

类别	内容
关键词	温度采集, I ² C时序, I ² C接口寄存器, 断线检测
摘要	ZAM6222B双通道热电阻测温模块, 主要用于温度传感器PT1000的温度测量, 最多可进行两通道测量, 内部采用24位的Sigma-Delta ADC进行信号采样, 实现温度测量的高分辨率和高精度。模块通过I ² C通信接口与外部设备通信, 并且模块具备断线检测功能。

修订历史

版本	日期	原因
V1.00	2023/07/26	创建文档

目 录

1. 功能简介	1
1.1 概述	1
1.2 产品特性	1
1.3 应用场合	1
2. 硬件说明	2
2.1 产品外观	2
2.2 引脚定义	2
2.3 使用说明	3
2.3.1 双路测温应用电路	3
2.3.2 单路测温应用电路	3
3. 功能配置	5
3.1 功能描述	5
3.2 接口说明	5
3.2.1 I ² C 总线工作状态	5
3.2.2 主机模拟 I ² C 配置	6
3.2.3 数据传输流程	7
3.2.4 I ² C 时序	9
3.2.5 片选功能时序	10
3.3 寄存器	12
3.3.1 Pointer 寄存器	12
3.3.2 Temperature 寄存器	12
3.3.3 Configurature 寄存器	13
3.3.4 THIGH 与 TLOW 寄存器	15
3.4 断线检测功能	16
4. 产品使用注意事项	18
5. 免责声明	19

1. 功能简介

1.1 概述

ZAM6222B 双通道热电阻测温模块，主要用于配合温度传感器 PT1000 进行温度测试，可同时进行两通道温度检测，内部采用 24 位的 Sigma-Delta ADC 进行信号采样，实现温度测试的高分辨率和高精度。模块通过 I²C 通信接口与外部设备通信。

1.2 产品特性

- ◆ 双通道 PT1000 热电阻
- ◆ -100°C~300°C测温范围
- ◆ 0.01°C测温分辨率
- ◆ 0.1%±0.1°C测温误差（二线制）
- ◆ 0.02%±0.1°C测温误差（三线制）
- ◆ 10ppm/°C温漂
- ◆ 断线检测功能
- ◆ I²C 通信接口
- ◆ 3.3V 供电电压
- ◆ 温度报警输出
- ◆ 工作环境 -40°C~+85°C

1.3 应用场合

- ◆ 工业恒温箱
- ◆ 充电桩
- ◆ 电池测试设备
- ◆ 温度监控器
- ◆ 测温仪表
- ◆ 钢铁重工
- ◆ 医疗设备
- ◆ 电力温度监控
- ◆ 石油化工
- ◆ 天然气管道
- ◆ 热循环系统

2. 硬件说明

2.1 产品外观

产品外观如图 2.1 所示。



图 2.1 产品外观图

2.2 引脚定义

ZAM6222B 模块灌封在 24.98×16.90×8.10mm 塑胶外壳内，产品引脚排列如图 2.2 所示。

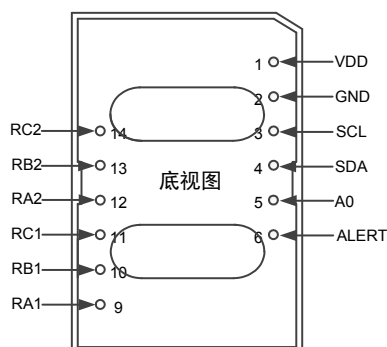


图 2.2 ZAM6222B 引脚底视图

ZAM6222B 各引脚功能如表 2.1 所示。

表 2.1 引脚功能描述

名称	引脚	I/O	功能描述
VDD	1	I	供电电源，3V 到 3.3V
GND	2	--	接地端
SCL	3	I	I2C 通信时钟引脚
SDA	4	I/O	I2C 通信数据引脚
A0	5	I	I2C 地址选择引脚，接地(0x48)，接 VDD(0x49)
ALERT	6	O	报警信号输出引脚
RA1	9	--	外部热电阻 PT1000 接口，外接 1 通道热电阻 A 端
RB1	10	--	外部热电阻 PT1000 接口，外接 1 通道热电阻 B 端
RC1	11	--	外部热电阻 PT1000 接口，外接 1 通道热电阻 C 端
RA2	12	--	外部热电阻 PT1000 接口，外接 2 通道热电阻 A 端
RB2	13	--	外部热电阻 PT1000 接口，外接 2 通道热电阻 B 端
RC2	14	--	外部热电阻 PT1000 接口，外接 2 通道热电阻 C 端

2.3 使用说明

2.3.1 双路测温应用电路

ZAM6222B 双路热电阻测温模块只需要一个 I²C 主机和一个 3.3V 直流电源即可完成两路温度采集。ZAM6222B 模块的 RA1、RB1、RC1 与一路三线制热电阻 PT1000 相连，模块的 RA2、RB2、RC2 与另一路三线制热电阻 PT1000 相连，I²C 主机通过标准 I²C 接口与 ZAM6222B 模块通信即可完成双路温度采集，三线制温度采集电路如图 2.3 所示。ZAM6222B 也支持二线制温度采集，只需将 RA1、RC1 和 RA2、RC2 分别与两路 PT1000 连接即可，二线制温度采集电路如图 2.4 所示。

根据 PT1000 三线制测温原理，建议 RA1~RC1 引脚到 PT1000 接线端子上的 PCB 布线做等长等宽处理，RA2~RC2 引脚到 PT1000 接线端子上的 PCB 布线做等长等宽处理。

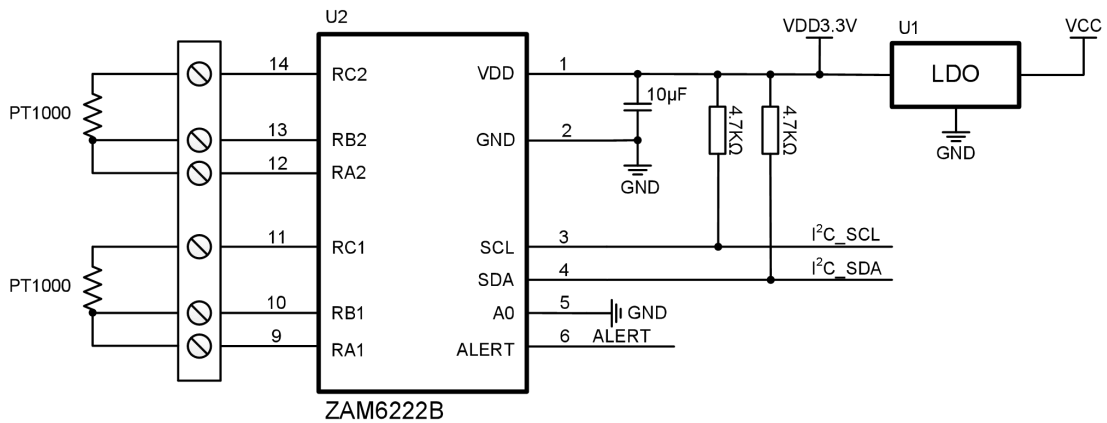


图 2.3 ZAM6222B 双路温度采集三线制典型电路

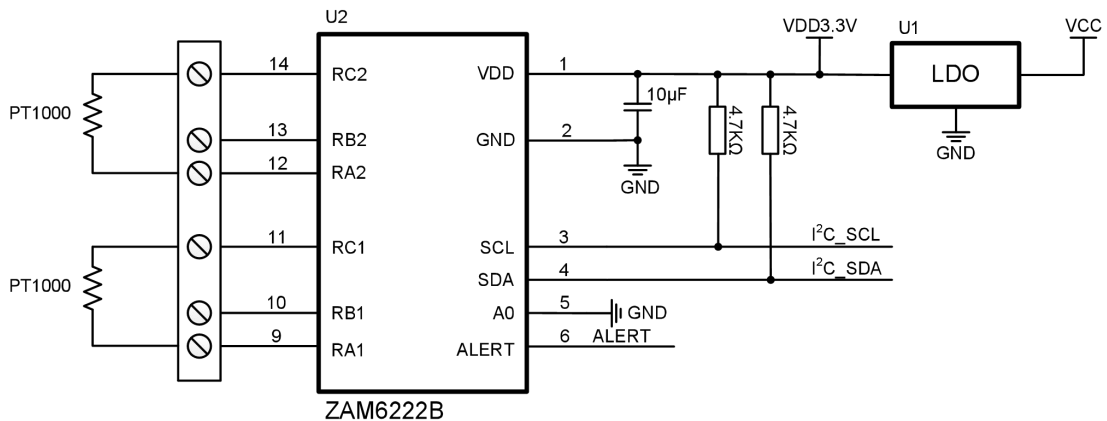


图 2.4 ZAM6222B 双路温度采集二线制典型电路

2.3.2 单路测温应用电路

ZAM6222B 双路热电阻测温模块兼容单路温度采集，使用单路温度采集，另一路热电阻必须短接。使用通道一做单路温度采集三线制典型电路如图 2.5 所示，通道二需要将 RA2、RB2、RC2 都短接到一起；使用通道一做单路温度采集二线制典型电路如图 2.6 所示，通道二需要将 RA2 和 RC2 短接。同理，使用通道二做单路采集时，通道一也需按相同方式短接。

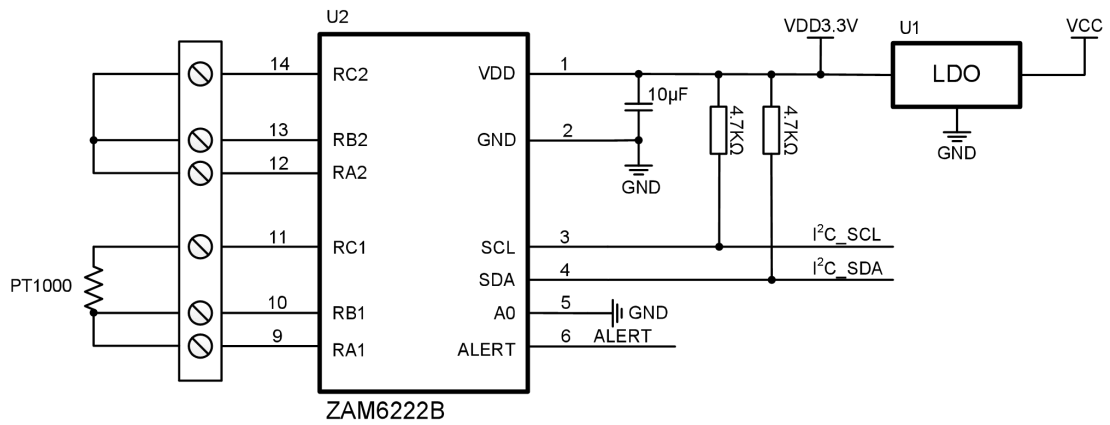


图 2.5 ZAM6222B 采集通道一温度三线制典型电路

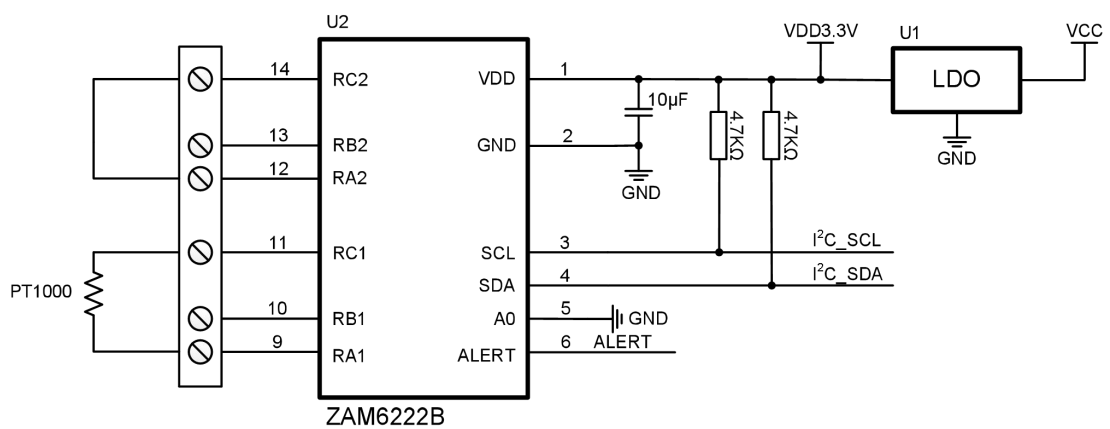


图 2.6 ZAM6222B 采集通道一温度二线制典型电路

3. 功能配置

3.1 功能描述

ZAM6222B 热电阻温度测量模块，只需接入 PT1000 热电阻，即可完成温度的采集，采用标准 I²C 接口直接输出以℃为单位的温度数据。模块同时具备温度报警功能和断线检测功能，并且支持级联模式。

3.2 接口说明

3.2.1 I²C 总线工作状态

ZAM6222B 使用标准 I²C 接口通信，I²C 总线工作状态说明如下：

1. 总线空闲

SDA 和 SCL 均保持为高电平。

2. 数据传输开始

SDA 的状态由高到低，SCL 保持为高，这个状态即为数据传输开始的状态。

3. 数据传输结束

SDA 的状态由低到高，SCL 保持为高，这个状态即为数据传输结束的状态。

4. 数据传输

- (1) 发送到 SDA 线上的每个字节必须是 8 位，每次传输可以发送的字节数量不受限制。每个字节后必须跟一个响应位。首先传输的是数据的最高位（MSB）。
- (2) 如果从机要完成一些其他功能后（例如一个内部中断服务程序）才能接收或发送下一个完整的数据字节，可以使时钟线 SCL 保持为低电平，迫使主机进入等待状态，当从机准备好接收一个数据并释放时钟线 SCL 后，数据传输继续。
- (3) 数据传输必须带响应，相应的响应时钟脉冲由主机产生。在响应的时钟脉冲期间，发送器释放 SDA 线，接收器必须将 SDA 线拉低，使它这个时钟脉冲 SCL 的高电平期间保持稳定的低电平。通常被寻址的接收器在接收到每个字节后，必须产生一个响应。
- (4) 当从机不能响应从机地址时（例如它正在执行一些实时函数，不能接收或发送），从机必须使数据线保持为高电平，主机然后产生一个停止条件终止传输或者产生重复起始条件开始新的传输。
- (5) 如果从机接收器响应了从机地址，但是在传输了一段时间后，不能接收更多的数据字节，主机必须再次终止传输。这个情况用从机在第一个字节后没有产生响应来表示。从机使数据线保持高电平，主机产生一个停止或者重复起始条件。
- (6) 如果传输中有主机接收器，它必须通过在从机发出的最后一个字节时产生一个响应，向从机发送器通知数据结束，从机发送器必须释放数据线，允许主机产生一个停止或重复起始条件。

6. 时钟同步

- (1) 如果从机希望主机降低传送速度，可以通过将 SCL 主动拉低延长其低电平时间的方法来通知主机。
- (2) 当主机在准备下一次传送发现 SCL 的电平被拉低时就进行等待，直至从机完成操

作并释放 SCL 线的控制权。

- (3) 主机实际上受到从机的时钟同步控制，SCL 线上的低电平时长由时钟低电平最长的器件决定，高电平的时间由高电平时间最短的器件来决定。这就是时钟同步，它解决了 I²C 总线的速度同步问题。

3.2.2 主机模拟 I²C 配置

若主机使用的是模拟 I²C，需对总线状态进行判断，具体驱动例程可参考程序清单 3.1。若主机使用的是硬件 I²C，则无需此配置。

程序清单 3.1 主机模拟 I²C 驱动例程

```

/**
 * \brief I2C 总线忙等待
 *
 * 等待 I2C 总线释放为空闲状态。若 SCL 线为高电平，I2C 总线处于空闲状态。
 *
 * \param[in] timeout: 超时时间, 单位 us
 *
 * \retval 0: 等待成功
 * \retval 1: 等待超时
 */
static uint8_t __gpio_i2c_busy_wait(const am_i2c_gpio_devinfo_t *p_devinfo,
                                   uint32_t timeout)
{
    uint32_t wait_time = 0;

    while (__GPI2C_SCL_VAL_GET(p_devinfo) == 0) {
        /* 等待时间到或者 timeout 为 0 时，直接退出 */
        if (wait_time >= timeout) {
            return 1;
        } else {
            am_udelay(1);
            wait_time++;
        }
    }

    return 0;
}

/* 读取单字节数据，指定本次数据读取后是否应答 */
uint8_t am_i2c_gpio_read_byte (am_i2c_gpio_dev_t *p_dev,

```

```

        bool_t      ack)
{
    int i;

    uint8_t data = 0;

    const am_i2c_gpio_devinfo_t *p_devinfo = p_dev->p_devinfo;

    __GPI2C_SDA_INPUT(p_devinfo);

    /**
     * \brief 主机尝试拉高 SCL 线。若没有从机占用总线，则 SCL 线此时为高电平；
     * 若存在从机占用总线，则 SCL 线此时为低电平，需要等待从机释放总线才能继续进行通信
     */

    __GPI2C_SCL_HIGH(p_devinfo);
    /* 等待从机释放总线 */
    if (__gpio_i2c_busy_wait(p_devinfo, 200)) {
        am_i2c_gpio_stop(p_dev);
        return data;
    }

    __GPI2C_USER_DEALY(p_devinfo);
    if (__GPI2C_SDA_VAL_GET(p_devinfo)) {
        data |= 1 << 7;
    };

    __GPI2C_SCL_LOW(p_devinfo);
    __GPI2C_USER_DEALY(p_devinfo);
    for (i = 6; i >= 0; i--) {
        __GPI2C_SCL_HIGH(p_devinfo);
        __GPI2C_USER_DEALY(p_devinfo);
        if (__GPI2C_SDA_VAL_GET(p_devinfo)) {
            data |= 1 << i;
        };
        __GPI2C_SCL_LOW(p_devinfo);
        __GPI2C_USER_DEALY(p_devinfo);
    }

    am_i2c_gpio_ack_put(p_dev, ack);

    return data;
}

```

3.2.3 数据传输流程

1. 主机发送数据流程

(1) 主机在检测到总线为“空闲状态”（即 SDA、SCL 线均为高电平）时，发送一个

启动信号“S”，开始一次通信的开始；

- (2) 主机接着发送一个写命令字节。该字节由 7 位的外围器件地址和一位写控制位 W 组成（此时 $W=0$ ）；
- (3) 相对应的从机接收到命令字节后，向主机回馈应答信号 ACK（ $ACK=0$ ）；
- (4) 主机接收到从机的应答信号后，接着发送一个 8 位的寄存器地址字节；
- (5) 从机接收到地址字节后，向主机回馈应答信号 ACK；
- (6) 主机接收到从机的应答信号后，开始发送第一个字节的数据；
- (7) 从机接收到数据后返回一个应答信号 ACK；
- (8) 主机接收到应答信号后，再发送下一个数据字节；
- (9) 当主机发送最后一个数据字节并收到从机的 ACK 后，通过向从机发送一个停止信号 P 结束本次通信并释放总线。从机收到 P 信号后也退出与主机之间的通信。

以上所述即为主机发送数据的步骤，图 3.1 为主机发送的数据格式。

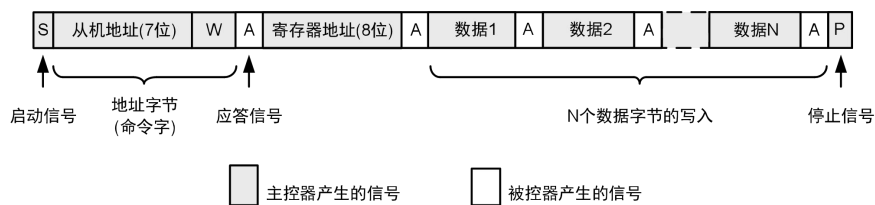


图 3.1 主机发送数据图示

主机通过发送地址码与对应的从机建立了通信关系，而挂接在总线上的其他从机虽然同时也收到了地址码，但因为与其自身的地址不相符，因此提前退出与主机的通信。主机的一次发送通信，其发送的数据数量不受限制。主机通过 P 信号通知发送结束，从机收到 P 信号后退出本次通信。主机的每一次发送后都是通过从机的 ACK 信号了解从机的接收情况，如果应答错误则重发。

2. 主机接收数据流程

- (1) 主机发送启动信号后，接着发送写命令字节（其中 $W=0$ ）；
- (2) 对应的从机收到命令字节后，返回一个应答信号；
- (3) 主机收到应答信号后向从机发送一个 8 位的寄存器地址字节；
- (4) 从机接收到地址字节后，向主机回馈应答信号；
- (5) 主机收到应答信号后向从机发送启动信号，接着发送读命令字节（其中 $R=1$ ）；
- (6) 对应的从机收到命令字节后，返回一个应答信号并向主机发送数据；
- (7) 主机收到数据后向从机反馈一个应答信号；
- (8) 从机收到应答信号后再向主机发送了下一个数据；
- (9) 当主机完成接收数据后，向从机发送一个“非应答信号（ $ACK=1$ ）”，从机收到 $ACK=1$ 的非应答信号后便停止发送；
- (10) 主机发送非应答信号后，再发送一个停止信号，释放总线结束通信。

以上所述即为主机接收数据的步骤，图 3.2 为主机接收数据的图示。

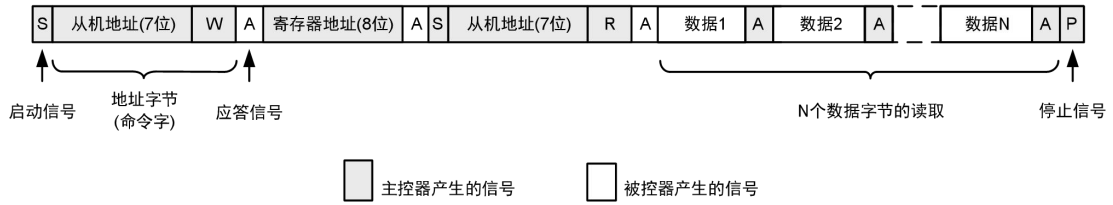


图 3.2 主机接收数据图示

主机所接收数据的数量是由主机自身决定，当发送“非应答信号”时从机便结束传送并释放总线（非应答信号的作用有两个，前一个数据接收成功，通知从机再次发送）。

3.2.4 I²C 时序

I²C 总线的传输速率限制如图 3.3 所示，时序传输时间统计如表 3.1 所示。

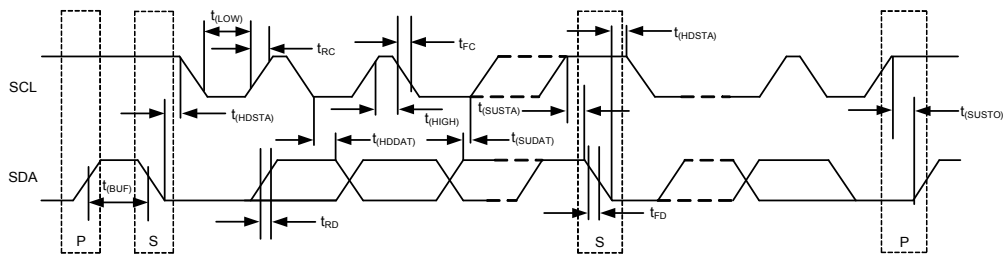


图 3.3 I²C 时钟与信号传输时间要求

表 3.1 时序传输时间统计

参数		最小值	最大值	单位
f _{SCL}	SCL 运行频率		100	kHz
t _{BUF}	总线空闲时间	4.7		μs
t _{HDSTA}	启动条件的保持时间	4		μs
t _{SUSTA}	启动条件的建立时间	4.7		μs
t _{SUSTO}	停止条件的建立时间	4		μs
t _{HDDAT}	数据保持时间（发送）	300		ns
t _{SUDAT}	数据建立时间（接收）	250		ns
t _{LOW}	SCL 时钟低电平保持时间	4.7		μs
t _{HIGH}	SCL 时钟高电平保持时间	4		μs
t _{RC} , t _{FC}	时钟线上升沿与下降沿时间		1000, 300	ns
t _{RD} , t _{FD}	数据线上上升沿与下降沿时间		1000, 300	ns

I²C 总线主机发送数据时序，如图 3.4 所示。

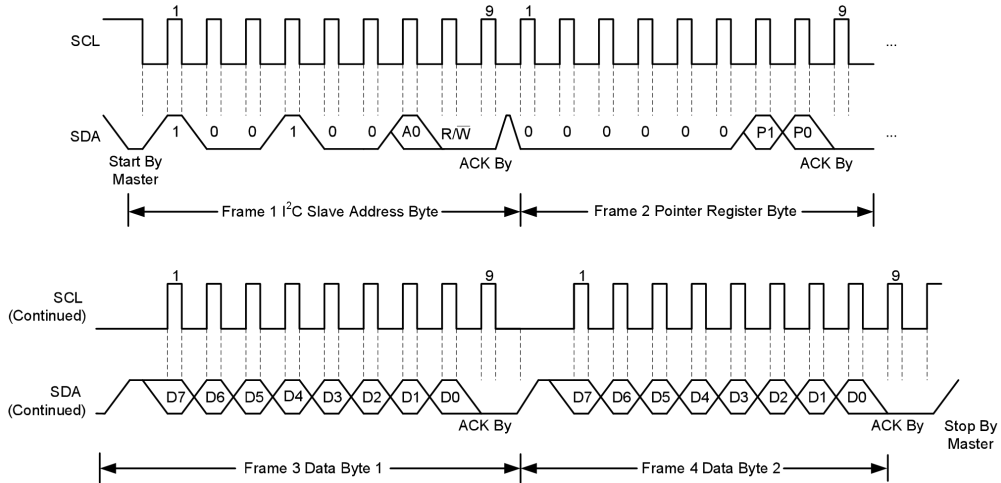


图 3.4 主机向 ZAM6222B 发送数据

温度测试模块 ZAM6222B 通过 I2C 向外进行通信时，测温模块作从机。主机接收 ZAM6222B 模块数据时序图如图 3.5 所示。

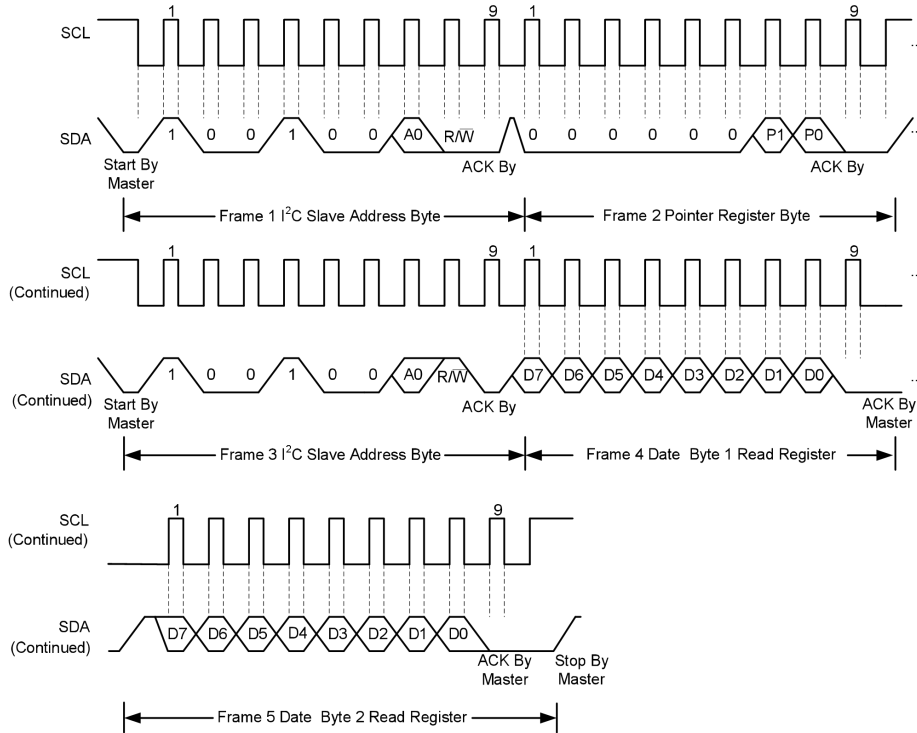


图 3.5 主机接收 ZAM6222B 数据

ZAM6222B 的地址设置为 100100x，即采用 7 位地址的方式，高 6 位固定为 100100。

3.2.5 片选功能时序

ZAM6222B 地址引脚 A0 除了常规地址功能，即通过上下拉地址引脚固定地址；还支持片选功能，主机 MCU 通过多个独立 I/O，分别控制各个 ZAM6222B 模块的地址引脚 A0，依次选通各模块的地址引脚，再读取各模块的数据。模块接线如图 3.6 所示。

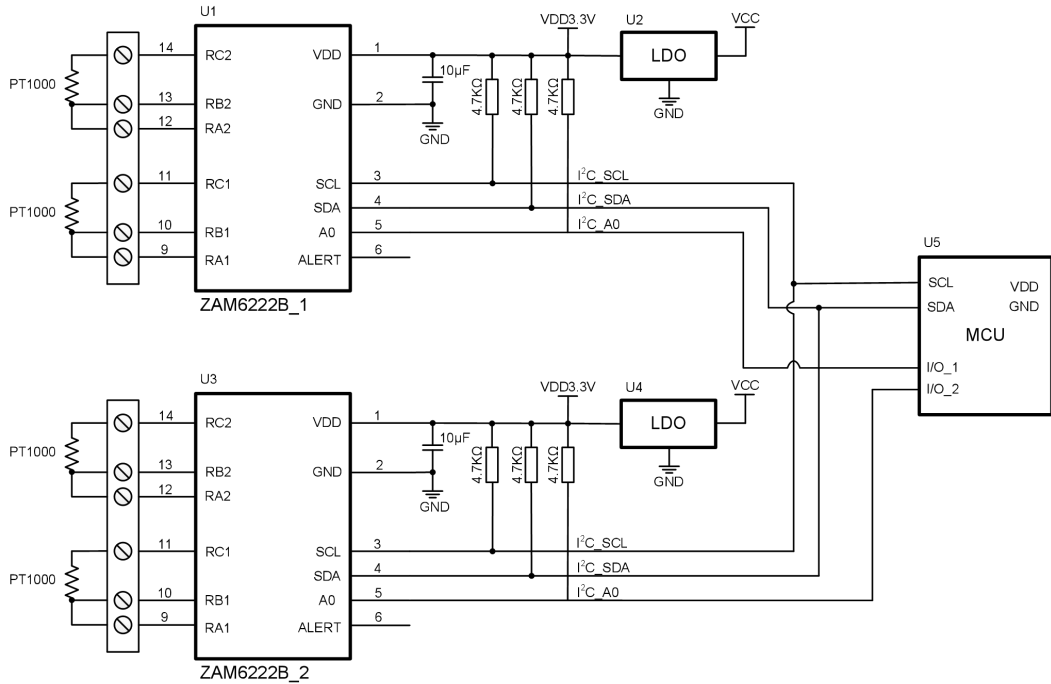


图 3.6 片选功能接线示意图

片选功能的地址引脚，支持高电平有效或低电平有效，可以根据需要选择。主机 MCU 通讯地址需与有效电平对应的地址一致，即高电平有效时对应的从机地址为 0x49，低电平有效时对应的从机地址为 0x48。以地址选通信号高电平有效为例，片选时序如图 3.7 与表 3.2 所示，主机发出地址选通信号上升沿 10ms 之后，才可以开始读取模块数据。当前模块的数据读完后，其对应地址 A0 的电平需拉低，下一个模块的地址选通信号上升沿，需与上一个模块的地址选通信号下降沿保持一定时间间隔，间隔时间需大于 25ms。

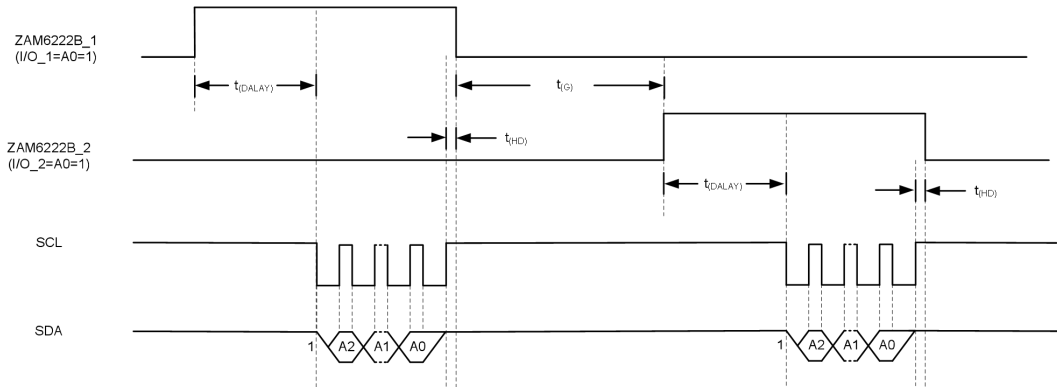


图 3.7 片选功能通讯时序图

表 3.2 时序传输时间统计

参数		最小值	最大值	单位
t_{DALAY}	地址翻转时间	10		ms
t_{HD}	读取数据后保持时间	0		ms
t_G	间隔时间	25		ms

3.3 寄存器

ZAM6222B 模块的 I²C 通信协议的寄存器结构如图 3.8 所示。主要分为 Pointer 寄存器、Temperature 寄存器、Configuration 寄存器、T_{LOW} 寄存器及 T_{HIGH} 寄存器。

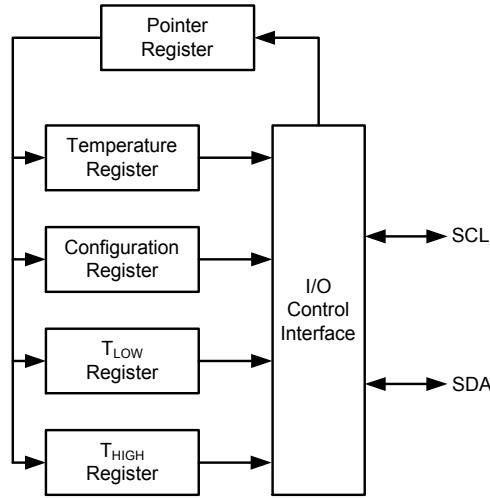


图 3.8 寄存器结构示意图

3.3.1 Pointer 寄存器

Pointer 寄存器的主要作用是指示 ZAM6222B 模块内部的其他四个寄存器地址。Pointer 寄存器的结构定义如表 3.3 所示，为一个字节的最后两位。

表 3.3 Pointer 寄存器内部结构

P7	P6	P5	P4	P3	P2	P1	P0
0	0	0	0	0	0	Register Bits	

Pointer 寄存器的指示定义如表 3.4 所示，分别指向图 3.8 中的四个寄存器。

表 3.4 Pointer 寄存器定义说明

P1	P0	读/写	寄存器
0	0	R only,default	Temperature Register
0	1	R/W	Configuration Register
1	0	R/W	TLOW Register
1	1	R/W	THIGH Register

3.3.2 Temperature 寄存器

Temperature 寄存器包含 2 个 24 位的寄存器，即包含 6 个字节，具体如下所示。表 3.5 为温度与寄存器位对应关系，Temperature 的两个寄存器前 3 个字节存储通道 1 的温度测试结果，后 3 个字节存储温度 2 的测试结果，通过 I²C 读取温度测试结果时，连续传输 6 个字节，先传输高字节，后传输低字节，下文写配置寄存器也是先传输高字节后传输低字节。

表 3.5 温度测试值寄存器

通道 1 温度寄存器	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	1	T23	T22	T21	T20	T19	T18	T17	T16
	Byte	D7	D6	D5	D4	D3	D2	D1	D0

续上表

通道 1 温度寄存器	2	T15	T14	T13	T12	T11	T10	T9	T8
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	3	T7	T6	T5	T4	T3	T2	T1	T0
通道 2 温度寄存器	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	4	T23	T22	T21	T20	T19	T18	T17	T16
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	5	T15	T14	T13	T12	T11	T10	T9	T8
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	6	T7	T6	T5	T4	T3	T2	T1	T0

表 3.6 温度与寄存器位对应关系

温度值 (°C)	数字输出	
	二进制	16 进制
1023.999878	0111 1111 1111 1111 1111 1111	7F FF FFH
0	0000 0000 0000 0000 0000 0000	00 00 00H
-0.000122	1111 1111 1111 1111 1111 1111	FF FF FFH
-1024	1000 0000 0000 0000 0000 0000	80 00 00H

如表 3.6 所示，为 ZAM6222B 模块每通道 24 位温度寄存器与温度值对应关系。24 位的分配如下，最高位为符号位，高 10 位作整数，低 13 位作小数显示，温度测量分辨率为 0.000122 °C。

两个通道六字节数据，前三位字节数据对应通道 1，后三位字节数据对应通道 2，每通道的数据转换关系如下：

$$value = Byte1 \ll 16 + Byte2 \ll 8 + Byte3$$

当 $value \geq 2^{23}$ 时，即当前测量温度为负温度值：

$$T = -(2^{24} - value) / 2^{13}$$

当 $value < 2^{23}$ 时，即当前测量温度为正温度值：

$$T = value / 2^{13}$$

3.3.3 Configuration 寄存器

Configuration 寄存器为 ZAM6222B 工作模式配置寄存器，主要用来配置测温模块的工作模式，以及采样率。具体的格式定义如表 3.7 所示。Configuration 包含两个字节，第一个字节配置通道 1，第二个字节配置通道 2。

表 3.7 Configuration 寄存器格式定义

BYTE	D7	D6	D5	D4	D3	D2	D1	D0
1	EN	ALERT	R0	F1	F0	POL	TM	SD
BYTE	D7	D6	D5	D4	D3	D2	D1	D0
2	EN	ALERT	R0	F1	F0	POL	TM	SD

SD: 控制 ZAM6222B 温度测试模块的测温线制配置，具体如表 3.8 所示。

表 3.8 测试方式配置

SD	测温线制	通道一连接端	通道二连接端
0	二线制	RA1、RC1	RA2、RC2
1	三线制	RA1、RA2、RC2	RA2、RA2、RC2

TM: 报警信号状态控制。当 TM=0 时，报警信号输出为比较模式，TM=1 时，报警信号输出为中断模式。

POL: 报警引脚输出信号状态控制，具体如图 3.9 所示。在比较模式下（Comparator Mode），即 TM=0，当 POL=0 时，ALERT 输出高电平，当测试温度大于 T_{HIGH} 时，ALERT 变为 0，直到测试温度小于 T_{LOW} ，则 ALERT 由变为 1。当 POL=1 时，ALERT 输出 0，当测试温度大于 T_{HIGH} 时，ALERT 变为 1，直到测试温度小于 T_{LOW} ，则 ALERT 由变为 0。在中断模式下（Interrupt mode），即 TM=1，当 POL=0 时，ALERT 输出为 1，当测试温度大于 T_{HIGH} 时，ALERT 输出为 0，当外部设备读取 Temperature 寄存器时，ALERT 输出状态被清除，变为初始状态为 1，直到下一次温度测试值小于 T_{LOW} ，ALERT 的状态再次改变。当 POL=1 时，ALERT 输出为 0，当测试温度大于 T_{HIGH} 时，ALERT 输出为 1，当外部设备读取 Temperature 寄存器时，ALERT 输出状态被清除，变为初始状态为 0，直到下一次温度测试值小于 T_{LOW} ，ALERT 的状态再次改变。

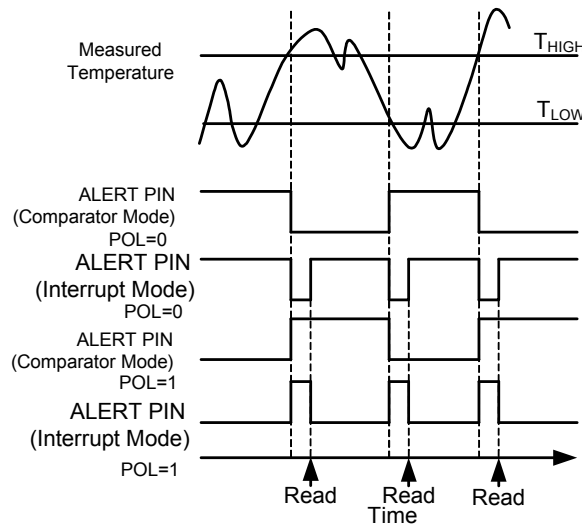


图 3.9 ALERT 信号输出状态设置图示

F1、F0: 当在过温或低温状态下，触发 ALERT 输出信号的温度值测试个数，具体如表 3.9 所示，例如，当 F1=1，F0=1 时，连续测试的温度值有 6 个温度值均在大于 T_{HIGH} 的情况下，才能触发 ALERT 输出相应的状态。

表 3.9 触发 ALERT 信号的温度值测试个数

F1	F0	连续触发个数
0	0	1
0	1	2
1	0	4
1	1	6

R0: 控制 ZAM6222B 温度测试模块的采样速率, 0 表示低速, 1 表示高速。具体如表 3.10 所示。

表 3.10 采样速率与转换时间设置

R0	SD	位数 (bit)	分辨率 ($^{\circ}\text{C}$)	采样率/通道 (SPS)	转换时间/通道 (s)
0	0	24	0.000122	1.25	0.8 (二线制)
1	0	24	0.000122	9	0.1 (二线制)
0	1	24	0.000122	0.6	1.6 (三线制)
1	1	24	0.000122	4.5	0.22 (三线制)

ALERT: 在比较模式下, 读取 ALERT 位的状态, 判断温度测试值是否正常。在刚开始进入比较状态时, ALERT 的输出状态与 POL 位相反。当 POL=1 时, ALERT=0, 直到测试温度值大于或等于 T_H , ALERT=1, 测试温度值小于 T_L 时, ALERT=0; 当 POL=0 时, ALERT=1, 直到测试温度值大于或等于 T_H , ALERT=0, 当测试温度值小于 T_L 时, ALERT=1。

判断报警的方式有两种, 一种是直接检测 ALERT 引脚的输出信号, 具体判断规则如图 3.9 所示, 另一种是通过 I²C 读取 ZAM6222B 的内部寄存器 Configuration 中的 ALERT 位来判断温度测试状态。

EN: EN=0 时, 该配置有效, EN=1 时, 该配置无效。设通道 1 的 EN 为 EN1, 通道 2 的 EN 为 EN2, 具体如表 3.11 所示。

表 3.11 模块的配置参数选择

EN1	EN2	配置参数选择	备注
0	0	选择通道 1 的配置参数	即模块的工作方式根据 Configuration 寄存器的字节 1 进行工作
1	1		
0	1		
1	0	选择通道 2 的配置参数	根据 Configuration 寄存器的字节 2 进行工作

对于 Configuration 寄存器, 用户没配置时, 默认为如表 3.12 所示。

表 3.12 默认配置参数

BYTE	D7	D6	D5	D4	D3	D2	D1	D0
1	EN	ALERT	R0	F1	F0	POL	TM	SD
	0	*(只读)	0	1	1	1	0	0
BYTE	D7	D6	D5	D4	D3	D2	D1	D0
2	EN	ALERT	R0	F1	F0	POL	TM	SD
	0	*(只读)	0	1	1	1	0	0

3.3.4 THIGH 与 TLOW 寄存器

这两个寄存器是设置测试温度值的上下限值, 均为 6 个 Byte。前 3 个 Byte 为通道 1 的上限或下限, 后 3 个 byte 是通道 2 的上限或下限, 寄存器存储格式如表 3.13 和表 3.14 所示。每次温度的测试值, 均与 T_{HIGH} 、 T_{LOW} 比较, 进而 ALERT 输出相应的信号。外部设备可以根据此信号作报警处理, 初始状态时, T_{HIGH} 默认为 7F FF FF H, T_{LOW} 默认为 FF FF FF H。

表 3.13 T_{HIGH} 温度寄存器定义

通道 1 T_{HIGH} 寄存器	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	1	T23	T22	T21	T20	T19	T18	T17	T16
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	2	T15	T14	T13	T12	T11	T10	T9	T8
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	3	T7	T6	T5	T4	T3	T2	T1	T0
通道 2 T_{HIGH} 寄存器	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	4	T23	T22	T21	T20	T19	T18	T17	T16
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	5	T15	T14	T13	T12	T11	T10	T9	T8
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	6	T7	T6	T5	T4	T3	T2	T1	T0

表 3.14 T_{LOW} 温度寄存器定义

通道 1 T_{LOW} 寄存器	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	1	T23	T22	T21	T20	T19	T18	T17	T16
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	2	T15	T14	T13	T12	T11	T10	T9	T8
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	3	T7	T6	T5	T4	T3	T2	T1	T0
通道 2 T_{LOW} 寄存器	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	4	T23	T22	T21	T20	T19	T18	T17	T16
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	5	T15	T14	T13	T12	T11	T10	T9	T8
	Byte	D7	D6	D5	D4	D3	D2	D1	D0
	6	T7	T6	T5	T4	T3	T2	T1	T0

外部主机设备直接连续设置 6 个字节 T_{HIGH} 温度寄存器，和 6 个字节的 T_{LOW} 温度寄存器。

3.4 断线检测功能

ZAM6222B 具备断线检测功能，可以判断热电阻 PT1000 与 ZAM6222B 的连接是否断开。ZAM6222B 与 PT1000 采用三线制连接方式，当任意通道的 RA、RB、RC 中的任意一根或多根线断开时，则模块对应断线通道输出的温度值为 -1000°C ，此时可以判断模块与 PT1000 之间处于断线状态。图 3.10 所示为其中两种断线状态，两个通道均断线状态下模块温度寄存器输出结果如表 3.15 所示。

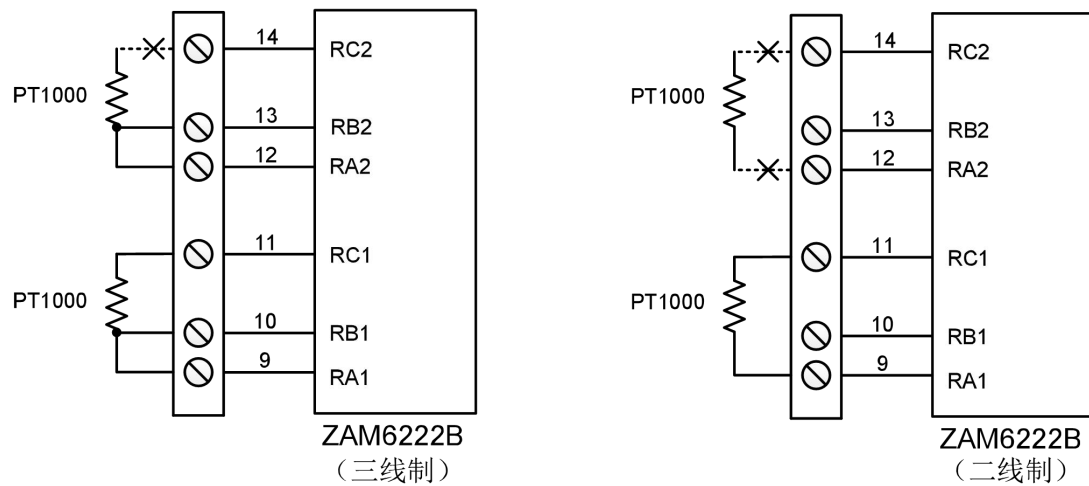


图 3.10 ZAM6222B 断线情况图示

表 3.15 断线下温度寄存器输出

通道	温度值 (°C)	数字输出	
		二进制	16 进制
通道 1 温度寄存器	-1000	1000 0011 0000 0000 0000 0000	83 00 00H
通道 2 温度寄存器	-1000	1000 0011 0000 0000 0000 0000	83 00 00H

4. 产品使用注意事项

- (1) 不支持热插拔。
- (2) ZAM6222B 产品的电气参数请参考《ZAM6222B 数据手册》。

5. 免责声明

本着为用户提供更好服务的原则，广州致远电子股份有限公司（下称“致远电子”）在本手册中将尽可能地向用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性，致远电子不能完全保证该文档在任何时段的时效性与适用性。致远电子有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，请尊敬的用户定时访问致远电子官方网站或者与致远电子工作人员联系。感谢您的包容与支持！

诚信共赢，持续学习，客户为先，专业专注，只做第一

广州致远电子股份有限公司

更多详情请访问
www.zlg.cn

欢迎拨打全国服务热线
400-888-4005

