

基于 Keil C51 的软件看门狗系统设计

范立南¹, 李震^{1,2}, 周胜来¹, 李荃高¹

(1. 沈阳大学 信息工程学院, 辽宁 沈阳 110044;

2. 东北大学 信息科学与工程学院, 辽宁 沈阳 110006)

摘要:介绍了一种基于 Keil C51 环境的软件看门狗设计方法。通过设置程序、喂狗程序和中断服务程序的有效配合, 解决软件模块的稳定性问题。设置程序在模块开始处对看门狗系统进行初值设定, 喂狗程序在模块结束时进行参数设置。通过 T0 中断服务程序监视软件模块的运行, T1 中断服务程序监视 T0 定时器的运行, 设置程序监视 T0 和 T1 的运行, 以达到看门狗系统的循环监视。此设计方法可以提高程序模块的稳定性, 特别是对于顺序控制结构的软件尤为适用。

关键词:软件模块; 看门狗; Keil C51

中图分类号: TP273+.5

文献标识码: A

文章编号: 1673-629X(2009)01-0154-03

Soft - Watchdog System Design Based on Keil C51

FAN Li-nan¹, LI Zhen^{1,2}, ZHOU Sheng-lai¹, LI Quan-gao¹

(1. School of Information Engineering, Shenyang University, Shenyang 110044, China;

2. School of Information Science & Engineering, Northeastern University, Shenyang 110006, China)

Abstract: Presents a kind of soft - watchdog based on Keil C51. The efficient cooperation between the setting program, dog - food program and interrupting serve program solves the problem of stability of software modules. Setting program is designed for setting initial value. Dog - food program is designed for setting parameters when software module runs to end. T0 interrupting serve program monitors software module, T1 interrupting serve program monitors T0 counter, and setting program monitors T0 and T1. This cycling monitoring system can enhance the stability of software modules, especially for the software which has a sequential structure.

Key words: software module; watch dog; Keil C51

0 引言

单片机在工业生产中已经被广泛应用,但由于其工作现场常常受到大功率器件干扰、电磁干扰的影响,则单片机系统容易出现程序运行紊乱甚至死机现象^[1]。针对这种情况,现在一般采用硬件看门狗和软件看门狗两种方案。

硬件看门狗应用简单,占用软件资源少,但其复位将使程序从头开始运行。此种做法不适用于那些不太重要但受干扰机会较多的场合^[2]。对于 C 语言所编写的程序,跳转指令 goto 只允许在本函数体内应用,这就影响了软件看门狗的使用。还有一种情况,如果由于干扰使定时器停止运行,那么中断响应程序就不能对程序进行正确的引导,看门狗系统就会失效^[3]。

这里提出如下问题:如何在 C 语言开发环境下有效地使用看门狗程序;如何在定时器失灵的情况下,保证看门狗的正确引导。

针对以上两个问题,笔者提供了一种基于 Keil C51 环境的全新的软件看门狗方法。该方法应用到了笔者开发的智能车设计中,成功地解决了由于电机电流扰动产生的电机控制模块失灵的问题。

1 看门狗技术

第 1 类,硬件看门狗技术,如图 1 所示。

通常的硬件看门狗技术是应用专用的看门狗芯片如 MAX691、MAX6316LUK29CY - T、TPS3823、MAX813L、W78E51B、X25045、X5043/X5045 等等。硬件看门狗是一些集成化的专用看门狗电路,它实际上是一个特殊的定时器,当定时时间到时,发出溢出脉冲^[4]。从实现角度上看,该方式是一种软件与片外专用电路相结合的技术,硬件电路连接好以后,在程序中

收稿日期:2008-05-28

基金项目:辽宁省教育科技基金项目(20060573)

作者简介:范立南(1964-),男,辽宁北票人,博士,教授,硕士生导师,研究方向为人工智能、智能仪器仪表、图像工程等。

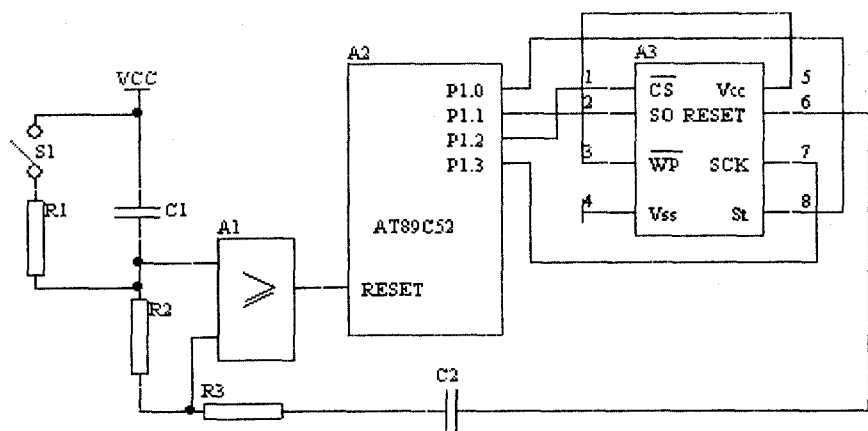


图1 硬件看门狗

适当地插入一些看门狗复位的指令,即“喂狗”指令,保证程序正常运行时看门狗不溢出;而当程序运行异常时,看门狗超时发出溢出脉冲,通过单片机的 RESET 引脚使单片机复位。

第2类是软件看门狗技术。

现有的软件看门狗技术通常是采用系统自带的定时器进行软件复位^[5]。传统的软件看门狗大都是针对汇编语言来实现的,由于汇编语言便捷多样的跳转指令,使软件看门狗的编辑变得比较灵活。通过定时器的溢出,在中断子程序中设置跳转指令,跳转到错误处理程序,从而实现了对跑飞程序的正确引导^[6]。

2 总体方案

由于硬件看门狗和基于汇编语言的软件看门狗存在一定的缺陷,文中提供了一种在 Keil C51 环境下针对 C 语言软件模块的新型看门狗设计方法。Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统,与汇编语言相比,C 语言在功能上、结构性、可读性、可维护性上有明显的优势。设计的主程序流程图如图2所示。在程序模块的初始位置对看门狗系统进行初值设定。在程序结束时进行喂狗程序设定,如果发现程序运行的时间超出规定的时间,则启动看门狗系统,使程序重新复位运行。

为了解决定时器失灵的问题,在主程序中如果发现 T0、T1 运行不正常,则将 T0、T1 恢复正常。T0 主要负责对主程序的监视。T1 负责对 T0 监视,如果发现 T0 工作异常,则尝试将其恢复。T0 和 T1 中断服务程序流程图如图3和图4所示。

3 看门狗系统软件的配置方法

单片机受干扰关闭单个定时器的机会较少,关闭两个定时器的机会则更少。使 T0、T1 协同工作,使 T0

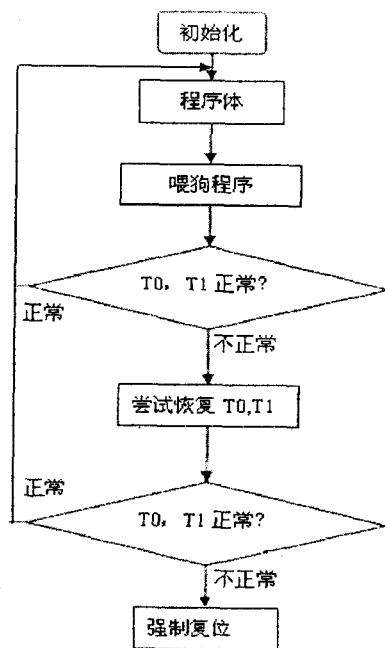


图2 主程序模块流程图

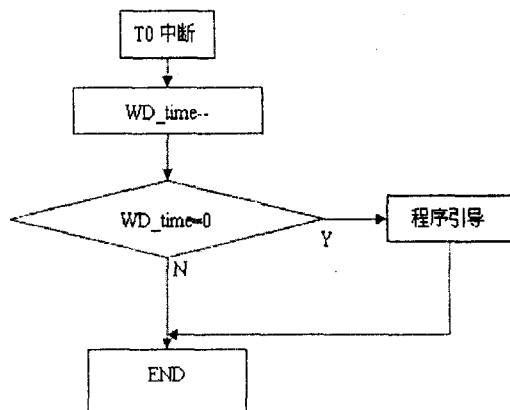


图3 T0 中断服务程序流程图

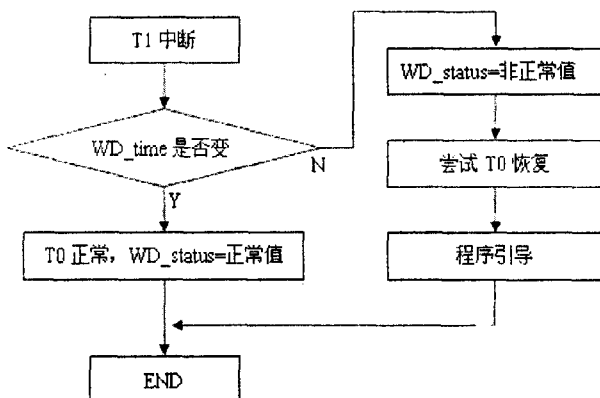


图4 T1 中断服务程序流程图

监视程序模块的运行,T1 监视 T0 的正常运行,在程序模块中插入设定程序,同时监视 T0 和 T1 的运行。这样即使 T0、T1、程序模块中的一个或者两个出现了问题,看门狗系统都可以将其进行正确的引导,而出现三者同时失控的现象是极少的^[7]。

这里先将 T1 的优先级设为最高, T0 次之。T1 设置的时间是 T0 的 1.5 倍以上。设置两个全局变量 WD_time 和 WD_status, 分别进行时间计数和看门狗状态记录。

编辑一个 T0 中断服务程序 WATCHDOG(), 进行软件模块引导; 编辑一个 T1 中断服务程序 SYS_WATCH(), 进行 T0 状态监视和 T0 失效时的软件模块引导; 编辑三个函数 WD_SET(time0, time1), WD_FOOD() 和 WD_SHUTDOWN()。其中 WD_SET(time0, time1) 和 WD_FOOD_SET() 分别为看门狗初始设置函数和看门狗喂狗函数, WD_SET(time0, time1) 函数主要是对 T0、T1 的启动和 WD_time 初值设定, 初值的设定要根据作用程序段的运行时间而定。WD_FOOD_SET() 函数为喂狗程序, 负责对 WD_time 的数值恢复和对 T0、T1 状态的监视, 在 T0、T1 失效的情况下尝试将其恢复, 如果恢复失败则进行强制复位。WD_SHUTDOWN() 函数负责关闭 T0 中断, 结束看门狗保护。

这里假设 MODULE() 函数是顺序程序中的一段。设置 WATCHDOG() 的内容如下:

```
{ WD_status + +;
.....
if(WD_time! = 0)
{ WD_time --;
  TH0 = XX;
  TL0 = XX;
  TR0 = 1;
}
else MODULE();
.....
}
```

T1 的中断服务程序 SYS_WATCH() 对 WD_status 状态进行周期检测, 如果发现异常情况, 则尝试恢复 T0 并且进行正确的程序引导。

在 MODULE() 内部首先设置看门狗 WD_SET(time0, time1), 进行时间参数设置, 这个时间要比 MODULE() 内运行的循环体单次运行的时间稍长, 大约是 1.2 到 1.5 倍。在单次循环的结束位置放入 WD_FOOD() 函数进行 WD_time 数据的恢复。在循环体外插入 WD_SHUTDOWN() 以关闭看门狗, 避免在没有干扰的时候看门狗误操作。这样如果正常运行的话, 每次在 WD_time 减到 0 之前都可以将 WD_time 重新赋值, 而 MODULE() 函数则不会被调用。一旦在循环中受到干扰 PC 指针受到影响, 程序跑飞, 则 WD_FOOD() 程序得不到调用, 在规定的时间内 WD_time 会减到 0, 从而再次调用 MODULE() 函数, 带来

一次引导, 使程序恢复正常工作。即使 T0 失效, T1 也可以进行补救, 进行正确的程序引导。如果 T0、T1 同时出现问题, 程序模块中的 WD_FOOD_SET() 程序可以进行有效的补救, 如果补救失败则进行强制复位。

这里需要注意一个问题。假设程序跑飞, 看门狗程序进行了一次引导, 那么在被引导的 MODULE() 成功结束时, 程序又会跳回跑飞的程序位置, 而不会正确地运行后续程序, 这必然造成又一次混乱。这种情况下可以进行如下处理。

假设程序为:

```
MODULE1();
MODULE2();
MODULE3();
```

要在 MODULE2() 程序中使用看门狗程序, 那么可以设计一个程序使其包含被处理程序段和它下面的所有程序。如:

```
void MODULE_ALL()
{
  MODULE2();
  MODULE2();
}
```

在 MODULE2() 程序段中使用以上所说的 WD_SET() 和 WD_FOOD_SET() 程序, 并在上述中断服务程序中将 MODULE() 替换为 MODULE_ALL(), 这样无论系统受到什么干扰, 程序总会运行到最末尾位置。如果需要循环到程序的开头位置, 可以在最末尾位置处通过 I/O 口线向 RESET 引脚发出信号, 重新复位单片机。

如果想在—个系统中使用两个不同的看门狗, 可以用两个不同的定时器, 用第三个定时器监视前两个定时器的工作。设定两套相应的看门狗设置程序、喂狗程序和看门狗关闭程序, 分别应用到不同的程序段。以 AT89C52 为例, 可以用 T1、T2 监视不同的程序模块, 而用 T0 监视 T1、T2。但要注意的是两套系统的嵌套, 避免调用时出现混乱。

4 结束语

在笔者开发的智能小车中, 应用了上述方法, 并取得了很好的效果。所设计的智能小车结构框图如图 5 所示, 通过 AT89C52 单片机实现对各个传感器数据的收集, 通过对数据的处理, 使单片机控制电机和显示模块输出。

在产品发现时发现在返程时时常出现运行紊乱, 转向系统失灵, 有时甚至发生死机现象。显然是因为返程时电机的负载加大, 而且电机正反转频繁。较

(下转第 160 页)

检验 $r = r'$ 是否成立。

以下证明 ECDSA 签名方案通过对域参数适当加以限制后在多用户环境下是安全的。

定理 4 令 E 为 F_q 上的椭圆曲线, 且 $n = \#E(F_q)$ 为素数, $n > q$, G 为 $E(F_q)$ 上阶为 n 的点。假定 F_q, E, G 和 n 是域参数, Hash 函数 H 是抗碰撞的, 那么在普通群模型中, ECDSA 在多用户环境下是安全的。

证明: 在普通群模型中, Hash 函数 H 是抗碰撞的, ECDSA 具有 GMR 安全性已有证明。在此, 只需证明该签名方案能够抵抗密钥置换攻击即可。

假定对手 E 成功地产生了不同于 A 的公开密钥 / 私有密钥对 (\bar{Q}, \bar{d}) , 使得 (r, s) 既是 A 又是 E 在消息 m 上的签名。既然两个签名都是有效的, 于是有 $x(s^{-1}H(m)G + s^{-1}rQ) \equiv x(s^{-1}H(m)G + s^{-1}r\bar{Q}) \pmod{n}$ 。由于 $n > q$, 则 F_q 中任一元素的二进制表示都小于 n , 因此, $x(s^{-1}H(m)G + s^{-1}rQ) = x(s^{-1}H(m)G + s^{-1}r\bar{Q})$, 于是有 $s^{-1}H(m)G + s^{-1}rQ = \pm(s^{-1}H(m)G + s^{-1}r\bar{Q})$ 。基于上述等式并结合私有密钥 \bar{d} , E 能够很容易确定 d 。由于求解离散对数问题尚没有有效的算法, 因此, 对手 E 就不可能产生一个置换密钥。

4 结束语

研究了签名方案的安全性现状, 分析了其中存在

的安全性隐患, 给出了签名方案在多用户环境下的安全性定义。在单用户环境下安全的签名方案能够通过公开密钥及消息摘要进行 Hash 以转换成在多用户环境下也是安全的。证明了 Schnorr, DSA, ECDSA 等签名方案通过对域参数做合理限制后在多用户环境下是安全的。

参考文献:

- [1] Bellare M, Rogaway P. Entity authentication and key distribution[C] // In Advances in Cryptology - Euro - Crypt' 93, LNCS773. Berlin: Springer - Verlag, 1993: 232 - 249.
- [2] Bellare M, Boldyreva A, Micali S. Public - key encryption in a multi - user setting: security proofs and improvements[C] // In Advances in Cryptology - Euro - Crypt' 2000, LNCS1807. Berlin: Springer - Verlag, 2000: 259 - 274.
- [3] Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361 - 396.
- [4] 王 纯. 数字签名安全性研究[J]. 电脑与电信, 2006(11): 43 - 45.
- [5] 王龙斌, 廉玉忠. 数字签名的安全性分析[J]. 信息工程大学学报, 2003(3): 90 - 92.
- [6] 卿斯汉. 安全协议[M]. 北京: 清华大学出版社, 2005: 215 - 220.
- [7] 杨 波. 现代密码学[M]. 北京: 清华大学出版社, 2003: 139 - 191.

(上接第 156 页)

大的电流冲击对单片机造成干扰, 导致程序指针赋值错误, 程序跑飞。针对这些情况, 在返程程序中嵌入了上述的软件看门狗程序, 使系统运行正常, 小车可以成功地行走目标位置, 从而解决了由于电机电流冲击引起的单片机程序跑飞现象。

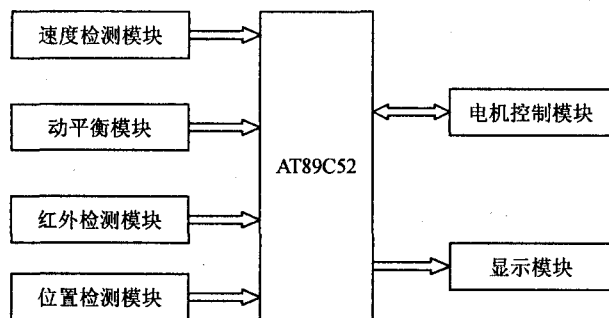


图 5 智能小车系统组成

文中介绍的方法既保证了局部程序的可靠运行, 又不必在此过程中复位单片机, 也就不必再次重复运行前面的程序段。在保证 C 语言快速开发性的同

时, 提高了程序段的可靠性。

参考文献:

- [1] 李烈彪, 李 仙. 计算机系统的可靠性技术[J]. 计算机技术与发展, 2007, 17(11): 142 - 145.
- [2] 范立南. 单片机原理及应用教程[M]. 北京: 北京大学出版社, 2006.
- [3] 卢大伟, 刘炳云. 用定时器实现软件看门狗应注意的问题[J]. 中国仪器仪表, 1998(4): 36 - 37.
- [4] Thomas F, Nayak M. A hardware/software codesign for improved data acquisition in a processor based embedded system[J]. Microprocessors and Microsystems, 2000, 24(3): 129 - 134.
- [5] 吴允平, 蔡声镇, 乐仁昌. 单片机任务型软件“看门狗”原理及应用[J]. 计算机工程与应用, 2004, 40(34): 122 - 123.
- [6] 刘芳芳, 黄会雄. 单片机测控系统抗干扰方法的研究和改进[J]. 电子质量, 2006(1): 62 - 63.
- [7] Zhou Y Q. Design and development of portable data acquisition system for vehicle roadway test[J]. Automation Instrumentation, 2006, 27(8): 61 - 63.