

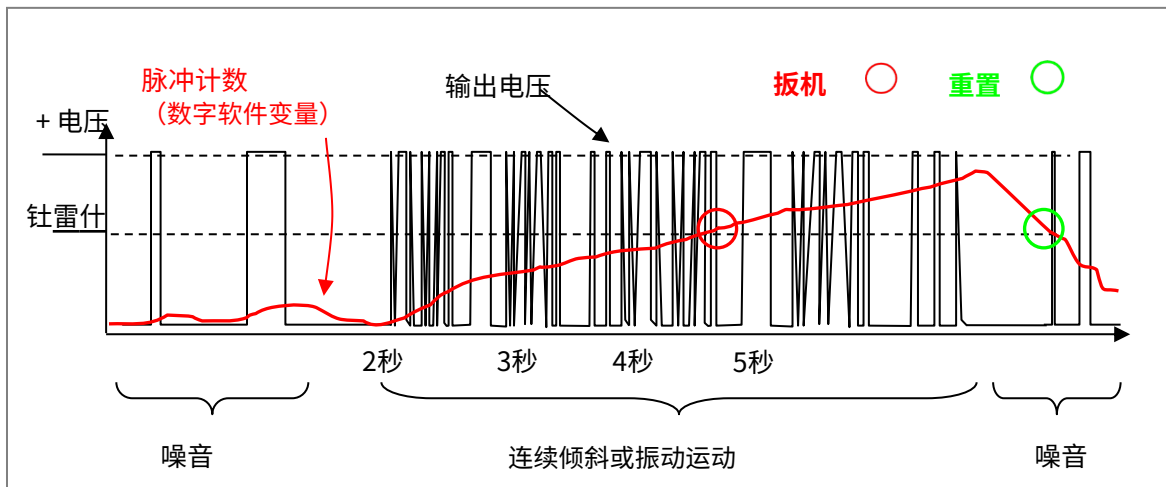
概述

为了解释传感器的脉冲串输出，许多客户结合使用“软件一次性”和“软件集成器”。这通常在固件中完成，但有时如果不使用微控制器，则在硬件中完成。这些简单、低功耗的算法消除了自然发生的传感器与传感器灵敏度差异的影响，并且可以轻松调整触发灵敏度。

本应用笔记描述了一种紧凑的算法，用于将 V_Out 的原始脉冲串输出（“SQ-SEN-200 应用电路”中看到的 V_Raw 或 V_Filt 信号）调节为表示给定时间段内运动体验量的定量信号。这是增强 SQ-SEN-200 输出信号的首选方法，以实现稳健、无误报、运动检测和唤醒功能。出于多种原因，固件例程优于模拟低通滤波器。

实施

- **电路：**请参阅标题为“SQ-SEN-200 应用电路”的应用笔记。
- **输入：**将微控制器上的中断或数字输入设置为边沿触发。将其连接到传感器的输出。如果边沿触发不可用，您可以使用“SQ-SEN-200 应用电路”中所示的交流耦合电路。也可以轮询传感器的信号，但如果轮询率较低，则可能会丢失脉冲。需要对以下注释进行一些调整。
- **软件一击：**当保持低功耗模式（即在 32 KHz 手表晶体上运行）时，寻找中断脉冲。第一次中断后，禁用进一步的中断 1/2 秒，从而仅允许识别每 1/2 秒时间片一个脉冲。此时不要唤醒处理器。
- **软件集成商：**如果单触发被触发，则将脉冲计数器变量增加 5。每 1/2 秒，将脉冲计数器变量减少 1，以随着时间的推移耗尽计数值。请勿切换至高功率模式。
- **临界点：**如果脉冲计数器值超过阈值 25（运动 5 秒），则设备正在移动。现在，切换到高功率状态、警报等。
- **笔记：**确保停止将脉冲计数器变量递增到某个最大值（尝试 50），这样它就不会随着连续运动而继续增长并导致变量的数值溢出。确保停止将脉冲计数器递减为零，这样就不会发生数值下溢。
- **示例**
- V_Out 信号（来自传感器）将类似于下面的黑色图表。
- Pulse_Count（数字软件滤波器变量）将类似于下面的红色信号。



操作

当单次触发（来自传感器的中断）时，将向 Pulse_Count 变量添加一个恒定值。每隔一段时间，从 Pulse_Count 变量中减去一个常数值，从而“泄漏”掉它的一些值。这可确保脉冲计数不会随着时间的推移而累积。经过一段时间没有运动后，Pulse_Count 将衰减为零。

- 该软件一次性禁止在接收到触发脉冲后的给定时间窗口内识别其他输入脉冲。这可以防止噪声信号在传感器静止时向 Pulse_Count 变量贡献较大的计数值。这还可以防止处理器针对给定的运动信号触发数千个中断。

我初始化和D定义

- Pulse_Count - 计数器累加器变量
- 触发器” - 用于发出唤醒或警报状态信号
- Dec_val - 定期减去的常数值
- Inc_val - 软件一次性触发时要添加的常量值
- Max_Count - 用于防止数值溢出
- Thresh - 与 Pulse_Count 进行比较的阈值
- 初始化 Pulse_Count = 0
- 初始化触发器 = False
- **磷罗格拉姆**
- 在中断信号上
 - 哦 如果 Pulse_Count < (Max_Count - Inc_val), 则禁用进一步中断 ½ 秒 // 这是软件一次性检查可能的溢出
 - 哦 脉冲计数 = 脉冲计数 + Inc_val // 每次转换时递增 Pulse_Count
 - 哦 别的
 - 没做什么 // 不进行增量以避免溢出
- 每 ½ 秒执行一次
 - 哦 重新启用中断 // 这是软件一次性 // 检查可能的下溢
 - 哦 如果 Pulse_Count > Dec_val
 - Pulse_Count = Pulse_Count - Dec_val // 无运动时递减 Pulse_Count
 - 哦 别的
 - 没做什么 // 不进行递减以避免下溢
 - 哦 如果 Pulse_Count > 阈值
 - 触发=真 // 现在起床, 闹钟等。
 - 哦 别的
 - 触发=假