

# 二次标定

## 一. 准备工作:

1. 进行二次标定前须保证模组在整机内处于**稳定状态**。  
模组的稳定状态表现为：当对着均匀面时，探测器均值稳定。
2. 模组状态稳定的前提条件：  
确保整机散热状况良好，并且模组在整机内的温度（快门温度、FPA 温度）不能过高且模组**受热均匀**。

## 二. 使用:

1. 获取对应模式下测温点的 nuc 值，设置测温精度，使用 **autoCalibrate** 接口
2. 标定若成功，需使用发送接口将 **c\_fix** 和 **shutter\_fix** 存储到模组用户区对应位置，并且发送 0x80ff 保存所得参数。
3. 标定成功后，温度表计算接口需要替换成 **thermometryT4Line\_auto\_fix**。

## 三. 注意事项

1. 若标定使用黑体发射率不是 0.98，须先将对应的发射率保存到模组中的用户区。其次，标定距离须为 1m。
2. 测温点的选择须将测温范围的首尾两端包括其中，中间测温点可根据实际测温表现进行选取。例如，测温范围在 10℃~120℃，第一和第三测温点可为 20 和 100，第二测温点可为中间点 60。
3. 精度（precision）设置说明。例如将精度设置为 3，表示在高增益模式下误差范围在 ±3，低增益模式下误差范围为 ±3%。
4. 模组状态稳定才可进行标定。到达稳定状态时间需要根据产品实际情况进行测算。

附录：

1. 用户区参数保存位置

参数名	起始位置
Correction（整体修正）	0
Reflection（发射温度）	4
Amb（环境温度）	8
Humidity（湿度）	12
Emissivity（发射率）	16
Distance（距离）	20
C_fix_NT（高增益 c_fix）	26
Shutter_fix_NT（高增益 shutter_fix）	30
C_fix_HT（低增益 c_fix）	34
Shutter_fix_HT（低增益 shutter_fix）	38

注：Distance（距离）为 unsigned short 类型，其余均为 float 类型。

下图为 Linux 发送接口示例，红框内即为参数存储位置。

```
void sendCorrection(float correction)
{
    unsigned char inputCo[4];
    memcpy(inputCo,&correction,sizeof(float));
    sendFloatCommand(0 * 4, inputCo[0], inputCo[1], inputCo[2], inputCo[3], 20, 40, 60, 80, 120);
}
```

下图为 Android 发送接口示例。

```
void UVCCamera::sendC_fix(float c_fix)
{
    unsigned char inputC_fix[4];
    memcpy(inputC_fix,&c_fix,sizeof(float));
    sendFloatCommand(26, inputC_fix[0], inputC_fix[1], inputC_fix[2], inputC_fix[3], 20, 40, 60, 80, 120);
}
```