**通信协议**

目录

[一．概述 2](#_Toc4179)

[二．通信接口硬件特性 2](#_Toc18990)

[三． MODBU RTU通信协议详述 2](#_Toc14936)

[3.1命令格式 2](#_Toc3482)

[四． 举例 4](#_Toc29121)

# **一．概述**

通信协议采用标准MODBUS RTU协议,本协议规定了应用系统中辅控系统与主机模块之间在应用层的通信协议。

# **二．通信接口硬件特性**

接口类型：异步串行RS485通讯口。

通信波特率为：9600bps。

数据传输格式：N,8,1。

说明：主机与RS485总线设备的通信时间间隔大于等于500ms，推荐值1s。

# **MODBU RTU通信协议详述**

## 3.1命令格式

（1）主机发送命令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址 | 功能码 | 起始地址  高位 | 起始地址  低位 | 数据个数  高位 | 数据个数  低位 | CRC16  校验 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 2字节 |

（2）终端模块返回信息

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址 | 功能码 | 字节长度 | 数据1 | 数据2 | … | CRC16校验 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 |  | 2字节 |

（3）功能码描述

|  |  |
| --- | --- |
| 功能码 | 功能描述 |
| 0x02 | 读输入门柜状态量 |
| 0x05 | 写入门柜开启指令 |

（4）寄存器地址分配表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地址 | 功能描述 | 读写 | 数据类型 | 初始值 |
| 0x0000 | 设备ID | read | Byte | 0x00 |
| 0x0001 | 门柜开关状态量 | read | Byte | 0x00 |
| 0x0002 | 门柜异常状态量 | read | Byte | 0x00 |
| 0x0003 | 门柜前门开启指令 | write | Byte | 0x00 |
| 0x0004 | 门柜后门开启指令 | write | Byte | 0x00 |

1. 寄存器描述

0x0000寄存器

|  |  |  |
| --- | --- | --- |
| Bit | 读写 | 描述 |
| 0-7 | read | 设备ID |

0x0001寄存器

|  |  |  |
| --- | --- | --- |
| Bit | 读写 | 描述 |
| 2-7 | read | 无效 |
| 1 | read | 后门状态：开为1；关为0 |
| 0 | read | 前门状态：开为1；关为0 |

0x0002寄存器

|  |  |  |
| --- | --- | --- |
| Bit | 读写 | 描述 |
| 2-7 | read | 无效 |
| 1 | read | 后门异常：异常为1；正常为0 |
| 0 | read | 前门异常：异常为1；正常为0 |

0x0003寄存器

|  |  |  |
| --- | --- | --- |
| Bit | 读写 | 描述 |
| 1-15 | write/read | 无效 |
| 0 | write/read | 前门开启：开启为1；关闭为0 |

0x0004寄存器

|  |  |  |
| --- | --- | --- |
| Bit | 读写 | 描述 |
| 1-15 | write/read | 无效 |
| 0 | write/read | 后门开启：开启为1；关闭为0 |

# **举例**

4.1读取设备ID – 0x02功能

* 主机发送命令：

[IO模块地址] [0x02] [0x00] [0x00] [0x00] [0x01] [CRC低位] [CRC高位]

* IO模块响应：

[IO模块地址] [0x02] [0x01] [数据字节1] [数据字节2] [CRC低位] [CRC高位]

说明：每个设备的ID占用1位。

返回错误：

[IO模块地址] [0x82] [0x00] [CRC低位] [CRC高位]

4.2 读取门柜开关状态量 – 0x02功能

* 主机发送命令：

[IO模块地址] [0x02] [0x00] [0x01] [0x00] [0x01] [CRC低位] [CRC高位]

* IO模块响应：

[IO模块地址] [0x02] [0x02] [数据字节1] [数据字节2] [CRC低位] [CRC高位]

说明：每个门柜开关状态量占用1位。

返回错误：

[IO模块地址] [0x82] [0x00] [CRC低位] [CRC高位]

4.3 读取门柜异常状态量– 0x02功能

* 主机发送命令：

[IO模块地址] [0x02] [0x00] [0x02] [0x00] [0x01] [CRC低位] [CRC高位]

* IO模块响应：

[IO模块地址] [0x02] [0x02] [数据字节1] [数据字节2] [CRC低位] [CRC高位]

说明：每个门柜异常状态量占用1位。

返回错误：

[IO模块地址] [0x82] [0x00] [CRC低位] [CRC高位]

4.4 读取设备0x0000 – 0x0002寄存器信息– 0x02功能

* 主机发送命令：

[IO模块地址] [0x02] [0x00] [0x00] [0x00] [0x03] [CRC低位] [CRC高位]

* IO模块响应：

[IO模块地址] [0x02] [0x06] [数据字节1] [数据字节2] … [数据字节6] [CRC低位]

[CRC高位]

说明：每个门柜开关状态量占用1位,每个门柜开关状态量占用1位,每个门柜异常状态量占用1位。

返回错误：

[IO模块地址] [0x82] [0x00] [CRC低位] [CRC高位]

4.5 写门柜开启指令 – 0x05功能

* 主机发送命令：

[IO模块地址] [0x05] [0x00] [0x03] [0x00] [0x01] [CRC低位] [CRC高位]

* IO模块响应：

原码返回。

说明：每个门柜开关指令占用1位。

注：本协议CRC校验统一采用CRC-16/MODBUS（x16+x15+x2+1）。

C语言代码：

unsigned int CRC16 ( unsigned char \*arr\_buff, unsigned int len)  
{  
  unsigned int crc=0xFFFF;  
  unsigned char i, j, Data;  
  for ( j=0; j<len;j++)  
 {  
    crc=crc ^\*arr\_buff++;  
    for ( i=0; i<8; i++)  
  {  
       if( ( crc&0x0001) >0)  
       {  
           crc=crc>>1;  
           crc=crc^ 0xa001;  
        }  
      else  
          crc=crc>>1;  }  
 }  
 return crc;  
}