

CD-ROM

DANPIANJI YU WAIWEI QIJIAN JIEKOU

单片机与外围器件接口 实例详解

杜树春 张体才 编著



中国电力出版社
www.cepp.com.cn

DANPIANJI YU WAIWEI QIJIAN JIEKOU

单片机与外围器件接口 实例详解

杜树春 张体才 编著



中国电力出版社

www.cepp.com.cn

内 容 提 要

本书通俗易懂,条理清晰,通过实际工程实例介绍了单片机和外围器件接口方面的知识,每一个实例都选用了一种接口芯片进行介绍。

本书共分 15 章,主要内容包括带 RAM 存储器的实时时钟芯片 PCF8583;实时时钟/日历芯片 DS1307;数字温度传感器及存储器 DS1624;数字温度传感器及实时时钟/日历芯片 DS1629;电可擦除的只读存储器(E²PROM)AT93C46;SPI 串行总线 E²PROM X25040;8 通道 10 位串行模数转换器 MAX192;低功耗 8 通道 12 位 A/D 转换器 AD7888;4 通道 12 位串行 D/A 转换器 MAX525;单通道 14 位串行 D/A 转换器 MAX544/545;单通道 16 位串行 D/A 转换器 MAX541;线性 256 抽头数字电位器 MAX5402;双路非易失性可变电阻器 DS3902;线性 32 抽头数字电位器 MAX5160;I²C 总线 8 位远程 I/O 口扩展芯片 PCF8574 等。书中 C51 高级语言和汇编语言的程序实例,全部经过实际调试。对于有一定基础的人,书中的模块可直接使用。

本书适用于单片机初学者和工程技术人员参考使用,也可作为大中专院校电子技术、自动化、计算机及相关专业的辅助教材和培训用书。

图书在版编目(CIP)数据

单片机与外围器件接口实例详解 / 杜树春, 张体才编著. —北京: 中国电力出版社, 2009
ISBN 978-7-5083-7961-6

I. 单… II. ①杜… ②张… III. 单片微型计算机—数据传输—接口设备 IV. TP368.147

中国版本图书馆 CIP 数据核字(2008)第 168577 号

责任编辑: 刘 焜

责任校对: 崔燕菊

责任印制: 郭华清

书 名: 单片机与外围器件接口实例详解

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 68362602 传真: (010) 68316497

印 刷: 北京市同江印刷厂

开本尺寸: 185mm × 260mm 印 张: 18.5 字 数: 436 千字

书 号: ISBN 978-7-5083-7961-6

版 次: 2009 年 1 月北京第 1 版

印 次: 2009 年 1 月第 1 次印刷

印 数: 0001—3000 册

定 价: 32.00 元(含 1CD)

敬告读者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

Preface

前言

单片机就是最小的计算机。单片机的应用极其广泛，目前它已经渗透到人们生活的各个领域。

单片机种类繁多，以字长来分，有 4 位机、8 位机、16 位机和 32 位机等；以用途来分，有专用型和通用型两类。专用型单片机这里不作解释，通用型单片机是指可供开发人员设定其功能的单片机，这种单片机配以不同的接口电路或不同的外围芯片，再编写不同的程序就可以实现不同的功能。一般情况下所说的单片机都是指通用型单片机。

对于初学单片机的人而言，在学习了单片机的工作原理后，从事实际的工程项目开发时，往往会感到无从下手。但如果在此基础上再学习一些单片机应用的实例，并且实际动手做一做，情形就会大大不同。实践证明，在学过工作原理的基础上，再学习一些单片机应用实例（单片机和外围芯片接口的实例）是学习单片机开发技术的捷径。

51 系列单片机是 8 位单片机，是目前应用最广泛、使用人数也最多的单片机。它常是初学者的首选。51 系列单片机包括由不同厂家生产的型号各异的单片机，常用的有 ATMELE 公司的 AT89C51、AT89C52、AT89C55、AT89C2051 等；INTEL 公司的 8031、8032、8751、8752 等；WINBOND 公司的 W77E58、W78E58、W78E516 等。这些单片机各有特点，但其基本内核相同，指令系统也完全兼容，换句话说，它们都能识别用 ASM51 汇编语言和 C51 高级语言编写的程序。本书各个实例选用的单片机是国内用得最多的 AT89C52，如果要把它应用于其他型号的 51 系列单片机时，绝大部分程序都可以原封不动的照搬，极少数需稍作修改。

主要特点

本书是通过实际例子介绍 51 单片机和外围器件接口方法的。每一种实例选用了一种接口芯片，介绍的顺序依次是：该单片机接口芯片功能简介；该芯片与单片机连接电路原理、汇编语言程序、被调汇编语言程序、调汇编语言的 C51 语言程序、不调汇编语言的 C51 语言程序（有些例子没有这一项）；使用该芯片的注意事项。

由于书中大部分实例涉及单片机 C51 高级语言和 ASM51 汇编语言混合编程，对其中一些规则和语法不熟练的读者，可参阅书末的附录 A “Keil C51 和 A51 接口所涉及的几个主要问题”。

本书对一种芯片提供三种不同的程序——汇编语言程序、调汇编语言的 C51 语言程序和不调汇编语言的 C51 语言程序，这扩大了使用本书的读者范围。即不论是单纯使用汇编语言编程的读者，还是单纯使用 C51 语言编程的读者，以及用 C51 语言和汇编语言混合编

程的读者都可以借鉴本书。

内容简介

本书共分 15 章，每一章介绍一个单片机和外围芯片连接的例子，一共 15 个例子。第 1 章介绍带 RAM 存储器的实时时钟芯片 PCF8583；第 2 章介绍实时时钟/日历芯片 DS1307；第 3 章介绍数字温度传感器及存储器 DS1624；第 4 章介绍 I^2C 温度传感器及实时时钟 DS1629；第 5 章介绍电可擦除的只读存储器（E² PROM）AT93C46；第 6 章介绍 SPI 串行总线 E² PROM X25040；第 7 章介绍 8 通道 10 位串行 A/D 转换器 MAX192；第 8 章介绍低功耗 8 通道 12 位 A/D 转换器 AD7888，第 9 章介绍 4 通道 12 位串行 D/A 转换器 MAX525；第 10 章介绍单通道 14 位串行 D/A 转换器 MAX544/545；第 11 章介绍单通道 16 位串行 D/A 转换器 MAX541；第 12 章介绍线性 256 抽头数字电位器 MAX5402；第 13 章介绍双路非易失性可变电阻器 DS3902；第 14 章介绍线性 32 抽头数字电位器 MAX5160；第 15 章介绍 I^2C 总线 8 位远程 I/O 口扩展芯片 PCF8574。

光盘说明

在本书所赠光盘中，使用了两种单片机集成开发环境软件，一种是目前最流行的 Keil C51 单片机集成开发环境软件，一种是国内用的较多的伟福 WAVE 单片机集成开发环境软件。在光盘中每一章下有三个文件夹，一个是图形夹，内放本章所介绍芯片和单片机连接电路原理图；一个是 Keil C51 单片机集成开发环境下载程序夹；一个是伟福 WAVE 单片机集成开发环境下载程序夹。书中每一个芯片的应用程序都已在这两种开发环境下调试通过，读者可以任择一种自己熟悉的集成开发环境。

适用读者

本书适用于 51 系列单片机包括 C51 高级语言和 ASM51 汇编语言的初学者，书中 C51 高级语言和汇编语言的程序实例全部经过实际调试。对于有一定基础的人，书中的模块可以直接使用。本书既适用于工程技术人员参考和套用，也可作为各工科高等院校和中等职业学校的单片机辅助或参考教材。

致谢

在本书编写过程中，得到杜朋、杜菲等同志的协助。在此，向他们表示衷心感谢。本书还参考了有关方面的应用实例，其来源都在书末的参考文献中给出，在此对有关作者表示诚挚的感谢。

限于作者水平，并且写作时间仓促，书中难免存在不足和错误，恳请读者批评指正。

作者

2008 年 8 月

Contents

目 录

前 言

第 1 章 带 RAM 存储器的实时时钟芯片 PCF8583	1
1.1 PCF8583 简介	1
1.2 PCF8583 和单片机 AT89C52 的连接	3
1.3 汇编语言程序	4
1.4 被调汇编语言程序	9
1.5 调用汇编的 C51 语言程序	15
1.6 C51 语言程序	16
1.7 使用 PCF8583 的注意事项	21
第 2 章 带 RAM 存储器的串行 I ² C 实时时钟芯片 DS1307	23
2.1 DS1307 简介	23
2.2 DS1307 和单片机 AT89C52 的连接	26
2.3 汇编语言程序	27
2.4 被调汇编语言程序	32
2.5 调用汇编的 C51 语言程序	38
2.6 C51 语言程序	40
2.7 使用 DS1307 的注意事项	47
第 3 章 数字温度传感器和存储器 DS1624	48
3.1 DS1624 简介	48
3.2 DS1624 和单片机的连接	52
3.3 汇编语言程序	53
3.4 被调汇编语言程序	59
3.5 调用汇编的 C51 语言程序	65
3.6 C51 语言程序	66
3.7 使用 DS1624 的注意事项	76
第 4 章 I ² C 数字温度传感器及实时时钟 DS1629	77
4.1 DS1629 简介	77
4.2 DS1629 和单片机的连接	81
4.3 汇编语言程序	81
4.4 被调汇编语言程序	89

4.5	调用汇编的 C51 语言程序	97
4.6	C51 语言程序	98
4.7	使用 DS1629 的注意事项	110
第 5 章	电可擦除的只读存储器 (E²PROM) AT93C46	111
5.1	AT93C46 简介	111
5.2	AT93C46 与单片机 AT89C52 的连接	113
5.3	汇编语言程序	114
5.4	被调汇编语言程序 1	121
5.5	被调汇编语言程序 2	127
5.6	调用汇编的 C51 语言程序 1	133
5.7	调用汇编的 C51 语言程序 2	134
5.8	使用 AT93C46 的注意事项	135
第 6 章	SPI 串行总线 E²PROM X25040	136
6.1	X25040 简介	136
6.2	X25040 与单片机 AT89C52 的连接	139
6.3	汇编语言程序	139
6.4	被调汇编语言程序	150
6.5	调用汇编的 C51 语言程序	156
6.6	使用 X25040 的注意事项	158
第 7 章	8 通道 10 位串行 A/D 转换器 MAX192	159
7.1	MAX192 简介	159
7.2	MAX192 与单片机 AT89C52 的连接	162
7.3	汇编语言程序	163
7.4	被调汇编语言程序	165
7.5	调用汇编的 C51 语言程序	167
7.6	C51 语言程序	168
7.7	使用 MAX192 的注意事项	171
第 8 章	低功耗 8 通道 12 位 A/D 转换器 AD7888	172
8.1	AD7888 简介	172
8.2	AD7888 和单片机的连接	174
8.3	汇编语言程序	175
8.4	被调汇编语言程序	176
8.5	调用汇编的 C51 语言程序	178
8.6	C51 语言程序	179
8.7	使用 AD7888 的注意事项	182
第 9 章	4 通道 12 位串行 D/A 转换器 MAX525	183
9.1	MAX525 简介	183
9.2	MAX525 与单片机的硬件连接	187
9.3	汇编语言程序	188
9.4	被调汇编语言程序	190
9.5	调用汇编的 C51 语言程序	191

9.6	C51 语言程序	192
9.7	使用 MAX525 的注意事项	194
第 10 章	单通道 14 位串行 D/A 转换器 MAX544/545	195
10.1	MAX544/545 简介	195
10.2	单片机 AT89C52 与 MAX544 的接口电路	197
10.3	汇编语言程序	198
10.4	被调汇编语言程序	199
10.5	调用汇编的 C51 语言程序	201
10.6	C51 语言程序	202
10.7	使用 MAX544 的注意事项	203
第 11 章	单通道 16 位串行 D/A 转换器 MAX541	205
11.1	MAX541 简介	205
11.2	单片机 AT89C52 与 MAX541 的接口电路	207
11.3	汇编语言程序	207
11.4	被调汇编语言程序	209
11.5	调用汇编的 C51 语言程序	210
11.6	C51 语言程序	211
11.7	使用 MAX541 的注意事项	213
第 12 章	线性 256 抽头数字电位器 MAX5402	215
12.1	MAX5402 简介	215
12.2	MAX5402 和单片机 AT89C52 的连接	217
12.3	汇编语言程序	217
12.4	被调汇编语言程序	219
12.5	调用汇编的 C51 语言程序	220
12.6	C51 语言程序	221
12.7	使用 MAX5402 的注意事项	222
第 13 章	双路非易失性可变电阻器 DS3902	224
13.1	DS3902 简介	224
13.2	DS3902 和单片机 AT89C52 的连接	228
13.3	汇编语言程序	228
13.4	被调汇编语言程序	235
13.5	调用汇编的 C51 语言程序	244
13.6	C51 语言程序	246
13.7	使用 DS3902 的注意事项	256
第 14 章	线性 32 抽头数字电位器 MAX5160	257
14.1	MAX5160 简介	257
14.2	MAX5160 和单片机 AT89C52 的连接	259
14.3	汇编语言程序	259
14.4	被调汇编语言程序	261
14.5	调用汇编的 C51 语言程序	262
14.6	使用 MAX5160 的注意事项	263

第 15 章 I ² C 总线 8 位远程 I/O 口扩展芯片 PCF8574	264
15.1 PCF8574 简介	264
15.2 PCF8574 与单片机的连接	267
15.3 汇编语言程序	267
15.4 被调汇编语言程序	270
15.5 调用汇编的 C51 语言程序	273
15.6 C51 语言程序 1	274
15.7 C51 语言程序 2	278
15.8 使用 PCF8574 的注意事项	281
附录 Keil C51 和 A51 接口所涉及的几个主要问题	283
参考文献	285

第 1 章

带 RAM 存储器的实时 时钟芯片 PCF8583

实时时钟/日历芯片种类很多,按其接口方式来分,可分为并行接口和串行接口两类。而串行接口又分为三线式(SPI)接口、两线式(I²C)接口和单线式(1-Wire)接口三种。本章介绍的串行时钟日历芯片 PCF8583 和第 2 章介绍的 DS1307 均属于两线式(I²C)接口芯片。

1.1 PCF8583 简介

PCF8583 是飞利浦(PHILIPS)公司生产的 CMOS 型实时时钟芯片,它带有 240 个字节 RAM,并且具有 I²C 串行总线接口。只需要通过单片机并行 I/O 口的两个引脚模拟 I²C 总线的数据线 SDA、时钟线 SCL,就可以完成对 PCF8583 的参数设置、读取日期和时间数据等操作。PCF8583 具有接口简单、占用单片机资源少等优点。因而被广泛应用在各类智能仪器仪表和电子工程项目中。

1. 特性

- (1) 工作电源: 2.5~6.0V。
- (2) 日历时钟工作电源和 RAM 数据保持电源电压: 1.0 ~ 6.0 V。
- (3) 240×8 位低电压 RAM, 自动字节地址增加。
- (4) 工作电流(在 f_{scl} = 0 Hz 时): 最大为 50μA。
- (5) 4 年日历时钟, 24 或 12 小时格式, 32.768 kHz 或 50Hz 时基。
- (6) 具有串行总线接口(I²C)。
- (7) 具有可编程的闹钟、定时和中断功能。
- (8) I²C 总线从地址: 读, 0A1H 或 0A3H; 写, 0A0H 或 0A2H。

2. PCF8583 的内部结构

PCF8583 是一个低功耗的 8 脚芯片,其内部结构包括振荡器和分频器电路、上电复位电路、串行 I²C 总线接口电路、地址寄存器、控制逻辑电路和 256 字节的静态 RAM 存储单元。其中地址为 00H 的 RAM 单元为控制/状态寄存器,地址为 01H~07H 的 RAM 单元为专用的时钟日历计数器;地址为 08H~0FH 的 RAM 单元可编程为空闲单元或闹钟寄存器;其余的 240 个字节可以作为通用的数据存储器。PCF8583 的整体内部结构如图 1.1 所示。PCF8583 的日历时钟寄存器及用户 RAM 的地址分布见表 1.1。

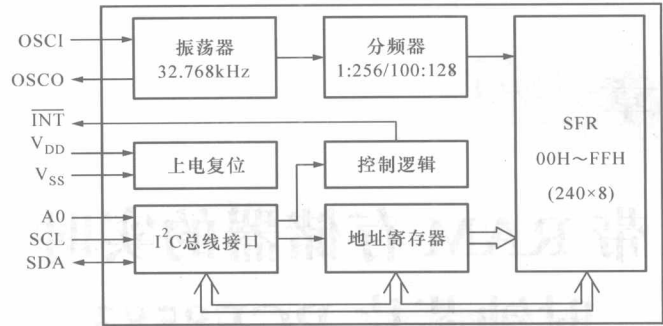


图 1.1 PCF8583 的整体内部结构图

表 1.1 PCF8583 的日历时钟寄存器及用户 RAM 的地址分布

地 址	内 容	地 址	内 容
00H	控制/状态寄存器	06H	星期/月寄存器
01H	百分之一秒寄存器	07H	定时器
02H	秒寄存器	08H	闹钟控制
03H	分寄存器	09H~0FH	闹钟寄存器或 RAM
04H	小时寄存器	10H~FFH	240 个字节的 RAM 区
05H	年/日寄存器		

3. PCF8583 的引脚功能

图 1.2 为 PCF8583 的引脚图，其各引脚功能为：

(1) V_{DD} 、 V_{SS} ：分别为工作电源（+5 V）和公共地（GND）引脚。

(2) OSC1、OSCO：时钟/计数脉冲引脚。在日历时钟方式下，OSC1、OSCO 引脚外接晶体振荡器，晶体频率可以选择 32.768 kHz 或 50 Hz。（通过设控制字选择）。在计数方式下，OSC1 引脚接计数脉冲，而 OSCO 引脚悬空。

(3) A0：地址输入引脚。

(4) SDA：I²C 总线的串行数据引脚。

(5) SCL：I²C 总线的串行时钟引脚。

(6) INT：中断输出信号引脚。

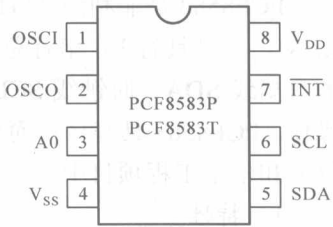


图 1.2 PCF8583 的引脚图

4. PCF8583 的控制/状态寄存器和专用时间数值寄存器

(1) PCF8583 的控制/状态寄存器。

PCF8583 的控制/状态寄存器位于其内部 RAM 的 00H 单元，是一个 8 位寄存器。

其中：

D7 为运行控制位。

D7=0 为启动运行（开始对振荡器或外部输入脉冲计数）；

D7=1 为停止运行（停止计数，复位分频器）；

D6 为计数器状态位。

D6=0 正常计数；

D6=1 保持、存储最新的（最后的）计数值。

D5、D4 为功能方式设置位。D5、D4=00 为 32.768kHz 的时钟方式；

D5、D4=01 为 50Hz 的时钟方式；

D5、D4=10 为事件计数方式；

D5、D4=11 为测试方式。

D3 为屏蔽位。 D3=0 读 05H、06H 单元内容时不屏蔽（正常读出全部内容）。 D3=1 读 05H、06H 单元时屏蔽年份、星期内容（只读月、日数值）

D2 为起闹允许位。 D2=0 禁止起闹。（禁止触发起闹标志）

D2=1 允许起闹。（08H 单元为起闹控制寄存器）

D1 该位在 D2 位=1 时为起闹标志位。 D1=0 表示未到闹钟设置时间。

D1=1 表示已到闹钟设置时间。

该位在 D2 位=0 时为分标志位。

D0 该位在 D2 位=1 时为计时器标志位。

该位在 D2 位=0 时为秒标志位。

向各个计时数值寄存器（01H~06H）送初值时，应使计时器停止运行（使控制寄存器的 D7 位=“0”）。而初值设置完毕，启动计时器时需要将控制寄存器的 D7 位设置为 1。

(2) PCF8583 的计时数值寄存器。

1) 1/100 秒数值寄存器[01H 单元]。本单元存放两位 BCD 码，其中：低位存放 1/100 秒数值，范围为 0~9。高位存放 1/10 秒数值，范围为 0~9。

2) 秒数值寄存器[02H 单元]。本单元存放两位 BCD 码，其中：低位存放秒数的个位数值，范围为 0~9。高位存放秒数的十位数值，范围为 0~5。

3) 分数值寄存器[03H 单元]。本单元存放两位 BCD 码，其中：低位存放分钟的个位数值，范围为 0~9。高位存放分钟的十位数值，范围为 0~5。

4) 小时数值寄存器[04H 单元]。本单元的 D3~D0 四位为小时的个位数值，用 BCD 码形式存放，数值范围为 0~9。

D5、D4 两位存放的是小时的十位数值，其数值范围受 D6、D7 位控制。

D6 位为 AM/PM 标志。D6=1 为 AM；D6=0 为 PM。

D7 位是计时类型选择位。D7=1 为 12 小时制（此时 D6 位有效并变化）。

D7=0 为 24 小时制（此时 D6 位无意义）。

5) 年/日数值寄存器[05H 单元]。该单元的 D3~D0 四位为日的个位数值，用 BCD 码形式存放，数值范围为 0~9。

D5、D4 两位存放日的十位数值，数值范围为 0~3。

D7、D6 两位存放的是年份的相对数值，其范围为 00~11。

由此可见，如果使用时间超过 4 年，要通过程序对 PCF8583 重新初始化。

6) 星期/月数值寄存器[06H 单元]。该单元的 D3~D0 四位为月的个位数值，用 BCD 码形式存放，数值范围为 0~9。

D4 存放月的十位数值。数值范围为 0~1。

D7~D5 三位存放星期数值，其数值范围为 000~110，分别表示星期日~星期六。

1.2 PCF8583 和单片机 AT89C52 的连接

在 MCS-51 系列单片机的基础型号中（如 80C31、AT89C51 等），没有专门的 I²C 接口，

在这种情况下可以使用 I/O 口（如 P1 口）中的两条 I/O 引脚来模拟 I²C 接口的 SDA 和 SCL 信号线，并通过相应的程序来控制这两条引脚上的信号波形来实现 I²C 总线上的起始、终止和数据传输时序，从而完成 PCF8583 的命令初始化和 I²C 总线上的数据传输。MCS-51 系列单片机 AT89C52 与 PCF8583 的硬件接口电路如图 1.3 和图 1.4 所示。

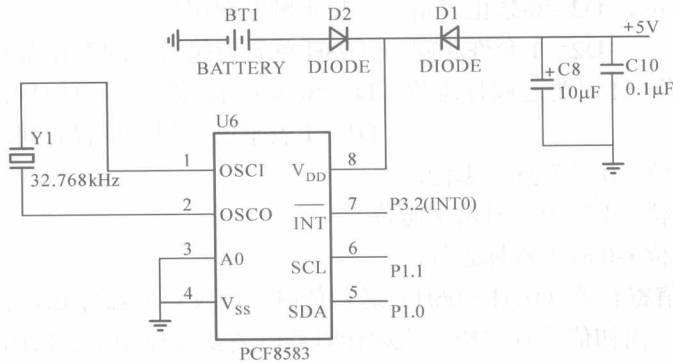


图 1.3 AT89C52 单片机与 PCF8583 接口电路的 PCF8583 部分

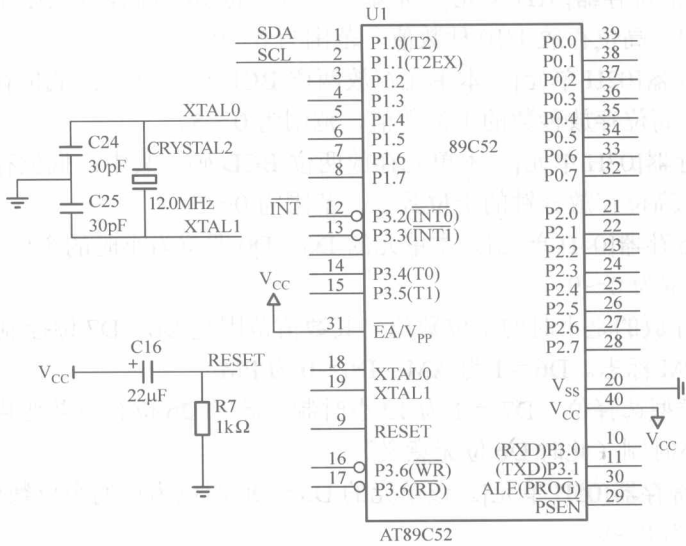


图 1.4 AT89C52 单片机与 PCF8583 接口电路的 AT89C52 部分

图 1.4 中 PCF8583 的 SDA 和 SCL 分别与单片机 AT89C52 的 P1.0 和 P1.1 相连,PCF8583 的 OSCI 和 OSC0 接 32.768kHz 的晶振, Vss 和 A0 接地, PCF8583 的 VDD 接 +5V 电源及电池。INT 接 AT89C52 的 P3.2 (INT0),以便接收 PCF8583 发来的中断信号。

由于 PCF8583 的芯片寻址字节为 101000 A0 R/W,所以按照上述电路,本章中 PCF8583 器的芯片写操作时地址为 0A0H,而读操作时地址为 0A1H。

1.3 汇编语言程序

```
;/*****  
;/* 这是读写 PCF8583 芯片的汇编语言程序, 文件名为 PCF85832.ASM
```

```

; /* 功能: 本程序主要是读 pcf8583 的日历时钟值 (共 6 个字节) 和 RAM 值以及写 PCF8583 的 */
; /*      日历时钟值和 RAM 值。 */
; /*****
SDA    EQU    P1.0
SCL    EQU    P1.1
DSEC   DATA 30H      ; 百分之一秒寄存器
SEC     DATA 31H      ; 秒寄存器
MIN     DATA 32H      ; 分寄存器
HOUR    DATA 33H      ; 时寄存器
DAY     DATA 34H      ; 日/年寄存器
MONTH   DATA 35H      ; 月/周寄存器

ORG 0000H              ; 主程序
LJMP MAIN
ORG 0030H
MAIN: MOV SP, #0CFH
      CLR EA
      LCALL INTTIME    ; 时间日历初始值
      LCALL INTRAM     ; RAM 初始值
      LCALL SETTIME    ; 当前时钟日历值设定
      LCALL SETRAM     ; RAM 区设定
      LCALL READTIME   ; 读出时钟芯片当前值
      LCALL READRAM    ; 读出 RAM 值
      SJMP $

; PCF8583 当前时间设定子程序
SETTIME:
      MOV R2, #00H     ; 设置 PCF8583 的控制寄存器 00H 为 00H, 以普通模式运行
      MOV R3, #0H
      LCALL WRITEDATA   ; 写单个字节数据程序
      MOV 38H, DSEC     ; 时间初始化——将百分之一秒至月的时间写入发送缓冲区
      MOV 39H, SEC
      MOV 3AH, MIN
      MOV 3BH, HOUR
      MOV 3CH, DAY
      MOV 3DH, MONTH
      LCALL WRITEM      ; 调用写多个字节数据程序
      RET

; 读出时钟芯片当前值子程序
READTIME:
      MOV R1, #01H     ; 取待读寄存器地址
      MOV R3, #6        ; 读 6 个时钟信息
      LCALL READDATA1   ; 调读多个字节数据程序
      MOV A, 40H        ; 取百分之一秒字节
      MOV DSEC, A       ; 送百分之一秒寄存器
      MOV A, 41H        ; 取秒字节
      ANL A, #7FH       ; 屏蔽无效位
      MOV SEC, A        ; 送秒寄存器

```



```

MOV A, 42H ;取分字节
ANL A, #7FH ;屏蔽无效位
MOV MIN, A ;送分寄存器
MOV A, 43H ;取小时字节
ANL A, #3FH ;屏蔽无效位
MOV HOUR, A ;送时寄存器
MOV A, 44H ;取天/年字节
ANL A, #3FH ;屏蔽无效位
MOV DAY, A ;送天寄存器
MOV A, 45H ;取月/星期字节
ANL A, #1FH ;屏蔽无效位
MOV MONTH, A ;送月寄存器
RET

;读出 RAM 值子程序
READRAM :
MOV R1, #10H ;取待读 RAM 地址
MOV R3, #8 ;读 8 个字节
LCALL READDATA1 ;调读多个字节数据程序
RET

;时间日历初始值
INTTIME :MOV 30H, #00H
MOV 31H, #01H
MOV 32H, #59H ;时间初值: 07-07, 07:59:01 : 00
MOV 33H, #07H
MOV 34H, #07H
MOV 35H, #07H
RET

;RAM 初始值
INTRAM:MOV 48H, #00H
MOV 49H, #11H
MOV 4AH, #22H
MOV 4BH, #33H
MOV 4CH, #44H
MOV 4DH, #55H
MOV 4EH, #66H
MOV 4FH, #77H
RET

DELAY: ;延时程序
NOP
NOP
NOP
RET

;起动位程序
BSTART:SETB SDA ;发出一个起始位
SETB SCL
LCALL DELAY
CLR SDA

```

```

    LCALL DELAY
    CLR SCL
    RET
;停止位程序:
BSTOP:                                ;发出一个停止位
    CLR SDA
    CLR SCL
    LCALL DELAY
    SETB SCL
    LCALL DELAY
    SETB SDA
    LCALL DELAY
    RET
WAITACK:MOV R7,#20                    ;等待应答
    SETB SDA
    LCALL DELAY
    SETB SCL
    LCALL DELAY
FLAG1:  MOV C,SDA
    JNC FLAG2
    DJNZ R7,FLAG1
    LCALL BSTOP
FLAG2:  CLR SCL
    LCALL DELAY
    RET
;输出 ACK ,每个字节传输完成, 输出 ACK=0,结束读数据, ACK=1
WRITEACK:
    MOV SDA,C
    LCALL DELAY
    SETB SCL
    LCALL DELAY
    CLR SCL
    RET
;读一个字节数据程序: 入口 R1 放待读数据地址
READDATA:
    LCALL BSTART                      ;START
    MOV A,#0A0H                      ;写命令
    LCALL SENDBYTE
    MOV A,R1                          ;写地址
    LCALL SENDBYTE
    LCALL BSTART                      ;START
    MOV A,#0A1H                      ;读命令
    LCALL SENDBYTE
    LCALL READBYTE
    SETB C
    LCALL WRITEACK
    LCALL BSTOP

```

RET
;读多个字节数据程序: 入口 R1 放待读数据首地址, R3 中放要读数据个数

```
READDATA1:
    LCALL BSTART      ;START
    MOV A,#0A0H       ;写命令
    LCALL SENDBYTE
    MOV A,R1           ;写地址
    LCALL SENDBYTE
    LCALL BSTART      ;START
    MOV A,#0A1H       ;读命令
    LCALL SENDBYTE
    MOV R1,#40H        ;40H 为数据暂存区首址
    MOV B,R3           ;R3 中放要读个数
    DEC B
LOOP: LCALL READBYTE
    MOV A,R2
    MOV @R1,A
    INC R1
    CLR C
    LCALL WRITEACK
    DJNZ B,LOOP
    LCALL READBYTE
    MOV A,R2
    MOV @R1,A
    SETB C
    LCALL WRITEACK
    LCALL BSTOP
    RET
```

;写字节程序

```
SENBYTE: MOV R0,#08H
LOOP1:   RLC A
    MOV SDA,C
    SETB SCL
    LCALL DELAY
    CLR SCL
    DJNZ R0,LOOP1
    LCALL WAITACK
    RET
```

;读字节子程序

```
READBYTE: MOV R0,#08H
    CLR A
    SETB SDA
LOOP4:   SETB SCL
    MOV C,SDA
    RLC A
    CLR SCL
```