

基于 AVR 单片机的曼彻斯特编解码及其应用

王建国, 孙敬华, 曹丙霞

(中国海洋大学工程学院, 青岛 266071)

摘要: 提出了一种在较高通信速率(10kbps)下用 AVR 单片机软件实现曼彻斯特码编解码的方法。介绍了小功率无线传输系统的工作原理和通信协议。详细阐述了用软件识别同步时钟、起始符、同步头和数据的技术方法。

关键词: 曼彻斯特编解码; 无线通信; AVR 单片机

Manchester Code/decode Based on AVR and Its Application

WANG Jianguo, SUN Jinghua, CAO Bingxia

(Engineering College, Ocean University of China, Qingdao 266071)

【Abstract】 The paper introduces a hi-efficient coding/decoding method to apply AVR MCU to code/decode Manchester at 10kbps successfully, principle and communication protocol in wireless communicating systems of low power. It provides in detail a technique about how to identify initial flag, synchronization code and data.

【Key words】 Manchester code/decode; Wireless communication; AVR

随着无线通信芯片技术的发展,越来越多的便携式或电池供电的无线传输设备进入人们日常生活中。例如,遥控车门开关(RKE)系统、汽车轮胎压力监视系统(TPMS)、无线内窥镜系统、蓝牙技术等。这种小功率无线传输系统的关键技术是在低电流消耗和信道干扰较强的情况下实现数据稳定可靠的传输。

曼彻斯特码由于其特殊的性能,被广泛应用于小功率无线传输系统中。曼彻斯特编码是串行数据传输的一种重要的编码方式。曼彻斯特编码最大的优点是:数据和同步时钟统一编码,曼码中含有丰富的时钟信号,直流分量基本为零,接收器能够较容易恢复同步时钟,并同步解调出数据,具有很好的抗干扰性能,这使它更适合于信道传输。

传统的曼彻斯特码的编解码一般采用专用芯片电路实现,数据的串/并转换和信号的合成都通过硬件电路实现,导致电路复杂,系统成本提高。考虑到目前微处理器功能不断提高,在不影响系统微处理器完成其它处理任务的情况下,完全可以采用微处理器来实现曼彻斯特码的编码和解码。本设计提出了一种简单有效的曼彻斯特码解调方法,并选用高速 AVR 单片机及软件编程实现了曼彻斯特码的编解码。

1 小功率无线传输系统工作原理

小功率无线传输系统一般由射频发射机和接收监视器两部分组成。射频发射机采集和发射信息;接收监视器接收信息,并根据接收到的信息执行相应的操作。

1.1 射频发射机模块

射频发射机一般采用电池供电,以遥控车门开关(RKE)系统为例,电池需要连续工作3~5年,低功耗设计是电路设计的关键。在此射频发射机主控 MCU 选用 Atmel 公司的高性能、低功耗微控制器 ATmega48V。该芯片工作电压最低可达 1.8V,具有 5 种省电休眠模式,适合低功耗应用场合。根据处理器工作状态的不同进入相应的休眠模式。在掉电模式下耗电小于 1 μ A,极大地降低了功耗。

射频发射机的系统框图如图 1 所示。MCU 首先对数据进行曼彻斯特码编码,然后把数据送到射频发射芯片发射。



图 1 发送模块原理框图

1.2 射频接收监视器模块

接收监视器的主控 MCU 采用 Atmel 公司的高性能、高速 RISC 微控制器 ATmega169。该芯片内嵌了 4*25 段的 LCD 驱动器,通过它可以在 LCD 上显示一些用户关心的状态信息,如轮胎气压信息等。ATmega169 单片机的 T/C1 具有一个捕捉单元,可用来捕捉外部事件,并为其赋予时间标记以说明此事件发生的时刻。外部事件发生的触发信号可由引脚 ICP1(输入捕捉引脚)输入,也可通过模拟比较器单元实现。本设计采用 T/C1 的输入捕捉单元接收数据,即曼彻斯特码数据从引脚 ICP1 输入。这种数据接收处理方法要求 MCU 对外部输入事件具有足够的处理能力,微控制器 ATmega169 工作在 16MHz 时性能高达 16MIPS。实验证明,当数据波特率为 10kHz 时,系统能够正确接收数据。

接收监视器的系统框图如图 2 所示。射频信号经接收芯片解调,输出的曼码数据经过引脚 ICP1 输入到 MCU,MCU 对数据进行解码处理后送到 LCD 显示。

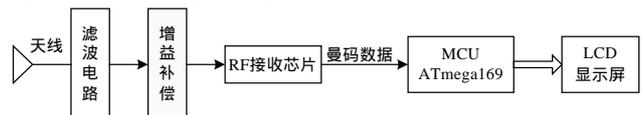


图 2 接收模块原理框图

利用单片机的 T/C1 的引脚捕捉功能虽然能够实现曼码数据电平时间精确测量,但存在一个明显的不足,就是对干

作者简介: 王建国(1954-),男,教授,主研方向:智能仪器仪表与计算机控制技术;孙敬华、曹丙霞,硕士生

收稿日期: 2005-11-08 **E-mail:** sunjinghua-2001@163.com

扰比较灵敏, 干扰严重时直接导致数据传输失败。本系统从软硬件两方面对系统的抗干扰能力进行了改善。软件方面详见下面介绍。在硬件方面加了带通滤波器。又因为接收机接收到的信号比较微弱, 所以又加入了增益补偿电路。此外还使用了 MCU 内部的输入捕捉单元的噪声抑制器。噪声抑制器通过一个简单的数字滤波方案提高系统的抗噪性。它对输入触发信号进行 4 次采样。只有当 4 次采样值相等时其输出才会送入边沿检测器。

目前市场上有很多种 RF 发射接收芯片, 表 1 列出几种常用的芯片。

表 1 RF 发射接收芯片

型号	类型	收/发频点	工作模式	生产厂家
T5744	接收芯片	315/433.92 MHz	ASK	Atmel
MAX1470	接收芯片	315MHz	ASK	Maxim
MC33594	接收芯片	315/434MHz	OOK/FSK	Freescle
T5754	发送芯片	433.92 MHz	ASK/FSK	Atmel
TDA5101	发送芯片	868/433 MHz	ASK/FSK	Infineon
MC33493	发送芯片	315/434/868/928MHz	OOK/FSK	Freescle
NRF401	收发芯片	433.93/434.33 MHz	FSK	Nordic
TRF6903	收发芯片	315/433/868/915MHz	OOK/FSK	TI

2 AVR 单片机实现曼彻斯特码编解码的方法

2.1 曼彻斯特码及数据帧的特征

在曼彻斯特编码中, 将一位时间一分为二, 位时间内发生低电平到高电平的变化表示 '0'; 高电平到低电平的变化表示 '1'; 由于跳变都发生在每一个码元的中间, 接收端可以方便地利用它作为位同步时钟, 因此, 这种编码也称为自同步编码。曼彻斯特编码如图 3 所示。

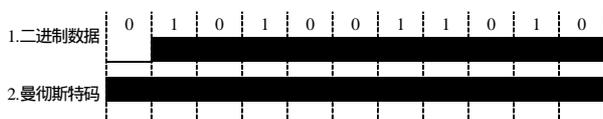


图 3 曼彻斯特编码

从图 3 可以看出曼码具有以下特征:

(1) 当数据位出现连 '0' 或连 '1' 时, 曼码的位速率 2 倍于的数据位速率, 在每个数据的起始点也出现跳变, 但都是无效跳变。

(2) 当数据位出现 '0'、'1' 交替时, 曼码的位速率等于数据位速率, 脉冲的前后沿均为有效跳变。

2.2 数据传输帧格式及曼彻斯特码编解码原理

2.2.1 数据传输协议

本系统采用的数据传输格式如下:

同步时钟	同步头	ID 号	数据包	校验字节	结束符
------	-----	------	-----	------	-----

同步时钟也就是数据解码时钟, 同时还是本次传输起始标志。同步头则用于确定信息包的开始, 本设计用 "@" 作为同步头数据。ID 号用于识别发射模块, 防止与其它发射模块串扰。数据包含有要传输的信息, 采用定长数据包格式。停止符为二进制数 '0', 停止符的作用是保证数据接收完整。因为此设计在上升沿中断时识别码元, 当最后一个字节为 1 时, 如果没有停止位, 就不存在上升沿, 则无法识别这个码元。用 data[] 表示该格式数组。

2.2.2 曼彻斯特编码原理

实现曼彻斯特软件编码比较简单, 根据曼码和二进制数据的对应关系, 一位二进制数据在编码后将占据两位空间。

例如: 1 在编码后变为 1、0; 0 在编码后变为 0、1。其中, 曼码数据低位在前, 高位在后; 被编码的数据字节的高 4 位编码后存放在一个字节中, 低 4 位编码后存放在相邻的下一个字节中。

2.2.3 曼彻斯特解码原理

曼彻斯特解码是编码的逆过程, 也就是从曼码数据中提取时钟, 并利用这个时钟还原二进制数据的过程。根据曼彻斯特解码过程中完成任务的不同, 可以把曼彻斯特解码过程分为起始符识别及获取同步时钟、识别同步头、提取数据信息 3 个阶段。

(1) 起始符识别及获取同步时钟

通常传输通道干扰比较严重, 所以无法用简单的起始位方法来判断何时开始接收数据。本系统采用连续发送 2 个字节 0xFF 的方法来标识一次数据传输的开始。2 个字节的 0xFF 转换成曼码后就相当于 16 个周期一定的矩形波。因为干扰信号连续出现几个周期相近矩形波的几率很小, 因此, 在本系统中如果连续接收到 8 个码元, 其周期在一定容差范围内, 则认为捕捉到起始符, 再根据捕捉到的信号周期计算实际波特率。由于在发送接收开始时波形畸变的机率比较大, 因此在波特率计算时进行了滤波。滤波算法中忽略了前 2 个数据, 采用后 6 个数据的平均值作为数据传输的波特率, 从而使系统具有波特率自适应的能力。该特性可很好地解决因发射模块供电电压变化导致波特率改变带来的不利影响。

(2) 同步头识别

同步头识别也就是识别数据 "@" 的问题。本系统通过信号高低电平所占的位宽来判断数据。当电平持续时间小于 3/4 的同步时钟周期时, 电平时间为半个位宽, 反之电平时间为 1 个位宽。因为在起始符识别时只用了 2 个字节 0xFF 中的一个, 所以在开始识别同步头时识别出的第 1 个数据位肯定是 '1', 由此决定了在码元识别时必须遵循以下 2 点。

1) 先采用下降沿触发方式获得高电平的持续时间, 然后把触发方式改为上升沿, 获得低电平的持续时间, 并进行码元判别。判别依据见表 2。

其中: S 代表电平持续时间为半个位宽, L 代表电平持续时间为一个位宽, 本次位个数是指从曼码的一个高低电平持续时间中提取出来二进制数据的位数。

表 2 码元判别

高电平时间	低电平时间	本次位值	本次位个数
S	S	与上次值相同	1
S	L	1, 0	2
L	S	1	1
L	L	1, 0	2

2) 在第 1 次进行码元识别时, 假定上次值为 '1'。当识别出来的位数据个数大于等于 8 时, 则开始判断接收到的数据是否包含 "@"。如果是则认为识别出同步头, 下次开始接收数据; 反之则把数据右移 1 位去掉最低位, 把下次接收到的位数据添加到最高位, 再判断, 依此类推, 逐位接收判断。如果接收数据位的个数大于 15, 但是还没有检测到同步头, 则此次数据接收无效, 回到步骤 1) 重新开始。

3) 提取数据信息。在此阶段数据识别方法与同步头识别时相同, 不同之处在于数据提取保存方面。通过识别同步头可以找到数据信息的起始位, 从起始位开始每 8 位整合为一个字节数据, 存放到一个字节变量中, 这样就提取出一个字节信息。因为每帧信息中的字节数是固定的, 所以当接收到足够个数的数据字节则认为此次数据接收完成。停止符不会

导致数据接收不正确。完成数据帧的接收后,MCU 进行 CRC 校验、ID 号识别以及数据处理等操作。

2.3 曼彻斯特编解码在 AVR 单片机中的实现

2.3.1 AVR 单片机中曼彻斯特编解的实现

曼彻斯特编码的子程序流程如图 4 所示。该程序仅对一个数据字节编码,循环调用此程序就可完成所有数据字节的编码。在编码前定义一个缓冲区 encode[],用来存放编码后的曼码数据,在调用编码程序前对其清零。其中:data[j]数组为未编码的帧数据,j 为本次被编码数据的次序(数组下标)。

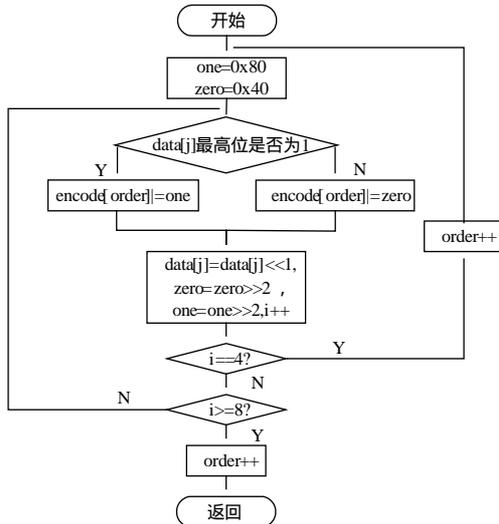


图 4 编码程序流程

数据编码完成后,采用定时器控制方式经通用 I/O 口发送数据。I/O 口与发射芯片的输入引脚相连,发射芯片把数据调制后发射。MCU 把位数据送到 I/O 口后开定时器,并立即进入省电模式。定时时间到引发定时器溢出中断唤醒 MCU 发送下一个位数据。这样就保证了数据传输波特率的准确性,并节省功耗。例如数据传输波特率为 10kHz,则设置定时器定时时间为 50μs。

2.3.2 AVR 单片机中曼彻斯特解码的实现

接收监视器的单片机程序需要完成数据接收、曼彻斯特解码、数据处理、显示、异常情况报警等任务。在接收数据时,因为小功率无线传输通道干扰比较严重,单片机又对所有的 ICP1 引脚沿跳变都响应,它的中断级别又比较高。大大加重了单片机的负担,严重时会造成单片机的其它任务得不到执行。所以在本系统软件设计中加入了滤波。当输入捕捉中断发生后,首先判断输入捕捉寄存器(ICP1)的值是否在一定范围内,如果不是,则不进行数据处理,中断处理返回。

表 3 过程分段

接收阶段 标志符	同步时钟识别 及获取、识别 起始符	计算波 特率	测高电平 持续时间	测低电平持 续时间及码 元识别
start	0	3	1	2

根据数据传输协议帧格式,曼彻斯特解码过程是一个顺序过程,分成 4 个阶段,系统设置不同的标志值以示区分(如表 3),采用 C 语言中的开关语句(switch)实现。

解码程序流程如图 5 所示。其中:start2=0 为同步头识别阶段,start2=1 是提取数据信息阶段,二者可统称为码元识别阶段。在此阶段为了测量信号高低电平的持续时间,定时器

捕捉中断的触发方式必须采用上升沿和下降沿交替进行。也就是说,在下降沿捕捉中断处理程序中(即 start=1 时)设置下次捕捉中断的触发方式为上升沿,在上升沿捕捉中断处理程序中(即 start=2 时)设置下次捕捉中断的触发方式为下降沿。在识别起始符阶段,中断处理程序的主要任务就是判断信号的周期是否在一定的容差范围内,可以选择任一种中断触发方式,在此选择下降沿触发方式。在捕捉到起始符的同时需要计算数据传输波特率和初始化码元识别阶段用到的变量,考虑到单片机的处理能力,把捕捉中断的触发方式改为上升沿,在上升沿引发的中断内计算数据传输波特率和初始化码元识别阶段用到的变量,并把中断触发方式改为下降沿,这样就保证了准确测量信号的高电平持续时间。

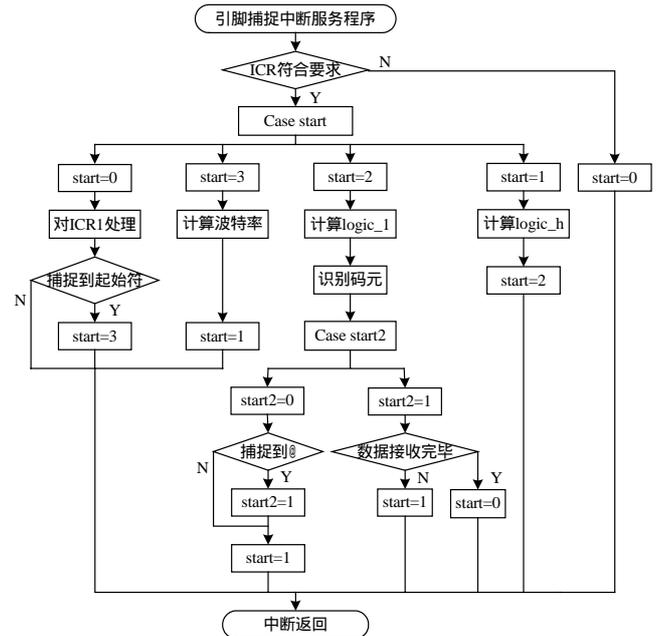


图 5 解码程序流程

3 结论

本文以一个实际工程的一部分为例,阐述了用 AVR 单片机软件编程实现曼彻斯特编解码方法。经过了实际验证,数据传输可靠性很高,系统运行稳定,样机已经成型,即将投入生产。与其它的数据传输方法和曼彻斯特编解码方式相比,本方法十分灵活方便,它可以自适应射频发射机电压变化引起的数据传输波特率的变化。可处理的码速率也很高,极限码速率与所采用的单片机的速度和振荡器的频率以及采用的编程语言都有关系。本设计中采用 C 编程,振荡器频率为 16MHz,数据传输率可达 10kbps,用汇编语言编程数据传输率会更高。

参考文献

- 1 罗伟雄,韩力,原东昌等.通信原理与电路[M].北京:北京理工大学出版社,1998.
- 2 李朝青.无线发送/接收 IC 芯片及其数据通信技术选编[M].北京:北京航空航天大学出版社,2004.
- 3 黄智伟.单片无线数据通信 IC 原理与应用[M].北京:北京航空航天大学出版社,2004.
- 4 沈文, Lee E, 詹卫前. AVR 单片机 C 语言开发入门指导[M].北京:清华大学出版社,2003.
- 5 黄智伟.无线发射与接收电路设计[M].北京:北京航空航天大学出版社,2004.

