

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04L 12/28

H04L 29/06



[12] 发明专利说明书

专利号 ZL 00103067.1

[45] 授权公告日 2005 年 10 月 12 日

[11] 授权公告号 CN 1223152C

[22] 申请日 2000.2.28 [21] 申请号 00103067.1

[71] 专利权人 永泰锋国际有限公司

地址 香港沙田小沥源安丽街 18 号达利广场
18 楼 11 室

[72] 发明人 黄 更

审查员 刘春霞

[74] 专利代理机构 中科专利商标代理有限责任公
司

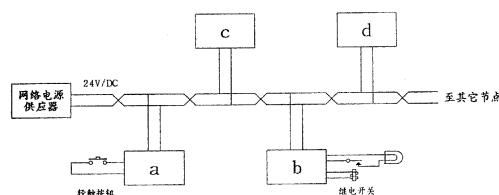
代理人 汤保平

权利要求书 2 页 说明书 18 页 附图 7 页

[54] 发明名称 智能家庭控制总线

[57] 摘要

本发明一种智能家庭控制总线，拥有至少一个输入节点(指令器)A 和至少一个输出节点(执行器) L，其中使用者的控制由 A 输入，经联结 A 和 L 的总线传送至 L，并由 L 产生使用者所期望的逻辑操作输出；本发明具有结构简单、成本低和功能灵活的优点。



I S S N 1 0 0 8 - 4 2 7 4

1. 一种智能家庭控制总线系统，拥有至少一个指令器和至少一个
5 执行器，其中控制信号由指令器输入，经总线传送至执行器，再由执行器实现操作输出，其特征在于：

1) 指令器是控制信号的输入者和数据包的发送者，指令器至少含有一个输入装置，可以用来输入控制信号；指令器同时至少含有一个事件数据包，该数据包在系统中对事件具有唯一性；当输入装置检测到有
10 信号输入时，指令器便将该数据包发送至总线之上；

2) 执行器是数据包的接收者和操作的执行者，执行器至少含有一个输出装置，可以用来执行某些操作的输出；执行器同时至少含有一个可读写的存储器，该存储器主要用来存放编程过程中产生的数据包；

3) 执行器在接收到来自总线的由指令器发出的事件数据包后，便
15 会将该事件数据包与执行器编程时存放在可读写存储器内的数据包相比较，如果两数据包相同或相关，执行器便执行与该数据包相对应的操作输出。

2. 根据权利要求 1 所述的智能家庭控制总线系统，其特征在于，其中指令器的数据包事先已经设定并存放在指令器内的非易失性存储器当
20 中，即是指令器在使用时不需编程。

3. 根据权利要求 1 所述的智能家庭控制总线系统，其特征在于，其中执行器还进一步包含有一个编程按钮，使用该编程按钮可以令执行器进入编程状态，亦即是说执行器可由用户编程。

4. 根据权利要求 1 所述的智能家庭控制总线系统，其特征在于，
25 其中指令器与执行器之间的控制逻辑关系的建立依赖于用户对执行器的编程，而与指令器的数据包的含义无关。

5. 根据权利要求 1 所述的智能家庭控制总线系统，其特征在于，其中执行器的可读写存储器是指包含在执行器内的数据存储或非易失性存储器。

30 6. 根据权利要求 1 所述的智能家庭控制总线系统，具特征在于，

其中执行器还进一步包含有至少一个指示灯，用来显示执行器的工作状态，至少用来指示执行器是否已进入编程状态。

7. 根据权利要求 2 所述的智能家庭控制总线系统，其特征在于，其中指令器内的非易失性存储器，是指包含在指令器内部的特定的中央
5 处理器或微处理器内部的只读存储器。

8. 根据权利要求 3 所述的智能家庭控制总线系统，其特征在于，其中编程状态是指这样的一种状态，在这种状态中，执行器将进一步等待接收来自指令器的事件数据包和任何执行器能接受和解释的来自编程
10 装置的操作指示。

9. 根据权利要求 8 所述的智能家庭控制总线系统，其特征在于，其中来自指令器的事件数据包，和来自编程器的相关操作指示，在编程完成后均会以数据的形式存放在执行器的可读写存储器之内。

10. 根据权利要求 1 所述的智能家庭控制总线系统，其特征在于，其中指令器是采用类曼彻斯特编码方式将数据包发送到总线上，该编码
15 方式是指二进制数字 1 用一个脉冲组成，而二进制数字 0 则用二个宽度相同而方向相反的脉冲组成。

11. 根据权利要求 10 所述的智能家庭控制总线系统，其特征在于，其中二进制数字 1 的一个脉冲的时间与二进制数字 0 的二个脉冲的时间之和相等，亦既是二进制数字 1 和 0 有相同的传输时间和传送速率。

12. 根据权利要求 10 所述的智能家庭控制总线系统，其特征在于，其中不同种类的事件数据包都具有相同的长度即 10 个字节，其中第 1
20 个字节为数据类别代码，在第 2 至第 9 个字节当中包含有节点身份代码，而第 10 个字节则为整个数据包的校验和。

智能家庭控制总线

5

技术领域：

本发明涉及一种家庭控制总线系统，特别是一种智能家庭控制总线系统，属于家庭自动化领域。

10

背景技术：

家庭自动化（HA）在我国正在兴起，一些欧美著名的家庭自动化标准，例如 LonWorks、CEBus 和 X-10 等都先后被业界的先驱们介绍引进，对我国在该领域的技术发展起到了积极的推动作用，有些亦事实上成为我国在 HA 方面的标准。与此同时，我国有不少机构和厂家亦积极参与，提出了许多有建设性的构想，并推出相关的产品，使系统集成商们和广大的用户都有了更多的选择，逐步形成百花齐放的局面。最近在国家重点科技攻关项目“智能化住宅及住宅小区试点工程”——“创世纪滨海花园小区”的自动化工程便选用的了一款名为“ApBus”的智能家庭控制总线产品。到底智能家庭控制总线是一项什么样的技术，有何可取之处，在云云众多的开放系统中又如何脱颖而出，寻找自己的生存空间呢？

家庭自动化技术早在七十年代便开始流行于美国，当时最有代表性的产品是 X-10，其后美国的埃施朗（Echelon）和美国电子工业协会（EIA）又相继推出 LonWorks 和 CEBus，并最终都成为美国电子工业协会（EIA）的家庭自动化网络标准。这些标准所采用的传输媒介虽然不尽相同，但无一例外都采用了控制网络的技术，并制订一个统一的通信协议（communication Protocol），通过该通信协议，网络系统中的各个设备便可实现资源的共享。控制网络因其布线简单、功能灵活，扩展容易而被人们广泛接受和应用。至今，X-10 产品的已销售已超过一亿个，

光在美国一个国家，便有超过 400 万个家庭在使用。LonWorks 亦被广泛应用于家庭自动化、建筑自动化（BA）及工业自动化（IA）方面，至 98 年中，全球安装 LonWorks 的节点（Node）数目已超过 500 万个。

然而 X-10 产品在我国的应用例子并不多见，究其原因可能是产品不太适合国内使用。X-10 产品在美国主要是用在旧房子的改造上，而国内则一般是在买新房子或在旧房子装修时才会考虑这个问题。如果房子是入墙式装修，要外加 X-10 功能模块在开关插座之上，多少会觉得不太协调。另外人们采用新技术时，又往往对新技术期望过高，X-10 的一些技术限制成为人们批评的藉口。例如反应速度慢（在 60 赫芝供电系统中，传送一个指令需时 0.883 秒），抗干扰性能差（这是由于信号载波在电力线中传送造成的，在我国这个问题比较严重）等，这都给 X-10 在我国的推广应用带来实质性的困难。

这些问题在美国同样亦存在。为了解决 X-10 技术上的不足，美国电子工业协会（EIA）于 1984 年 4 月开始组织开发新的家居网络标准，于 1992 年正式推出，命名为 CEBus（Consumer Electronic Bus），并定为 IS-60 / EIA-600 标准。参与研究开发 CEBus 的公司多达几百间，象 Microsoft、IBM、Compaq Computer Corp、AT&T、Bell Labs、Honeywell、Panasonic、Sony、Thomson Consumer Electronics、Leviton、Pacific Gas & Electric 等都是它的会员。CEBus 是一个较完整的开放系统，它定义了几乎所有传送媒体（Medium）中信号的传输标准，并要求控制信号在所有的媒体中都要以相同的速度（10KBPS）传送，从而有效地避免信号传输中可能出现的“瓶颈”问题，任何符合 CEBus 标准的电器产品都可以直接互连及通信，其寻址能力超过 40 亿，足可以与互联网相比美。CEBus 的抗干扰能力比 X-10 强，控制功能亦十分丰富，成为 X-10 的“最大挑战者”。但由于 CEBus 接口技术比较复杂，价钱亦非常之贵，因此使用 CEBus 产品的用户相对较少，CEBus 在中国的应用亦不多见。

相反地，LonWorks 在国内的某些领域则具有较高的知名度，不少专家学者和研究机构对 LonWorks 都很有兴趣，成为一个热门的研究课题。在应用上，LonWorks 也取得比较好的成绩，特别在信息采集和控制方面已成为事实上的行业标准。LonWorks 是由美国 Echelon 公司耗巨资，于

1990年12月开发成功的全分布式智能控制网络技术，参与研究开发的人员多达700人。Echelon的创办人为Mike Markkula，他亦是苹果电脑的主要创办人之一。LonWorks最基本的部件是同时具有通信与控制功能的Neuron芯片，该芯片内置三个微处理器、并将操作系统、运作时间数据库以及通信协议等都固化在芯片之内。1997年7日Echelon允许在牌照管制的条件下将LonWorks通信协议用于其它CPU上，并委托Adept systems, Inc. 在Motorola的32位微处理器68360上开发相关软件。同年8月LonWorks被EIA的集成家庭系统技术委员会(Integrated Home Systems Technical Committee)定为家庭网络(Home Networking)的标准，编号为EIA/IS-709。

尽管LonWorks是EIA定义的家庭网络标准，而在中国则主要用于BA方面，在HA领域的应用则要逊色一些。主要原因恐怕是价钱还未能被普通家庭所接受。LonWorks最大的优点是开放性、分散性和互换性，一个具有LonMark标志的节点(Node)可以互操作(Interoperability)亦可以互替换，但这是有条件的，要达到这个目的，必须尽量把产品细分，增加控制器变量的种类，简化及固定操作流程等，这又不可避免地增加控制节点的数量从而增加了系统的成本。在家庭自动化中最基本的操作应该是电器的开关控制，而最方便可行的方法是使用可控插座，例如美国的Leviton便生产一款有LonMark标志的插座，但其价钱相当贵，一个包含有三个8位微处理器(Neuron芯片)或一个32位微处理器(Motorola 68360)的插座，试想会有多少个家庭负担得起？

于是国内的一些LonWorks产品开发厂家便反其道而行之，来一个功能上的大集成，例如将Neuron芯片在外围上尽量扩展，或者用一般的微处理器作控制部件，用Neuron芯片作通信部件，设计成类似多功能控制盒之类的产品。这个方法确实可以起到降低成本，增加功能的作用，其做法无可非议。但这类产品多数没有互操性和互换性，更加谈不上申请LonMark标志，使LonWorks的开放性光彩尽失。这样的系统亦失去了分布式系统灵活多变的优点，成了一个中央式或半中央式控制系统了。

纵观上述几类家庭自动化网络技术都是各有其特点，但没有哪一个

能完全适合中国普通家庭使用，市面上可供选择的产品亦非常有限，这些因素或多或少地影响了这些技术在国内 HA 方面的进一步普及和应用。

5 本发明的目的在于提供一种智能家庭控制总线，其具有结构简单、成本低和功能灵活的优点。

发明内容：

10 本发明一种智能家庭控制总线系统，拥有至少一个指令器和至少一个执行器，其中控制信号由指令器输入，经总线传送至执行器，再由执行器实现操作输出，其特征在于：

1) 指令器是控制信号的输入者和数据包的发送者，指令器至少含有一个输入装置，可以用来输入控制信号；指令器同时至少含有一个事件数据包，该数据包在系统中对事件具有唯一性；当输入装置检测到有信号输入时，指令器便将该数据包发送至总线之上；

15 2) 执行器是数据包的接收者和操作的执行者，执行器至少含有一个输出装置，可以用来执行某些操作的输出；执行器同时至少含有一个可读写的存储器，该存储器主要用来存放编程过程中产生的数据包；

3) 执行器在接收到来自总线的由指令器发出的事件数据包后，便会将该事件数据包与执行器编程时存放在可读写存储器内的数据包；相比较，如果两数据包相同或相关，执行器便执行与该数据包相对应的操作输出。

其中指令器的数据包事先已经设定并存放在指令器内的非易失性存储器当中，即是指令器在使用时不需编程。

25 其中执行器还进一步包含有一个编程按钮，使用该编程按钮可以令执行器进入编程状态，亦即是说执行器可由用户编程。

其中指令器与执行器之间的控制逻辑关系的建立依赖于用户对执行器的编程，而与指令器的数据包的含义无关。

其中执行器的可读写存储器是指包含在执行器内的数据存储或非易失性存储器。

30 其中执行器还进一步包含有一个指示灯，用来显示执行器的工作

作状态，至少用来指示执行器是否已进入编程状态。

其中指令器内的非易失性存储器，是指包含在指令器内部的特定的中央处理器或微处理器内部的只读存储器。

其中编程状态是指这样的一种状态，在这种状态中，执行器将进一步等待接收来自指令器的事件数据包和任何执行器能接受和解释的来自编程装置的操作指示。

其中来自指令器的事件数据包，和来自编程器的相关操作指示，在编程完成后均会以数据的形式存放在执行器的可读写存储器之内。

其中指令器是采用类曼彻斯特编码方式将数据包发送到总线上，该编码方式是指二进制数字 1 用一个脉冲组成，而二进制数字 0 则用二个宽度相同而方向相反的脉冲组成。

其中二进制数字 1 的一个脉冲的时间与二进制数字 0 的二个脉冲的时间之和相等，亦既是二进制数字 1 和 0 有相同的传输时间和传送速率。

其中不同类型的事件数据包都具有相同的长度即 10 个字节，其中第 1 个字节为数据类别代码，在第 2 至第 9 个字节当中包含有节点身份代码，而第 10 个字节则为整个数据包的校验和。

附图说明：

为进一步说明本发明的技术内容，以下结合附图对本发明作一详细描述，其中：

- 图 1 是本发明系统结构示意图；
- 图 2 是现有常规系统结构逻辑图；
- 图 3 是本发明系统结构逻辑图；
- 图 4 是本发明系统工作 / 编程原理示意图；
- 图 5 是本发明编程原理逻辑图；
- 图 6 是本发明数字编码脉冲波形图；
- 图 7 是本发明数据包时序图。

具体实施方式

和 LonWorks 一样，本发明智能家庭控制总线（又叫 ApBus 总线）

亦是一种全分布式的智能控制网络技术，智能家庭控制总线产品具有双向通信能力，以及互操作性和互换性，其控制部件都可以编程。典型的智能家庭控制总线采用双绞线总线结构，各网络节点（Node）可以从总线上获取供电（12-24V / DC），亦通过同一总线实现节点间的无极性（Polarity Insensitive）、无拓扑逻辑限制的互连（Free Topology Wiring）和通信，智能家庭控制总线的信号编码方式与 LonWorks 相似（Manchester-5 Like），信号传输速率和系统容量则与 CEBus 一样，分别为 10Kbps 和 4G。典型的系统结构如图 1 所示。

在分析智能家庭控制总线的技术特点之前，我们首先分析一下其他控制系统，并以 LonWorks 为例作一个比较。典型的 LonWorks 逻辑结构如图 2 所示。图中把节点分为两大类，即执行器（Listener，广义上亦称为听者）和指令器（Announcer，广义上亦称为讲者），其中 a 是指令器，b 是执行器，c 和 d 既是执行器又是指令器。LonWorks 的地址编码较为复杂，有物理地址（Neuron_ID），亦有逻辑地址（Domain, Subnet, Group），有源地址（Source），亦有目标地址（Destination）。我们假设 a 是一个按钮（传感器），b 是一个继电器开关（执行器），当 a 按下时 b 动作。根据 LonTalk 协议，a 会送出一个数据包，内含 b 的地址及事先定义的指令，以令 b 可进行指定的操作。在该例中，a 是指令器，是主动一方；b 是执行器，是被动的一方，b 只能按 a 的指令行事，如果要让 b 执行不同的操作，a 必须发出不同的指令。同样地，如果 a 要控制更多的节点，原则上亦要发出更多的指令，以实现对不同对象（执行器）进行不同的操作。一句话，系统功能编程是针对 a 来进行的。

图 3 是本发明智能家庭控制总线系统结构逻辑图。我们发现，智能家庭控制总线结构较为简单。为了方便描述，这里亦把节点分为执行器（L）和指令器（A），不过由于系统并未定义目标地址，因此数据包会送到每一个节点，智能家庭控制总线系统在逻辑上是一个整体。由于不分组（Grouping），系统没有需要由用户设定的逻辑地址，节点的身份则主要由工厂预设的代码（Node_ID）来区分和识别。假设 a 是按钮（指令器），b、c、d 是执行部件（执行器），当按钮按下时，a 会送出一个带有自身身份代码的数据包，b、c、d 以及其它网络节点都会同时收到该

数据包并同步执行各自不同的动作。

一个数据包要令多个节点执行多个动作，通常的做法是在该数据包中指定所有被控节点的地址和所要执行的操作指令，如果事情是这样，则智能家庭控制总线便失去独到之处了。事实上是，在智能家庭控制总线系统中，a 所送的数据是固定的，它不是一个真正的指令，它代表的
5 的是一个事件的发生（例如按钮被按下），至于该事件会引致系统怎么样的反应并不是 a 所要考虑的事情。a 的职责是，将该事件以独有（即唯一）的数据包的方式传送到总线之上，再通过总线将该事件（数据包）通知总线上的所在其他节点。

10 那么，智能家庭控制总线又如何去定义一个事件，是不是象 CEBus 或 LonWorks 那样对所有的事件、控制对象和操作都来一个详细的划分？答案是否定的。对控制过程变量种类和参数的过分细微的划分，必会增加系统节点的种类和数目，使系统的成本增加，同时亦因为流程过于固定而削弱了系统的功能并且亦会降低系统的灵活性。为了解决这个问题，
15 智能家庭控制总线进行了前所未有的尝试，没有去定义任何事件，只是要求事件所对应的数据包必须具有唯一性，同时为了方便系统的管理，智能家庭控制总线对数据的种类和长度都作了适当的定义，仅此而已。

20 那么，智能家庭控制总线又如何让接收到数据包的节点明白所发生的事情，并产生正确的反应呢？这正是智能家庭控制总线技术的关键所在，但其原理却又比想象中简单。原来智能家庭控制总线的每一个执行部件（执行器）都有智能，即具有学习和记忆功能，只要在系统组成后的编程设定过程中，让相关的执行部件将事件重演一次，编程（学习）过程便宣告完成，执行部件会将有关事件代码及对事件的反应（操作）
25 记入自身的存储器（EEPROM）中，这些数据包在节点中即使断电亦可保留超过 100 年。我们看到，智能家庭控制总线与 LonWorks 等现有网络系统的一个最大的不同之处是智能家庭控制总线针对执行器而非指令器编程，指令器的一句话可以令多个执行器跟随动作，智能家庭控制总线无疑具有较高的智慧和效率，这便是家居控制总线被冠以“智能”的
30 由来。

那么，与 LonWorks 等系统相比，智能家庭控制总线的互操性和互换性又如何呢？其实我们从上述的分析中已有答案。因为执行器具有学习能力，互操性和互换性问题便可轻易解决，任何部件在被更换之后只需要让相关的执行部件（执行器）重新学习一次便可认识该新“搭档”并知道如何跟它配合，“隔膜”在智能家庭控制总线系统的节点之间并不存在。

对用户来说，智能家庭控制总线似乎很简单（任意接驳），很灵活（具有智能），功能很强（取决于执行部件），有互操作性（要学习一次）和互换性（要重学一次）。但从另一个角度来看，为了达到上述目的，智能家庭控制总线必须解决更多的技术难题，例如信号与直流混合传输问题；无极性接驳问题；混合拓扑布线问题；信号优先权问题；无屏蔽工频干扰问题等等。原来对这些问题智能家庭控制总线都有其独到的解决方案，因篇幅所限，这里就不介绍了。值得一提的是，对于网络系统最基本的问题，即信号传输过程中的数据冲撞问题，智能家庭控制总线有新的突破，它是在以太网所使用的 CSMA / CD 信号传输控制技术的基础上作了关键性的改进，使各节点可以实现无冲撞的多节点随机数据传输，这项技术进一步提高了系统的通信效率和数据的可靠性，智能家庭控制总线将该方案定义为 CSMA / CF (Carrier Sense Multiple Access with ColliSion-Free)。

从上述我们可以看到，智能家庭控制总线是对现有 HA 技术作出改进后推出的一个家庭总线技术，其目的是使 HA 技术在网络化的同时再进一步简单化和实用化，以方便设计及生产出符合我国国情的，能为普通百姓家庭接受的智能家居产品。下面我们就来看看智能家庭控制总线产品的特点。

与智能家庭控制总线兼容的产品都具有“ApBus Compatible”标志，具有该标志的产品都具有互操性和互换性，可以任意组合使用。该标志的拥有者是 Actpro Int’l (HK) Ltd。由于智能家庭控制总线的接口相对简单，人们甚至可以把智能家庭控制总线的技术应用到电工产品例如开关插座之中，给这些家庭中最古老而又用得最多的电工产品带来新的意义。例如在最近推出的智能家庭控制总线产品中便有这样一类带智能的

开关插座，可以用一个开关控制多个灯光电器，而同一个灯光电器又可以在不同地方用不同的开关去控制，即是编程和联动，非常有趣。同时这些开关插座可接受其它智能家庭控制总线产品例如定时器、遥控器的控制，以及电话、电脑的控制，非常方便灵活，而价钱又在普通家庭可接受的范围之内。

目前在市面上智能家庭控制总线产品从功能上可分为五类，下面将逐类分析：

第一类是基本控制产品，主要包括总线电源供应器，电话遥控 / 报警接口，无线遥控接口和安防控制接口等，这些产品是智能家庭控制总线系统的基础，为其他智能家庭控制总线控制产品提供一个接口平台。如果用户在房子装修时把这几个产品考虑在内并按一定的要求布线，那么这个家居智能化的将来便有了保障，因为住户可以随时根据自己的预算来增加控制的内容，从而可以实现有计划有步骤的系统扩展和升级。

第二类是灯光控制产品，以轻触式电子开关 / 调光器为代表。这一类产品有一个特色就是其外观尺寸与常规开关相似，可以直接替换原有的开关便可实现照明系统的智能化改造，可以调光，也可以遥控，更可以用来产生不同的灯光组合以满足不同的照明需要。万一一个别电子开关有故障，所受影响的也仅仅是该开关所连接的那一部分，即使没有备用产品也可以换回原来的开关便可恢复原来的功能，实现手动操作。

第三类是电器控制产品，这类产品即是前面所提到过的可控插座。与第二类的产品相似，这类产品也可以通过直接替换原有的插座来实现对电器控制的智能化改造。将电饭锅、热水炉、空调机、洗衣机等电器的电源接在该插座上，用户便可以通过电子开关、遥控器或电话等来控制这些电器，非常方便。该可控插座分自动和手动两种操作方式，万一系统失灵，亦可以用手动去操作。

第四类是安防控制产品，这类产品主要包括人体红外传感器，煤气泄漏传感器，烟雾火灾传感器和多功能报警器等。这些产品的安装和使用比其他同类产品都要简单，因为有智能家庭控制总线兼容标志，可以直接连接于在总线之上，并可利用基本系统中的电话遥控 / 报警接口实现远程报警。另外，通过基本系统中的安防控制接口，普通的传感器也

可以用在智能家庭控制总线系统之中。

第五类是其他控制产品，这类产品主要用于实现一些较特殊或较高级的控制。例如智能家庭控制总线提供的一款万用遥控器除了可以控制几百款不同的电视机、录影机和音响设备之外，还可以控制家中所有的智能家庭控制总线产品，起到集中控制的作用。另外有一款万用定时器，可以设定 100 组时间，足能够为家里的智能家庭控制总线控制产品制定一个详细的执行时间表，例如早上六点钟开灯，六点十分开电视，六点二十分煮咖啡，六点三十分开多士炉等，一切都可以由用户设定。

智能家庭控制总线的出现对我国家庭自动化行业无疑是件值得庆贺的事情，至少可以给用户多一个选择。智能家庭控制总线是一个新事物，要人们广泛认识和使用，或许需要一定的时间，但随着经济的发展，人们生活水平的提高，以及对家庭自动化认识的加深，智能家庭控制总线走向千家万户的梦想，是有希望实现的。

下面我们将通过一个具体实施例（如图 4 所示）说明一个完整的系统操作以及系统编程的过程。

在图 4 中，节点 A 是指令器，用一个按钮组成一个输入装置，该装置有二个逻辑状态（开路和闭路），分别对应着两个事件（按钮按下和释放），每个事件都有一个唯一的数据包（D1 / D2）来对应，数据包在总线上的传输代表着事件的发生。根据智能家庭控制总线的定义，指令器 A 无需编程，数据包 D1 和 D2 在 A 出厂时预设，并存放在非易失性存储器内，见表 1 所示。

表 1 指令器 A 的事件与数据包的关系

| 事件 | 数据包 |
|------|-----|
| 按钮按下 | D1 |
| 按钮释放 | D2 |

我们再来看看另一个节点 L，图 4 中 L 是执行器，L 有一个输出装置，即由一个继电器去控制一个灯的开和关。我们假设用 A 去控制 L，当 A 按钮按下时 L 继电器吸合，L 灯点亮，而当 A 按钮释放时继电器触

点开路，L 灯熄灭。要实现这一逻辑关系，必须对 L 进行编程，将数据包 D1 和 D2 存入 L 的工作表中，并指定 D1、D2 与灯亮、灭之间的关系，具体见表 2 所示。

5 表 2 执行器 L 的数据包与操作的关系

| 数据包 | 操作 |
|-----|----|
| D1 | 开灯 |
| D2 | 关灯 |

在图 4 中，“ApBus 总线”是一种物理传输媒介，指令器的事件数据包可通过该媒介传输给执行器，在系统中最常用的传输媒介是双绞线，图 4 上画的亦是一段双绞线。图中指令器 A 和执行器 L 都有二个接线端子，分别接到该双绞线上，即是 A 和 L 都同时并联到 ApBus 总线上，这是本发明智能家居总线系统的一个典型连接方法。在应用上，ApBus 总线还会使用不同的传输媒介，例如电力线、平行线、同轴电缆、光纤以及红外、无线电波等媒介。使用不同的传输媒介，A 和 L 与其连接的方法也不一样，但系统的工作原理和编程方法是一样的。

15 当执行器 L 收到一个数据包时，便会跟其工作表中的数据包相比较，如果数据包与 D1 相同，则执行开灯操作；如果数据包与 D2 相同，则执行关灯操作。在实际应用上，我们存放入执行器 L 工作表中的数据包不一定是 D1 和 D2，有时会将 D1 和 D2 简化，但简化后的数据包与 D1 和 D2 存在着某种对应关系，使得执行器 1 能判别出所收到的数据包是
20 与 D1 相关还是与 D2 相关，从而确定要执行哪一种操作。

我们发现，数据包 D1 和 D2 在系统中起只起到传送事件信息的作用，是一个无意义而又不重复的字串，如果把该数据包忽略则系统操作逻辑如表 3 所示。

25 表 3 智能家庭控制总线系统操作逻辑关系（从 A-L）

| 指令器 A | 执行器 L |
|-------|-------|
| 按钮按下 | 开灯 |

| | |
|------|----|
| 按钮释放 | 关灯 |
|------|----|

从以上分析中我们发现，数据包本身并不是系统中最重要东西，而系统编程最重要的反而是如何将 A（指令器）的事件数据包置入 L（执行器）的工作表中，并指定这些数据包与 L 所要执行的动作的关系。为了达到这一目的，智能家庭控制总线在执行器上加一个编程按钮（或其它输入装置），当该编程按钮被按下（或输入装置被触发）时，节点 L 进入编程状态。如果这时 A 有事件发生（按钮按下或释放），相应的数据包（D1 和 D2）便通过总线传送到 L，而处于编程状态的 L 节点会自动将该数据包存入自身的工作表（RAM、EEPROM 或 FLASH-ROM 等可读写存储器）中，继而等待接收与该数据包（事件）相关的操作指示。这些操作指示可通过二个途径传送给 L，一是通过总线，二是直接与 L 相接。无论通过那一个途径，发出操作指示的设备这里将统称为“编程装置”或“编程器”。

如果通过总线编程，则必须有“数据接口”与总线连接，该数据接口是编程器与待编程节点 L 间的一个数据桥梁。如果直接对 L 编程，则 L 亦要有一个与编程器相连的数据通道。编程器可以是电脑、电话、数据键盘或其它可输入数据的装置，这些装置发出的数据即为“操作指示”，这些操作指示被待编程节点 L 接收并解释出一系列指定动作，例如开 / 关、上 / 下、升 / 降、快 / 慢，回路选择，数据保存或放弃等等。一句话，节点 L 必须能接收到编程器的数据，并与编程器有共同的语言。编程器的目的，是在待编程节点的工作表中跟随事件数据后输入相关的操作数据。有关编程原理逻辑图如图 5 所示。

首先由编程按钮发出编程指示，然后由待编程节点接收事件数据，最后由编程装置发出操作指示从而完成整个编程过程，这是智能家居总线常规的编程法，但事实上三者的时间顺序可以任意编排，只要这三个步骤存在，能确定因（事件）果（操作）关系，编程便可实现。

同样地，编程指示除了可用编程按钮来产生之外，亦可用其它输入装置，例如开关、红外触发等来实现。另外还可以将编程指示放在编程装置中，通过总线或其它数据传输通道触发待编程节点而令该节点进入

编程状态。在这种情况下，待编程装置必须懂得解释该指令，如果是通过总线触发，该编程指示还必须包含有待编程节点的身份代码，以确定是那一个执行器接受编程（如果一个系统内有多个执行器的话）。

由上述的分析我们发现，在总线中流通的数据除无含义的事件数据包外，还包括有含义的数据，例如编程指示数据包（表 15）和操作指示数据包（表 13）等。

事实上，智能家居控制总线将数据包作了一个分类，具体如表 4—23 所示。

10

表 4 通用事件数据包格式(第 0 类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|---------|------------|-----|
| 0000-1111/ '0F' H | 0000-0000/ '0' H | 8×6=48 位 | 8×0=0 位 | 8 位 | 8 位 |

表 5 通用事件数据包格式(第 1 类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|---------|------------|-----|
| 0001-1110/ '1E' H | 0000-0000/ '0' H | 8×5=40 位 | 8×1=8 位 | 8 位 | 8 位 |

表 6 通用事件数据包格式(第 2 类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|----------|------------|-----|
| 0010-1101/ '2D' H | 0000-0000/ '0' H | 8×4=32 位 | 8×2=16 位 | 8 位 | 8 位 |

15

表 7 通用事件数据包格式(第 3 类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|-------------|------------|------------|--------|------------|-----|
| | | | | | |

| | | | | | |
|----------------------|---------------------|----------|----------|-----|-----|
| 0011-1100/ '3C' H | 0000-0000/ '0' H | 8×3=24 位 | 8×3=24 位 | 8 位 | 8 位 |
|----------------------|---------------------|----------|----------|-----|-----|

表 8 通用事件数据包格式(第 4 类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|----------|------------|-----|
| 0100-1011/ '4B' H | 0000-0000/ '0' H | 8×2=16 位 | 8×4=32 位 | 8 位 | 8 位 |

表 9 通用事件数据包格式(第 5 类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|----------|------------|-----|
| 0101-1010/ '5A' H | 0000-0000/ '0' H | 8×1=8 位 | 8×5=40 位 | 8 位 | 8 位 |

5

表 10 通用事件数据包格式(第 6 类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|----------|------------|-----|
| 0110-1001/ '69' H | 0000-0000/ '0' H | 8×0=0 位 | 8×6=48 位 | 8 位 | 8 位 |

表 11 专用数据包格式(时间类)

| 数据包类别 代码 | 系统保留 字节 | 年/月/日 | 时/分/秒 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|----------|----------|------------|-----|
| 0111-1000/ '78' H | 0000-0000/ '0' H | 8×3=24 位 | 8×3=24 位 | 8 位 | 8 位 |

10

表 12 专用数据包格式(功能类)

| 数据包类别 代码 | 系统保留 字节 | 功能类别 | 控制指令 | 数据流控制 字 | 校验和 |
|-------------|------------|------|------|------------|-----|
|-------------|------------|------|------|------------|-----|

| | | | | | |
|----------------------|----------------------|-----|-----|-----|-----|
| 1000-0111/ '87' H | 8×5=40Bits, 全 '0' | 8 位 | 8 位 | 8 位 | 8 位 |
|----------------------|----------------------|-----|-----|-----|-----|

表 13 专用数据包格式(操作指示)

| 数据包类别 代码 | 系统保留 字节 | 操作指示 | 参数 | 数据流控制 字 | 校验和 |
|----------------------|--------------------|------|-----|------------|-----|
| 1001-0110/ '96' H | 8×5=40 位, 全 '0' | 8 位 | 8 位 | 8 位 | 8 位 |

表 14 专用数据包格式(反馈类)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|----------|------------|-----|
| 1010-0101/ 'A5' H | 0000-0000/ '0' H | 8×4=32 位 | 8×2=16 位 | 8 位 | 8 位 |

5

表 15 专用数据包格式(编程指示)

| 数据包类别 代码 | 系统保留 字节 | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|------------|----------|------------|-----|
| 1011-0100/ 'B4' H | 0000-0000/ '0' H | 8×4=32 位 | 8×2=16 位 | 8 位 | 8 位 |

表 16 专用数据包格式(保留)

| 数据包类别 代码 | 系统保留 字节 | 保留 | 保留 | 数据流控制 字 | 校验和 |
|----------------------|---------------------|----------|----------|------------|-----|
| 1100-0011/ 'C3' H | 0000-0000/ '0' H | 8×3=24 位 | 8×3=24 位 | 8 位 | 8 位 |

10

表 17 自定义数据包格式(用户类)

| 数据包类别 代码 | 数据长度 (N) | 节点身份 代码 | 用户自定义区 | 数据流控制 字 | 校验和 |
|-------------|-------------|------------|--------|------------|-----|
|-------------|-------------|------------|--------|------------|-----|

| | | | | | |
|----------------------|-----|----------|----------|-----|-----|
| 1111-0000/ 'F0' H | 8 位 | 8×4=32 位 | 8×(N-8)位 | 8 位 | 8 位 |
|----------------------|-----|----------|----------|-----|-----|

表 18 数据类数据包格式(第 1 类)

| 数据包类别 代码 | 数据格式及子 类别 | 节点身份 代码 | 数据 | 数据流控制 字 | 校验和 |
|----------------------|-------------------|------------|---------|------------|-----|
| 1110-0001/ 'E1' H | 00-B5B4B3- 001 | 8×5=40 位 | 8×1=8 位 | 8 位 | 8 位 |

表 19 数据类数据包格式(第 2 类)

| 数据包类别 代码 | 数据格式及 子类别 | 节点身份 代码 | 数据 | 数据流控制 字 | 校验和 |
|----------------------|-------------------|------------|----------|------------|-----|
| 1110-0001/ 'E1' H | 00-B5B4B3- 010 | 8×4=32 位 | 8×2=16 位 | 8 位 | 8 位 |

5

表 20 数据类数据包格式(第 3 类)

| 数据包类别 代码 | 数据格式及 子类别 | 节点身份 代码 | 数据 | 数据流控制 字 | 校验和 |
|----------------------|-------------------|------------|----------|------------|-----|
| 1110-0001/ 'E1' H | 00-B5B4B3- 011 | 8×3=24 位 | 8×3=24 位 | 8 位 | 8 位 |

表 21 数据类数据包格式(第 4 类)

| 数据包类别 代码 | 数据格式及 子类别 | 节点身份 代码 | 数据 | 数据流控制 字 | 校验和 |
|----------------------|-------------------|------------|----------|------------|-----|
| 1110-0001/ 'E1' H | 00-B5B4B3- 100 | 8×2=16 位 | 8×4=32 位 | 8 位 | 8 位 |

10

表 22 数据类数据包格式(第 5 类)

| 数据包类别 代码 | 数据格式及 子类别 | 节点身份 代码 | 数据 | 数据流控制 字 | 校验和 |
|-------------|--------------|------------|----|------------|-----|
|-------------|--------------|------------|----|------------|-----|

| | | | | | |
|----------------------|-------------------|---------|----------|-----|-----|
| 1110-0001/ 'E1' H | 00-B5B4B3- 101 | 8×1=8 位 | 8×5=40 位 | 8 位 | 8 位 |
|----------------------|-------------------|---------|----------|-----|-----|

表 23 数据类数据包格式(第 6 类)

| 数据包类别 代码 | 数据格式及 子类别 | 节点身份 代码 | 数据 | 数据流控制 字 | 校验和 |
|----------------------|-------------------|------------|----------|------------|-----|
| 1110-0001/ 'E1' H | 00-B5B4B3- 110 | 8×0=0 位 | 8×6=48 位 | 8 位 | 8 位 |

其中表 4-10 为通用事件数据包，从第 0 类至 6 类共分 7 类，另外
5 还有 6 类专用数据包（表 11-16），6 类数据类数据包（表 18-23）和 1 类自定义数据包（表 17）。后三类的数据包均有含义或由用户自定义。

从上述图表中我们知道，除自定义数据外，其它所有的数据包都由
10 个字节组成，其中第 1 个字节为数据包类别代码，第 2 个字节为系
10 统保留字节，第 3 至第 8 个字节为节点身份代码或参数或其它用途，第
9 个字节为数据流控制字，最后一个字节（第 10 字节）为较验和。

最后介绍一下智能家庭控制总线的数字编码方式，即是说用什么样的
脉冲波形组成一个数据。智能家庭控制总线要用类似曼彻斯特
(Manchester-Like) 的编码方式，其脉冲波形如图 6 所示，图中我们可
15 以看到，智能家庭控制总线的的一个字节由 8 个数据位，一个奇偶较验位
和一停止位组成，而在每一个数据包的开始还含有一个开始位，所有位
的宽度（即时间长度“T”）都是一样的，二进制数字“1”由一个时间
长度为 T 的正向或反向脉冲组成，而二进制数字“0”则由二个时间长度
为 $1/2 T$ 的方向相反的脉冲组成，总长度亦为 T。

20

表 24 智能家庭控制总线的数字编码时间参数(标称值)

| 开始位/停止位 | 数字 '0' | 数字 '1' |
|-----------|-----------------------|-----------|
| T(100 μs) | T/2(50 μs)+T/2(50 μs) | T(100 μs) |

表 2 4 是数字编码时间数据 T 的标称值，智能家庭控制总线为 1 0 0 μ s，如果是二进制数字“0”则由二个 5 0 μ s 组成。由于时间 T 固定，因此智能家庭控制总线数据传送速率亦固定，为 1 0 Kbps（即每秒 1 0 K 个位元）。

5

表 25 智能家庭控制总线的数字编码时间参数(发送器)

| 开始位/停止位 | 数字‘0’ | 数字‘1’ |
|---|------------------------------------|-----------------------------------|
| 90 μ s \leq 开始位/停止位 <110 μ s | 40 μ s \leq T/2 < 60 μ s | 90 μ s \leq T < 110 μ s |

表 2 5 是智能家庭控制总线发送器在发送数据时的时间精度要求，表中我们可以看出，智能家庭控制总线要求每个脉冲的误差都应小于 1 0 0 μ s，无论是二进制“0”还是二进制数字“1”。

表 26 智能家庭控制总线的数字编码时间参数(接收器)

| 开始位/停止位 | 数字‘0’ | 数字‘1’ |
|---|------------------------------------|-----------------------------------|
| 75 μ s \leq 开始位/停止位 <150 μ s | 25 μ s \leq T/2 < 75 μ s | 75 μ s \leq T < 150 μ s |

表 2 6 是智能家庭控制总线数接收器在接收数据时每个脉冲容许的时间宽度。我们发现，智能家庭控制总线数在接收数据时的宽容度比发送数据时要大得多，几乎有 5 0 % 的容错能力。

图 7 是智能家庭控制总线数数据包的时序图，图中我们发现，每个数据包之间都有一个静止间隙，并以一个负脉冲作为一个数据的开始。

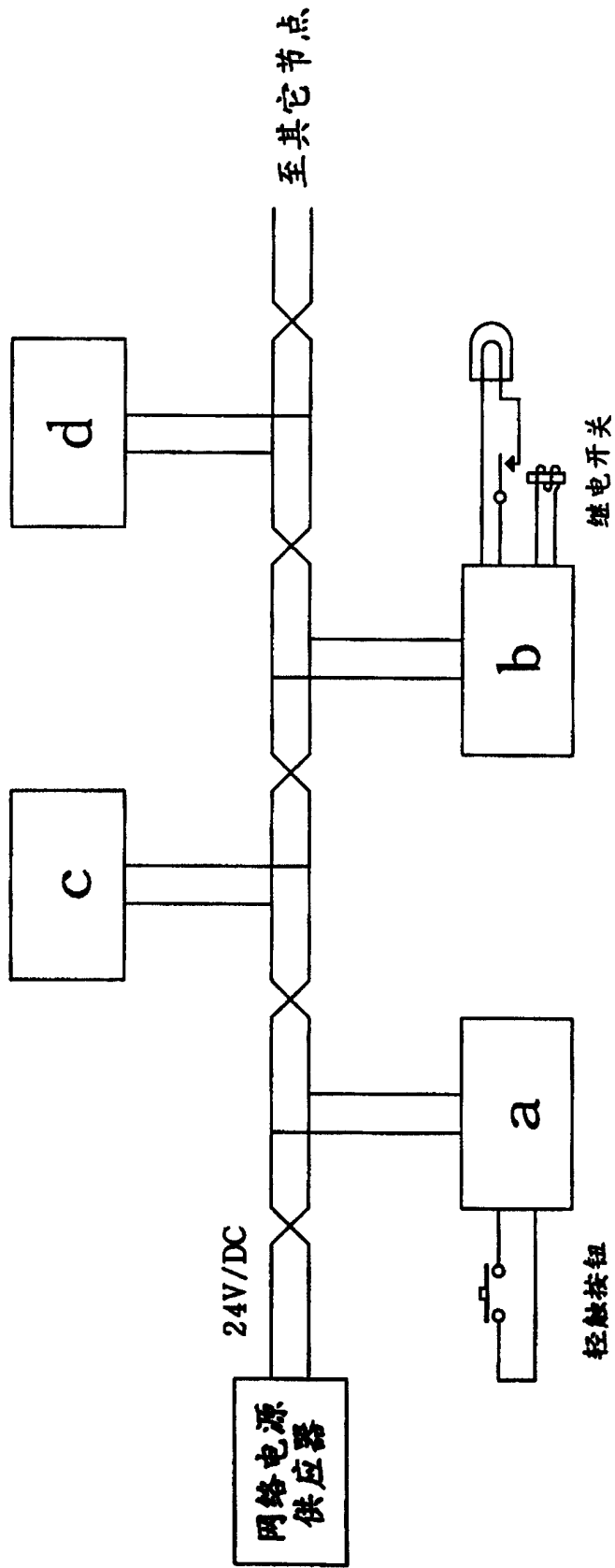


图 1

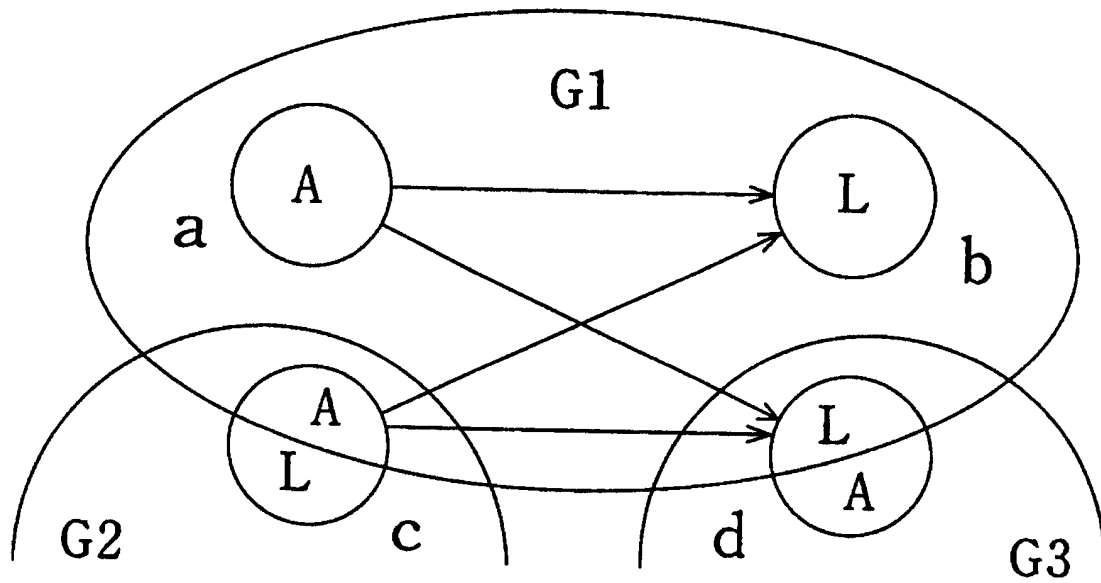


图 2

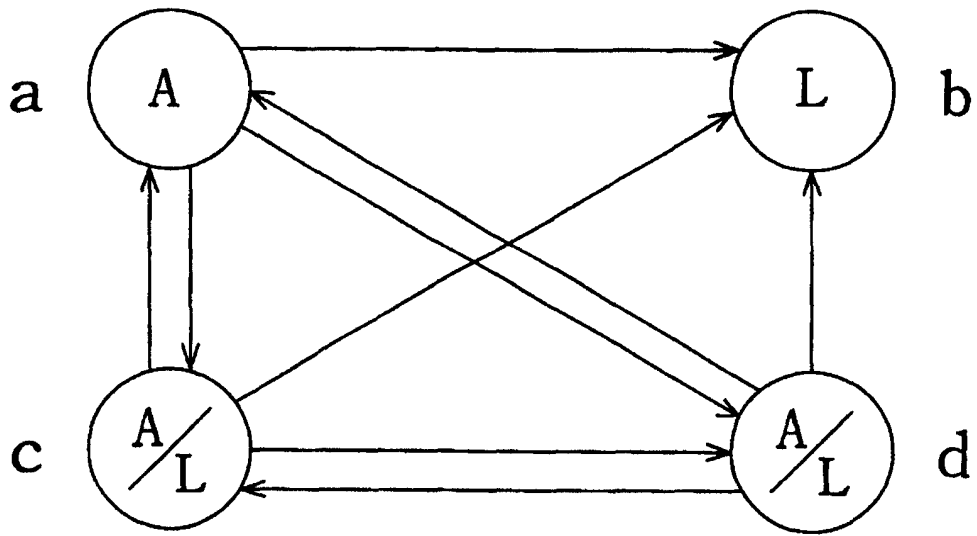


图 3

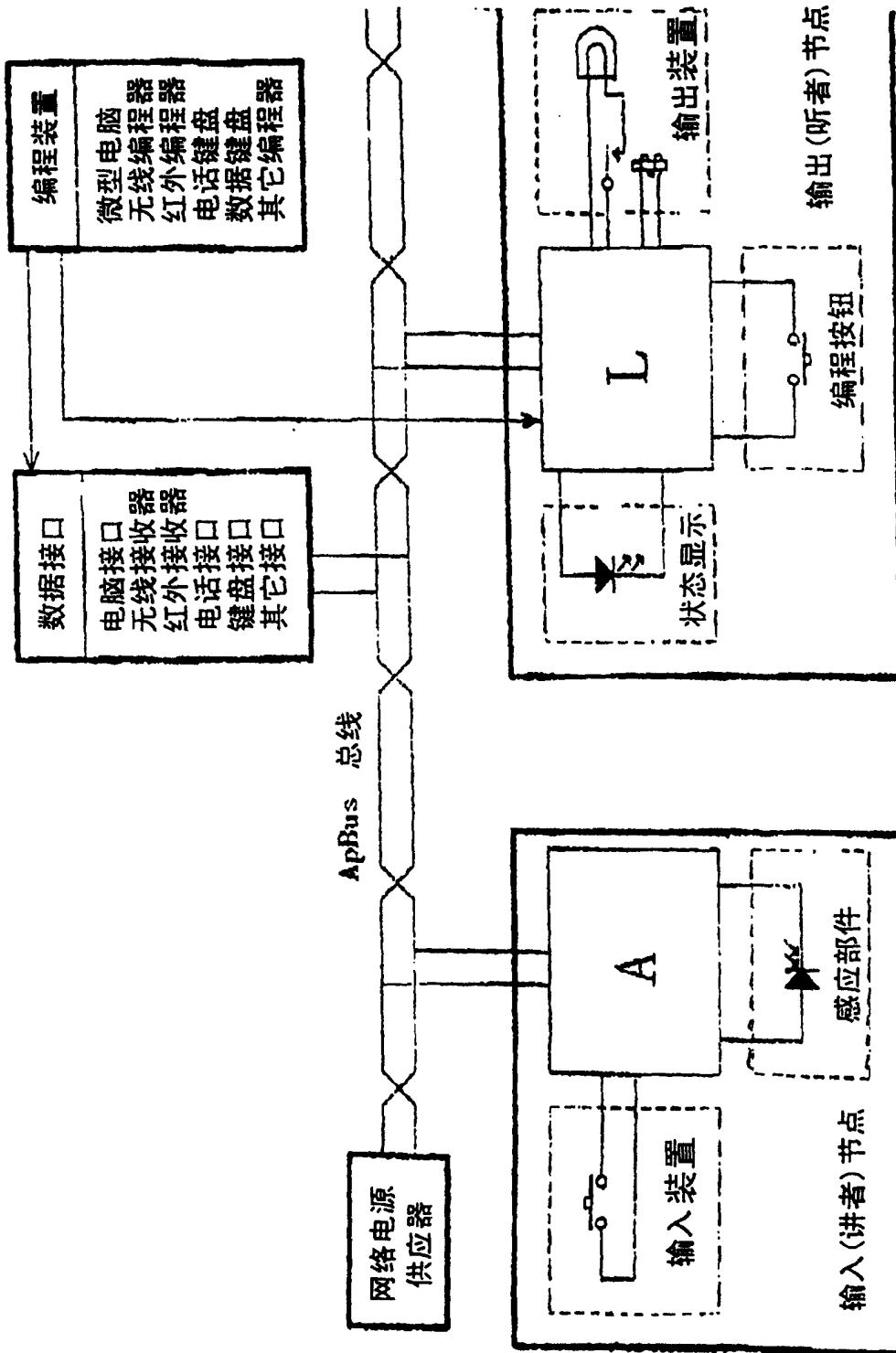


图 4

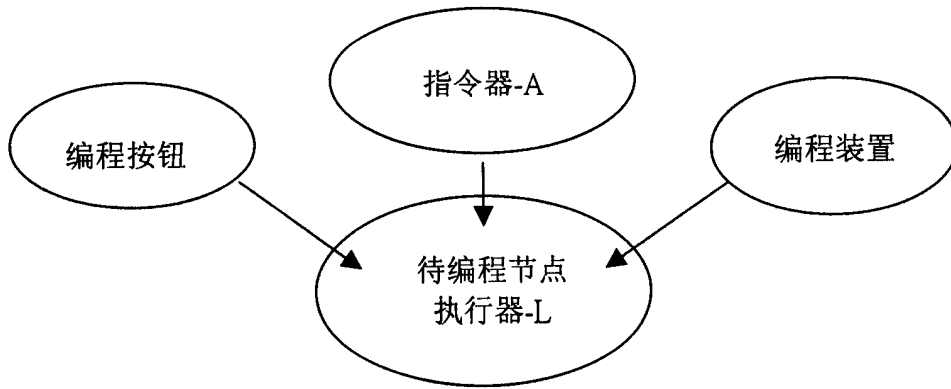


图 5

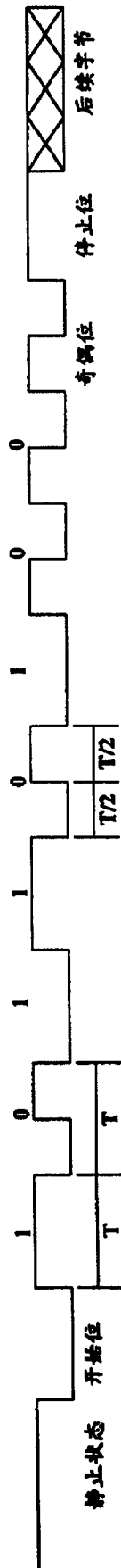


图 6



图7