



(12) 发明专利申请

(10) 申请公布号 CN 104503390 A

(43) 申请公布日 2015. 04. 08

(21) 申请号 201410703498. 8

(22) 申请日 2014. 11. 30

(71) 申请人 天津理工大学

地址 300384 天津市西青区宾水西道 391 号
天津理工大学主校区

(72) 发明人 陈在平 孙逊 贾超 倪建云

(74) 专利代理机构 天津佳盟知识产权代理有限公司 12002

代理人 李益书

(51) Int. Cl.

G05B 19/418(2006. 01)

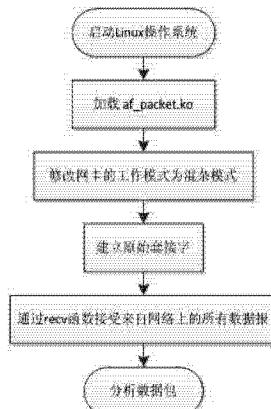
权利要求书1页 说明书4页 附图3页

(54) 发明名称

一种集成工业以太网从站自主识别主站的方法

(57) 摘要

本发明公开了一种集成工业以太网从站自主识别主站的方法，应用在嵌入有 Linux 操作系统的集成从站上。该方法基于工业以太网主从站之间通信前，主站会向从站发送识别信息这一前提，采用在 Linux 操作系统下原始套接字编程的方法，抓取通信网络中主站发出的识别信息，并对抓取到的识别信息进行区分判断，从而确定主站的类型并加载相应的从站程序。通过该方法，可以实现工业以太网集成从站自主识别与之相连接的主站类型，并完成自动加载运行与之对应的从站程序。



1. 一种集成工业以太网从站自主识别主站的方法,该方法应用在工业控制现场,运行有嵌入式 Linux 操作系统的集成工业以太网从站,以下简称集成从站;所述的集成从站集成了 Modbus-TCP 和 POWERLINK 从站程序;此方法通过分析判断来自通信网络中的主站识别信息,来确定与集成从站相连接的主站类型,从而选择加载运行对应的集成从站程序,实现了集成从站对主站类型的自主识别和集成从站程序的自动加载,摆脱了传统的手动加载的方法;此方法的具体实现步骤如下:

第 1 步、抓取工业以太网通信网络中主站发送的识别数据包;

第 1.1 步、启动集成从站中的 Linux 操作系统;

第 1.2 步、设置集成从站以太网芯片的工作模式为混杂模式,这样集成从站就不会只接受目的地址为它的数据包,而是全部经过它的数据包,方便获得主站发送来的识别信息数据包;

第 1.3 步、通过 socket 套接字函数创建一个原始套接字的文件描述符;

第 1.4 步、利用 recv() 函数来接受来自通信网络中的数据包,并将接收到的数据包赋值到一个字符型指针变量中,方便后续对其进行分析;

第 2 步、分析抓取到的通信网络中的数据包;

第 2.1 步、将第 1 步中抓取到的数据包划分格式,因为所抓取的数据包赋值给某一变量时,是没有划分格式的,数据包从左向右依次为以太网报文头、IP 报文头、TCP 报文头或者 UDP 报文头,在区分 Modbus-TCP 和 POWERLINK 工业以太网只需要知道以太网报文头即可;将数据包所复制的变量强制转化为 struct ether_header 结构体,那么数据包的以太网报文头便被保留下来;

第 2.2 步、判断 struct ether_header 结构体中的 ether_type 成员变量,如果变量的值为 0x0806 则可确定数据包来自 Modbus-TCP 主站,与集成从站相连的是 Modbus-TCP 主站,结束第 2 步,进入第 3 步;如果 ether_type 的值,不为 0x0806,继续分析数据包;

第 2.3 步、继续判断 struct ether_header 结构体中的 ether_type 成员变量,如果变量的值为 0x88ab 则能够确定数据包来自 POWERLINK 主站,与集成从站相连的是 POWERLINK 主站,结束第 2 步,进入第 3 步;如果 ether_type 的值,不为 0x88ab,继续分析数据包;

第 2.4 步、抛弃所抓取的数据包,返回第 1 步,重新抓取通信网络中其他数据包;

第 3 步、根据第 2 步的分析结果,加载相应的从站程序;

此步骤根据第 2 步中对 ether_type 成员变量的分析判断结果不同,选择加载运行不同的工业以太网从站程序,或者不加载运行;

第 3.1 步、对于 Modbus-TCP 主站,Modbus-TCP 作为 Linux 下的可执行程序,在可执行文件中通过 system() 函数来完成 Modbus-TCP 从站程序的运行;

第 3.2 步、对于 POWERLINK 主站,POWERLINK 从站程序代码被交叉编译后生成一个 Linux 内核模块,需要通过 insmod 命令加载到 Linux 内核中才能运行;这里,通过编写一个 shell 脚本文件,在可执行程序中通过执行 system() 来执行脚本文件,从而完成 POWERLINK 从站程序的加载运行。

一种集成工业以太网从站自主识别主站的方法

技术领域

[0001] 本发明属于工业控制领域,涉及工业控制现场,运行有嵌入式 Linux 操作系统的集成工业以太网从站,与不同种类工业以太网主站进行通信时,进行自主识别与之连接的主站种类,进行相应从站程序的选择运行。

背景技术

[0002] 以太网技术向工业现场的推广,为生产效率的提高,产品的优化等等多方面都起到了极大的促进作用。目前世界上有多种工业以太网标准,种类繁多,各有优点,在某一生产过程中,往往需要几种工业以太网配合使用会产生更大的收益。所以,在工业控制现场比较流行使用集成工业以太网从站,但是,集成从站内对于从站程序的选择运行,一直采用传统的手动加载运行,即在上电前确定集成从站的性质,这给大规模的工业生产带来了很多不便。目前,急需一种方法,能够实现集成从站自主识别与之连接的主站类型,并自动运行与之对应的从站程序。

发明内容

[0003] 本发明的目的是为了解决现有技术上存在的上述问题,提供一种集成工业以太网从站能够识别主站类型并自主选择加载运行相应从站程序的方法。

[0004] 本发明提供的集成工业以太网从站自主识别主站的方法,应用在工业控制现场,运行有嵌入式 Linux 操作系统的集成工业以太网从站,以下简称集成从站;所述的集成从站集成了 Modbus-TCP 和 POWERLINK 从站程序;此方法通过分析判断来自通信网络中的主站识别信息,来确定与集成从站相连接的主站类型,从而选择加载运行对应的集成从站程序,实现了集成从站对主站类型的自主识别和集成从站程序的自动加载,摆脱了传统的手动加载的方法;

该方法具体的实现步骤如下:

第 1 步、抓取工业以太网通信网络中主站发送的识别数据包;

第 1.1 步、启动集成从站中的 Linux 操作系统;

第 1.2 步、设置集成从站以太网芯片的工作模式为混杂模式,这样集成从站就不会只接受目的地址为它的数据包,而是全部经过它的数据包,方便下面获得主站发送来的识别信息数据包。

[0005] 第 1.3 步、通过 socket 套接字函数创建一个原始套接字的文件描述符;

第 1.4 步、利用 recv() 函数来接受来自通信网络中的数据包,并将接收到的数据包赋值到一个字符型指针变量中,方便后续对其进行分析。

[0006] 第 2 步、分析抓取到的通信网络中的数据包;

第 2.1 步、将第 1 步中抓取到的数据包划分格式,因为所抓取的数据包赋值给某一变量时是没有划分格式的,数据包从左向右依次为以太网报文头、IP 报文头、TCP 报文头或者 UDP 报文头,在区分 Modbus-TCP 和 POWERLINK 工业以太网只需要知道以太网报文头即可。

将抓取到的数据包强制转化为 struct ether_header 结构体,那么数据包的以太网报文头便被保留下。

[0007] 第 2.2 步、判断 struct ether_header 结构体中的 ether_type 成员变量,如果变量的值为 0x0806 则可确定数据包来自 Modbus-TCP 主站,与集成从站相连的是 Modbus-TCP 主站,结束第 2 步,进入第 3 步;如果 ether_type 的值,不为 0x0806,继续分析数据包。

[0008] 第 2.3 步、继续判断 struct ether_header 结构体中的 ether_type 成员变量,如果变量的值为 0x88ab 则可确定数据包来自 POWERLINK 主站,与集成从站相连的是 POWERLINK 主站,结束第 2 步,进入第 3 步;如果 ether_type 的值,不为 0x88ab,继续分析数据包。

[0009] 第 2.4 步、抛弃所抓取的数据包,返回第 1 步,重新抓取通信网络中其他数据包。

[0010] 第 3 步、根据第 2 步的分析结果,加载相应的从站程序;

此步骤根据第 2 步中对 ether_type 成员变量的分析判断结果不同,选择加载运行不同的工业以太网从站程序,或者不加载运行:

第 3.1 步、对于 Modbus-TCP 主站,Modbus-TCP 作为 Linux 下的可执行程序,在可执行文件中通过 system() 函数来完成 Modbus-TCP 从站程序的运行;

第 3.2 步、对于 POWERLINK 主站,POWERLINK 从站程序代码被交叉编译后生成一个 Linux 内核模块,需要通过 insmod 命令加载到 Linux 内核中才能运行;这里,通过编写一个 shell 脚本文件,在可执行程序中通过执行 system() 来执行脚本文件,从而完成 POWERLINK 从站程序的加载运行。

[0011] 本发明的优点和有益效果

通过本发明采用的技术方案可以看出,本方法能够实现集成工业以太网从站对与之相连接主站进行判断识别,并加载相应的从站程序,与主站建立连接,完成后续通信的任务。摆脱了传统的手动选择方式所带来的工作量大,不确定因素多等问题。现在的办法既方便又安全可靠,有更高的实用价值。

附图说明

[0012] 图 1 是数据包抓取流程图;

图 2 是以太网控制芯片设置分析图;

图 3 是 Modbus-TCP 和 POWERLINK 主站报文分析图;

图 4 是 Modbus-TCP 和 POWERLINK 报文以太网头分析图;

图 5 是数据包分析流程图。

具体实施方式

[0013] 本发明提供的集成工业以太网从站自主识别主站的方法,具体步骤如下:

第 1 步、抓取工业以太网通信网络中主站发送的识别数据包;

图 1 是集成从站嵌入式 Linux 操作系统下,如何运行程序来抓取通信网络上的数据包。

[0014] 首先启动 Linux 操作系统,提供软件程序的工作环境,加载 af_packet.ko 原始套接字功能模块,一般的嵌入式 Linux 操作系统中往往不具有这一模块,需要自行加载到内核中。接着,将与集成从站相连的以太网芯片的工作模式设置成混杂模式,因为以太网芯片的缺省工作模式为直接模式或广播模式,缺省工作模式下,以太网控制芯片会判断每个物

理数据帧目的地是否为本站地址,如果不是就将它丢弃。然而,集成从站需要在从站程序运行前,完成对通信网络中主站发出的带有识别信息的数据包进行抓取和分析。因此,需要将适配器网卡设置成为混杂模式,这样就可以达到对于网络信息监视捕获的目的。通过命令“ifconfig eth0 promisc”完成以太网卡模式的修改,图2上下两部分显示出了以太网控制芯片设置前后的状态。

[0015] 第2步、分析抓取到的通信网络中的数据包;

这里涉及到的图3和图4都是通过wireshark抓包工具,抓取到的来自通信网络中数据包。

[0016] 图3中方框标注的内容分别为Modbus-TCP和POWERLINK工业以太网主站发出的识别信息数据包,观察比较可知,二者在协议类型(protocol选项)上存在差异,Modbus-TCP是ARP,POWERLINK是ep1,以此作为区标志。

[0017] 图4是这两种协议以太网报文头的分析图。本发明方法对抓取到的数据包进行分析,所涉及到的内容是在数据链路层,主要判定以太网报文头,对上层网络包括网络层和传输层,不作分析判定。Modbus-TCP和POWERLINK为两种工业以太网通信协议,使用的均为Ethernet II帧格式,这里之所以要突出是使用的Ethernet II型数据帧,是因为它的12字节之后为类型选项,而Modbus-TCP和POWERLINK这两种协议在这里有所区分。Modbus-TCP协议是基于TCP/IP的工业以太网,它的以太网报文头中此项内容为0x0806;POWERLINK协议不是基于TCP/IP的以太网,它有自己的数据链路层协议标准,因此它的以太网报文头的类型内容为0x88ab。因此,就可以通过判断抓取到的数据包的以太网数据报文类型,来确定与集成从节点相连接的通信主站类型。

[0018] 图5是分析判断所抓取到数据包的流程图。首先,根据数据包的构成形式,将抓取到的数据包划分格式,因为所抓取的数据包给某一变量是没有划分格式的,数据包从左向右依次为以太网报文头、IP报文头、TCP报文头或者UDP报文头,根据上面的分析,可知区分Modbus-TCP和POWERLINK工业以太网只需要知道以太网报文头即可。将数据包赋值到某一字符指针变量中,并将此变量强制转化为struct ether_header结构体,那么数据包的以太网报文头便被保留下。struct ether_header结构体是以太网报文头的函数体形式,具体构成如下:

*****以太网报文头的结构体*****

```
struct ether_header
{
    u_int8_t    ether_dhost[ETH_ALEN];
    u_int8_t    ether_shost[ETH_ALEN];
    u_int16_t   ether_type;
};
```

接着,通过判断语句,判断以太网报文头中的ether_type这一项,根据图3的报文分析可知,如果以太网头的类型为0x0806,就是Modbus-TCP主站的识别信息,那么就加载Modbus-TCP从站程序;如果以太网头的类型为0x88ab,就是POWERLINK主站的识别信息,那么就加载POWERLINK从站程序;如果不是这两种的一个,那么就是垃圾包,将其丢弃,并通过while函数返回到第1步,重新抓取分析数据包。

[0019] 第 3 步、根据第 2 步的分析结果,加载相应的从站程序

此步骤根据第 2 步中对 ether_type 成员变量的分析判断结果不同,选择加载运行不同的工业以太网从站程序,或者不加载运行。由于 Modbus-TCP 和 POWERLINK 的从站程序的特点不同,它们运行的方式也是不同的。Modbus-TCP 作为 Linux 下的可执行程序,在可执行文件中通过 system () 函数来完成 Modbus-TCP 从站程序的运行;而 POWERLINK 从站程序代码被交叉编译后生成一个 Linux 内核模块,需要通过 insmod 命令加载到 Linux 内核中才能运行,但是,从可执行文件中是无法执行 insmod 命令来加载内核模块,这里,通过编写一个 shell 脚本文件,在可执行程序中通过执行 system () 来执行脚本文件,从而间接地完成 POWERLINK 从站程序的加载运行。

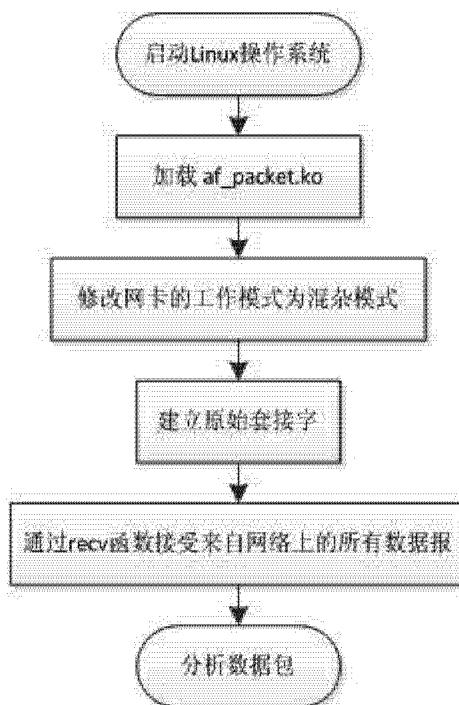


图 1

<pre> eth0 Link encap:Ethernet Hwaddr 10:23:45:67:89:AB inet addr:192.168.100.113 Bcast:192.168.100.255 Mask:255.255.255.0 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:5 errors:0 dropped:0 overruns:0 frame:0 TX packets:13 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:623 (623.0 B) TX bytes:714 (714.0 B) Interrupt:51 Base address:0x6000 </pre>	修改前
<pre> eth0 Link encap:Ethernet Hwaddr 10:23:45:67:89:AB inet addr:192.168.100.113 Bcast:192.168.100.255 Mask:255.255.255.0 UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1 RX packets:17 errors:0 dropped:0 overruns:0 frame:0 TX packets:13 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:4383 (4.2 Kib) TX bytes:714 (714.0 B) Interrupt:51 Base address:0x6000 </pre>	修改后

图 2

Source	Destination	Protocol
dc:0e:a1:6c:44:16	Elitegro_32:6b:37	ARP
Elitegro_32:6b:37	dc:0e:a1:6c:44:16	ARP
dc:0e:a1:6c:44:16	Broadcast	ARP
dc:0e:a1:6c:44:16	Broadcast	ARP
50:46:5d:cf:c9:4b	Broadcast	ARP

Source	Destination	Protocol
Elitegro_32:6b:37	EPLv2_SoA	EPL

图 3

- Ethernet II, Src: Applicom_07:a6:5f (00:a0:91:07:a6:5f),
 + Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 + Source: Applicom_07:a6:5f (00:a0:91:07:a6:5f)
 Type: ARP (0x0806)
- Ethernet II, Src: Elitegro_32:6b:37 (00:21:97:32:6b:37)
 + Destination: EPLv2_SoA (01:11:1e:00:00:03)
 + Source: Elitegro_32:6b:37 (00:21:97:32:6b:37)
 Type: ETHERNET Powerlink v2 (0x88ab)

图 4

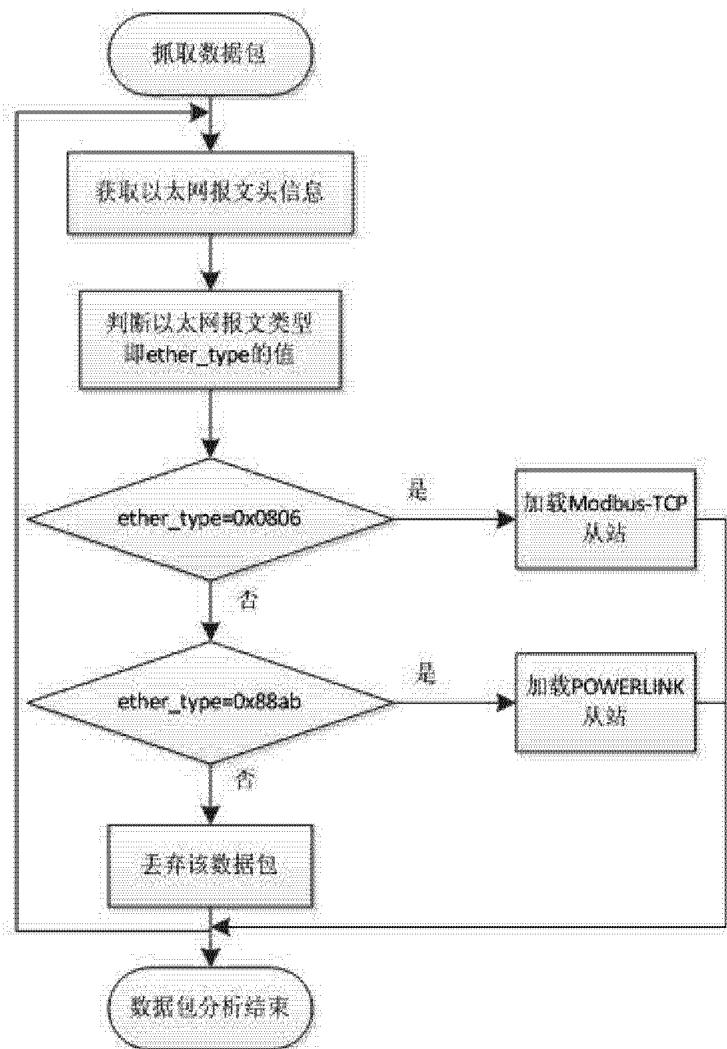


图 5