LGT8F684A

FLASH Based 8bit Microcontroller

应用笔记

V1.0.1

2016/7/9

LGT8F684A 应用笔记

内容概述

文档主要介绍如何利用 LGT8F684A 自带校准系数，实现高精度的 ADC 应用。

本应用比较只针对使用内部参考作为 ADC 转换参考电压的应用。使用外部参考或者 VCC 作为参考的情况，不适用本文档的算法。

概述

LGT8F684A 内部集成的 12 位 ADC 具有较高的精度和线性度。但由于芯片生产工艺的偏差，可能会导致有个别芯片出现较大的转换误差。经批量测试， 误差最大到±20mV。误差产生的原因主要是制造的一致性和因此而造成的 ADC 内部的失调误差。

为满足对 ADC 的精度有更高要求的应用，我们通过在测试时对芯片逐个校准，测量出用于校准 ADC 误差的系数。 软件可以利用这些系数，配合一个简单的算法， 实现高精度的 ADC 转换结果。

经我们批量测试， 校准后的芯片，转换误差可以保证在 5mV 以内，其中 90%以上的误差在 2mV 左右。

需要注意的是，早期提供的芯片内部没有包含 ADC 校准参数。

校准参数包含了一个有效标志位，我们可以通过检测这个标志位，确定校准参数是否可用。

文档涉及到的算法只适用于使用 LGT8F684A 内部 1.2V 参考的 ADC 应用。

ADC 校准参数的定义：

ADC 校准参数被映射到存储空间的 0xD6~0xD9 四个字节，含义如下：

校准系数	地址	功能说明
RATIOV	0x0D6	校准比例系数
RATIOF	0x0D7	RATIOV 有效标志位 0x55 = RATIOV 参数有效
ADCVZV	0x0D8	校准误差系数
ADCVZF	0x0D9	ADCVZV 有效标志位 0x55 = ADCVZV 参数有效

RATIOV 和 ADCVZV 均为一个字节大小的有符号数，在使用前，请用如下方式进行定义：

ADC 校准系数定义

```
//-----------------------------------------
// LGT8F684A test case
//-----------------------------------------
#include "lgt8f684a.h"

volatile signed char  RATIOV  @ 0x0D6;
volatile signed char  RATIOF  @ 0x0D7;
volatile signed char  ADCVZV  @ 0x0D8;
volatile signed char  ADCVZF  @ 0x0D9;
```

请特别需要注意，这些系数均为有符号数据，确认声明为有符号数。在后面将要介绍的 ADC 校准算法，也均为有符号运算。 另外，由于这部分地址超出常规的数据空间，因此编译器会对以上地址的定义给出警告，此处可忽略。

ADC 校准算法

校准算法原理非常简单：基于 LGT8F684A 内置 ADC 具有非常高的线性，我们只需要对转换的结果经行一个比例的调整，既可以得到非常精确的结果。

校准参数的产生，本身经过了较为复杂的运算，这样大大降低了应用程序校准的开销。在获得高精度转换结果的同时，也不用付出速度和代码空间的浪费。

ADC 校准算法 （在 ADC 转换函数中实现）

```
//-----------------------------------------
// LGT8F684A ADC Calibration
//-----------------------------------------
u16 read_adc()
{
    u16 tmp, tmp2;

    GO = 1;
    NOP();
    while(nDONE);
    tmp = (ADRESH << 8) | ADRESL;

    if(RATIOF == 0x55) {
        tmp -= ADCVZV;
        tmp2 = (tmp >> 2) * RATIOV;
        tmp2 >>= 9;
        tmp -= tmp2;
    }

    return tmp;
}
```

以上函数完成对 ADC 转换结果的校准，配合适当的滤波算法，即可得到非常稳定和精确的采样结果。

版本历史

版本	作者	日期	版本日志
1.0.1	LGT	2016/7/09	更改补偿算法中数据的类型为 16 位无符号数据
1.0.0	LGT	2016/6/27	The first edition