04)	207
12.3.4	Bit Timing Register (addresses 0x07 & 0x06)	207
12.3.5	Test Register (addresses 0x0B & 0x0A)	208
12.3.6	BRP Extension Register (addresses 0x0D & 0x0C)	208
12.3.7	IFx Command Request Registers	209
12.3.8	IFx Command Mask Registers	209
12.3.9	IFx Mask Registers	211
12.3.10	IFx Arbitration Registers	212
12.3.11	IFx Message Control Registers	212
12.3.12	IFx Data A and Data B Registers	214
12.3.13	Interrupt Register (addresses 0x09 & 0x08)	214
12.3.14	Transmission Request Registers	215
12.3.15	New Data Registers	215
12.3.16	Interrupt Pending Registers	215
12.3.17	Message Valid 1 Register	216
12.3.18	内存数据帧报文结构示意图	216
12.4	CAN 测试模式	216
12.4.1	静音模式	216
12.4.2	回路模式	217
12.4.3	回路静音模式	217
12.4.4	基本模式	218
12.5	CAN 波特率设置	218
12.6	CAN 调试例程	223
例程 12-1	CAN 双节点通信	223
例程 12-2	CAN 多消息通信	231
例程 12-3	CAN 交互收/发通信	236
例程 12-4	CAN 远程帧控制	244
12.7	CAN 外围电路	252

第 13 章 SMBUS 总线

13.1	I ² C 总线简介	255
13.2	SMBUS 概述	257
13.3	SMBUS 器件连接	257

13.4 SMBUS 协议	258
13.4.1 SMBUS 时序	258
13.4.2 SMBUS 仲裁	259
13.4.3 SMBUS 传输模式	260
13.5 SMBUS 寄存器	262
13.5.1 SMB0CN(SMBUS0 Control Register)	263
13.5.2 SMB0CR(SMBUS0 Clock Rate Register)	265
13.5.3 SMB0DAT(SMBUS0 Data Register)	265
13.5.4 SMB0ADR(SMBUS0 Address Register)	266
13.5.5 SMB0STA(SMBUS0 Status Register)	266
13.6 SMBUS 调试例程	268
13.6.1 利用配置向导完成 SMBUS 初始化	268
13.6.2 SMBUS 主发从收调试过程	280
13.6.3 SMBUS 主收从发调试过程	285
13.6.4 应用 SMBUS 实现内存互访	290

第 14 章 SPI 总线

14.1 SPI 信号描述	301
14.1.1 MOSI 控制线	301
14.1.2 MISO 控制线	301
14.1.3 SCK 时钟线	301
14.1.4 NSS 片选线	302
14.2 SPI0 传输模式	303
14.2.1 SPI0 主机模式	303
14.2.2 SPI0 从机模式	304
14.3 SPI0 中断源	304
14.4 SPI0 时钟	305
14.5 SPI 寄存器	306
14.5.1 SPI0CFG(SPI0 Configuration Register)	307
14.5.2 SPI0CN(SPI0 Control Register)	308
14.5.3 SPI0CKR(SPI0 Clock Rate Register)	309
14.5.4 SPI0DAT(SPI0 Data Register)	309
14.6 SPI 调试例程	310

第 15 章 UART0 串行通信

15.1 UART0 工作模式	316
15.1.1 模式 0	316
15.1.2 模式 1	317
15.1.3 模式 2	318
15.1.4 模式 3	319

15.2	UART0 多机通信	319
15.2.1	屏蔽地址控制	320
15.2.2	广播地址控制	320
15.3	UART0 标准波特率设置	321
15.4	UART0 寄存器	322
15.4.1	SCON0(UART0 Control Register)	323
15.4.2	SSTA0(UART0 Status and Clock Selection Register)	324
15.4.3	SBUF0(UART0 Data Buffer Register)	325
15.4.4	SADDR0(UART0 Slave Address Register)	325
15.4.5	SADEN0(UART0 Slave Address Enable Register)	325
15.5	UART0 调试例程	326
例程 15-1	UART0 模式 1 全双工通信	326
例程 15-2	UART0 模式 3 多机通信	332

第 16 章 UART1 串行通信

16.1	UART1 增强型波特率产生器	340
16.2	UART1 工作模式	344
16.2.1	8 位工作模式	345
16.2.2	9 位工作模式	345
16.3	UART1 多机通信	346
16.4	UART1 寄存器	346
16.4.1	SCON1(Serial Port 1 Control Register)	346
16.4.2	SBUF1(Serial UART1 Port Data Buffer Register)	347
16.5	UART1 调试例程	348
例程 16-1	UART1 8 位模式双机通信	348
例程 16-2	UART1 多机通信	353

第 17 章 定时器

17.1	T0 和 T1	359
17.1.1	13 位定时/计数模式(MODE0)	359
17.1.2	16 位定时/计数器(MODE 1)	363
17.1.3	8 位自动重载定时/计数器(MODE 2)	363
17.1.4	8 位双定时/计数器(MODE3)	364
17.2	T0 和 T1 寄存器	365
17.2.1	TCON(Timer Control Register)	365
17.2.2	TMOD(Timer Mode Register)	366
17.2.3	CKCON(Clock Control Register)	367
17.2.4	TL0(Timer 0 Low Byte)	368
17.2.5	TL1(Timer 1 Low Byte)	368
17.2.6	TH0(Timer 0 High Byte)	368

17.2.7 TH1(Timer 1 High Byte)	368
17.3 T2,T3 和 T4 定时/计数器	369
17.3.1 T2,T3 和 T4 双向计数	369
17.3.2 捕捉模式	369
17.3.3 自动重载模式	370
17.3.4 方波输出模式	371
17.4 T2,T3 和 T4 寄存器	371
17.4.1 TMRnCN(Timer n Control Registers)	371
17.4.2 TMRnCF(Timer n Configuration Registers)	372
17.4.3 RCAPnL(Timer n Capture Register Low Byte)	373
17.4.4 RCAPnH(Timer n Capture Register High Byte)	374
17.4.5 TMRnL(Timer n Low Byte)	374
17.4.6 TMRnH(Timer n High Byte)	374
17.5 定时器调试例程	375
例程 17-1 T3 可调频率方波输出	375
例程 17-2 T 法测频	377
例程 17-3 T4 捕捉测脉宽	387

第 18 章 PCA 可编程计数器阵列

18.1 PCA 定时/计数器	393
18.2 捕捉/比较模块	394
18.2.1 边沿捕捉模式	395
18.2.2 定时模式	396
18.2.3 高速输出模式	397
18.2.4 频率输出模式	397
18.2.5 8 位 PWM 输出模式	398
18.2.6 16 位 PWM 输出模式	398
18.3 PCA0 寄存器	399
18.3.1 PCA0CN(PCA Control Register)	399
18.3.2 PCA0MD(PCA0 Mode Register)	400
18.3.3 PCA0CPMn(PCA0 Capture/Compare Mode Registers)	401
18.3.4 PCA0L(PCA0 Counter/Timer Low Byte)	402
18.3.5 PCA0H(PCA0 Counter/Timer High Byte)	402
18.3.6 PCA0CPLn(PCA0 Capture Module Low Byte)	402
18.3.7 PCA0CPHn(PCA0 Capture Module High Byte)	403
18.4 PCA 调试例程	403
例程 18-1 PCA 捕捉测脉宽	403
例程 18-2 PCA 高速输出模式	408
例程 18-3 PCA 可调频率方波输出	413
例程 18-4 PCA 8 位 PWM 脉宽调制	416

例程 18-5 可变周期 PWM 脉宽调制	418
例程 18-6 用 PCA 检测 UART0 波特率	421

第 19 章 LCD 输入/输出子系统

19.1 LCD 控制器 LC1621	428
19.2 LC1621 操作原理	432
19.3 LC1621 指令描述	436
19.4 LC1621 单片机接口操作	439
19.5 LCD 调试例程	443
例程 19-1 LCD 虚拟时序实现和显示操作	444
例程 19-2 用库函数 printf() 实现 LCD 显示操作	449
例程 19-3 输入/输出子系统与 LCD 菜单操作实现	454

第 20 章 单片机应用

20.1 串行可擦写 ROM AT24C256 操作	469
20.2 HT1380 时钟芯片控制	477
20.3 直流电机驱动控制	488
20.4 步进电机控制	493
20.5 舵机控制	499
20.6 超声波传感器测距	505
20.7 红外编码/解码	510
20.7.1 红外遥控原理	510
20.7.2 红外编码/解码电路	512
20.7.3 红外编码/解码调试代码	513
20.7.4 用 MCS-51 实现红外遥控	521

附录 A CIP51 指令集

附录 B C8051F040 封装及引脚说明

附录 C 中断矢量表

附录 D 随书光盘内容、各种资料资源及软件安装说明

附录 E C8051F 单片机开发工具

参考文献

第 1 章 Keil C 编程

1.1 Keil C 快速入门

本节主要讲述 Keil C 编程中常见问题的处理,而不再对 C 语言语法进行详细解释。若想系统地理解 Keil C 的语言和优化编译,请参见 Keil 公司的 *Cx51 Compiler* 白皮书,以及马忠梅等编写的《单片机的 C 语言应用程序设计》^[1] 和徐爱钧等编写的《单片机高级语言 C51 Windows 环境编程与应用》^[2]。且以下例子均是采用 insight 的 medwin+Keil C 6.0 编译进行分析的,参数均采用默认参数,若分析与读者的结果有出入,请谅解,因为不同的集成开发工具所设的默认参数可能有所不同,从而产生不同的编译结果。

1.1.1 文件载入

就 C 语言而言,要对寄存器操作,需先定义寄存器的地址,不像汇编语言可直接通过指令对寄存器进行操作。对 AT89C51/52 等单片机来说,其寄存器的地址定义文件可在 Keil C 的 INC(include)目录下找到,分别为 REG51.H 和 REG52.H,只要在程序的开头引入该文件即可。就 C8051F 系列单片机而言,则需将其厂商提供的头文件加进去。例如 C8051F040,其寄存器定义的头文件是 c8051F040.h。

【例 1-1】 单片机是 AT89C51,在 test.c 中要实现功能:读端口 P1 的值,并将其取反。

以下是 test.c 的代码:

```
#include "REG51.H"
typedef unsigned char uchar;    //将 unsigned char 关键字定义成 uchar,以便书写方便
uchar readp1(){                //对 P1 的读操作,返回所读的值
    P1=0xff;                   //一般对端口的读操作往往要求先置 1,后读
    return P1;
}
void writep1(uchar temp){      //对 P1 写操作
    P1=temp;                   //将 temp 的值赋给 P1
}
main(){
    uchar plval;
    plval=readp1();
    plval=~plval;              //字节取反指令
    writep1(plval);
}
```

注：也可用 `#include "REG52.H"`，因为 REG52.H 对各寄存器的地址定义与 REG51.H 是相同的，只是增加了 51 不具备的寄存器的地址定义。

值得注意的是，当多个文件存在于一个项目中时，加载文件时要避免对寄存器的重复定义，否则编译不会通过。

比如对 `io.h` 文件编写如下：

```
#include "REG51.H"
typedef unsigned char uchar;    //将 unsigned char 关键字定义成 uchar,以便书写方便
unsigned char readp1(){         //对 P1 的读操作,返回所读的值
    P1=0xff;                    //一般对端口的读操作往往要求先置 1,后读
    return P1;
}
void writep1(uchar temp){       //对 P1 写操作
    P1=temp;                    //将 temp 的值赋给 P1
}
```

对 `test.c` 的编写如下：

```
#include "REG51.H"
#include "io.h"
main(){
    uchar plval;
    plval=readp1();
    plval=~plval;               //字节取反指令
    writep1(plval);
    T0=0x00;                    //对 T0 定时器写 0
}
```

则编译时会提示重复定义寄存器错误。应将 `test.c` 中的 `#include "REG51.H"` 去掉或将 `io.h` 中的 `#include "REG51.H"` 去掉，当然也可采用条件编译技术解决此问题。

更值得注意的是，Keil C 是按照 `main()` 所含文件的 `include` 顺序进行编译的，若 `test.c` 中语句的顺序为：

```
#include "io.h"
#include "REG51.H"
```

则在编译 `io.h` 时，会出现寄存器未定义错误。

【例 1-2】 假如出了一款新的单片机，其指令内核与 51 兼容，多了 T3MOD, T3CON, T3L 和 T3H 四个寄存器来控制 T3，其地址分别为 0xc0, 0xc1, 0xc2 和 0xc3，其他寄存器地址与 C52 一样。若厂商不提供寄存器定义头文件，试改写 REG52.H，使之能用于该单片机。

只需在 REG52.H 的结尾加上以下代码，并另存为 REG52plus.H，以免混用。

```
sfr T3MOD= 0xc0;
sfr T3CON=0xc1;
sfr T3H=0xc3;
```