

## LIN总线及其在智能家居

## 控制系统中的应用

颜自勇, 王辉堂, 金凯鑫, 陈文芾  
(厦门大学 机电系, 福建 厦门 361005)



第一作者: 颜自勇  
(1982), 男, 硕士研究生, 研究方向为单片机嵌入式系统、现场总线控制技术。

**摘要:** 介绍了 LIN 总线协议, 及其在智能家居控制系统中的应用。LIN 总线是新近出现在汽车行业的一种串行通信总线, 其协议对硬件的依赖程度低, 可基于普通单片机的通用串口等硬件资源以软件方式实现, 成本低廉, 可广泛应用于汽车行业以外对实时性要求不高的其他领域。利用 LIN 总线在智能家居系统中构建内部网络, 以实现对各电器节点的控制, 相较于传统控制方式具有诸多优势, 将更为用户所接受。

**关键词:** LIN 总线; 智能家居; 软件实现

中图分类号: TU 855 TP273<sup>+</sup>.5 文献标识码: A 文章编号: 1001-5531(2006)

11 0033 04

## Local Interconnect Network Bus and Its Application in Control System of Smart Home

YAN Zi-yong, WANG Hui-tang, JIN Kai-xin, CHEN Wen-xiang

(Department of Mechanical & Electrical Engineering, Xiamen University, Xiamen 361005, China)

**Abstract** The LIN bus protocol and its application in controlling system of the smart home were introduced. The LIN bus is a new serial communication bus which is emerging in existing automotive industry with features of protocol less depend upon hardware, implementing low cost software model by using hardware source of standard UART/SCI interface of simple MCU, widely used in other fields with less real time demand etc. Using LIN bus to build internal network of the smart home and implement controlling node of every electrical apparatus will be well received by users for many advantages over the conventional controlling mode.

**Key words** LIN bus, smart home, software realization

### 0 引言

局部互联网络 (Local Interconnect Network, LIN) 协议是新出现的一种新型低成本串行通信总线。它最开始出现于汽车行业, 作为 CAN 的辅助总线, 用于车身控制网络的低端场合, 实现汽车车身网络的层次化, 以降低汽车网络的复杂程度。LIN 的目标应用不需要 CAN 的性能带宽及复杂性的低端系统。LIN 协议的突出特点是协议对硬件的依赖程度低、成本低, 可基于普通单片机的通用串口等硬件资源以软件方式实现, 几乎所有的微控制都具备 LIN 必需的硬件资源, 因此, 可广泛应用于汽车行业以外的其他领域 (如智能家居系统内部网络的数据通信、节点控制等)。本文对 LIN 总线协议进行了研究, 介绍了 LIN 总线在智

能家居系统中家用电器控制的应用。

智能家居系统一般应具有安防报警、自动抄表、家用电器控制功能。目前, 较多的智能家居系统对家用电器的控制采用如下方式来实现: ① 传统的点对点的控制; ② 利用 RS-485 总线; ③ 利用 CAN 总线。第 1 种方式采用集中式控制方案, 由单一的电子控制单元通过大量的线束连接到各电器控制单元。随着电器数量的增加, 庞大的线束会增加系统成本, 降低的可靠性; 而 RS-485 总线是一种物理层标准, 未提出统一的协议标准, 对数据帧无统一的格式标准, 因而无法检测总线上的数据是否出现错误及处理错误; CAN 总线具有高抗电磁干扰性和高传输可靠性, 在很多场合都得到广泛应用, 但其成本也相对较高。对于智能家居系统中的电器控制, 其实时性要求不高, 但节

点数多, 布置分散, 对成本比较敏感, 故用 LN 代替 CAN 可降低成本、提高系统性能<sup>[1]</sup>。

## 1 LN 总线协议

LN 协议标准包括传输协议规范、传输媒体规范、开发工具接口规范和用于软件编程的接口。LN 在硬件和软件上保证了网络节点的互操作性, 并有可预测 EMC 的功能。

### 1.1 LN 总线的特点<sup>[2]</sup>

LN 总线是一种基于普通串行接口的新型串行通信协议, 其目标定位是作为 CAN 总线的辅助总线, 具有以下主要特点:

(1) 采用单主控制器 多从设备的通信模式, 无须总线仲裁, 数据的优先级由主机节点确定, 可以根据需要灵活改变;

(2) 基于普通的 UART /SCI 低成本硬件和软件, 低成本的单总线实现;

(3) 确定性的信号传输, 并保证信号传输的延时时间;

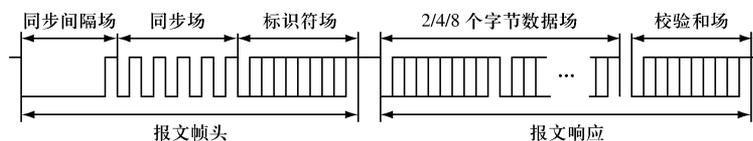


图 1 LN 报文帧

报文帧的报文头包括 1 个同步间隔场、1 个同步场和 1 个标识符场。报文帧的响应则由 3 ~ 9 Byte 场组成: 2、4 或 8 Byte 的数据场和一个校验和场。

LN 网络中的所有通信都是由主机任务发起, 从节点在接受标识符并经滤波后, 其中的一个节点被激活并且开始本消息的应答传输。针对每一个标志符只有一个节点发出响应。信息标志符标志信息的内容而不是目的地址, 该特点使得网络中的数据交换能以多种方式进行: 主节点既可以与每一个从节点进行单独通信(上载和下载数据), 也可以进行网络广播。网络中的数据交换方式有:

(1) 数据由从机到主机通信。指从机对主机的查询帧的报文响应, 只有在主机发送报文头后, 由标识符一致的从机节点应答传输, 其他节点不响应;

(4) 具有监控总线、数据校验和、标识符双重奇偶校验等错误检测功能, 以及对网络中故障节点的错误检测, 保证数据传输的可靠性;

(5) 具有进入睡眠状态和唤醒功能, 可降低系统功耗;

(6) 网络传输速率最高可达 20 kb/s 最大传输距离不超过 40 m; 网络中的节点数受标志符的数量及总线物理特性的限制, 实际应用中不高于 16 个, 即 1 个主机和 15 个从机。

### 1.2 LN 总线的通信机制及帧结构<sup>[3]</sup>

LN 网络中的每个节点都有一个从机任务模块, 主节点还包含一个主机任务模块。LN 总线上的报文传输是由报文帧的格式形成和控制的。报文帧由主机任务向从机任务传送同步和标识符信息, 并将一个从机任务的信息传送到所有其他的从机任务。一个报文帧是由一个主机节点发送的报文头和一个主机或从机节点发送的响应组成, 如图 1 所示。

(2) 数据由主机到从机通信。指主机节点的主机任务发送报文头及其从机任务发送数据场和校验和场, 此通信方式可用来对所有从机节点进行广播;

(3) 数据由从机到从机通信。指 LN 网络中的一个从机节点在向主机节点应答传输后, 通过主机节点中的从机任务将应答传输的数据转发给同一网络中的其他从机节点, 利用主机节点转发应答数据, 主机任务得重新发送报文头。

### 1.3 LN 的物理接口

LN 总线的物理层接口如图 2 所示。LN 物理接口是基于通用 UART /SCI 的硬件接口, 由于 UART /SCI 接口是几乎所有微处理器都集成的模块, 故使用 LN 十分方便。LN 采用单线传输数据, 每个节点可通过上拉电阻与总线进行连接。与上拉电阻串联的二极管可以防止电子控制单元 ECU 在本地电池掉电时通过总线上电。

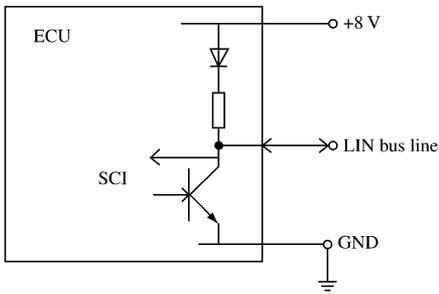


图 2 LIN 总线的物理层接口

#### 1.4 LIN 总线的软件实现

LIN 协议的最大优点在于不需要专用的硬件控制器,只是在 UART/SCI 接口的基础上添加了一套 C 语言 API 函数。用户在开发 LIN 应用程序时,无须直接读写微控制器内部的寄存器,只要

调用相应的 API 函数即可完成同功能的操作。

目前存在两类 API 函数:LIN API 和 Motorola API。本系统采用的是 LIN API。LIN API 共 20 个函数,可以完成包括系统初始化、数据读写、节点上线、离线、中断屏蔽等功能。LIN API 基于信号模式,应用程序代码只能访问信号,不能直接访问整个信息帧数据。一个信号由一到几个数据位构成,驱动程度提供了访问不同长度(1 bit~8 bit, 9~16 bit)信号的能力。应用 LIN API 时需要一个完整的信号描述文件,该文件包含了网络中每一个节点的信号描述,并通过工具将其转换成 C 程序头文件,再与应用程序代码编译连接。表 1 是 LIN 总线软件实现中 LIN API 的几个最常用到的函数调用。

表 1 几个最常用的 LIN API 函数调用

函数功能	函数名	备注
初始化	<code>l_bool l_sys_init( void )</code>	用户在使用其他任何 API 函数前必须第一个调用的函数
接口调用		
接口初始化	<code>void l_ifc_init( l_ifc_handle iii );</code>	用户在初始化后,使用相关的 API 函数接口前要首先调用的函数
接口连接	<code>l_bool l_ifc_connect( l_ifc_handle iii );</code>	调用此函数会将接口 iii 连接到 LIN 网络上
读调用	<code>void l_u8_ptr_rd( l_signal_handle sss, l_u8 * rd_ptr, l_u8 num );</code>	读取信号 sss 的当前值,返回到指定的数组中
写调用	<code>void l_u8_ptr_wr( l_signal_handle sss, l_u8 * wr_ptr );</code>	将信号 sss 的当前值设置成指定的数组变量值
标志调用		
读标志位	<code>l_bool l_flg_tst( l_flg_handle fff );</code>	返回名字为 fff 指定的标志当前的状态
清标志位	<code>void l_flg_clr( l_flg_handle fff );</code>	将名字为 fff 的标志的当前值置零
过程调用	<code>l_u8 l_sch_tick( l_ifc_handle iii );</code>	调用该函数把要发送的进度表项的号码返回,只能在主机节点使用
	<code>void l_sch_sel( l_ifc_handle iii, l_schedule_handle sch, l_u8 en );</code>	调用该函数来设定将要发送的进度表及起始表项,只能在主机节点使用

## 2 LIN 在电器控制中的应用

智能家居控制系统的电器控制是指利用电话、以太网等远程通信方式对一些主要的家用电

器进行监控。对于家居控制系统需要处理的事情包括:与外部网络的远程通信和对内部网络中节点的控制。家用电器的控制对实时性要求不高,但一定要能可靠控制。本文介绍的智能家居系统

的电器控制模块采用电话与外部网络通信,接收用户远程命令,经过控制器的处理后,利用 LIN 总线构成的内部网络对各个家用电器节点进行监控,其电器控制模块结构框图如图 3 所示。

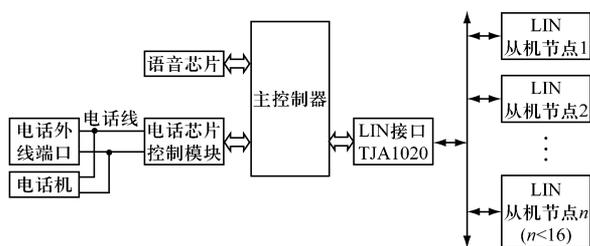


图 3 智能家居系统电器控制模块结构框图

用户通过拨打户内电话,在无人接听一段时间后,主控制器会自动接通电话,进入电器控制界面,根据预置的语音提示,用户可以对电器节点的状态进行查询及进行启动或关闭控制。

使用 LIN 总线进行通信,可在主节点无需对从节点控制时,使所有的从节点进入睡眠模式,降低系统的功耗。主控制器在接收到远程的控制命令后,再往各节点发送唤醒帧。在唤醒信号发送到总线上后,所有的节点都运行启动过程,并等待主机任务发送报文头,接收并做出相应的响应。

本系统中,定义了睡眠帧、唤醒帧、命令帧、查询帧 4 种帧命令来对电器从机节点进行控制。电器控制模块只是作为智能家居系统的一部分,其模块程序流程图如图 4 所示。

### 3 结 语

利用 LIN 总线协议在智能家居系统中构建内部网络,实现对各电器节点的控制,被证明具有以下几个优点:① 良好的扩展性。无须改变 LIN 从节点的硬件和软件就可以在网络上增加节点,方便用户添置新电器;② 可靠性好。具有故障错误检测功能,可以可靠地控制家电的启动和关闭;③ 低成本。基于普通串口硬件的低成本硬件实现,低成本的单线设备;④ 可进入睡眠模式,降低系统功耗,节约能源。综上所述,利用 LIN 总线实现

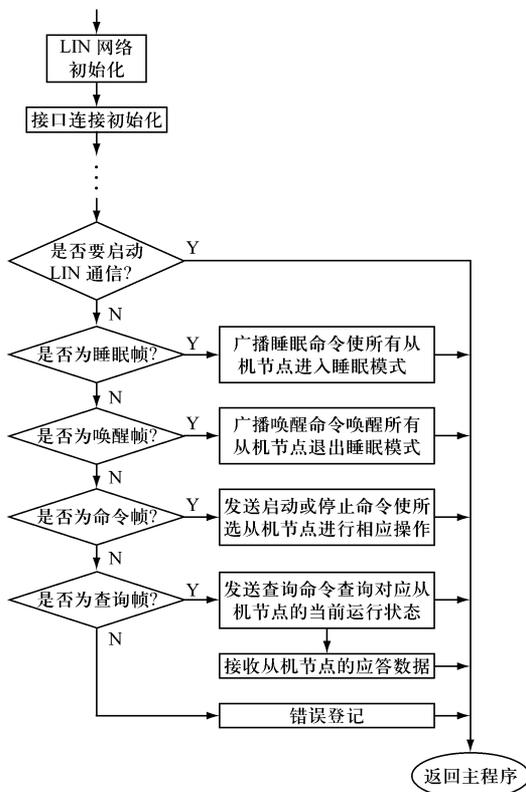


图 4 主电器模块程序流程图

智能家居系统的电器控制在同类产品中具有很多优势,将更为用户所接受。LIN 总线的应用也将在更多的工业领域推广。

### 【参考文献】

- [ 1 ] 张文超,卢可义,刘振方.几种可用于测控仪器的串行总线研究和应用选择原则[ J].测控技术,2005 24(2):40-45.
- [ 2 ] 佟为明,孙凡金,赵志衡.LIN 总线技术[ J].低压电器,2004(2):26-29.
- [ 3 ] 广州周立功单片机发展有限公司.LIN 规范 1.2 [ DB OL]. <http://www.zlgmcu.com>.
- [ 4 ] 广州周立功单片机发展有限公司.Philips 微控制器在 LIN 中的应用[ DB OL]. <http://www.zlgmcu.com>.
- [ 5 ] 李万成,陈辉堂.家庭自动化系统的研制[ J].系统工程与电子技术,1997(12):78-82.

收稿日期:2006-05-20