

一线总线器件异步读写实现方法

Implementing Method of Asynchronous Read-write for 1-wire Device

于志豪¹ 常 龙¹ 肖林京¹ 张瑞雷¹ 魏瑞托²

(山东科技大学机械电子工程学院¹, 山东 青岛 266590; 山东科技大学信息与电气工程学院², 山东 青岛 266590)

摘要: 针对一线总线读写时序过多占用微控制器(MCU)运算资源的问题,对一线总线器件异步读写方法进行了研究。以MCU作为一线总线器件的主控器,通过一个定时器的四个中断实现一线总线器件的读写时序。在进行一线总线数据读写前,MCU程序将读写内容译成相应的时序信息后以指令队列的形式存储。在一线总线读写过程中,定时器中断程序按事先存储于指令队列中的时序信息顺序修改定时器的计数值,从而完成一线总线的异步读写。该方法将主要的延时工作交由定时器实现,最大程度地减少了CPU的占用率。

关键词: 一线总线 定时器中断 异步读写 指令队列 CPU占用率

中图分类号: TP212+.9 **文献标志码:** A

Abstract: Aiming at the problem of read/write timing of 1-wire bus excessively occupies operational resources of microcontroller; the asynchronous read-write method of the 1-wire bus devices is researched. With MCU as the main controller of the 1-wire bus device, the read/write time sequence of 1-wire bus device is implemented through four interrupts of one timer. Before performing data read/write of 1-wire bus, the contents of read/write are translated into corresponding timing information, and stored as instruction queue form by MCU program. During the read/write process of 1-wire bus, the timing information sequence pre-stored in instruction queue is modified into counting value of the timer by the interrupt program in timer, thus the asynchronous read-write of 1-wire bus is completed. With this method, the major time delay is implemented by the timer, so the occupation rate of the CPU is minimized.

Keywords: 1-wire bus Timer interrupt Asynchronous read-write Instruction queue Occupation rate of CPU

0 引言

一线总线技术将地址线、数据线、控制线、电源线合为一根信号线,这根信号线上允许挂数百个使用一线总线技术的测控对象,较大规模地减少了测控设备的引线数量^[1]。一线总线结构的数字式温度传感器以其检测精度高、抗干扰力强、硬件结构简单等诸多优点得以广泛应用^[2-3]。

在基于MCU的一线总线驱动程序设计中,因一线总线器件的时序要求相对严格,大多采用CPU延时等待的方法来实现精确延时^[4-5]。这种延时属于典型的同步程序风格。在整个一线总线操作期间,CPU延时等待方法直接导致了CPU大量消耗于总线时序的处理而无法顾及其他工作^[6-7]。

本文以Arm-Cortex芯片作为一线总线器件的主控器,通过一个定时器的四个中断实现一线总线器件的读写时序。将一线总线单个数据位读写过程中的四个关键时间点作为MCU系统的中断点,在中断程序中由CPU完成总线状态位的修改,从而实现一线总线数据位的读写。该方法充分发挥了MCU系统外设的工作能力。在定时器的协助下,以极低的CPU占用率实现了一线总线的异步读写。

1 一线总线读写时序

18B20是工业现场广泛采用的数字式温度传感器。该器件采用了一线总线技术,可以大规模地减少多点温度测量系统中测温传感器的引线数量。

使用基于一线总线结构的数字式温度传感器DS18B20测量温度,特别是在多点测量领域,必须首先确定总线上所有器件的64位的序列号^[8]。典型的温度读取方法是通过搜索算法,查找出所有总线上的器件的64位ID号,并固化于MCU内。在温度读取过程中,一线总线控制器先向总线上所有器件下发温度转化指令,然后控制器通过指令逐个锁定总线器件,并读取各个传感器的温度数据^[9]。

山东省科技发展计划基金资助项目(编号:2012GSF11606);

山东省教育厅课题基金资助项目(编号:JI0LG17);

青岛市关键技术攻关计划基金资助项目(编号:12-4-1-31-gx)。

修改稿收到日期:2013-04-28。

第一作者于志豪(1978-),男,2004年毕业于山东大学电机与电器专业,获硕士学位,讲师;主要从事新能源电机车控制系统及锂电池动力电源管理系统的研究。

准确完成单只 18B20 的温度读取应经过如下几步操作:①主机发送复位指令;②主机发送 MachRom 指令;③主机发送 64 位 ID 号;④主机发送温度转化指令;⑤延时 500 ms 以上;⑥主机发送复位指令;⑦主机发送 MachRom 指令;⑧主机发送 64 位 ID 号;⑨主机发送读取温度指令;⑩主机读取温度。

如果总线上存在多个 18B20,则上述工作可简化为:①主机发送复位指令;②主机发送 SkipRom 指令;③主机发温度转化指令;④延时 500 ms 以上;⑤主机发送复位指令;⑥主机发送 MachRomN 指令;⑦主机发送 64 位 ID 号;⑧主机发送读取温度指令;⑨主机读取温度 N ,直到总线上所有器件温度读取完成^[9]。

一线总线主机读 0 时序如图 1 所示。

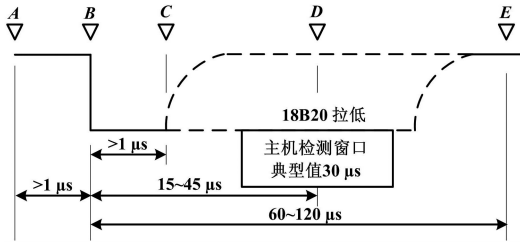


图 1 一线总线主机读 0 时序

Fig. 1 Time sequence of read "0" in host computer

总线完成上述操作所需时间为: $2.6 + 500 + (1 + 96 \times 0.1)N$ ms。取 $N = 100$ 可知,当总线上存在 100 只 18B20 时,完成一次总线温度读取用时应不小于 1 562.6 ms。无论采用何种算法,该时间消耗是确定的,若要想全部读取总线上的 100 只器件的温度值,用时必须大于 1 562.6 ms。对于基于 MCU 的测控系统来说,如果全部使用空指令实现这些延时,将会对 CPU 资源产生极大消耗并最终导致 MCU 无法完成其他任务。

2 一线总线异步读写方法

2.1 时序分析

以一线总线主机从总线读取一位数据为例,其时序示意图如图 2 所示。

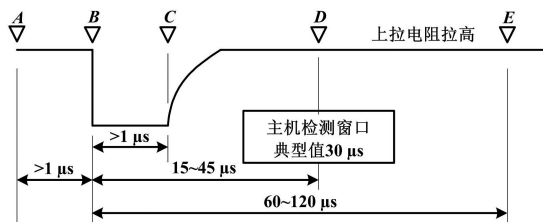


图 2 主机读一位数据操作时序

Fig. 2 Time sequence of read one data bit in host computer

由图 2 可以看出,从 A 点开始主机释放总线,总线电平被上拉电阻自然拉至高电平。间隔 1 μs 后,从 B 点开始主机拉低总线并持续大于 1 μs 的时间,到 C 点主机释放总线。此后一线总线设备响应总线时序向总线写入 0 或 1,主机会在由 B 点开始后的 15 ~ 45 μs 的时间窗口内读取总线上的数据,并认为从 B 点开始后 60 μs 以后的 E 点为主机完成总线数据的读取。主机读取并存储当前数据位后进入下一位数据位的读取。

上述工作(主机读取一位数据)如交由 MCU 处理,可简要描述如下:A 点,计时开始;B 点拉低总线;C 点,释放总线;D 点,读取总线数据;E 点,标记当前位读取完成。

该工作的简要描绘如表 1 所示。表 1 同时也给出了主机向总线写 0 或写 1 的操作时序。由表 1 可见,一线总线的读写过程中主机的操作内容略有不同,但时序有很强的共性。完成总线数据读取只需 MCU 在 B、C、D、E 四个时间点上对总线进行相应操作即可,且各个关键时间点上的 MCU 工作极其简单,完全可以放在中断程序中直接实现,且不会对其他程序的运行产生明显影响。

表 1 主机读写总线时序与 CPU 的主要工作

Tab. 1 Main tasks of CPU in the time sequence of bus read/write in host computer

位置	$t/\mu\text{s}$	主机读 0 或 1	主机写 1	主机写 0
A 点	0	计时开始	计时开始	计时开始
B 点	1	拉低总线	拉低总线	拉低总线
C 点	5	释放总线	释放总线	释放总线
D 点	30	读取数据	无动作	无动作
E 点	70	标记完成	标记完成	标记完成

总线的复位操作时序要求与总线读 0 操作过程非常相似,不同之处仅在于操作时间较长。

总线复位的主要时序与主机应完成的工作如图 3 和表 2 所示。

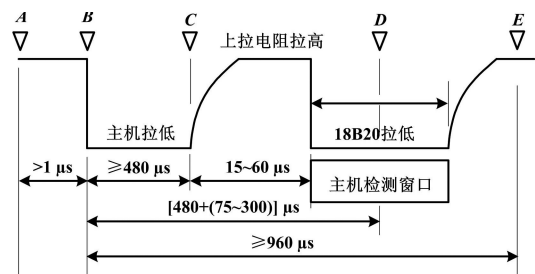


图 3 主机复位时序图

Fig. 3 The reset time sequence of host computer

表 2 主机复位时序与 CPU 的主要工作

Tab.2 Main tasks of CPU in reset time sequence of host computer

位置	时间/ μs	主机复位
A 点	0	计时开始
B 点	1	拉低总线
C 点	500	释放总线
D 点	700	读取数据
E 点	1 000	标记完成

2.2 异步读写实现

由上述分析可知,无论是复位或是总线读写,均可以将总线操作化成四个关键时间点的 CPU 动作,且每个时间点上工作都极为简单,可以在中断程序中直接实现。

本文以 ST 公司的 Arm-Cortex 芯片 STM32F103RCT6 为例,实现上述一线总线的操作方法。

STM32F103RCT6 提供了两个高级控制定时器,每个高级控制定时器提供了四个独立的输出比较通道,每个输出比较通道均可以产生中断。

表 1、表 2 所示的主要时序可以通过如图 4 所示的高级定时器的四个输出比较通道实现。以表 1 所示的主机读取一个总线数据位为例。程序初始化时将定时器时钟的时基配置为 $1\ \mu\text{s}$,即计数器的时间与以微秒为单位的计时同步。A 点,程序将四个输出比较通道输出比较寄存器值预设为 1、5、30、70(与表 1 中的 B、C、D、E 四个时间点对应),计数器清 0 后开启各中断,启动定时器。随后即会顺次发生图 4 所示的 B、C、D、E 四个时间点所对应的中断,在每个中断的中断程序中完成表 1 所示的对应工作即可。

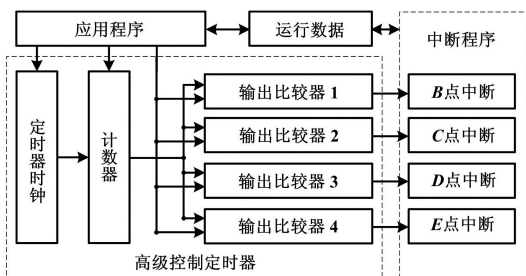


图 4 高级控制定时器的四个输出比较通道

Fig.4 Four comparison channels of the advanced control timer

2.3 总线数据处理

上节所述的一线总线异步数据读定方法仅读写了一线总线上的一个数据位,实际的一线总线操作多以字节(8 个数据位)为一个操作单元,只有复位操作可以认为是一个数据位的操作。一线总线操作的典型数据如表 3 所示。

表 3 一线总线操作的典型数据

Tab.3 Typical data of 1-Wire bus operation

操作类型	主机任务	实现形式
复位操作	复位总线	读 1 位
搜索操作	总线读写 1 位	读或写 1 位
总线指令	写 1 字节	写 8 位
读取数据	读 1 字节	读 8 位

表 3 中,复位操作和搜索操作可以按读写 1 位总线数据位的形式以 2.2 节所讨论的方法实现。总线指令和读取数据两种操作是以连续读写 8 个数据位(1 字节)的形式进行的。总结表 3 所有可能会出现的一线总线操作数据类型,可给出如下数据结构。

```

Struct BusDataUnit {
    Int bitLength;           //读写数据的位数
    Bool bRW;               //指令是读或是写
    Byte Result;           //读取的结果
    Byte data;             //写入总线的数据
    Byte presentBit;      //当前读写的数据位
}
    
```

无论是复位,还是总线指令读写都可用上述数据结构的形式进行表达。例如,主机读取总线上的某一传感器的温度转化数据,其操作应包含如下顺序执行的 5 个步骤:①主机发送复位指令;②主机发送 MachRom 指令;③主机发送 64 位 ID 号;④主机发送读取温度指令;⑤主机读取 16 位温度。

这 5 步操作中,64 位的 ID 号和 16 位的温度数据会因传感器的不同而变化,其他操作中的数据相对固定。将这 5 步操作以 BusDataUnit 的形式进行表达,并以指令队列的形式进行存储,得到的指令队例如图 5 所示。此后用户程序可将该指令队列交由定时器中断程序进行解析,当指令队列中的所有指令执行完成后,定时器中断程序产生完成标记,CPU 控制程序发现指令队列完成标记后,对数据进行进一步处理。

读温度数据	ReadRam	WriteIDBit	MachRom	复位
Length=9	Length=8	Length=64	Length=8	Length=1
ReadBus	WriteBus	WriteBus	WriteBus	ReadBus
Result=?	Result=?	Result=?	Result=?	Result=?
Data=?	Data=0xBE	Data=IDx	Data=0x55	Data=0x01
presentBit=0	presentBit=0	presentBit=0	presentBit=0	presentBit=0

图 5 读某一传感器温度值的指令队列

Fig.5 The instruction queue of reading one temperature value of sensor

指令队列为异步处理流程如图 6 所示。主程序完成的主要工作是指令队列的生成,主程序按总线操作

要求生成指令队列后即返回。中断处理程序在中断程序中顺序实现了指令队列中各个数据位的读写。当指

令队列完成后,中断程序会产生一个完成标记,主程序在收到该完成标记后认为总线读写完成。

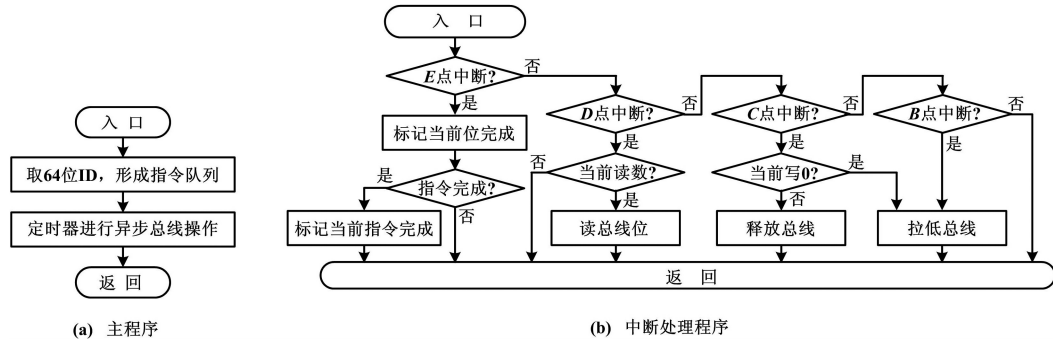


图6 指令队列的异步处理流程

Fig.6 Asynchronous processing of instruction queue

3 试验过程

试验系统采用了 ST 公司的 Arm-Cortex 芯片 STM32F103RCT6 作为一线总线的主机^[10], MCU 主频为 72 MHz。一线总线采用了三线制(地线、电源线、数据线),总线上接入了 100 只温度传感器 DS18B20。总线的读写用 STM32F103RCT6 的 I/O 口直接驱动。一线总线的控制程序采用 Keil RealView-Arm MDK 开发。

试验测试了主机从一线总线上完整读取 100 只 18B20 的温度所对应的时间消耗。为准确获取一线总线操作所占用的 CPU 时间,主机以 Arm-Cortex 的 SysTick 定时器作为系统运行计时器。主程序及中断程序的入口和出口点记录 SysTick 的当前值作为相关程序对 CPU 时间占用的依据。主要的时间消耗如表 4 所示。

表 4 异步数据读取时间

Tab.4 Time cost for reading asynchronous data

主机工作	执行次数	CPU 占用时间/ μ s	任务用时/ μ s
总线复位与指令下达	1	37	2 665
延时 500 ms	1	24	500 047
主机读取器件数据	100	8 203	1 071 000

表 4 中,总线复位与指令下达包括的主机操作包括了总线复位、主机发 SkipRom 指令和主机发温度转化指令三种顺序执行的主要操作。主机读取器件数据包括的主机操作包括了主机发送复位指令、主机发 MachRom 指令、主机发 64 位 ID 号、主机发读取温度指令和主机读取温度 5 种顺序执行的主要操作。另外,表 4 所给出的数据未能将 SysTick 中断程序对 CPU 消耗计算在内,所得出的数据较之于实际数据偏大。由表 4 可见,一线总线完成 100 只 18B20 温度转化及数据读取用时约

1.6 s。采用异步数据读写程序,上述工作对 CPU 的时间占用仅 8.26 ms,CPU 的占用率不到 0.6%。

4 结束语

本文以 MCU 作为一线总线主机,实现了一线总线器件异步读写方法。在分析了一线总线主要工作时序共性的基础上,给出了实现一线总线时序的基本程序结构,进而形成了一线总线的异步数据读写算法,并给出了主要的程序实现思路。以 STM32F103RCT6 作为一线主机,实现了一线总线的异步数据读写方法。试验数据表明,该方法可有效降低一线总线读写操作对 CPU 的占用率,提高 MCU 控制系统的实时性。

参考文献

- [1] 李敏,汪春,孟臣. 数字化粮情检测智能温湿度传感器[J]. 农业机械学报,2005,36(4):99-102.
- [2] 姜幼卿,左国华,陈勤学,等. 基于 CAN 总线的温室自动控制系统的通信设计[J]. 仪表技术与传感器,2005(4).
- [3] 陈汉臻,王爱美,罗进文,等. CC2430 芯片和单片机的暖气控制系统设计与实现[J]. 自动化仪表,2012,33(9):17-20.
- [4] Huang W L, He L L. Study on timing sequences of interface program of digital temperature sensor DS18B20 [J]. Lecture Notes in Electrical Engineering,2012,13(8):1329-1334.
- [5] 张晓玲,梁伟,于海斌,等. 无线传感器网络传输调度方法综述[J]. 通信学报,2012,33(5):143-157.
- [6] 魏学哲,戴海峰,孙泽昌. 汽车嵌入式系统开发方法、体系架构和流程[J]. 同济大学学报:自然科学版,2012,40(7):1064-1070.
- [7] 刘啸滨,郭兵,沈艳,等. 基于 ARM 处理器的嵌入式软件能耗统计模型[J]. 电子科技大学学报:工学版,2012,41(5):770-774.
- [8] 鲁正林,陈一飞,孙倩,等. 1-wire 技术在温室集成控制系统中的应用[J]. 农业机械学报,2009,40(S1):119-222.
- [9] 陈良光. DS18X20 在多点温度检测中的地址码设计及其在线纠错[J]. 传感技术学报,2005,18(3):157-161.
- [10] 黄家才,李宏胜,郭婧. 嵌入式油田无线远程监控系统的研究和设计[J]. 自动化仪表,2011,32(11):46-48.