Application Note

RS200 软件编程指南

智能雨刮模块（安防类）

AN02041301 1.0.01 Date:2020/12/25

类别	内容
关键词	软件编程
摘要	介绍RS200软件编程

修订历史

版本	日期	原因
V1.0	2020/01/10	创建文档
V1.1	2020/12/25	更新文档

目 录

1. 简介

RS200 模块是广州致远电子股份有限公司推出的针对安防摄像头的智能雨刮模块。RS200 模块安装在摄像头玻璃罩内表面，当雨滴落在玻璃罩外表面时，RS200 模块能检测雨量大小，通过串口将雨量状况发送给主机，主机控制雨刮，刮掉玻璃罩外表面的雨滴，保证摄像头视野的持续清晰。

1.1 产品特性

- 基于光学系统，能准确检测玻璃表面的雨量状态，并通过 UART 发送给主机；
- 使用 HALIOS®-SD 专利技术，具有超强抗太阳光干扰能力；
- 模块安装在玻璃罩内部，不与外界自然环境接触，减少环境干扰，增长使用寿命；
- 超小体积，直径仅为 12.5±0.15mm，轻松适配各种摄像头；
- 供电范围为 3.1~3.5V，正常工作模式功耗约为 31ma；
- UART 通信波特率默认为 115200；
- 使用带 CRC-8 校验的通信协议，提高通信抗干扰能力；
- 可通过 UART 配置部分可调参数；
- 模块错误自检测，主动通过 UART 向主机发送自身错误状态；
- 光学系统自校准；
- 自然环境光检测，为摄像头提供更多辅助参数；
- 板载温度测量，优异的温度线性关系保证测量的准确度；
- 支持光学睡眠模式，延长光学器件使用寿命。

1.2 产品图片

RS200 雨量检测模块实物图如图 1.1 所示。

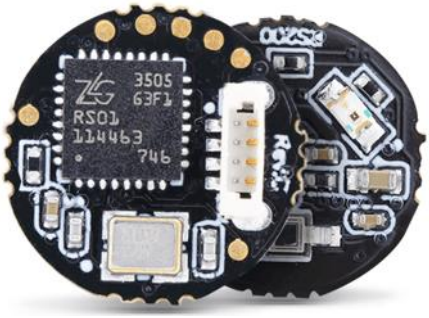

图 1.1 RS200 实物图

2. RS200 电路设计

RS200 采用 0.8mm 的线到板板端插座将供电和通信管脚引出，方便用户使用，如图 2.1 所示。其通信管脚为二线串口，能直接与主机进行通信，发送雨量状态数据，其管脚定义如图 2.1 所示。

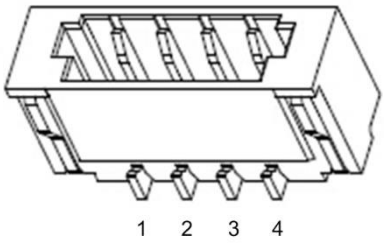

图 2.1　线到板板端插座

表 2.1　接口定义

引脚序号	信号名称	功能	备注
1	VCC	3.3V 电源	——
2	UART_RX	UART 接收	与主机 UART 的 TX 引脚相连
3	UART_TX	UART 发送	与主机 UART 的 RX 引脚相连
4	GND	电源地	——

RS200 应用电路图如图 2.2 所示，RS200 的供电电压为 3.3V，通过 UART 接口与主机连接。

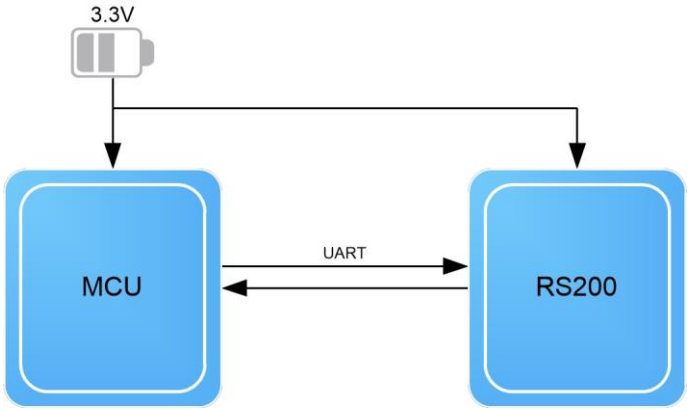

图 2.2　RS200 应用电路图

3. 通信协议

RS200 通过 UART 与主机之间进行交互，串口配置参数如表 3.1。所有功能的实现依赖 UART 收发指令。

表 3.1　串口配置参数

波特率	115200
数据位	8bit
停止位	1bit
校验位	NONE

每一帧数据的格式见表 3.2。帧头固定为 0x3A，见表 3.3；帧标识和帧数据共同代表了一帧数据的意义，见表 3.4。RS200 和主机在发送或接收一帧数据的时候都要通过 CRC-8 校验，帧校验为帧标识和帧数据的 CRC-8 校验值，见表 3.5。

表 3.2　串口数据帧结构

帧头	帧标识	帧数据	帧校验 CRC-8
1Byte	1Byte	2Byte（16 进制，低位在前）	1Byte

表 3.3　帧头定义

帧头[7:0]	帧头定义
0x3A	每一帧 5 Byte 数据固定以 0x3A（ASCII 对应的字符为冒号“:”）开头

表 3.4　帧标识与帧数据定义

帧标识[7]：数据读写属性	帧标识[6:0]：数据编号	帧数据[15:0]：数据值	说明
0(读) 1(写)	0（固件版本）	X	帧数据[15:8]主版本号， 帧数据[7:0]副本号
	1（雨量状态）	0	无雨
		1	小雨
		2	中雨
		3	大雨
	2（系统状态）	0	系统正常
		1	RS200 内通信错误
		2	LEDA 损坏
		3	LEDB 损坏
		4	光学系统校准不理想
		5	参数配置失败
		6	串口通信异常（串口校验错误）
		7	低压警告（低压阈值 2.8V）
	3（光学系统）	0	执行光学系统校准
		1	发送光学系统校准值
	4（进入实时雨量）	0	退出实时雨量模式
		1	进入实时雨量模式

续上表

帧标识[7]：数据读写属性	帧标识[6:0]：数据编号	帧数据[15:0]：数据值	说明
0(读) 1(写)	5（雨量状态输出频率或使能）	0~9	雨量状态输出频率，默认值为 1，代表 50ms；可修改。每增加或减少 1 代表增加或者减少 50ms（当为 0 时禁用输出）
	6（无雨与小雨的阈值 V1）	0~65535	无雨与小雨的阈值 V1
	7（小雨与中雨的阈值 V2）	0~65535	小雨与中雨的阈值 V2
	8（中雨与大雨的阈值 V3）	0~65535	中雨与大雨的阈值 V3
	9（无雨与小雨的阈值 S1）	0~65535	无雨与小雨的阈值 S1
	10（小雨与中雨的阈值 S2）	0~65535	小雨与中雨的阈值 S2
	11（中雨与大雨的阈值 S3）	0~65535	中雨与大雨的阈值 S3
	12（10 次中判定为大雨的次数阈值 N1）	1~10	10 次中判定为大雨的次数阈值 N1
	13（10 次中判定为中雨的次数阈值 N2）	1~10	10 次中判定为中雨的次数阈值 N2
	14（10 次中判定为小雨的次数阈值 N3）	1~10	10 次中判定为小雨的次数阈值 N3
	15（环境光测量模式）	0	RS200 退出环境光测量模式
		1	RS200 进入环境光测量模式
	16（主机读取 RS200 温度）	0	主机读取一次 RS200 模块温度
	17（RS200 光学睡眠模式）	0	RS200 退出睡眠，进入雨量测试
		1	RS200 进入睡眠模式

表 3.5　帧校验定义

帧校验[7:0]	多项式（HEX）	数据反转	初始值（HEX）	异或值（HEX）
CRC-8	x8+x5+x4+1（0x31）	MSB First	0xFF	0x00

4. 软件编程指南

RS200 目前已接入 ZLG 的 AMetal 平台，提供了相应的驱动，可以直接使用相应的 API 完成雨量检测模块的配置及雨量状态的获取，用户无需关心底层的通信协议，即可快速使用 RS200 检测雨量状态。相关源码已开源，本节将详细介绍提供的 RS200 功能函数。

4.1 芯片初始化

AMetal 已经支持的 RS200 雨量检测模块，这里以 ZLG116 作为主控驱动 RS200 为例，使用其它各功能函数前必须先完成初始化，初始化函数的原型（am_rs200.h）为：

```
am_rs200_handle_t am_rs200_init (
        am_rs200_dev_t          *p_dev,
        const am_rs200_devinfo_t  *p_devinfo,
        am_uart_handle_t          uart_handle);
```

其中，p_dev 为指向 am_rs200_dev_t 类型实例的指针，p_devinfo 为指向 am_rs200_devinfo_t 类型实例信息的指针，uart_handle 为 UART 句柄。

1. 实例

定义 am_rs200_dev_t 类型（am_rs200.h）实例如下：

```
am_rs200_dev_t g_rs200_dev;                                    // 定义一个 RS200 实例
```

其中，g_rs200_dev 为用户自定义的实例，其地址作为 p_dev 的实参传递。

2. 实例信息

实例信息主要描述了与 RS200 通信时，与 UART 通信、缓冲区等相关的信息。其类型 am_rs200_devinfo_t 的定义（am_rs200.h）如下：

```
typedef struct am_rs200_devinfo {
    uint8_t     *p_uart_rxbuf;          // 用于串口接收的缓冲区，建议大小在 32 以上
    uint8_t     *p_uart_txbuf;          // 用于串口发送的缓冲区，建议大小在 32 以上
    uint16_t     rxbuf_size;            // 用于串口接收的缓冲区大小
    uint16_t     txbuf_size;            // 用于串口发送的缓冲区大小
    uint32_t     timeout_ms;            // 用于设置串口接收超时
} am_rs200_devinfo_t;
```

为了提高数据处理的效率和确保接收数据不会因为正在处理事务而丢失。UART 发送和接收都需要一个缓冲区，用于缓存数据，缓冲区的实际大小由用户根据实际情况确定，建议在 32 字节以上，一般设置为 128 字节。p_uart_rxbuf 和 rxbuf_size 描述了接收缓冲区的首地址和大小，p_uart_txbuf 和 txbuf_size 描述了发送缓冲区的首地址和大小。如分别定义大小为 128 字节的缓冲区供发送和接收使用，定义如下：

```
uint8_t g_rs200_uart_txbuf[128];
uint8_t g_rs200_uart_rxbuf[128];
```

其中，g_rs200_uart_txbuf[128]为用户自定义的数组空间，供发送使用，充当发送缓冲区，其地址（数组名 g_rs200_uart_txbuf 或首元素地址&g_rs200_uart_txbuf[0]）作为实例信息中 p_uart_txbuf 成员的值，数组大小（这里为 128）作为实例信息中 txbuf_size 成员的值。同理，g_rs200_uart_rxbuf[128]充当接收缓冲区，其地址作为实例信息中 p_uart_rxbuf 成员的值，数组大小作为实例信息中 rxbuf_size 成员的值。

为了保证通信出错时，避免 UART 接收数据不完整而出现死等，可以设置串口接收超时时间。timeout_ms 描述串口接收超时时间，单位为 ms，如超过超时时间仍未完整接收一帧数据，为通信错误。

基于以上信息，实例信息可以定义如下：

```
#define    RS200_RX_BUF_SIZE    16                          // 接收环形缓冲区大小，应该为 2^n
#define    RS200_TX_BUF_SIZE    16                          // 发送环形缓冲区大小，应该为 2^n
am_local uint8_t __rs200_rxbuf[RS200_RX_BUF_SIZE];      // UART 接收环形缓冲区
am_local uint8_t __rs200_txbuf[RS200_TX_BUF_SIZE];      // UART 发送环形缓冲区
am_const am_local struct am_rs200_devinfo __g_rs200_devinfo = {
    __rs200_rxbuf,
    __rs200_txbuf,
    RS200_RX_BUF_SIZE,
    RS200_TX_BUF_SIZE,
    200
};
```

其中，__g_rs200_devinfo 为用户自定义的实例信息，其地址作为 p_devinfo 的实参传递。

3. UART 句柄 uart_handle

若使用 ZLG116 的 UART2 与 RS200 通信，则 UART 句柄可以通过 ZLG116 的 UART2 实例初始化函数 am_zlg116_uart2_inst_init()获得。即：

```
am_uart_handle_t uart_handle =am_zlg116_uart2_inst_init();
```

获得的 UART 句柄即可直接作为 uart_handle 的实参传递。

4. 实例句柄

RS200 初始化函数 am_rs200_init()的返回值即为 RS200 实例的句柄。该句柄将作为其它功能接口函数的 handle 参数的实参。

其类型 am_rs200_handle_t（am_rs200.h）定义如下：

```
typedef am_rs200_dev_t *am_rs200_handle_t;
```

若返回值为 NULL，说明初始化失败；若返回值不为 NULL，说明返回了一个有效的 handle。基于模块化编程思想，将初始化相关的实例、实例信息等的定义存放到 RS200 的配置文件(am_hwconf_rs200.c)中，通过头文件（am_hwconf_rs200.h）引出实例初始化函数接口，源文件和头文件的程序范例分别详见程序清单 4.1 和程序清 4.2。

程序清单 4.1　RS200 实例初始化函数实现

```
1    #include "ametal.h"
2    #include "am_rs200.h"
3    #include "am_zlg116.h"
4    #include "am_zlg116_inst_init.h"
5
6    #define    RS200_RX_BUF_SIZE    16                          // 接收环形缓冲区大小，应该为 2^n
7    #define    RS200_TX_BUF_SIZE    16                          // 发送环形缓冲区大小，应该为 2^n
8
9    am_local uint8_t __rs200_rxbuf[RS200_RX_BUF_SIZE];      //UART 接收环形缓冲区
10   am_local uint8_t __rs200_txbuf[RS200_TX_BUF_SIZE];      //UART 发送环形缓冲区
```

```
11
12  am_const am_local struct am_rs200_devinfo __g_rs200_devinfo = {
13          __rs200_rxbuf,
14          __rs200_txbuf,
15          RS200_RX_BUF_SIZE,
16          RS200_TX_BUF_SIZE,
17          200
18  };
19  am_local struct am_rs200_dev __g_rs200_dev;                    //RS200 设备结构体定义
20  am_rs200_handle_t am_rs200_inst_init (void)                    //RS200 设备实例初始化
21  {
22          return am_rs200_init(  &__g_rs200_dev,
23                                 &__g_rs200_devinfo,
24                                 am_zlg116_uart2_inst_init());
25  }
```

程序清 4.2　RS200 实例初始化函数声明

```
1   #include "ametal.h"
2   #include "am_rs200.h"
3   am_rs200_handle_t am_rs200_inst_init (void);
```

后续只需要使用无参数的实例初始化函数即可获取到 RS200 的实例句柄。即：

```
am_rs200_handle_t rs200_handle = am_rs200_inst_init();
```

4.2　功能函数接口

RS200 控制接口用于控制 RS200，完成所有与 RS200 相关的控制，如设置雨量状态输出频率、设置雨量状态阈值等。其函数原型（am_rs200.h）为：

```
int am_rs200_ioctl(am_rs200_handle_t handle, int cmd, void *p_arg);
```

其中，cmd 用于指定控制命令，不同的命令对应不同的操作。p_arg 为与命令对应的参数。当命令不同时，对应 p_arg 的类型可能不同，因此，这里 p_arg 使用的类型为 void *，其实际类型应该与命令指定的类型一致。常见命令（am_rs200.h）及命令对应的 p_arg 类型详见表 4.1。

表 4.1　支持的命令及命令对应的 p_arg 类型

序号	操作	命令	p_arg 类型
1	获取 RS200 固件版本	AM_RS200_VERSION_GET	uint16_t*
2	读取雨量状态	AM_RS200_RAIN_STA_GET	uint16_t*
3	读取系统状态	AM_RS200_SYS_STA_GET	uint16_t*
4	读取光学系统校准值	AM_RS200_OPTICAL_SYS_VALUE_GET	uint16_t*
5	设置雨量状态输出频率	AM_RS200_RAIN_STA_OUT_FRE_SET	uint16_t
6	读取雨量状态输出频率	AM_RS200_RAIN_STA_OUT_FRE_GET	uint16_t*
7	设置无雨与小雨的阈值 V1	AM_RS200_THRESHOLD_V1_SET	uint16_t
8	读取无雨与小雨的阈值 V1	AM_RS200_THRESHOLD_V1_GET	uint16_t*
9	设置小雨与中雨的阈值 V2	AM_RS200_THRESHOLD_V2_SET	uint16_t

续上表

序号	操作	命令	p_arg 类型
10	读取小雨与中雨的阈值 V2	AM_RS200_THRESHOLD_V2_GET	uint16_t*
11	设置中雨与大雨的阈值 V3	AM_RS200_THRESHOLD_V3_SET	uint16_t
12	读取中雨与大雨的阈值 V3	AM_RS200_THRESHOLD_V3_GET	uint16_t*
13	设置无雨与小雨的阈值 S1	AM_RS200_THRESHOLD_S1_SET	uint16_t
14	读取无雨与小雨的阈值 S1	AM_RS200_THRESHOLD_S1_GET	uint16_t*
15	设置小雨与中雨的阈值 S2	AM_RS200_THRESHOLD_S2_SET	uint16_t
16	读取小雨与中雨的阈值 S2	AM_RS200_THRESHOLD_S2_GET	uint16_t*
17	设置中雨与大雨的阈值 S3	AM_RS200_THRESHOLD_S3_SET	uint16_t
18	读取中雨与大雨的阈值 S3	AM_RS200_THRESHOLD_S3_GET	uint16_t*
19	设置 10 次判定为大雨的阈值 N1	AM_RS200_THRESHOLD_N1_SET	uint16_t
20	读取 10 次判定为大雨的阈值 N1	AM_RS200_THRESHOLD_N1_GET	uint16_t*
21	设置 10 次判定为中雨的阈值 N2	AM_RS200_THRESHOLD_N2_SET	uint16_t
22	读取 10 次判定为中雨的阈值 N2	AM_RS200_THRESHOLD_N2_GET	uint16_t*
23	设置 10 次判定为大雨的阈值 N3	AM_RS200_THRESHOLD_N3_SET	uint16_t
24	读取 10 次判定为大雨的阈值 N3	AM_RS200_THRESHOLD_N3_GET	uint16_t*
25	设置环境光测模式	AM_RS200_OPTICAL_TEST_MODE_SET	uint16_t
26	读取环境光值	AM_RS200_OPTICAL_TEST_VALUE_GET	uint16_t*
27	读取芯片温度	AM_RS200_TEMP_GET	uint16_t*
28	设置光学睡眠状态	AM_RS200_LIGHT_SLEEP_MODE_SET	uint16_t*

在绝大部分应用场合下，默认值即可正常工作，因此，若无特殊需求，可以直接使用读雨量状态接口完成雨量状态的检测。一般地，也尽可能使用少量命令完成一些特殊应用需求。下面详细介绍各个命令的使用方法。

4.2.1 获取固件版本

RS200 版本号使用两个字节表示，高字节表示主版本号，低字节表示副版本号。获取 RS200 固件版本的范例程序详见程序清单 4.3。

程序清单 4.3 获取固件版本

```
1  uint16_t version;
2  am_rs200_ioctl (handle, AM_RS200_VERSION_GET, (void*)version);
```

4.2.2 读取雨量状态

RS200 根据设置的雨量阈值将雨量状态的分为无雨、小雨、中雨、大雨四种状态。读取雨量状态的范例程序详见程序清单 4.4。

程序清单 4.4 获取雨量状态

```
1  uint16_t status;
2  am_rs200_ioctl (handle, AM_RS200_RAIN_STA_GET, (void*)status);
```

4.2.3 读取系统状态

系统状态分为系统正常、内部通信错误、LED 损坏、参数配置失败、串口通信错误、低压警告。通过获取系统状态，从而判别获取的数据是否正常。读取系统状态的范例程序详

见程序清单 4.5。

程序清单 4.5　获取系统状态

```
1    uint16_t status;
2    am_rs200_ioctl (handle, AM_RS200_SYS_STA_GET, (void*)status);
```

4.2.4 设置/读取雨量状态输出频率

雨量状态输出频率，默认值为 1，代表 50ms；可修改。每增加或减少 1 代表增加或者减少 50ms，当为 0 时禁用输出。

设置雨量状态输出频率的范例程序详见程序清单 4.6。

程序清单 4.6　设置雨量状态输出频率

```
1    uint16_t freq = 2;
2    am_rs200_ioctl (handle, AM_RS200_RAIN_STA_OUT_FRE_SET, (void*)freq);
```

获取雨量状态输出频率的范例程序详见程序清单 4.7。

程序清单 4.7　获取雨量状态输出频率

```
1    uint16_t freq;
2    am_rs200_ioctl (handle, AM_RS200_RAIN_STA_OUT_FRE_GET, (void*) freq);
```

4.2.5 设置/读取雨量状态阈值

雨量状态阈值分为 V 阈值、S 阈值和 N 阈值。V 值代表玻璃表面水滴的变化情况，S 值代表玻璃表面水滴的静态情况。没有理想的绝对静止的水滴情况，RS200 使用两个维度识别玻璃表面水滴的变化趋势。N 阈值代表 10 次判定有效的阈值。

设置雨量状态阈值的范例程序详见程序清单 4.8。

程序清单 4.8　设置雨量状态阈值

```
1    uint16_t threshold = 3;
2    am_rs200_ioctl (handle, AM_RS200_THRESHOLD_N1_SET, (void*) threshold);
```

读取雨量状态阈值的范例程序详见程序清单 4.9。

程序清单 4.9　读取雨量状态阈值

```
1    uint16_t threshold;
2    am_rs200_ioctl (handle, AM_RS200_THRESHOLD_N1_GET, (void*) threshold);
```

4.2.6 设置环境光检测模式

RS200 支持环境光（白光）检测功能，该功能与雨量测试资源复用，需要使用指令控制 RS200 进入环境光检测功能。RS200 进入环境光检测功能后，会按固定频率输出环境光值。环境光反馈值范围为（十进制）0~1024。光强越强，反馈值越低；反之反馈值越高。

设置环境光检测模式范例程序详见程序清单 4.10。

程序清单 4.10　设置环境光检测模式

```
1    uint16_t mode = 1;
2    am_rs200_ioctl (handle, AM_RS200_OPTICAL_TEST_MODE_SET, (void*) mode);
```

4.2.7 读取环境光值

RS200 进入环境光检测功能后，会按固定频率输出环境光值。通过调用功能接口读取环境光值。读取环境光值范例程序详见程序清单 4.11。

程序清单 4.11 读取环境光值

```
1  uint16_t value;
2  am_rs200_ioctl (handle, AM_RS200_OPTICAL_TEST_VALUE_GET, (void*) value);
```

4.2.8 读取芯片温度

RS200 支持环境温度检测功能，该功能使用片内集成温度传感器实现。测量数据从环境温度-40℃到 85℃，步进 5℃获得，线性度良好，详见图 4.1。计算公式中 y 代表 RS200 反馈的温度值（RS200 反馈为 16 进制，计算公式为 10 进制）；x 代表环境温度。

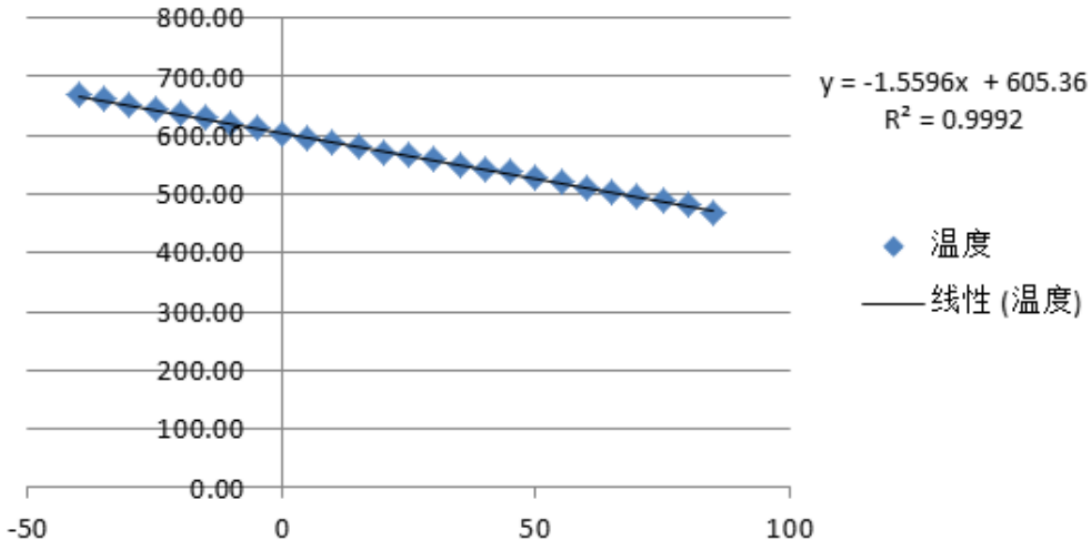

图 4.1 RS200 温度反馈值

读取芯片温度范例程序详见程序清单 4.12。

程序清单 4.12 读取芯片温度

```
1  uint16_t temp;
2  am_rs200_ioctl (handle, AM_RS200_TEMP_GET, (void*) temp);
```

4.2.9 设置光学睡眠状态

RS200 支持光学睡眠模式，延长光学器件使用寿命。进入光学睡眠状态的范例程序详见程序清单 4.13。

程序清单 4.13 设置光学睡眠状态

```
1  uint16_t mode = 1;
2  am_rs200_ioctl (handle, AM_RS200_LIGHT_SLEEP_MODE_SET, (void*) mode);
```

4.2.10 雨量状态获取和数据接收接口

雨量状态获取通过指令查询的方式实现了雨量状态的获取。数据接收通过从环形缓冲区

获取接收的数据。接口函数详见表 4.2。

表 4.2　雨量状态获取和数据接收接口函数

函数原型	功能简介
int am_rs200_rain_sta_get (am_rs200_handle_t handle, uint16_t *p_data);	获取雨量状态
int am_rs200_recv (am_rs200_handle_t handle, uint8_t *p_buf, uint32_t len);	接收数据

查询雨量状态函数原型为：

```
int am_rs200_rain_sta_get (am_rs200_handle_t handle, uint16_t *p_data);
```

其中，handle 为 RS200 初始化时获取的 handle，p_data 为指向存放雨量状态的地址。若返回值为 AM_OK，表示获取雨量状态成功，返回其他表示失败。获取雨量状态的范例程序详见程序清单 4.14。

程序清单 4.14　获取雨量状态

```
1    uint16_t data;
2    am_rs200_rain_sta_get (handle, &data);
```

4.2.11　接收数据

当雨量变化时，达到雨量状态阈值，RS200 会发送雨量状态数据，通过环形缓冲区接收数据可以实时了解雨量状态的变化，而不需要不断的查询，提高雨量检测的效率。接收数据的范例程序详见程序清单 4.15。

程序清单 4.15　接收数据

```
1    uint8_t buff[16];
2    am_rs200_recv (handle, buff, 5);
```

4.2.12　应用示例

RS200 初始化后，设置雨量状态输出频率，接收数据检测雨量状态的变化。范例程序详见程序清单 4.16。

程序清单 4.16　雨量状态检测范例程序

```
1    #include "ametal.h"
2    #include "am_rs200.h"
3    #include "am_hwconf_rs200.h"
4
5    int am_main (void)
6    {
7        uint8_t data[5];
8        am_rs200_handle_t rs200_handle =am_rs200_inst_init();
9        /* 设置雨量状态输出频率 */
```

```
10          am_rs200_ioctl(rs200_handle, AM_RS200_RAIN_STA_OUT_FRE_SET, (void*)2);
11
12          while(1) {
13                  if (am_rs200_recv (handle, data, 5) == 5) {
14                          AM_DBG_INFO("rs200 data: 0x%x 0x%x 0x%x 0x%x 0x%x\r\n\r\n",
15                          data[0], data[1],data[2], data[3], data[4]);
16                  }
17          }
18          return 0;
19  }
```

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为用户呈现详实、准确的产品信息。但介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