

DM20171203-BA45F5240-独立烟感（双发）

功能说明

IC 型号：BA45F5240-2

封装：16 NSOP-A

编程语言：C

IDE 版本：HT-IDE3000 8.05

1. **供电方式**：DC 3.0V。
2. **上电自检**：上电后红灯闪一下，蜂鸣器响一声，若设备没有标定，红灯保持常亮。
3. **零点标定模式**：短接 BD 端口上电会进入标定模式，红灯每秒闪烁一次，标定成功进入待机模式，失败红灯常亮。
4. **按键标定**：按住测试按键上电后，连接三下测试按键可触发零点标定；
5. **待机模式**：无其他异常状态时设备会进入待机模式，红灯每 40s 闪烁一次。
6. **低压状态**：电压低于 2.7 V 时会判定为低电压状态，该状态会在待机模式的基础上每 40s 蜂鸣器响一声。
7. **报警模式**：设备检测到一定的烟雾浓度时会触发报警模式，报警后红灯持续闪烁，蜂鸣器响 3 声停 2s 周期循环。白烟报警灵敏度为 0.3dB/m 左右；
8. **消音模式**：报警模式下按下测试按键进入消音模式，该模式下红灯保持闪烁，但蜂鸣器不响，持续时间 90s。
9. **测试模式**：在正常待机状态按下测试按键会触发测试模式，该模式会模拟报警状态，按键松开退出该模式。
10. **故障报警**：当检测到发射、接收管故障时，蜂鸣器每 40s 响两声，红灯闪两下。
11. **内阻检测**：未开启；
12. **Debug 输出**：输出端口 PA0，按住按键上电后自动输出，波特率 115200。

EEPROM 地址参数说明:

注: 以下参数均为会在运行过程实时读取, 改变 EEPROM 的参数会立即生效;

;白烟通道

- ;000H: 零点标定下限 (标定零点值小于该值无法通过标定)
- ;001H: 零点标定上限 (标定零点值大于该值无法通过标定)
- ;002H: 标定记录 (标定时自动记录, 不能修改)
- ;003H: 标定零点 (标定时自动记录, 不能修改)
- ;004H: 漂移补偿量 (不能修改)
- ;005H: 设定报警阈值
- ;006H: 标定校验码 (标定时自动记录, 不能修改)
- ;007H: 白烟通道 SDPGAC0
- ;008H: 白烟通道 SDPGAC1
- ;009H: ISGDATA0
- ;00AH: IR 开发射时间 (发射时间 $t = ((1.5 * x) + 1.5) * 2$)
- ;00BH: 温度补偿上限 (当 R_TEMP_ADC 大于该值时取代, 防止溢出)
- ;00CH: 温度补偿下限 (当 R_TEMP_ADC 小于该值时取代, 防止溢出)
- ;00DH: 高温补偿起始点 (当 R_TEMP_ADC 大于该值开始补偿)
- ;00EH: 低温补偿起始点 (当 R_TEMP_ADC 小于该值开始补偿)
- ;00FH: 补偿参数 K, 0FH 高字节, 10H 低字节
- ;011H: 补偿参数 B, 11H 高字节, 12H 低字节
- ;013H: 通道使能 (20H: 使能, 其他值: 除能)

;黑烟通道

- ;020H: 零点标定下限 (标定零点值小于该值无法通过标定)
- ;021H: 零点标定上限 (标定零点值大于该值无法通过标定)
- ;022H: 标定记录 (标定时自动记录, 无需修改)
- ;023H: 标定零点 (标定时自动记录, 无需修改)
- ;024H: 漂移补偿量 (不能修改)
- ;025H: 设定报警阈值
- ;026H: 标定校验码 (标定时自动记录, 无需修改)

;027H: 白烟通道 SDPGAC0
;028H: 白烟通道 SDPGAC1
;029H: ISGDATA1
;02AH: IR 开发射时间 (发射时间 $t = ((1.5 * x) + 1.5) * 2$)
;02BH: 温度补偿上限 (当 R_TEMP_ADC 大于该值时取代, 防止溢出)
;02CH: 温度补偿下限 (当 R_TEMP_ADC 小于该值时取代, 防止溢出)
;02DH: 高温补偿起始点 (当 R_TEMP_ADC 大于该值开始补偿)
;02EH: 低温补偿起始点 (当 R_TEMP_ADC 小于该值开始补偿)
;02FH: 补偿参数 K, 2FH 高字节, 30H 低字节
;031H: 补偿参数 B, 31H 高字节, 32H 低字节
;033H: 通道使能 (20H: 使能, 其他值: 除能)

通用

;038H: 白烟、黑烟变化量判定比值 A (为实际的 10 倍, 用于黑烟报警额外判定)
;039H: 白烟、黑烟变化量判定比值 B (为实际的 10 倍, 用于黑烟报警额外判定)
;当前比值大于 B 且小于 A 判定为黑烟报警, 大于 A 为白烟报警, 小于 B 不判定报警
;03AH: 烟感待机检测周期 (待机时的烟感检测周期, 取值范围 1 ~ 255)

温度补偿说明

温度补偿对象为标定零点值, 计算公式如下:

$$\text{温度补偿系数} = \text{温度 AD} \times K + B$$

$$\text{补偿后的标定零点} = \text{原标定零点} \times \text{温度补偿系数}$$

$$\text{补偿后的报警阈值} = \text{原报警阈值} \times \text{温度补偿系数}$$

补偿起始点: 当温度 AD 值大于高温补偿起始点或小于低温补偿起始点开始补偿, 若需全温补偿, 可将高温补偿起始点设为 000H, 低温补偿起始点设为 0FFH;

温度补偿上下限: 当温度 AD 值低于温度补偿下限或高于温度补偿上限时将使用设置的限值取代温度 AD 值进行运算;

标定温度校准: 为了避免不同温度下进行零点标定引起的误差, 进行零点标定时会自动将当前实际的零点换算成为参考温度的零点作为标定保存。

串口输出格式:

```

DebugIOEnable();
R_RS_CHECKSUM = 0x00;
DebugSend(0xAA); //0 固定的起始码
DebugSend(0x35); //1 数据长度

DebugSend(R_SMKRAM_POOL.smARefDat); //2 无发射的基准值
DebugSend(R_SMKRAM_POOL.smAIrDat); //3 IR发射值
DebugSend(R_SMKRAM_POOL.smADtDat); //4 IR发射差值
DebugSend(R_SMKRAM_POOLC.smAFinlBd0); //5 标定零点值
DebugSend(R_SMKRAM_POOL.smABdAlarm); //6 标定报警值
DebugSend(R_SMKRAM_POOL.smARunFlg.byte); //7 烟感相关标志位

DebugSend(R_SMKRAM_POOL.smBRefDat); //8 无发射的基准值
DebugSend(R_SMKRAM_POOL.smBIrDat); //9 IR发射值
DebugSend(R_SMKRAM_POOL.smBDtDat); //10 IR发射差值
DebugSend(R_SMKRAM_POOLC.smBFinlBd0); //11 标定零点值
DebugSend(R_SMKRAM_POOL.smBBdAlarm); //12 标定报警值
DebugSend(R_SMKRAM_POOL.smBRunFlg.byte); //13 烟感相关标志位

DebugSend(modeFlg.byte); //14 工作模式
DebugSend(R_BAT_ADC); //15 电压AD值
DebugSend(R_TEMP_ADC); //16 温度AD值
DebugSend(R_SMKRAM_POOLC.smABRatio); //17 白烟、黑烟变化量比值
DebugSend(R_SMKRAM_POOLC.smATemp0ftBd0); //18 白烟通道温度补偿后的标定零点值
DebugSend(R_SMKRAM_POOLC.smBTemp0ftBd0); //19 黑烟通道温度补偿后的标定零点值
DebugSend(R_SMKRAM_POOLC.smADft0ftBd0); //20 白烟通道漂移补偿后的标定零点值
DebugSend(R_SMKRAM_POOLC.smBDft0ftBd0); //21 黑烟通道漂移补偿后的标定零点值
DebugSend(R_SMKRAM_POOL.smABdZero); //22
DebugSend(R_SMKRAM_POOL.smBBdZero); //23
DebugSend(R_SMKRAM_POOLC.smADftStat); //24
DebugSend(R_SMKRAM_POOLC.smBDftStat); //25
DebugSend(R_SMKRAM_POOLC.smAHistMaxChg); //26
DebugSend(R_SMKRAM_POOL.smAlaSta.byte); //27

DebugSend(R_NEROTIME[2]); //28
DebugSend(R_NEROTIME[1]); //29
DebugSend(R_NEROTIME[0]); //30

DebugSend((unsigned char)(C_SOFT_VERSION >> 8)); //31 软件版本号[15:8]
DebugSend((unsigned char)(C_SOFT_VERSION >> 0)); //32 软件版本号[7:0]

DebugSend(0x55); //33
DebugSend(R_RS_CHECKSUM); //34 校验和
  
```