

一种提高 4G 视频网管可靠性机制的设计^①

张 鑫, 刘钊远, 王 钊

(西安邮电大学 计算机学院, 西安 710061)

摘 要: 针对嵌入式系统在 4G 的环境下把视频流推送到云端、从云端拉取、尤其是进行远程更新或调试的过程中存在大量的干扰现象, 从而导致嵌入式系统不能正常工作的问题, 提出了一种基于“watchdog”的技术(俗称“看门狗”)。主要分析了使用状态机, 多线程, 远程更新等技术的过程中怎么使系统更可靠的进行工作, 并将此方案应用到 OMAP(德州仪器(TI)公司的开放式多媒体开发板)进行实验, 试验结果表明, 它能很有效的提升 4G 视频网管的可靠性。

关键词: 看门狗; 多线程; 可靠性; 远程更新

Design of the 4G Network for Improving Video Reliability Mechanism

ZHANG Xin, LIU Zhao-Yuan, WANG Zhao

(College of Computer, Xi'an University of Posts & Telecommunications, Xi'an 710061, China)

Abstract: Embedded systems pushes the video stream to the cloud and pulls it from the cloud in 4G environments. Especially in remote update and debugging, there is so much interference that the system is not working properly. Therefore, we propose a kind of technology based on watchdog (commonly known as “watchdog”). This paper mainly analyzes how to make the system work more reliably in the process of using the state machine, the multiple threads and the remote update and so on, and this scheme applies to OMAP (TI's open multimedia development board) to make experiments. Test results show that it is effective to enhance the reliability of the 4G network video.

Key words: watchdog; multi-threading; reliability; remote update

随着科学技术的发展, 互联网已经渗透到生活中的方方面面。尤其是 4G 的高速发展极大的改善了人们的生活方式。而随着“流媒体”^[1]技术的发展, 人们可以通过一些音/视频直播等方式来参与其中。本文所涉及方案也是基于“流媒体”视频的基础上设计的。

在嵌入式系统运行的过程中, 尤其是在 4G 网络的情况下, 对嵌入式的可靠运行要求更高, 我们需要采取一定的措施来保障系统的可靠性, 即使由于干扰或其它原因导致系统异常, 系统也能够自恢复。这样才能保证嵌入式系统长时间不间断的工作。

提高嵌入式系统可靠性^[2]有很多种方法, 所以本文从实际出发采用的是硬件看门狗方案。因为“软件看门狗”是基于内核的定时器实现, 当内核或中断出现异常时, “软件看门狗”将会失效。而“硬件看门狗”由自身的硬件电路控制, 独立于内核。无论当前系统状

态如何, “硬件看门狗”在设定的时间间隔内没有被执行写操作(“喂狗”操作), 仍会重新启动系统。所以“硬件看门狗”能很好的弥补“软件看门狗”的一些缺点^[3], 从而提高了嵌入式系统的可靠性。

比如论文“一种远程在线更新 FPGA 程序的方法”^[4]需要有远程更新模块, 这就提高了成本, 而且操作不便。而笔者提出的是“ssh 反向端口转发”的远程更新策略, 这种方法不需要额外的模块, 操作方便, 而且可靠性更高, 能很好的满足远程更新过程中对可靠性的要求。

1 背景介绍

如图 1 所示, 系统主要分为三部分: 移动设备端、服务端、设备端(包含 OMAP 板子和带云台的摄像机)。其中移动端的作用是使用 APP(应用程序)来操控设备

^① 收稿时间:2016-05-21;收到修改稿时间:2016-06-21 [doi: 10.15888/j.cnki.csa.005595]

端的摄像头(命令设备端推送视频流、转动、调焦等),并接收服务端发送的视频流;服务端起到转发的作用,当多路设备同时推送视频流的时候还可以起到“负载均衡”作用;设备端主要实现具体的功能,包括视频的采集、摄像头的转动,调焦等.其中设备端使用的是4G的网络进行视频的采集和推送.由于此设备不在我们的可触及范围之内,所以不能进行人为的干预,这就导致了一些问题,比如:

① 如果设备端 4G 网络突然中断,当 4G 网络通畅时设备能不能不需要人为的干预重新连接到服务端?

② 怎么检测程序运行中的异常,怎么解决?

③ 尤其是当需要对设备进行远程更新时,此时看门狗该如何工作才能保证系统正常运行?

为了解决以上的几个问题,设计如下方案来提高设备在运行和更新的过程中的可靠性.

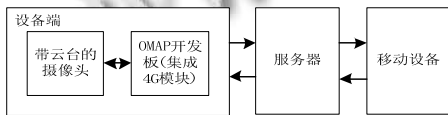


图 1 系统整体结构

2 可靠性方案的设计

本文设计的方案主要是以硬件“看门狗”芯片为基础^[5],以软件^[6]中的“状态机”、“多线程”^[7]、“ssh 反向端口转发”等技术为辅助,充分考虑嵌入式系统在工作过程中所遇到的状况,构成一个可实现自恢复、检测、远程更新等功能的可靠性系统^[8,9].具体技术方案如下文描述.

2.1 “状态机”技术

状态机是表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型,对状态机一个极度确切的描述是它是一个有向图形,由一组节点和一组相应的转移函数组成,状态机通过响应一系列事件而“运行”.比如,此文所介绍的嵌入式系统在启动时首先要在“开始”状态(建立 socket, 连接服务端);“开始”状态成功后,转移到下一个“登录”状态;登录成功后就要转移到下一个“处理命令”状态(如果在规定时间内没有收到来自服务端的命令,此时就转移到“心跳”状态,确保嵌入式设备没有断线).即使在某个状态出现错误(比如,服务端出现错误,嵌入式系统会一直在登录状态循环直到服务端恢复正常),嵌入式设备也不

会中断.所以此方案主要是形成闭环的防错机制,对比单方向的监测,在干扰情况发生时,闭环中的任一部分出现异常都能进行监测恢复.图 2 所示为状态机的工作流程.

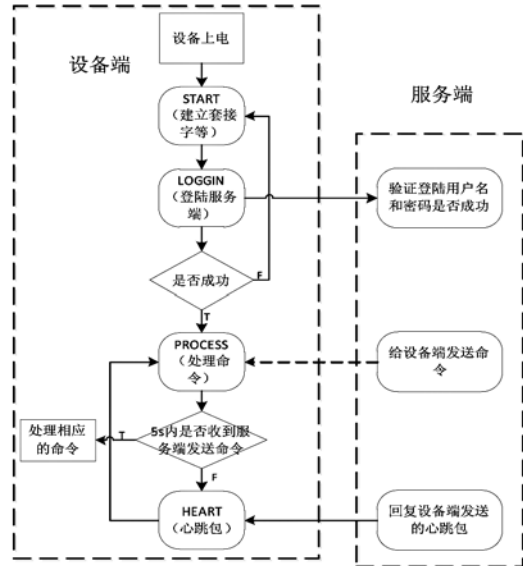


图 2 “状态机”工作流程

2.2 “看门狗”技术

为了使系统在异常情况下能自恢复,笔者使用了看门狗电路.看门狗电路其实就是一个计数器.当看门狗启动后,计数器开始自动计数,经过一定时间计数器溢出就会对 CPU 产生一个复位信号使系统重启.系统正常运行时,需要在看门狗允许的时间间隔内对看门狗计数器清零(“喂狗”),不让复位信号产生.本方案通过添加“看门狗”模块,一方面监测系统状态,另一方面监测此应用中其它模块的工作状态.系统出现故障时能自恢复.其中“看门狗”模块会随着系统的启动而启动,自动提供检测服务.这类服务应用通常是多线程的.笔者的作法是在应用中加入一个“看门狗”线程,此线程主循环的工作步骤如下:

(1) 启动看门狗模块(此方案中的看门狗芯片的 EN, WDI 管脚是和 OMAP 上的 GPIO 相连的).当系统启动时,把 EN 所对应的 GPIO 电平拉低,看门狗模块就开始工作.

(2) “喂狗”操作,进入 while(1)循环中进行喂狗操作,如果对 WDI 所对应的 GPIO 执行写操作,表明当前系统运行良好.如果超过规定的时间内(这个时间是根据硬件芯片规定的,典型的为 1.6s)没有对所对应的

GPIO 执行写操作, 系统将会自动重启。

(3) 当系统或者其它线程出现异常, 不再进行“喂狗”操作, 并等待看门狗定时器溢出, 此时“看门狗”电路复位整个系统。

2.3 “多线程”技术

由于多线程具有能够很好的利用 CPU 资源, 程序设计更简单, 程序响应更快等优点, 所以此方案采用的多线程的机制。图 3 所示为看门狗监测线程监测^[9]其它线程工作状态的流程图。看门狗线程检测多个线程的工作状态, 只有当所有线程工作状态正常时, “看门狗”线程向硬件看门狗电路发送喂狗信号。若某个线程出现故障, 该线程会自动停止对硬件看门狗的喂狗, 硬件看门狗发出 RESET 信号重启嵌入式系统。

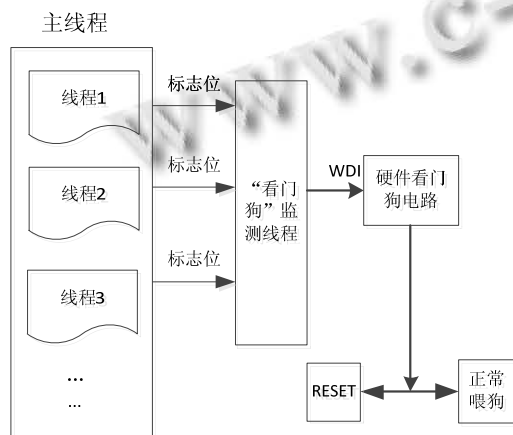


图 3 “看门狗”线程工作流程

2.4 基于 ssh 反向端口转发的技术

反向端口转发^[10]是 ssh 最强大的特性之一, 也是本文的重点, 笔者的方案是通过在 OMAP 开发板设置端口转发, 将流量转发到一个服务器上, 然后我们可以登录到这个服务器上, 进而可以通过此服务器登录到 OMAP 开发板的操作系统, 对板子进行操作。其工作流程为:

(1) 在本地主机上使后台程序向 OMAP 板子发送开启反向端口转发的命令, 然后嵌入式系统的应用程序打开反向端口(前提是 OMAP 必须能够无密码登录到特定的服务器上)登录到此特定服务器上。

(2) 本地主机可以登录到这个特定的服务器上, 通过服务器登录到 OMAP 开发板上, 对开发板进行更新或调试等。

我们对 OMAP 上的应用程序进行远程更新^[10]调

试时看门狗是不能工作的。不然我们更新调试程序时肯定没办法在规定的时间内“喂狗”, 进而系统会重启。这不是我们想要的。所以在打开远程端口进行反向端口转发时应该关闭“看门狗”模块(把看门狗芯片 EN 管脚所对应的 GPIO 的电平拉高)。在更新调试完成以后还要使“看门狗”电路工作(把看门狗芯片 EN 管脚所对应的 GPIO 的电平拉低)。这样“看门狗”就不会在我们调试的时候重启系统。图 4 所示为 ssh 远程更新工作流程图。

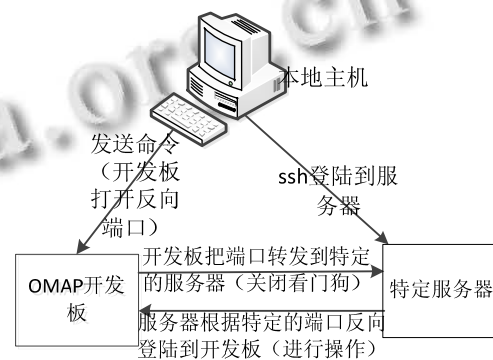


图 4 远程更新流程图

3 实验设计与结果分析

3.1 实验环境

为了验证此方案的确能提高 4G 视频网管的可靠性, 设置如表 1 的实验环境。

表 1 实验环境和工具

| 物理机 | 开发板 | 测试工具 | 数据库工具 |
|-------------------------|-----------------|------------------------|----------------------------|
| Ubuntu 12.04, 内核版本 3.10 | OMAP(内核版本 3.10) | vim, cat, top, tail等命令 | Navicat for MySQL(浏览数据库内容) |

3.2 测试方法

为了测试该方法的正确性和可用性, 笔者首先和没有使用此方案的方法在功能上做了对比, 在功能上主要对比以下三个功能点:

功能点 1. 能否在 4G 断线恢复后自动登录到服务端。

功能点 2. 设备端能否长时间运行。

功能点 3. 能否进行可靠的远程更新调试。

对于以上功能采取以下的方案进行测试。

方案 1: 人为的使 4G 断线和重启, 在服务端观测设备端的嵌入式系统是否重新启动并登录到服务端正常运行。

方案 2: 系统长时间运行测试, 不需要人为干预, 测试系统能否长时间正常稳定的工作.

方案 3: 远程更新测试, 首先本地主机经过 ssh 登录到特定的服务器上, 再在特定的服务器上通过反向端口转发登录到 OMAP 开发板进行更新和调试.

3.3 实验结果

对上述三种方案进行对比测试, 测试结果如表 2 所示, 其中 board2 是用了此方案的开发板, board3 是未用此方案的开发板.

表 2 功能对比

| | 断线重启 | 长时间运行 | 远程更新 |
|--------|------|-------|------|
| board2 | 是 | >72小时 | 是 |
| board3 | 否 | <72小时 | 否 |

```

May 23 18:55:52 i2255hzmcoqZ board2[2944]: BOARD2:DEVICE board2 EXIT!!!!
May 23 18:55:52 i2255hzmcoqZ server[2944]: SERVER:CLOSE THE FD 45
May 23 18:56:07 i2255hzmcoqZ server[2944]: SERVER:recv sockfd:45
May 23 18:56:07 i2255hzmcoqZ server[2944]: SERVER:recv length:36
May 23 18:56:07 i2255hzmcoqZ server[2944]: SERVER:COMMAND: f2 6 24 62 6f 61 72 0
May 23 18:56:07 i2255hzmcoqZ server[2944]: SERVER:old sockfd:0
May 23 18:56:07 i2255hzmcoqZ server[2944]: SERVER:NEW sockfd:45
May 23 18:56:07 i2255hzmcoqZ board2[2944]: BOARD2:DEVICE board2 REGISTER SUCCESS!
May 23 18:56:12 i2255hzmcoqZ board2[2944]: BOARD2:recv sockfd:45
May 23 18:56:12 i2255hzmcoqZ board2[2944]: BOARD2:recv length:6
May 23 18:56:12 i2255hzmcoqZ board2[2944]: BOARD2:COMMAND: f2 7 5 1 ff 7f

```

图 5 设备端断线和重启

方案 1 测试结果如图 5 所示. 图 5 中, 首先设备端运行正常, 在设备端人为的使 4G 断线和重启, 设备端退出(board2 EXIT!!!), 经过一段时间设备端 board2 又重新登录到服务端并且向服务端发送“心跳包”, 设备端运行正常.

方案 2 测试结果如图 6, 图 7 所示.

① 使用此方案的测试结果如图 6 所示. 图 6 中, 设备(board2)4 月 26 登录到服务端直到发文时还未断线.

| 对象 | module_name | occur_time | info |
|--------|-------------|---------------------|-------------|
| board2 | | 2016-04-26 11:19:43 | dev_on_line |

图 6 使用此方案测试结果

| | | |
|--------|---------------------|--------------|
| board3 | 2016-04-29 18:00:39 | dev_on_line |
| board3 | 2016-04-30 03:32:52 | dev_off_line |

图 7 未使用此方案的测试结果

② 未使用此方案的测试结果如图 7 所示. 图 7 中, 设备(board3)只连接了九个半小时左右, 使用此方案效果显著.

方案 3 测试结果如图 8 所示. 图 8 中, ssh 首先登录到一台特定的服务器(112.74.89.131), 再在这台特定的服务器上通过反向端口转发(ssh -p)登录到 OMAP 开发板, 可以进行更新和调试(如上图已登录到开发板).

```

[root@localhost work]# ssh zhangxin@112.74.89.131
zhangxin@112.74.89.131's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-67-generic

 * Documentation:  https://help.ubuntu.com/

323 packages can be updated.
294 updates are security updates.

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Welcome to aliyun Elastic Compute Service!

Last login: Mon May 23 19:29:50 2016 from 113.140.86.66
zhangxin@iZ94jt4x5c9Z:~$ ssh -p 9000 media@localhost
[zx@localhost ~]$ cd work/
[zx@localhost work]$ ls
board_client  remote_client

```

图 8 远程更新操作

综合上面所有测试结果可知, 本文提出的对 4G 视频网管的可靠性设计的方案是正确的, 并且在实际生产环境下是可用的.

4 总结

本文提出了一种在 4G 推送视频流的情况下的一种嵌入式系统可靠性机制的策略. 该策略有如下特点: 看门狗电路简单, 硬件只需一块看门狗芯片, 不需复杂的外围逻辑电路, 成本较低, 而且对芯片操作也比较简单, 此方案不仅能够对各个线程的运行状态进行监测. 而且能够在远程更新的过程中保证嵌入式系统的可靠性. 并且在运行的过程中不需要人为的干预, 使系统可靠性得到很大的提高, 具有很高的实用价值.

参考文献

- 胡敏. 流媒体技术在网络中的应用与发展探究. 电子测试, 2014,3:131-132.
- 谢雷. 对开发嵌入式系统软件的可靠性设计及分析[学位论文]. 长沙:中南大学, 2011.
- 刘金福, 武宏伟, 杨胜姚. 一种远程在线更新 FPGA 程序的方法. 单片机与嵌入式系统应用, 2012,12(2).
- 曹廷发, 吉华, 马江, 付耀国. Linux/RTLinux 下实时多线程实时性检测方法. 计算机技术与发展, 2010,20(3):113-115.
- 郑培余, 李汉斌, 姚绍文. 支持多线程监控的外置 Watchdog 监控组件的设计. 计算机工程, 2004,30(24):177-179.
- 王帆. 软件维护中的成本估算和质量保证技术研究[博士学位论文]. 杭州:浙江大学, 2011.
- 杨永, 高嵩, 陈超波, 何宁. 多线程编程技术在数控测井软件中的应用. 电子设计工程, 2011.
- 张虹, 姜明明, 黄百乔. 软件可靠性分析方法研究与应用. 测控技术, 2011,30(5):101-105.
- 李金麒, 徐建平. 嵌入式系统软件可靠性设计与测试方法. 计算机系统应用, 2013,22(1):74-78.
- 陈松. 一种基于 Telnet/SSH 的网络终端管理的方法. CN, CN 103716179 A. 2014.