如何对 NDIR LARK-1 进行负漂补偿

使用 LARK-1 传感器的客户应该发现给 LARK-1 通入零点气的情况下，LARK-1 输出的读数有时可能为负数 (*LARK-1 系列输出的读数 Reading 即是待测气的浓度，以 ppm 为单位)。LARK-1 输出负读数有一定的迷惑性，因为气体浓度不可能为负，LARK-1 输出负读数是不是有故障呢？并非如此，LARK-1 输出负读数其实反映了传感器的零点负漂情况，市面上的 NDIR 传感器都存在一定程度的零点负漂现象，需要定期进行标定，只是有的公司对 NDIR 传感器负读数设置了死区(Dead Band)，所以用户看不到。

客户使用 LARK-1 过程中若发现零点负漂现象，可对其进行零点标定使 LARK-1 输出准确的读数。为减少人工标定的工作量，亦可根据本文档使用主控设备对 LARK-1 输出的读数进行负漂补偿。需要注意的是**负漂补偿并不能代替零点标定功能**，NDIR 传感器均需要定期进行标定。

1. 应用背景

如图 1 所示，主控设备给 LARK-1 发送获取数据命令，LARK-1 才会给主控设备返回其查询的数据，如何操作 LARK-1 可以参考诺联芯的 AN-001 和 AN-003。

主控设备可以是电脑、显示屏、用户 MCU 等，而主控设备内部程序是用户可以操作的。这篇笔记正是告诉用户如何在自己的主控设备内对 LARK-1 返回的读数进行负漂补偿，该方案不会影响 LARK-1 输出的读数，只是用户程序根据负漂补偿算法对该读数进行了修正。

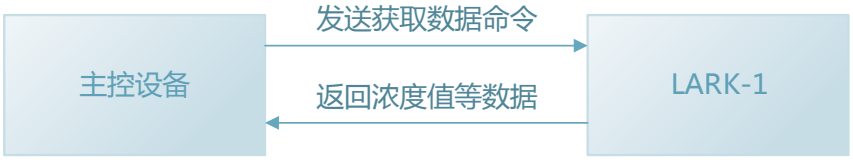

图 1. 主控设备与 LARK-1 数据流图

2. 何为零点负漂

图 2 是我们的工程师用 LARK-1 i-C4H8 通气所测得的数据，给 LARK-1 上电后通入纯净的 N_2 气，半小时后通入 300 ppm i-C4H8 标气。在通入零点气(纯净 N_2 气)的半小时内，读数从-230 继续负漂至-300 ppm 以下，以至于通入 300 ppm 标气时读数仍然是负数。

图 2. 零点负漂现象示例

如图 2 所示，在给 LARK-1 通入零点气时输出读数为负值的现象即是零点负漂，LARK-1 通常在开机 30 分钟后负漂值趋于稳定，即稳定在一个负读数，这就是零点负漂值。

3. 负漂补偿的作用

根据图 2 不难看出，零点负漂值不仅仅影响零点读数，通入 300 ppm i-C4H8 标气时输出的读数也被零点负漂值拉低，导致 LARK-1 输出的读数无法准确反映待测气浓度。如果我们已经知道了零点负漂值，那么用这个负漂值去修正 LARK-1 输出的读数，不就能够得到准确的待测气浓度了吗？比如图 2 中，零点负漂值为-304 ppm，通入 300 ppm i-C4H8 标气时，LARK-1 输出的读数是-18 ppm，用零点负漂值补偿：-18 – (-304) = 286 ppm，绝对误差为-14 ppm。

负漂补偿的理论依据是：NDIR 传感器的零点数值和测量点数值之间的差是相对恒定的。也就是说，NDIR 传感器灵敏度不变，所以可以用零点负漂值去修正测量点数值。

上面提到的这种情况，同样可以通过零点标定使 LARK-1 输出准确的浓度值，用户可以根据自己的实际情况选择方案。对 LARK-1 进行零点标定当然是相对完善的方案，而进行负漂补偿则可以认为是一种快速简洁的备选方案。

4. 负漂补偿方法

我们提出的负漂补偿方法，需要用户在自己的主控设备中编制运行负漂补偿程序。由于 LARK-1 的零点负漂值需要一段时间才能趋于稳定，所以负漂补偿程序需要实时更新，我们目前采用的是 100ms 执行一次负漂补偿程序，用户也可以根据自己的实际情况确定执行周期。负漂补偿程序需要根据 LARK-1 输出的负读数计算零点负漂值，并用该值修正读数，因此执行该程序前，主控设备需要向 LARK-1 发送查询读数的命令以获得最新的读数。

另外，在§<2. 何为零点负漂>中提到 LARK-1 通常在开机 30 分钟后，负漂值才能趋于稳

定，所以**采用负漂补偿的用户应在开机后半小时内通入零点气，不可通入待测气。**通入稳定的零点气，负漂补偿程序才能计算到准确的零点负漂值，从而对读数做出有效的修正。

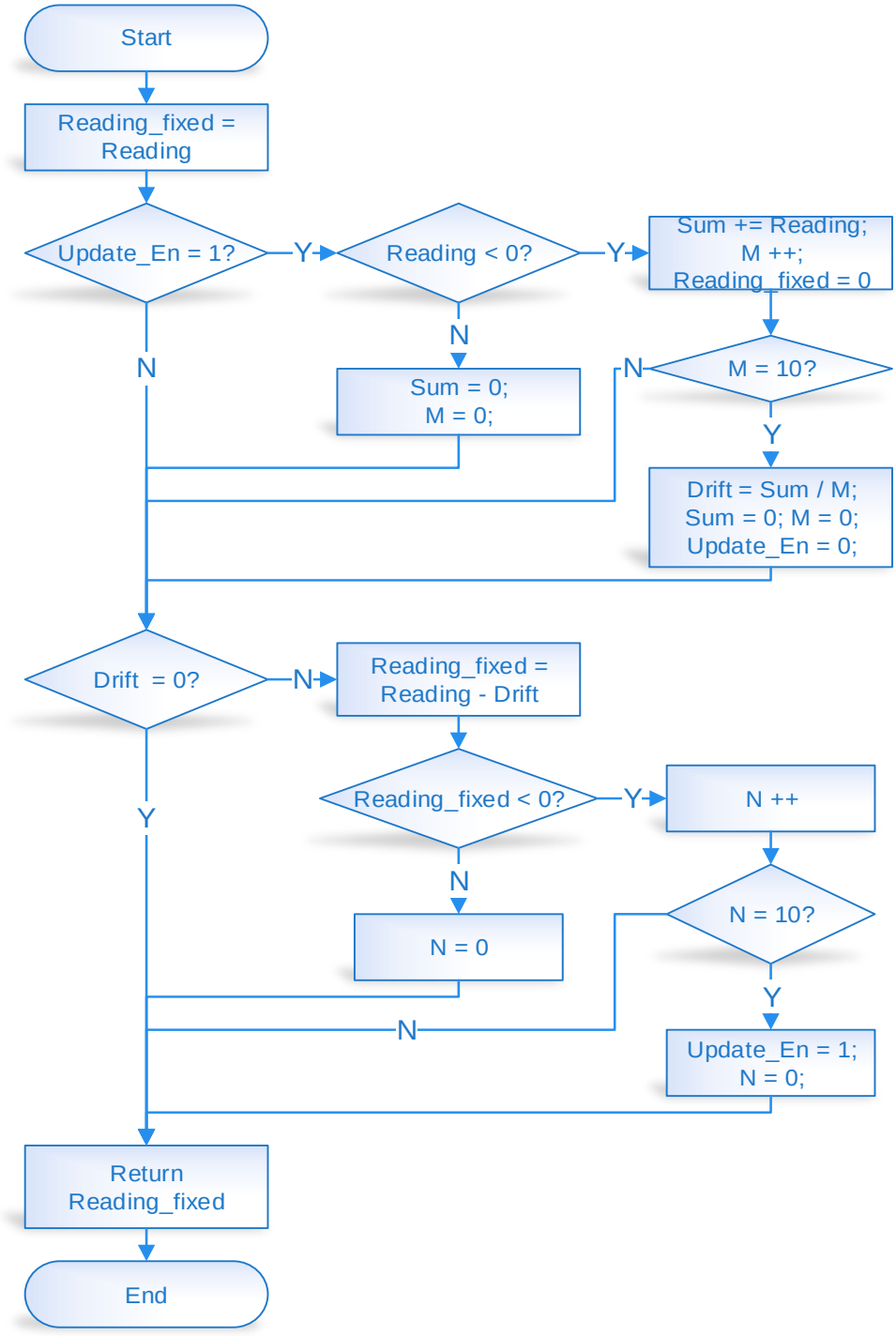

图 3. 负漂补偿程序框图

图 3 是负漂补偿程序的流程图，包括两大功能：

(1) 计算零点负漂值，用 Update_En 参数作为是否计算并更新零点负漂值（Drift）的标志，该部分程序检测到 LARK-1 输出的读数有连续 10 个为负值的情况，就取这 10 个负读数的平均值为零点负漂值，并将 Update_En 标志清零；

(2) 用计算到的零点负漂值（Drift）修正读数，得到修正后的读数 Reading_fixed，这段程序还要检测修正后读数仍然为负值且有连续 10 个为负值的情况，这种情况的发生也是因为 LARK-1 的零点负漂在开机后仍会持续下降，检测到这种情况后采取的措施是将 Update_En 置位，在下次执行该程序时启动零点负漂值的重新计算和更新。

5. 负漂补偿效果

图 4 是我们的工程师用 LARK-1 i-C4H8 通气所测得的数据对比曲线，灰色曲线是 LARK-1 输出的读数曲线，蓝色曲线是用主控设备对 LARK-1 输出读数进行负漂补偿后的读数曲线。

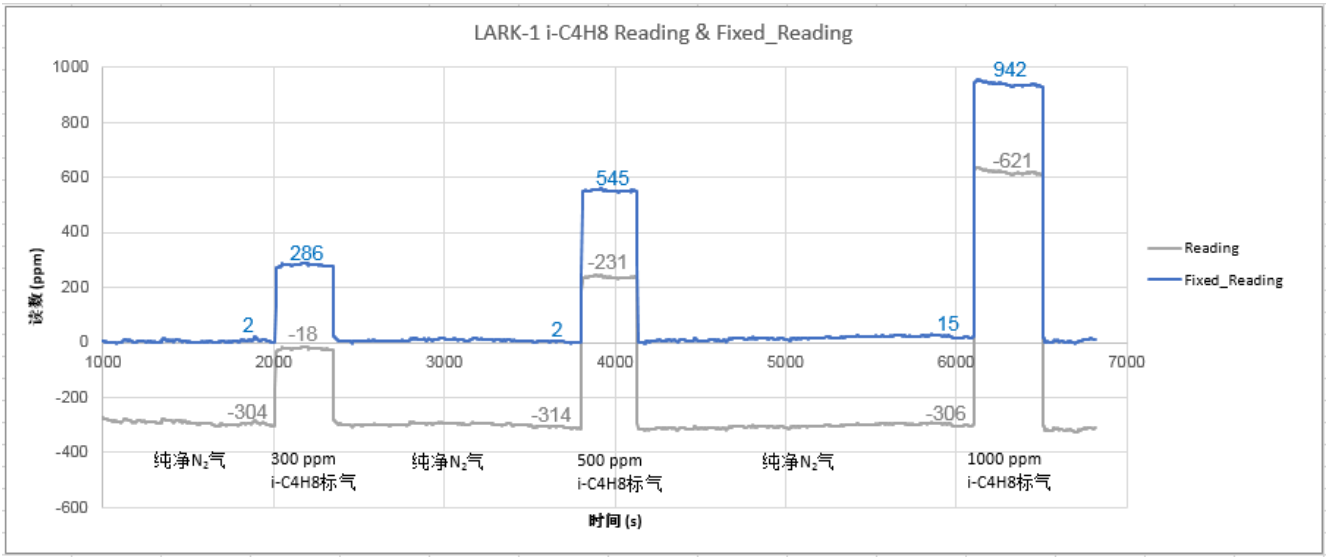

图 4. LARK-1 i-C4H8 读数与修正后数据比较曲线

根据图 4 所示的对比曲线，不难看出这只 LARK-1 i-C4H8 的零点负漂情况比较严重，导致测得的浓度值完全不准确了，但是进行负漂补偿后能够做到±100ppm 以内的绝对误差。

6. 参考代码

```
/*******************************************************************************
***  --- 函数名：NegDrift_Compensation
***  --- 形　参：LARK-1 返回的读数 reading
***  --- 返回值：返回修正后的读数 reading_fixed
***  --- 功　能：用零点负漂值对 LARK-1 读数进行修正，获得准确的待测气浓度
***  --- 说　明：该函数应在用户的主控设备里编写运行，以 100ms 为执行周期，运行该函数
***  ---          前主控设备应向 LARK-1 发送查询读数的命令，以获得最新的读数值
*******************************************************************************/
```

```
int32_t NegDrift_Compensation(int32_t reading)
{
    static uint8_t drift_update_enable =1;      // 允许计算并更新负漂值标志
    static uint8_t neg_reading_num = 0;         // 读数连续为负值的数量
     static uint32_t neg_reading_sum = 0;       // 连续 10 个负读数的累加和
    static uint32_t zero_neg_drift = 0;         // 零点负漂值
    static uint32_t reading_fixed = 0;          // 用负漂值修正后的读数
    static uint8_t neg_fixed_num = 0;           // 修正后读数连续为负值的数量

   //将读数赋值给修正后读数，作为其初始化参数
    reading_fixed = reading;

    // 计算零点负漂值，取 10 个连续的负读数之和的平均值
    if(drift_update_enable)
    {
        if (reading < 0)
        {
            neg_reading_sum += reading;
            neg_reading_num ++;
            reading_fixed = 0;
            // 有连续 10 个读数为负值，取其平均值为零点负漂值
            if(neg_reading_num >= 10)
            {
                zero_neg_drift = neg_reading_sum / neg_reading_num;
                drift_update_enable = 0;
                neg_reading_num = 0;
                neg_reading_sum = 0;
            }
        }
        // 累加负读数过程中，若遇读数为正的情况，则将累加参数清零
        else
        {
            neg_reading_num = 0;
            neg_reading_sum = 0;
        }
    }
```

```
    // 用零点负漂值< zero_neg_drift >修正读数
    if(zero_neg_drift)
    {
        reading_fixed = reading - zero_neg_drift;
        // 检测修正后的读数，连续有 10 个为负值，则重新取读数计算零点负漂值
        if(reading_fixed < 0)
        {
            neg_fixed_num ++;
            if(neg_fixed_num >= 10)
            {
                drift_update_enable = 1;
                neg_fixed_num = 0;
            }
        }
        else
            neg_fixed_num = 0;
    }
    // 该函数返回修正后的读数
    return reading_fixed;
}
```