

烧录器对烧录芯片内置 BG 的 ADC 值做 offset 校正

一、目的：

针对带 ADC MCU 芯片内置 Bandgap 电压 ADC 值 Offset 校正，可提升电源电压量测计算精度。

二、说明：

烧录芯片时由烧录器提供高精度参考电压 1.225V 及稳定的精准电压。计算得出每一颗待测芯片的 BG-ADC-Offset 值，并且烧录至指定地址。客户应用例程在读取 BG 的 ADC 值时后再增减 BG-ADC-Offset 值。此方法可提升电源电压量测计算精度。

三、适用芯片：

PFS122

四、烧录工具版本：

P003Bx Serial Writer

五、IDE 版本：

0.97B9 开始支持

六、使用烧录机台的环境注意事项/要求：

1. 预先对烧录器做校验以确认精度及功能；
2. 烧录机台与烧录器共地，降低干扰；
3. 对于烧录器与探针的电缆长度及材质阻抗，建议用屏蔽线，并且屏蔽线隔离地需妥善接地；
4. 烧录探针寿命及接触电阻需检查；
5. 建议在烧录探针侧的 IC 电源线加 103 电容；
6. 烧录机台与外围环境不能有噪声及其他干扰源；
7. 相关烧录注意事项请参照相关 AP-Note。

七、对策一：

优点：只做一次烧录，省时。

缺点：系统将占用较多的用户程序空间。（需于应用程序中插入对应的代码）

1. 于 Adjust_IC 的命令行插入新的参数：

字段一：运行函数名称 RUN=Add_Run_Func

字段二：运行函数所需时间（:40ms）

字段三：运行函数时的芯片电压（:3.2V）

字段四：BG Offset 写入 / 读取地址（:0x7FC）

字段五：PA4 上高精度 1.225V 的 ADC 误差范围值（:0x0A）

字段五：BG 1.200V 的 ADC 误差范围值（:x2F）

RUN=Add_Run_Func:40mS:3.2V:0x7FC:0x0A:0x2F;

Example:

.ADJUST_IC SYSCLK=IHRC/4, IHRC=16MHz, VDD = 4.8V, **RUN=Add_Run_Func:40mS:3.2V:0x7FC:0x0A:0x2F;**

2. 校正例程如下：

```
#define target_Add_Run_Volt    4800
#define target_ExtV1225_adc    (1225*4096/(target_Add_Run_Volt))
#define target_BGV1200_adc    (1200*4096/(target_Add_Run_Volt))

//byte data; //分辨率为 8 测得的 ADC 值
word data, get_bg1200_data, get_v1225_data; //分辨率为 12 测得的 ADC 值
word VDD_data;

void ADC_Run1(void) //初始化
{
    byte ilop = 2, Sample_cnt;
    word tmp_adc;
    Eword _Temp;

    $ ADCM /8 //时钟源选择（系统时钟/X）X 有 /1, /2, /4, /8, /16, /32, /64, /128
    //注：时钟源选择建议选用 500K（/2）

    do
    {
        // Enable; 停用: Disable //通道选择: PA4 or BANDGAP
        if (ilop == 2)
            $ ADCC Enable, PA4; //启用 ADC 功能，AD 转换的输入通道选择 PA4
        else
            $ ADCC Enable, BANDGAP; //启用 ADC 功能，AD 转换的输入通道选择 Bandgap

        .delay 1000*10; //延时
    }
```

```

//注：时钟源选择其他时，延迟时间请参考 datasheet
Sample_cnt = 0xff;      //测量 256 笔数据
_Temp = 0;              //256 笔数据之和
do
{
    AD_START = 1;        //开始 ADC 转换
    while(!AD_DONE)      //等待 ADC 转换结果
        NULL;
    data$1 = ADCRH;
    data$0 = ADCRL;
    data = data >> 4;
    _Temp += data;
}while(Icnt--);
data = _Temp >> 8; //取平均值
if (ilop == 2)
    get_v1225_data = data;
else
    get_bg1200_data = data;
}while(--ilop);
}

void Add_Run_Func (void)
{
    .delay 1000*4*5;      //delay 5ms
    while (1)
    {
        ////////////////////////////////// do something, here
        ADC_Run1 ();
        word ad_offs; // hi_byte
        word bg_offs; // lo_byte
        word tmp;

//-----External PA4 V1225 ADC offset
        if (get_v1225_data > target_ExtV1225_adc)
            { ad_offs = get_v1225_data - target_ExtV1225_adc;}
        else
            { ad_offs = target_ExtV1225_adc - get_v1225_data;}

        while(ad_offs$1) //PA4_V1225 Voltage Error
            { nop; }

//-----Internal BG V1200 ADC offset
        if (get_bg1200_data > target_BGV1200_adc) //
            { bg_offs = get_bg1200_data - target_BGV1200_adc;
              bg_offs$0 &= 0x7F;
            }
    }
}
```

```
    }
else
{   bg_offs = target_BGV1200_adc - get_bg1200_data;
    bg_offs$0 |= 0x80;
}

while(bg_offs$1) //BG_V1200 Voltage Error
{ nop; }

////////////////////////////////////
pa.6 = 1;          // after test done, notify writer via pa6 hi
.wait0 PA.3;       // wait writer ACK and then send data to writer

tmp$1 = ad_offs$0; tmp$0 = bg_offs$0;
byte cnt=16;

// send ad_offs (hi_byte) + bgoffs (lo_byte) to writer
do {
    tmp <<= 1;

    if (CF) pa.6 = 1;
    else    pa.6 = 0;
    .wait1 PA.3;
    .wait0 PA.3;
} while (--cnt);

.wait1 PA.3; // next voltage if need
}
}
```

```
//----- Normal ADC Scan -----  
void ADC_mea(void)//测量数据  
{  
    word tmp_adc;  
    byte bg_offset;  
    byte lcnt = 16;//测量 16 笔数据  
    Eword _Temp = 0;//16 笔数据之和  
    while(lcnt--)  
    {  
        AD_START = 1;        //开始 ADC 转换  
        while(!AD_DONE)      //等待 ADC 转换结果  
            NULL;  
        data$1 = ADCRH;  
        data$0 = ADCRL;  
        data = data >> 4;  
        if (adc_ch == 0x0F)// ADC Measure channel is BG  
        {  
            dc _SYS(OP.CALL_A) + 0x7FC;  
            bg_offset = A;  
            if (bg_offset.7 == 0) // bg adc value is positive  
                data -= bg_offset; //  
            else  
            {  
                bg_offset &= 0x7F;  
                data += bg_offset; // bg adc value is negative  
            }  
        }  
        _Temp += data;  
    }  
    data = _Temp >> 4;//取平均值  
    data = data + pre_data;  
    data >>= 1;  
    pre_data = data;  
    A= data$1;  
    UART_Send();  
    A= data$0;  
    UART_Send();  
}
```

```
void FPPA0 (void)
{
    .ADJUST_IC    SYSCLK=IHRC/4, IHRC=16MHz, VDD=3.8V, RUN=Add_Run_Func:100mS:Add_Run_Volt:0x7FC:0x0C:0x18;
    PAPH = 0xFF;  PBPB = 0x7F;
    $ UART_Out out;
    while (1)
    {
        ADC_mea();
        ADC_calcu();
        .delay 4*1000*500;
    }
}
```

八、对策二：

优点：系统占用较少的用户程序空间只做一次烧录，省时。

缺点：烧录较耗时，分成 2 次烧录，第一次烧录校正程序，第二次烧录客户应用程序

1. 于应用程序中插入下列命令

.Writer PDK_ADD BG_AD_offs_Test

2. 将烧录校正程序 BG_AD_offs_Test.PDK 与应用程序置放于同一个目录。

3. 应用程序 BG_ADC offset 补偿计算例程：

```
//----- Normal ADC Scan -----
void ADC_mea(void)//测量数据
{
    word tmp_adc;
    byte bg_offset, Scancnt = 16;//测量 16 笔数据
    Eword _Temp = 0;//16 笔数据之和
    while(1cnt--)
    {
        AD_START = 1;          //开始 ADC 转换
        while(!AD_DONE)        //等待 ADC 转换结果
            NULL;
        data$1 = ADCRH;
        data$0 = ADCRL;
        data = data >> 4;
        if (adc_ch == 0x0F) // ADC Measure channel is BG
        {
            dc _SYS(OP.CALL__A) + 0x7FC;
            bg_offset = A;
            if (bg_offset.7 == 0) // bg adc value is positive
                data -= bg_offset; //
            else
            {
                bg_offset &= 0x7F;
            }
        }
    }
}
```

```
        data += bg_offset;    // bg adc value is negative
    }
}
_Temp += data;
}
data = _Temp >> 4; //取平均值
data = data + pre_data;
data >>= 1;
pre_data = data;
A= data$1;
UART_Send();
A= data$0;
UART_Send();
}
```

4. 校正例程如下：

```
#define target_Add_Run_Volt    4800
#define target_ExtV1225_adc    (1225*4096/(target_Add_Run_Volt))
#define target_BGV1200_adc    (1200*4096/(target_Add_Run_Volt))

word data, get_bg_v1200_data, get_v1225_data; //
word VDD_data;

void ADC_init(void) //初始化
{
    byte ilop = 2;
    word tmp_adc;
    word Icnt;
    Eword _Temp;

    $ ADCM/2; //时钟源选择（系统时钟/X）//X有/1,/2,/4,/8,/16,/32,/64,/128
                //注：时钟源选择建议选用 500K (/2) //参考电压为 VDD
    do
    {
        if (ilop == 2)
            $ ADCC Enable, PA4;
        else
            $ ADCC Enable, BANDGAP; //启用 ADC 功能，信道选择（AD 转换的输入信号）
                //启用：Enable；停用：Disable //通道选择：PA4 or BANDGAP

        .delay 1000*10;                //延时 10ms
                //注：时钟源选择其他时，延迟时间请参考 datasheet

        Icnt = 1024;                //测量 1024 笔数据
        _Temp = 0;                //1024 笔数据之和
        do
        {
```

```
AD_START = 1;          //开始 ADC 转换
while(!AD_DONE)        //等待 ADC 转换结果
    NULL;

data$1 = ADCRH;
data$0 = ADCRL;
data = data >> 4;

    _Temp += data;
}while(Icnt--);

if (ilop == 2)
    get_v1225_data = _Temp >> 10; //取平均值
else
    get_bg_v1200_data = _Temp >> 10; //取平均值

}while(--ilop);
}

void Add_Run_Func (void)
{
    .delay 1000*10;      //delay 10ms
    while (1)
    {
        ///////////////////////////////////
        // do something, here
        ADC_init();
        word ad_offs; // hi_byte
        word bg_offs; // lo_byte
        word tmp;

//-----External PA4 V1225 ADC offset
        if (get_v1225_data > target_ExtV1225_adc)
        {
            ad_offs = get_v1225_data - target_ExtV1225_adc;
        }
        else
        {
            ad_offs = target_ExtV1225_adc - get_v1225_data;
        }

        while(ad_offs$1)
        {
            nop;
        }

//-----Internal BG V1200 ADC offset
        if (get_bg_v1200_data > target_BGV1200_adc)
```

```

    {
        bg_offs = get_bg_v1200_data - target_BGV1200_adc;
        while(bg_offs$1)
        {
            nop;
        }
        bg_offs$0 &= 0x7F;
    }
else
    {
        bg_offs = target_BGV1200_adc - get_bg_v1200_data;
        while(bg_offs$1)
        {
            nop;
        }
        bg_offs$0 |= 0x80;
    }

////////////////////////////////////
pa.6 = 1;          // after test done, notify writer via pa6 hi
.wait0 PA.3;       // wait writer ACK and then send data to writer

tmp$1 = ad_offs$0;  tmp$0 = bg_offs$0;
byte cnt=16;

// send ad_offs (hi_byte) + bgoffs (lo_byte) to writer
do {
    tmp <<<= 1;

    if (CF) pa.6 = 1;
    else    pa.6 = 0;
    .wait1 PA.3;
    .wait0 PA.3;
} while (--cnt);

.wait1 PA.3; // next voltage if need
}
}

voidFPPA0 (void)
{
    .ADJUST_IC    SYSCLK=IHRC/16, IHRC=16MHz, VDD = 4.8V;

    .OutFile      BG_AD_offs_Test          // outfile auto add PDK ext name
                // mS, Volt, Adr, 0x_hi_lo_offs
    .Reserved_User_Info { WORD 1000, WORD target_Add_Run_Volt+5, WORD 0x7FC, WORD 0x0A1F }

```

```
//==== Don't modify =====  
    PAC.3 = 0;  
    pA.6 = 0;    PAC.6 = 1;  
    .wait1 PA.3;  
//=====
```



```
    Add_Run_Func();  
    goto $;
```



```
}
```

在烧录器上做烧录校正 BG ADC Offset 测试时，若测试失败将会出现下列讯息：

PC 软件上会显示 “ Check Ext.Run fail ”

P003Bx 烧录器上的 LCD 屏上会显示 “ IC not work:Run ”

若您在使用上有任何疑惑，请就近洽询我们的代理商，或与我们 fae@padauk.com.tw 联系。