

doi: 10.3969/j.issn.1006-1576.2011.03.018

基于 RTC 的实时串口通信

樊鹏, 董琦昕

(中国兵器工业第 58 研究所 军品部, 四川 绵阳 621000)

摘要: 为实现多帧异步数据的实时发送, 结合某分布式火控系统的性能检测要求, 提出一种 PC104 平台下串口数据精确实时发送的控制方法。介绍了实时钟及 PC104 串口通信的基本原理, 设计了合理的中断控制方案。其通信软件的设计主要包括实时钟中断挂接及中断处理、串口中断挂接及中断处理、检测数据生成等关键部分。该方法已在某火控系统中得到实际应用, 能将各节点的检测数据在时间上严格对齐, 提高了火控系统的分析精度。

关键词: 火控系统; 实时钟; PC104; 串行通信

中图分类号: TP206 **文献标志码:** A

Real-Time Serial Communication Based on RTC

Fan Peng, Dong Qixin

(Dept. of Armament Products, No. 58 Research Institute of China Ordnance Industries, Mianyang 621000, China)

Abstract: To transmit several frames of asynchronous data immediately, combined with the requirements of performance testing of a distributed fire control system. This paper proposes a method of transmitting serial data immediately with accurate time control for PC104 platform. The basic principles of real-time clock and serial communication on PC104 are discussed, and a reasonable interrupt control scheme was designed. The design of communication software includes real-time clock interrupt installation and handling, serial port interrupt installation and handling, and generation of testing data. The method has been applied to a fire control system in which testing data of each node can be aligned strictly on time line, and the precision of performance analysis was improved.

Keywords: fire control system; real-time clock; PC104; serial communication

0 引言

某分布式火控系统中, 各控制节点异步工作, 为了检测系统性能, 要求各控制节点通过串行通信端口实时发送本节点的检测数据、数据有效时刻与发送时刻之间的延迟量。笔者在以 PC104 为平台的节点中结合实时钟 (Real-time clock, RTC), 实现串口数据的精确实时发送。

1 系统概述

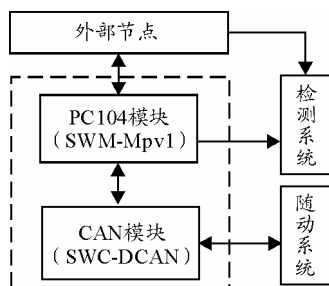


图 1 系统组成框图

系统组成框图如图 1, 其中, PC104 模块为节点的主控模块, 其主要功能是接收外部节点发送的目标参数完成射击诸元解算。本节点接收外部节点发送的校时命令, 实现与整个分布式系统的时钟统一。检测系统使用独立的时钟, 接收各节点检测数

据并进行延时补偿, 将各节点数据统一到检测系统时钟。本节点需要发送的检测数据包括诸元检测数据以及随动系统检测数据 (方位/高低系统检测数据), 其中, 诸元检测数据由主控模块通过计算得到, 随动系统检测数据通过 CAN 模块实时采集得到。

2 基本原理简介

2.1 实时钟运行原理

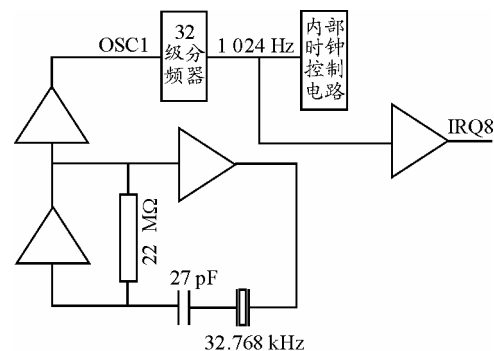


图 2 RTC 运行原理

从 AT 机开始, PC 机中配置了 RTC 芯片, 通常采用 Motorola 公司的 MC146818 芯片, 在允许周期中断的条件下, 32 级分频器输出有效, 作为 IRQ 信号经反相器送到 PC 机中断系统 8259A 从片的 IR0, 即实时钟中断请求 IRQ8, 如图 2。周期性中

收稿日期: 2010-11-16; 修回日期: 2010-12-29

作者简介: 樊鹏 (1984—), 男, 湖北人, 工学硕士, 从事火力控制系统研究。

断发生频率的可编程范围是 2~8 192 Hz。因此，能使 IRQ8 和 RTC 周期中断，并设置周期中断频率，即可向 CPU 申请相应频率的外部中断。

2.2 PC104 串口通信

PC/AT 兼容的串行端口一般由通用异步接收发送器 (UART) 控制，典型的 UART 芯片通常使用 INS8250 系列芯片。通过对 8250 的编程，可以实现特定格式的串行通信。另外，8250 芯片具有很强的中断管理能力，按优先级管理 4 种中断：接收数据错中断、接收缓冲器满中断、发送保持器空中断和 MODEM 状态改变中断，可以通过中断方式处理数据来提高性能。

PC104 平台 SWM-Mpvl 板载 16550 兼容的串行通信口。UART 芯片 16550 兼容 8250 系列，同时还具有 FIFO 控制器，可以使用 FIFO 进行中断发送和接收，避免 CPU 循环查询 UART 芯片状态，进一步提高了软件效率。为了使用中断发送功能，初始化时禁止发送中断，当有数据需要发送时使能发送中断；在中断服务程序 (ISR) 中向 FIFO 中一次写入 1~16 字节，数据发送完成后禁止中断。

3 设计方案

3.1 工作原理

检测数据实时发送的基本原理是 RTC 模块维持系统时钟，数据生成模块准备检测数据并记录数据产生时刻，数据发送模块读取发送时刻时钟计算数据延迟量并发送数据，如图 3。

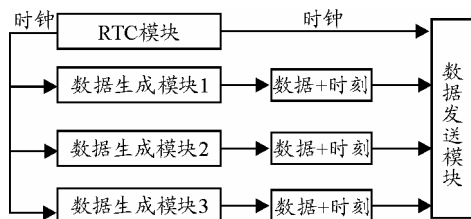


图 3 基本工作原理

当某帧数据正在被发送时，即使发送 FIFO 可写，也不能再写入下一帧数据，因为其发送时刻不确定。利用发送中断的产生原理，在 ISR 中进行数据发送，当发送完成后产生一次发送保持器中断即可发送下一帧数据。因此，设计合理的中断控制方式是实现多帧数据实时发送的关键。

3.2 软件设计

3.2.1 实时钟中断挂接及中断处理

RTC 基本 I/O 端口是 0x70~0x71，0x70 被用

作 RTC 芯片内部寄存器的地址索引端口，而端口 0x71 则被用作 RTC 芯片内部寄存器的数据端口。在读写一个 RTC 寄存器之前，必须先把该寄存器在 RTC 芯片内部的地址索引值写到端口 0x70 中。因此，读写一个 RTC 寄存器的宏定义如下：

```

#define RTC_READ(addr) ({
    outportb((addr),0x70);
    inportb(0x71);
})
#define RTC_WRITE(val, addr) ({
    outportb ((addr), 0x70);
    outportb ((val), 0x71);
})
  
```

RTC 芯片具有 A、B、C、D 等 4 个控制寄存器。其中，寄存器 A 控制计时的基频和输出频率；寄存器 B 控制 RTC 的工作方式；寄存器 C 是一系列的标志，反映了芯片向 CPU 申请中断的情况；D 是 RAM 有效位的标记寄存器。要控制和接管实时钟，要了解寄存器 A、B 中相关位的含义，见表 1、表 2。

表 1 状态寄存器 A 的位定义

D7	D6	D5	D4	D3	D2	D1	D0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

表 2 状态寄存器 B 的位定义

D7	D6	D5	D4	D3	D2	D1	D0
TE	PIE	AIE	UIE	SQWE	DM	M24	DSE

表 1 中，UIP 为周期更新标志，DV2~DV0 为选择 22 级分频器的输入基准频率，系统初始化为 010，即 32 768 Hz；RS3~RS0 为选择 32 级分频器的输出信号频率，系统初始化为 0110，即 1 024 Hz；

表 1 中，PIE 为周期中断允许位，PIE=1 表示允许周期中断，PIE=0 表示禁止周期中断。

1) 实时钟中断挂接

首先，使能 RTC 周期中断，必须将 RTC 寄存器 B 的 PIE 位置 1。然后使能 IRQ8，由于 IRQ8 由 8259A 从片控制，因此还必须使能 IRQ2。

2) 实时钟中断处理

实时钟中断频率为 1 024 Hz，主要功能是维持系统时钟 count，每中断一次产生一个时钟滴答 (TICK)，系统时钟 count 增 1。在 ISR 中必须读 RTC 的寄存器 C，否则 RTC 将不再产生下一次中断。由于 RTC 中断由 8259A 从片控制，因此，中断结束时必须向 2 片 8259A 发送中断结束信号 (EOI)。主要代码为：

```

void interrupt new_int70(...){
    BYTE reg1;
  
```

```

count++;
reg1 = inportb(0x70);
RTC_READ(0xb);
outportb(0x70, reg1);
outportb(0x20, EOI);
outportb(0x0a0, EOI);}

```

3.2.2 串口中断挂接及中断处理

1) 串口中断挂接

串口中断挂接时, 禁止 UART 中断, 但要使能串口相应的 IRQ (如 IRQ4)。另外, 必须使能 UART 寄存器 MCR 中的 OUT2 位, 否则 UART 芯片的中断不能向 8259A 发送中断信号。由于使用了发送 FIFO, 还必须将寄存器 FCR 的 BIT0 置 1 以发送 FIFO, 其流程图如图 4。

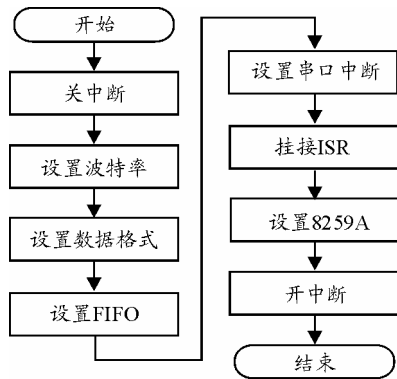


图 4 挂接串口中断

2) 串口中断处理

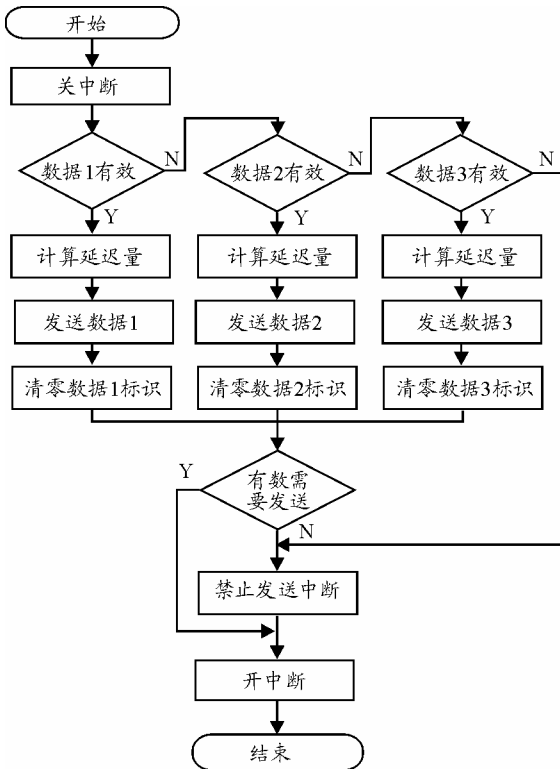


图 5 串口中断处理

串口 ISR 主要完成数据的发送和中断禁止控制。在 ISR 中, 查询数据标识是否有效, 数据有效时读取当前时钟, 并根据数据时刻计算延迟量; 将数据和延迟量按帧格式写入发送 FIFO; 最后判断是否还有数据需要发送, 无数据需要发送时禁止中断。其流程如图 5。

3.2.3 检测数据生成

完整的检测数据包括数据以及数据时刻, 其数据结构定义为:

```

typedef struct{
    INT16 flag; // 数据有效标识
    UINT32 rt; // 数据有效时刻
    BYTE data[DATALEN];
}DATA_STRUCT;

```

由于系统时钟是 32 位数据, 对于 16 位系统, 需要两条指令才能读取到系统时钟, 因此, 在主程序中记录数据时刻时需进行关中断处理, 否则可能造成时钟读取错误。另外, 由于数据是在 ISR 中发送, 在生成数据时必须在数据、数据时刻记录好之后再置数据有效标识, 否则多帧数据处理时可能造成数据错乱。在其他 ISR 中 (如本系统的 CAN 模块) 生成数据时, 需要根据是否允许中断嵌套决定是否进行以上处理。

完成以上主要模块设计后, 即可在主程序中循环查询数据标识, 当数据有效时使能串口发送中断。CPU 将响应中断, 执行串口 ISR 完成数据发送。

4 结论

该方法已在某火控系统中得到实际应用, 实现了多帧异步数据的实时发送。通过计算数据延迟量, 使检测系统能推算精确的数据有效时刻, 将各节点的检测数据在时间上严格对齐, 对分析火控系统精度具有重要意义。

参考文献:

- [1] 冯清华, 史文浩, 刘鸿雁. 基于 CMOS 时钟的实时控制[J]. 西安理工大学学报, 2004, 20(2): 206-209.
- [2] 田振清, 李改梅. 用 PC 机 RT_CMOS 周期中断实现定时间隔数据采集[J]. 内蒙古师大学报, 2001, 30(3): 228-232.
- [3] 尹彦芝. C 语言高级实用教程[M]. 北京: 清华大学出版社, 1992: 335-338.
- [4] 马兴, 童卓, 周丽娟. 基于 VB 的 PC 机与单片机间串口通讯及程序设计[J]. 兵工自动化, 2010, 29(10): 94-96.
- [5] 高成强, 王汉功, 刘小方. 基于 USB 接口的超声检测数据通讯[J]. 兵工自动化, 2010, 29(1): 85-86.