

一种高效率的定时器管理模块设计*

陈晓炜, 石江宏

(厦门大学 通信工程系, 厦门 361005)

摘要: 首先说明常见定时器管理方案的优缺点, 然后针对通信协议的特点, 设计了一种高效率的定时器管理模块。该模块通过使用一个辅助定时器, 大大减少了定时器中断处理的时间开销。

关键词: 软件定时器; 相对定时器; 定时器管理; 通信协议

中图分类号: TP336 文献标识码: A

An Efficient Timer Management Module

Chen Xiaowei, Shi Jianghong

(Department of Communication Engineering, Xiamen University, Xiamen 361005, China)

Abstract: The advantages and disadvantages of common timer management programs are described. Based on the characteristics of communication protocol stack, an efficient timer management module is designed. This module greatly reduces the interrupt overhead time by the use of an auxiliary timer.

Key words: software timer; relative timer; timer management; communication protocol stack

引言

定时器是通信协议正常运行的基本要素之一, 主要用于各种定时和帧重传的任务。通信协议在单片机系统上实现所使用的定时器, 定时精度要求不高, 但数量要求比较大。由于硬件资源有限, 不可能为每一个单独任务分配一个硬件定时器, 只能通过单个硬件定时器模拟多个软件定时器的方法, 来满足协议中的定时应用需要。

用一定的数据结构将这些软件定时器组织起来, 并提供统一的调用接口, 称为“定时器管理”。目前定时器管理主要有 2 种实现方法:

静态数组法。将定时器节点存储在数组中。优点是逻辑简单, 占用 ROM 较少。但这种方案有明显的缺点: 当硬件定时器中断发生时, 要对所有定时器节点进行减法操作, 时间开销很大, 且时延不确定(与定时器数目相关)。

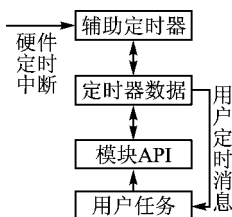
delta 链表法。按照定时器的定时值升序排列, 形成链表。后一个定时器的定时值是前面所有定时器的值加上本节点的值。这样, 在每次的时钟中断处理中, 只需对第 1 个定时器节点进行减法操作, 大大减少了时间开

销。但是, 该方案逻辑复杂, ROM 用量大, 需要频繁分配回收内存, 容易形成内存碎片。

1 定时器管理模块的设计

定时器管理模块的设计基于静态数组法。使用一个定时器节点数组来保存所有的定时请求, 数组的每一项代表一个可用的定时器节点。每一个定时器节点都有一个状态项, 表示该定时器正处于空闲、使用或溢出状态。定时器的定时值和定时器超时要发送的消息也存储在定时器节点中, 从而实现用一个硬件定时器为用户提供多个软件定时器。

为了解决中断处理时间开销大的问题, 在模块中引入一个辅助定时器, 辅助定时器的值总是等于所有定时器节点中的最小定时值。在硬件定时器中断处理中, 仅对辅助定时器进行减法操作, 从而大大缩短了中断处理的时间。设计原理如图 1 所示。图 1 定时器管理模块



2 定时器管理模块的实现

2.1 数据结构和函数接口

定时器管理模块使用的相关数据结构定义如下:

```
enum TIMER_STATE { T_FREE, T_INUSE, T_OVERFLOW
```

* 福建省自然科学基金项目(A0710021), 重点科技计划项目(2008H0037)。

```
};
struct TIMER {
    enum TIMER_STATE state;
    uint16 count; // 计数值
    MESSAGE *msg; // 指向定时时间结束时要发送的消息
};
static struct TIMER T[MAX_TIMER_NUM];
static uint16 current_cnt; // 辅助定时器
```

字段 state 保存了定时器节点的状态,可能取值为空闲(T_FREE)、使用(T_INUSE)或溢出(T_OVERFLOW)。

字段 count 保存了定时器节点的定时值,最大取值为 65 535。如果设置硬件定时器中断为 10 ms,则软件定时器最大定时约为 655 s,可以满足大多数应用需要。

字段 msg 指向定时器的用户消息。在启动定时器时,指向消息的指针被保存在此字段。当定时时间结束后,中断处理函数会自动发出这个消息以通知用户任务。

由于数组的下标是唯一的,可作为识别节点的唯一 ID 号。下文中提到的节点 ID 号均表示节点在数组中的下标。

MAX_TIMER_NUM 表示系统允许的最大定时器数,其值取决于具体应用需要。

本模块提供的关键接口函数如下:

```
void Init Timer(); // 定时器初始化
uint8 Set Timer(uint16 count, MESSAGE *msg); // 启动一个定时器
void Clear Timer(uint8 id); // 删除一个定时器
void Timer Tick(); // 定时器的驱动
```

2.2 定时器的初始化

使用定时器管理模块前,需要进行定时器的初始化。主要是初始化定时器节点数组,将每一个定时器节点设置为空闲状态,同时将辅助定时器置零,辅助 ID 指向 0xFF (表示空)。

2.3 定时器的启动

启动一个定时器,主要是将节点数组中一个空闲状态的节点置为使用状态。如果这个新启用的定时器,是所有定时器中定时值最小的,还要更新辅助定时器。函数以指向定时器消息的指针和定时值为参数,启动定时器流程如图 2 所示。成功启动定时器后,返回该定时器节点的 ID 号。

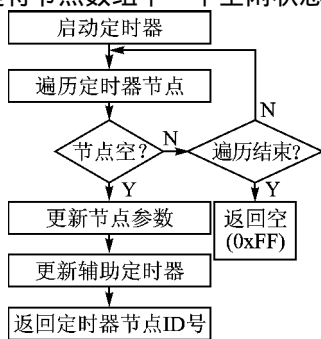


图 2 启动定时器流程

2.4 定时器的删除

在目标定时器到期之前,由于某种原因用户可能会要求取消定时器。如重发定时器,用户在发送数据帧后启动该定时器,并等待对方返回响应帧。如果在定时时间结束时没有收到响应帧,用户就需要重发原数据帧;如果在定时时间结束之前收到响应帧,用户就需要马上取消该定时器,然后进行下一次通信过程。

删除定时器函数以定时器节点 ID 号作为输入参数,将定时器节点设为空闲状态,并根据需要更新辅助定时器,流程如图 3 所示。

2.5 定时器的驱动

软件定时器的驱动由硬件定时器提供。在硬件定时器中断中,首先将辅助定时器的值减 1。如果辅助定时器值为 0,则表示定时值最小的定时器已经超时,应将对应的消息发送给用户任务,将节点置为空闲状态,并重新计算其他节点的定时值,同时查找定时值最小的节点,更新辅助定时器。驱动流程如图 4 所示。

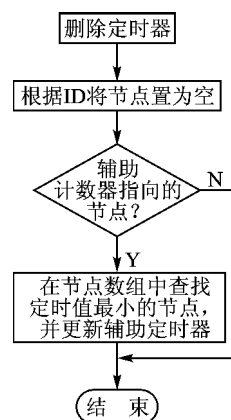


图 3 删除定时器流程

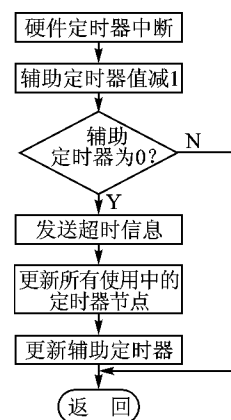


图 4 辅助定时器驱动流程

关键代码如下:

```
void Timer Tick() {
    if(current_cnt == 0) return;
    // 辅助定时器值为 0,表明定时器全部处于空闲状态
    current_cnt--;
    if(current_cnt == 0){ // 判断辅助定时器是否超时
        uint8 i;
        ..... // 发送用户定时消息
        current_cnt = 0xFFFF;
        for(i = 0; i < MAX_TIMER_NUM; i++){
            // 查找余下定时器中定时时间最短的
            if(T[i].state == TIMER_INUSE){
                T[i].count -= T[current_id].count;
                if(T[i].count == 0){
                    // 判断该定时器是否超时
                }
            }
        }
    }
}
```

得 ns 级的时钟同步成为可能。

DP83640 是一款基于 IEEE 1588 标准的时钟同步芯片,采用硬件和软件结合的方式提供最高的精确度实时工业的时钟同步,可确保分布式上各节点能按照主机时钟的时间同步定时,并确保各节点之间的时间偏差不会超过 8 ns。一旦线路上有 PTP 包,即被 DP83640 的精密 PHYTER 所读取。

DP83640 具有几个内部时钟,包括本地参考时钟、1 个以太网接收时钟和 1 个 PTP 时钟信号源;同时,还包括 1 个内部的 PTP 数字计数器,以及可以控制数字计数器和 PTP 时钟速率(频率)的逻辑。

在同步以太网交换机的方案中,通过替换以太网层并增加 IEEE 1588 PTP 软件实现。如图 4 所示,CPU、交换芯片和 DP83640 通过 MII/RMII 连接起来组成一个系统。交换机成为以太网中同步的一个器件,使得交换机所形成的以太网及该网络下所挂的器件都满足 IEEE 1588 协议,最终形成同步以太网。

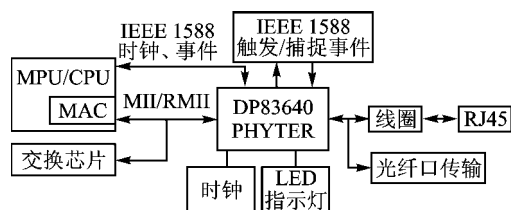


图 4 基于 DP83640 的同步以太网交换机应用框图

结 语

从目前的原型实验和应用来看,IEEE 1588 中标准化的精确时间协议可以达到亚微秒级的同步精度,并且有可能达到更高的精度。IEEE 1588 为基于多播技术的标准以太网的实时应用提供了有效的解决方案,但同时也存在一些尚待进一步研究的问题,如主时钟的容错性能、振荡器的稳定性对时钟的影响等。相信今后该标准会更加完善,也会有更多的具体应用可以参考。

编者注:本文为期刊缩略版,全文见本刊网站 www.mesnet.com.cn。

参考文献

- [1] IEEE Std. 1588—2002 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems[S].
- [2] Kendal R, Sivaram B, Anatoly M. The Application of IEEE 1588 to a Distributed Motion Control System[OL]. [2009-08]. <http://iee1588.nist.gov/>.
- [3] Roes F E, Vaman D R. IsoEthernet: An Integrated Services LAN[J]. IEEE Communications Magazine, 1996, 34(8).
- [4] 韦乐平. 电信级以太网的崛起[OL]. [2009-08]. <http://www.Chinatelecom.com.cn>.
- [5] 胡昌军. 电信级以太网技术漫谈[OL]. [2009-08]. <http://cartr.com.cn>.

(收稿日期:2009-08-31)

31

```

..... //发送用户定时消息
continue;
}
if(T[i].count < current_cnt){
    current_cnt = T[i].count;
}
}
}
if (current_cnt == 0xFF)
    current_cnt = 0;
//余下的定时器都是空闲的
}
}

```

结 语

本文设计的定时器管理器模块,具有逻辑简单、效率高的特点,在实验室自主研发的无线传感器网络中

得到应用。在共包含 30 个节点的温、湿、光信息采集系统中,使用该定时器管理模块的无线传感器网络协议运行高效、稳定。

参考文献

- [1] 张丽云,周洪利. 一种嵌入式时钟管理器的设计与实现[J]. 电子技术应用,2006,32(8).
- [2] 李光. 大型有限状态机系统中的定时器设计[J]. 无线电工程,2005,35(6).
- [3] 于伟,杜新华. 通信设备中相对定时器管理方法[J]. 电子测量技术,2006,29(3).
- [4] 严蔚敏. 数据结构(C语言版)[M]. 北京:清华大学出版社,2006.

陈晓伟(硕士),主要研究方向为无线传感器网络;石江宏(副教授),主要研究方向为无线网络技术。

(收稿日期:2009-09-27)