

1. 一种在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,包括步骤:

(1) 将红外遥控接收器的输出端 OUT 引脚连接到嵌入式微处理器的一个具有沿中断能力的通用输入/输出端口 GPIO 引脚,以便红外遥控接收器接收到的红外遥控脉冲序列通过 GPIO 引脚进入嵌入式微处理器;

(2) 初始化程序,把 GPIO 中断方式设为下降沿中断,解码状态机状态置为“引导码低开始”;

(3) 当红外遥控接收器接收到的红外遥控脉冲序列通过 GPIO 引脚并发生下降沿中断时,触发微处理器内的解码程序运行;

(4) 利用 GPIO 中断和预先设定的引导码和连发码的编码规则完成对引导码或连发码的有效性判断;本步骤具体包括步骤:

(41) 当第一个 GPIO 中断产生时,判断 GPIO 解码状态机状态是否为“引导码低开始”,是则将下一个 GPIO 中断设为上升沿中断,解码状态机状态置为“引导码低结束”,等待下一个 GPIO 中断的发生;

(42) 当下一个 GPIO 中断产生时,判断解码状态机状态是否为“引导码低结束”,是则把 GPIO 中断方式设为下降沿中断;进一步判断红外遥控脉冲序列的低电平长度是否合法,低电平长度合法则将解码状态机状态置为“引导码高结束”,低电平长度不合法则将解码状态机状态置为“引导码低开始”;最后等待下一个 GPIO 中断的发生;

(43) 当下一个 GPIO 中断产生时,判断解码状态机状态是否为“引导码高结束”,是则进一步判断红外遥控脉冲序列的高电平长度是否合法,高电平长度合法则引导码有效,同时将解码状态机状态置为“数据码解码”;高电平长度不合法则判断连发码是否有效,若连发码有效则将连发码入队,并将解码状态机状态置为“引导码低开始”,若连发码无效则直接将解码状态机状态置为“引导码低开始”;最后等待下一个 GPIO 中断的发生;

(5) 若步骤(4)中引导码有效,则继续利用 GPIO 中断和预先设定的数据位“0”和数据位“1”的编码规则完成对数据码的解码。

2. 根据权利要求1所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,步骤(1)中红外遥控接收器的 OUT 引脚和嵌入式微处理器的 GPIO 引脚之间,进一步包括电平转换芯片,用以实现 OUT 引脚与 GPIO 引脚之间的电平转换。

3. 根据权利要求1所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,步骤(5)具体包括步骤:

(51) 当 GPIO 中断产生时,判断解码状态机状态是否为“数据码解码”,非“数据码解码”状态则等待下一个 GPIO 中断的发生;是“数据码解码”状态则进一步判断红外遥控脉冲序列是否为数据位“0”,若是数据位“0”则记录该数据位“0”;

(52) 若非数据位“0”则判断是否为数据位“1”,若是数据位“1”则记录该数据位“1”,否则将解码状态机状态置为“引导码低开始”,等待下一个 GPIO 中断的发生;

(53) 记录数据位“0”或数据位“1”后,判断 32 位数据码解码是否完成,若未完成则等待下一个 GPIO 中断的发生;若完成则将数据码入队,同时将解码状态机状态置为“引导码低开始”,等待下一个 GPIO 中断的发生。

4. 根据权利要求1至3任一项所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,对数据位“0”或数据位“1”的解码包含步骤:

a. 初始时把 GPIO 中断方式设为下降沿中断,并将解码状态机设为“状态 1”;

b. 当 GPIO 中断 1 到来时,设置解码状态机的下一个状态为“状态 2”;

c. 当 GPIO 中断 2 到来时,首先判断 GPIO 中断 1 与 GPIO 中断 2 的时间间隔 interval 是否满足如下第一条件: $(1.12\text{ms}-\text{delta}) < \text{interval} < (1.12\text{ms}+\text{delta})$, 其中 $\text{delta} > 0$, 表示允许的误差值,若满足第一条件则将当前时间间隔 interval 内的脉冲序列解码为数据位“0”;不满足第一条件则判断 interval 是否满足如下第二条件: $(2.25\text{ms}-\text{delta}) < \text{interval} < (2.25\text{ms}+\text{delta})$, 若满足第二条件则将当前时间间隔 interval 内的脉冲序列解码为数据位“1”;不满足第二条件则将解码状态机设为初始时的“状态 1”,意味着脉冲序列无效,需要丢弃。

5. 根据权利要求 4 所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,所述时间间隔 interval 的获取使用嵌入式微处理器中的时钟周期计数寄存器来实现,实现方法是:

分别记录 GPIO 中断 1 与 GPIO 中断 2 到来时的计数寄存器的值,两个值的差值与时钟频率的倒数的乘积就得到时间间隔 interval 的值。

6. 根据权利要求 4 所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,所述时间间隔 interval 的获取使用嵌入式微处理器中的硬件定时器来实现,实现方法是:

将硬件定时器设置为不超时的连续模式,分别记录 GPIO 中断 1 与 GPIO 中断 2 到来时的硬件定时器的值,两个值的差值与时钟频率的倒数的乘积就得到时间间隔 interval 的值。

7. 根据权利要求 4 所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,所述 delta 参数的值不是一个固定值,而是在一个误差范围内波动的值,根据具体的系统对 delta 参数进行调整。

8. 根据权利要求 4 所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,所述 GPIO 中断的优先级在微处理器中被设置为最高级。

9. 根据权利要求 1 至 3 任一项所述的在嵌入式系统中用软件实现红外遥控解码的方法,其特征在于,所述 GPIO 中断的优先级在微处理器中被设置为最高级。

一种在嵌入式系统中用软件实现红外遥控解码的方法

技术领域

[0001] 本发明涉及红外遥控解码技术,具体的说,涉及一种在嵌入式系统中基于GPIO(通用输入/输出端口)中断用软件实现红外遥控信号解码的方法。

背景技术

[0002] 红外遥控已广泛应用于各种消费类电子设备中,其解码技术目前主要有如下两类:1. 硬件解码。这种解码技术采用硬件解码模块进行红外遥控的解码,该硬件解码模块或者集成在主芯片中或者是一个独立的硬件模块。该方式存在的问题是一般的嵌入式微处理器并不具备红外遥控的硬件解码模块,如果要单独购买一个现成的硬件解码模块,一方面会增加产品的成本,同时还会增加系统的复杂度。另外,硬件解码模块一般只能支持特定格式的红外遥控信号的解码,如果需要更换遥控器则可能需要重新选择硬件解码模块,灵活性较差。2. 软件解码。这种解码技术是采用一个独立的单片机用软件实现红外遥控的解码。这种方式存在的问题是,如果该单片机仅仅是为了完成红外遥控的解码,显然是比较浪费的,将无谓的增加产品的成本;另一方面,这样做无疑也会增加系统的复杂度。

发明内容

[0003] 本发明的目的是为了在嵌入式系统中有效降低红外遥控信号解码的成本和系统复杂度,从而提出了一种在嵌入式系统中用软件实现红外遥控解码的方法。

[0004] 本发明为解决上述技术问题所采用的方法包括步骤:

[0005] (1) 将红外遥控接收器的输出端 OUT 引脚连接到嵌入式微处理器的一个具有沿中断能力的通用输入/输出端口 GPIO 引脚,以便红外遥控接收器接收到的红外遥控脉冲序列通过 GPIO 引脚进入嵌入式微处理器;

[0006] (2) 初始化程序,把 GPIO 中断方式设为下降沿中断,解码状态机状态置为“引导码低开始”;

[0007] (3) 当红外遥控接收器接收到的红外遥控脉冲序列通过 GPIO 引脚并发生下降沿中断时,触发微处理器内的解码程序运行;

[0008] (4) 利用 GPIO 中断和预先设定的引导码和连发码的编码规则完成对引导码或连发码的有效性判断;本步骤具体包括步骤:

[0009] (41) 当第一个 GPIO 中断产生时,判断解码状态机状态是否为“引导码低开始”,是则将下一个 GPIO 中断的设为上升沿中断,解码状态机状态置为“引导码低结束”,等待下一个 GPIO 中断的发生;

[0010] (42) 当下一个 GPIO 中断产生时,判断解码状态机状态是否为“引导码低结束”,是则将把 GPIO 中断方式设为下降沿中断;进一步判断红外遥控脉冲序列的低电平长度是否合法,低电平长度合法则将解码状态机状态置为“引导码高结束”,低电平长度不合法则将解码状态机状态置为“引导码低开始”;最后等待下一个 GPIO 中断的发生;

[0011] (43) 当下一个 GPIO 中断产生时,状态判断解码状态机状态是否为“引导码高结

束”，是则进一步判断红外遥控脉冲序列的高电平长度是否合法，高电平长度合法则引导码有效，同时将解码状态机状态置为“数据码解码”；高电平长度不合法则判断连发码是否有效，若连发码有效则将连发码入队，并将解码状态机状态置为“引导码低开始”，若连发码无效则直接将解码状态机状态置为“引导码低开始”；最后等待下一个 GPIO 中断的发生。

[0012] (5) 若步骤 (4) 中引导码有效，则继续利用 GPIO 中断和预先设定的数据位“0”和数据位“1”的编码规则完成对数据码的解码。

[0013] 上述方法中，步骤 (1) 中红外遥控接收器的 OUT 引脚和嵌入式微处理器的 GPIO 引脚之间，进一步包括电平转换芯片，用以实现 OUT 引脚与 GPIO 引脚之间的电平转换。

[0014] 上述方法中，步骤 (5) 具体包括步骤：

[0015] (51) 当 GPIO 中断产生时，判断解码状态机状态是否为“数据码解码”，非“数据码解码”状态则等待下一个 GPIO 中断的发生；是“数据码解码”状态则进一步判断红外遥控脉冲序列是否为数据位“0”，若是数据位“0”则记录该数据位“0”；

[0016] (52) 若非数据位“0”则判断是否为数据位“1”，若是数据位“1”则记录该数据位“1”，否则将解码状态机状态置为“引导码低开始”，等待下一个 GPIO 中断的发生；

[0017] (53) 记录数据位“0”或数据位“1”后，判断 32 位数据码解码是否完成，若未完成则直接等待下一个 GPIO 中断的发生；若完成则将数据码入队，同时将解码状态机状态置为“引导码低开始”，等待下一个 GPIO 中断的发生。

[0018] 本发明的有益效果是：同现有技术中的红外遥控信号的硬件解码和软件解码相比较，本发明在嵌入式多任务系统中用软件实现了红外遥控的可靠解码，不需要增加额外的硬件设备，有效的降低系统的成本和复杂度，使得系统更灵活、适应性更强。

附图说明

[0019] 图 1 是本发明的硬件连接图。

[0020] 图 2 是 NEC 格式数据位“0”和数据位“1”的编码波形图。

[0021] 图 3 是 NEC 格式 32 位二进制码组的编码波形图。

[0022] 图 4 是 NEC 格式遥控连发信号波形图。

[0023] 图 5 是 NEC 格式引导码和连发码的波形图。

[0024] 图 6 是本发明的软件解码流程图。

[0025] 图 7 是数据位“0”和“1”的解码原理图。

具体实施方式

[0026] 下面结合附图和具体实施例对本发明的原理进行详细说明。

[0027] 本发明的思路是利用嵌入式微处理器的 GPIO 接口接收来自红外遥控接收器的脉冲信号，基于 GPIO 中断用软件实现对红外遥控脉冲序列的解码。

[0028] 如图 1 所示为本发明的硬件连接图。将红外遥控接收器的输出端 OUT 引脚连接到嵌入式微处理器的一个具有沿中断能力的通用输入 / 输出端口 GPIO 引脚，以便红外遥控接收器接收到的红外遥控脉冲序列通过 GPIO 引脚进入嵌入式微处理器；红外遥控接收器与嵌入式微处理器的 GND 引脚连接在一起并接地。如果 OUT 引脚与 GPIO 引脚之间的电平要求是不匹配的，则红外遥控接收器的 OUT 引脚和嵌入式微处理器的 GPIO 引脚之间，进一步

包括电平转换芯片,用以实现 OUT 引脚与 GPIO 引脚之间的电平转换。另外,红外遥控接收器由电源 VCC1 提供电能支持,嵌入式微处理器由电源 VCC2 提供电能支持。

[0029] 如图 2、图 3、图 4 和图 5 所示为一个典型的红外遥控信号编码格式 (NEC 格式) 的波形图,本发明将以该格式的编码格式的波形图为例对本发明的具体实施方式进行说明,为了便于本领域的技术人员理解本发明,现在结合附图对 NEC 格式进行详细的说明。

[0030] 由于 NEC 格式运用比较广泛,解码比较容易,因而一般家庭用的 DVD、VCD、音响都使用这种编码方式,由于红外遥控信号的所有波形的接收端与发射端相反,因而以接收端的波形为例进行说明,如图 2 所示,NEC 格式的脉冲信号采用脉宽调制的串行码,以脉宽为 0.565ms、间隔 0.56ms、周期为 1.125ms 的组合表示二进制的数字位“0”;以脉宽为 0.565ms、间隔 1.685ms、周期为 2.25ms 的组合表示二进制的数字位“1”。上述数字位“0”和数字位“1”组成的 32 位二进制码经 38kHz 的载频进行二次调制以提高发射效率,达到降低电源功耗的目的,然后再通过红外发射二极管产生红外线向空间发射红外遥控脉冲序列。如图 3 所示,红外遥控脉冲序列的遥控编码是连续的 32 位二进制码组,其中前 16 位为用户识别码,也称系统码,能区别不同的电器设备,防止不同机种遥控码互相干扰。后 16 位为 8 位数据码 (功能码) 及其反码,以确保数据码不会被错误的解码。遥控器在按键按下后,周期性地发出同一种 32 位二进制码,周期约为 108ms。如图 4 所示为遥控连发信号波形图,一组码本身的持续时间随它包含的二进制数字位“0”和数字位“1”的个数不同而不同,大约在 45 ~ 63ms 之间。如图 5 所示,当一个键按下超过 36ms,将发射一组 108ms 的编码脉冲,这 108ms 发射代码由一个低电平 (9ms) 和一个高电平 (4.5ms) 组成的引导码,其后是低 8 位地址码 (9ms ~ 18ms),高 8 位地址码 (9ms ~ 18ms),8 位数据码 (9ms ~ 18ms) 和这 8 位数据的反码 (9ms ~ 18ms) 组成的数据码。如果键按下超过 108ms 仍未松开,接下来发射的代码 (连发码) 将仅由低电平 (9ms) 和高电平 (2.25ms) 组成的连发码。

[0031] 以下通过 NEC 格式的红外遥控编码的解码过程来说明本发明的软件解码流程图,如图 6 所示,本发明主要包括的步骤包括:

[0032] (1) 将红外遥控接收器的输出端 OUT 引脚连接到嵌入式微处理器的一个具有沿中断能力的通用输入 / 输出端口 GPIO 引脚,以便红外遥控接收器接收到的红外遥控脉冲序列通过 GPIO 引脚进入嵌入式微处理器。

[0033] (2) 初始化程序,把 GPIO 中断方式设为下降沿中断,解码状态机状态置为“引导码低开始”,并完成 GPIO 中断优先级设置,GPIO 中断挂载等初始化工作。

[0034] (3) 当红外遥控接收器接收到的红外遥控脉冲序列通过 GPIO 引脚并发生下降沿中断时,触发微处理器内的解码程序运行,该程序将首先清除 GPIO 中断状态寄存器。

[0035] (4) 利用 GPIO 中断和预先设定的引导码和连发码的编码规则完成对引导码或连发码的有效性判断。

[0036] 从“引导码低开始”状态经过“引导码低结束”状态到达“引导码高结束”状态时,即可判定是引导码还是连发码。如果判定是引导码,接下来就会进入“数据码解码”状态,在该状态下将完成 32 位数据码的解码工作。

[0037] 对引导码或连发码有效性判断包括步骤:

[0038] (41) 当第一个 GPIO 中断产生时,判断解码状态机状态是否为“引导码低开始”,是则将下一个 GPIO 中断的设为上升沿中断,解码状态机状态置为“引导码低结束”,等待下一

个 GPIO 中断的发生；

[0039] (42) 当下一个 GPIO 中断产生时,判断解码状态机状态是否为“引导码低结束”,是则将把 GPIO 中断方式设为下降沿中断;进一步判断红外遥控脉冲序列的低电平长度是否合法,低电平长度合法则将解码状态机状态置为“引导码高结束”,低电平长度不合法则将解码状态机状态置为“引导码低开始”;最后等待下一个 GPIO 中断的发生;

[0040] (43) 当下一个 GPIO 中断产生时,状态判断解码状态机状态是否为“引导码高结束”,是则进一步判断红外遥控脉冲序列的高电平长度是否合法,高电平长度合法则引导码有效,同时将解码状态机状态置为“数据码解码”;高电平长度不合法则判断连发码是否有效,若连发码有效则将连发码入队,并将解码状态机状态置为“引导码低开始”,若连发码无效则直接将解码状态机状态置为“引导码低开始”;最后等待下一个 GPIO 中断的发生。

[0041] (5) 若步骤 (4) 中引导码有效,则继续利用 GPIO 中断和预先设定的数据位“0”和数据位“1”的编码规则完成对数据码的解码。

[0042] 对数据码的解码包括步骤:

[0043] (51) 当 GPIO 中断产生时,判断解码状态机状态是否为“数据码解码”,非“数据码解码”状态则等待下一个 GPIO 中断的发生;是“数据码解码”状态则进一步判断红外遥控脉冲序列是否为数据位“0”,若是数据位“0”则记录该位;

[0044] (52) 若非数据位“0”则判断是否为数据位“1”,若是数据位“1”则记录该位,否则将解码状态机状态置为“引导码低开始”,等待下一个 GPIO 中断的发生;

[0045] (53) 记录数据位“0”或数据位“1”后,判断 32 位数据码解码是否完成,若未完成则直接等待下一个 GPIO 中断的发生;若完成则将数据码入队,同时将解码状态机状态置为“引导码低开始”,等待下一个 GPIO 中断的发生。

[0046] 在步骤 (51) 和 (52) 中对数据位“0”或数据位“1”的解码又具体包含如下步骤,现结合图 7 对以下步骤进行描述。

[0047] a. 初始时把 GPIO 中断方式设为下降沿中断,将解码状态机设为“状态 1”;

[0048] b. 当 GPIO 中断 1 到来时,设置解码状态机的下一个状态为“状态 2”;

[0049] c. 当 GPIO 中断 2 到来时,首先判断 GPIO 中断 1 与 GPIO 中断 2 的时间间隔 interval 是否满足如下第一条件: $(1.12\text{ms}-\text{delta}) < \text{interval} < (1.12\text{ms}+\text{delta})$ (其中 $\text{delta} > 0$, 表示允许的误差值,该值可根据具体的系统进行调整),若满足第一条件则将脉冲序列解码为数据位“0”;不满足第一条件则判断 interval 是否满足如下第二条件: $(2.25\text{ms}-\text{delta}) < \text{interval} < (2.25\text{ms}+\text{delta})$,若满足第二条件则将脉冲序列解码为数据位“1”;不满足第二条件则将解码状态机设为初始时的“状态 1”,意味着脉冲序列无效,需要丢弃。

[0050] 上述时间间隔 interval 的获取可以使用嵌入式微处理器中的时钟周期计数寄存器来实现,实现方法是:分别记录 GPIO 中断 1 与 GPIO 中断 2 到来时的计数寄存器的值,两个值的差值与时钟频率的倒数的乘积就得到时间间隔 interval 的值。

[0051] 在上述方法中,时间间隔 interval 的获取也可以使用嵌入式微处理器中的硬件定时器来实现,实现方法是:将硬件定时器设置为不超时的连续模式,分别记录 GPIO 中断 1 与 GPIO 中断 2 到来时的硬件定时器的值,两个值的差值与时钟频率的倒数的乘积就得到时间间隔 interval 的值。

[0052] 在上述方法中, delta 参数需要根据具体的系统进行调整。这是因为对于嵌入式多任务系统来讲, 中断延迟是难免的, 而且该值对于不同的嵌入式多任务系统来说也是不同的; 另外, 对于同一个嵌入式多任务系统来说, 该值也不是一个固定值, 而是在一个范围内波动的。这就要求我们需要根据具体的系统对 delta 参数进行调整, 以准确反映该系统的中断延迟特性, 确保解码的正确性。

[0053] 在上述方法中, 有两种情况可能导致键码的丢失。一种是当嵌入式多任务系统负载较重时, 中断延迟可能会超出正常的波动范围, 从而直接导致解码错误, 也就是将正确的键码丢失。对于支持优先极中断的嵌入式系统来讲, 为了降低丢失率, 可以采取的措施是将 GPIO 中断的优先级设到最高。对于我们实测的 Linux 2.6 系统来讲, 通过将 GPIO 中断的优先级设到最高, 丢失率处于一个很低的水平, 完全能够达到产品级的可靠性要求。对于其他比 Linux 2.6 体量更小的系统来讲, 通过正确的设置, 也应该完全能够达到产品级的可靠性要求。另一种可能导致键码的丢失的情况是由于遥控器操作原因导致红外遥控接收器未收到相应的脉冲信号, 从而导致对应的键码丢失。如果丢失的是连发码, 问题尚不严重; 但如果丢失的是数据码, 则我们必须丢弃该数据码后续的连发码, 以避免应用程序对连发码的错误解释。图 6 中, 在步骤 (43) 中判定“该连发码有效?”就是为了完成这一工作的。

[0054] 另外还有一个在图 6 中不便描述的问题是对遥控键释放的处理。当操作者释放一个遥控键时, 软件解码程序需要给出一个特殊键码以供应用程序用作“键释放”事件。采用一个软件定时器与解码状态机配合起来, 就可以完成对遥控键释放的处理。

[0055] 本领域的普通技术人员将会意识到, 这里所述的实施例是为了帮助读者理解本发明的原理, 应被理解为发明的保护范围并不局限于这样的特别陈述和实施例。凡是根据上述描述做出各种可能的等同替换或改变, 均被认为属于本发明的权利要求的保护范围。

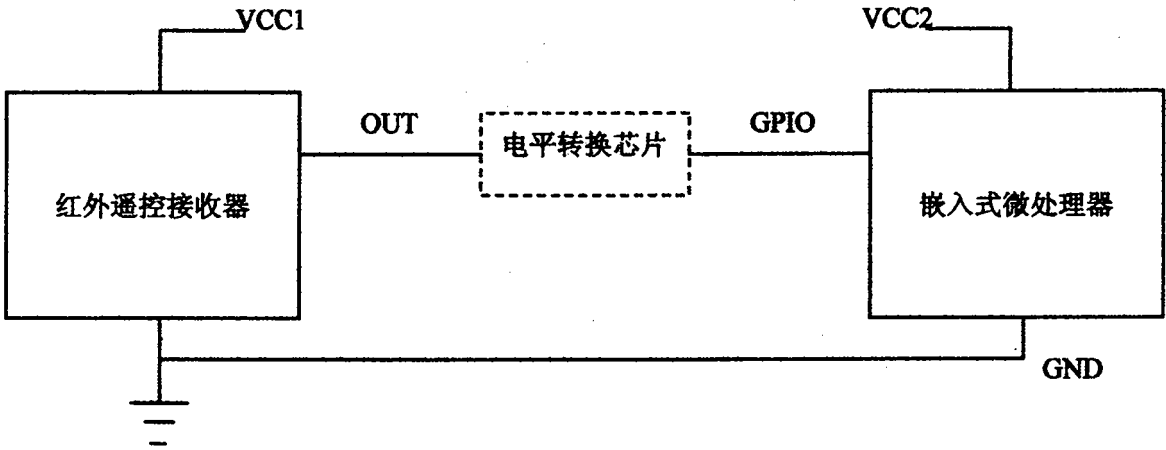


图 1

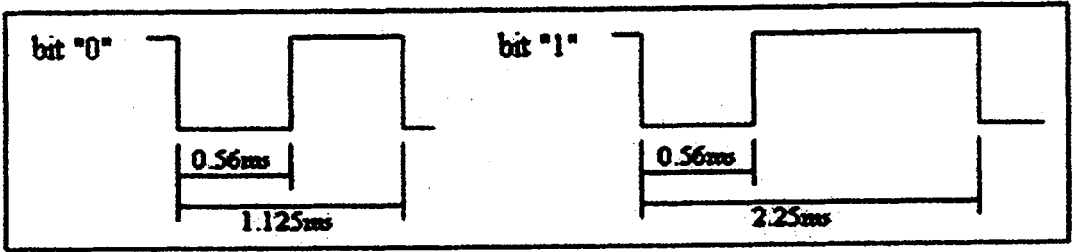


图 2

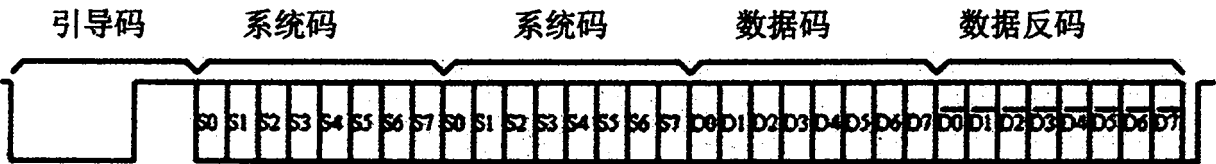


图 3

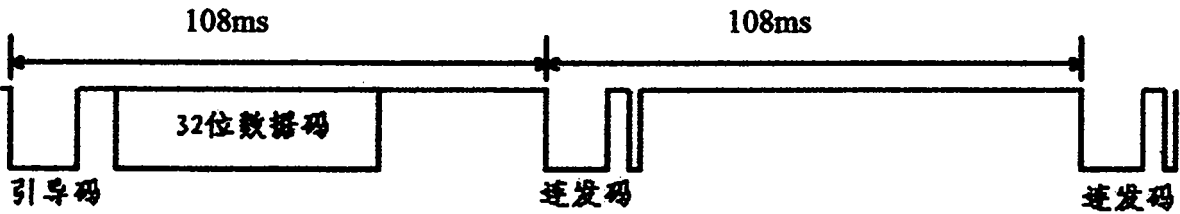


图 4

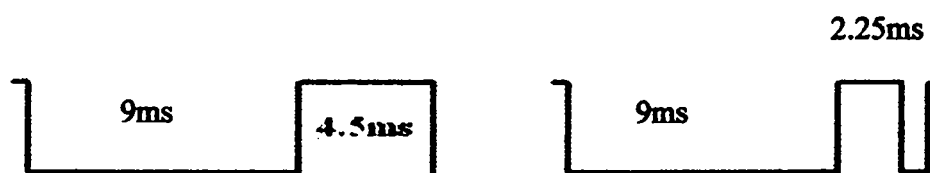


图 5

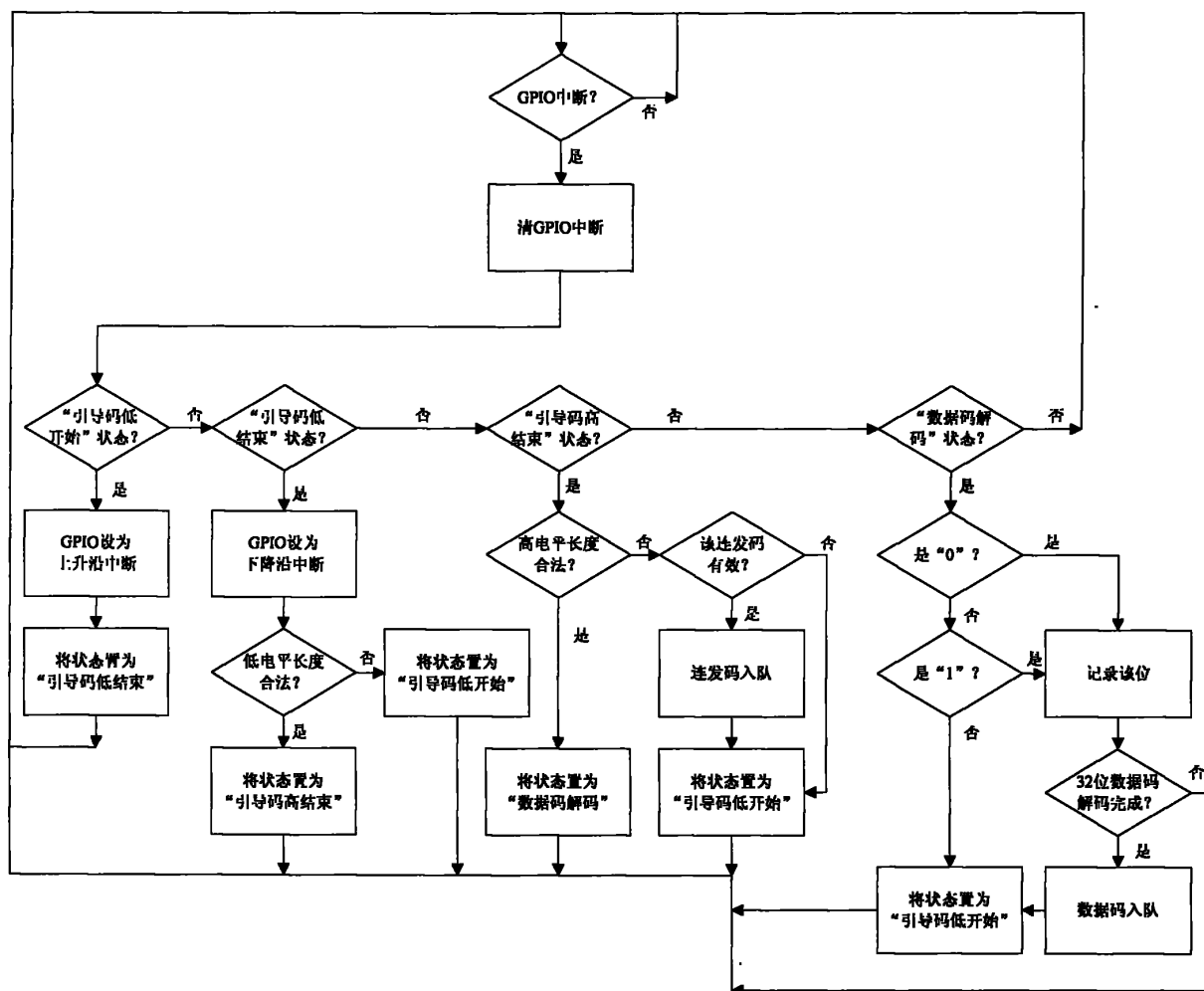


图 6

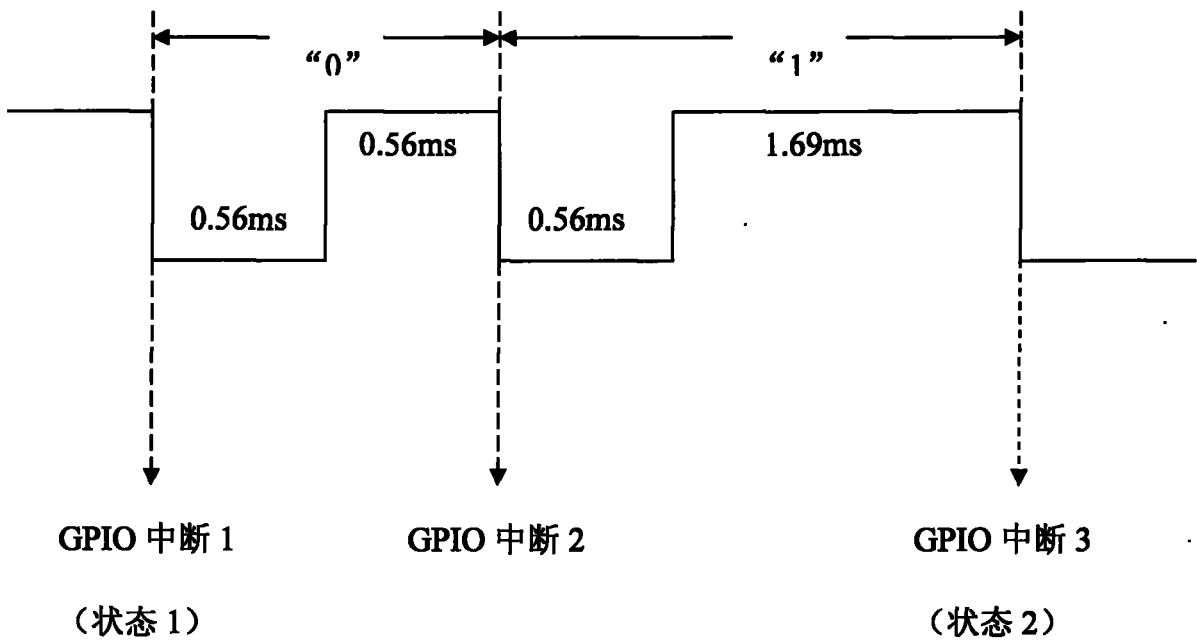


图 7