**讯音科技Modbus协议定义**

# 简介

讯音公司Modbus协议支持标准Modbus-RTU方式，波特率115200，8bit数据，1个起始位和一个停止位，没有奇偶校验。收发包中数据排列以及对齐方式均可参照Modbus标准协议。

Modbus标准协议文档和实用工具如下：

|  |  |
| --- | --- |
| 描述 | 连接 |
| 一般介绍 | <http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf> |
| Modbus帧解析器 | <http://modbus.rapidscada.net/> |
| Modbus CRC 生成器 | <https://www.lammertbies.nl/comm/info/crc-calculation.html> |

# Modbus地址

每台设备地址缺省地址为0x04，可以使用讯音公司的专用工具修改为0~15之间的任何值。

# Modbus功能码

|  |  |
| --- | --- |
| 功能码 | 描述 |
| 0x03 | 读寄存器 |
| 0x06 | 写单个寄存器 |
| 0x10 | 写多个寄存器 |

讯音公司Modbus协议支持功能码0x03，0x06和0x10完成对device寄存器和数据的远程读写。查询包异常的时候device不做响应，主机则通过超时判断查询是否失败。异常包括通讯异常，功能码异常，CRC异常以及内容异常等等。如果查询包被device正常接收并执行，则会返回如下面章节定义的数据包。

# 读寄存器功能码0x03

读寄存器功能码0x03查询包和响应包定义如下例，例1中地址为0x80，查询数据为12345，对应16进制数为0x3039；其中回应包中byte数量为查询寄存器数量\*2。

**例1：**

查询

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **地址MSB** | **地址LSB** | **寄存器个数MSB** | **寄存器个数LSB** | **CRC LSB** | **CRC MSB** |
| 0x04 | 0x03 | 0x00 | 0x80 | 0x00 | 0x01 | 0x85 | 0xB7 |

回应

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **Bytes数量** | **内容MSB** | **内容LSB** | **CRC LSB** | **CRC MSB** |
| 0x04 | 0x03 | 0x02 | 0x30 | 0x39 | 0xA0 | 0x56 |

例2为读取连续2个寄存器的查询包和回应包，寄存器起始地址为0x80，查询数据分别为0x1234和0x5678;

**例2：**

查询

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **地址MSB** | **地址LSB** | **寄存器个数MSB** | **寄存器个数LSB** | **CRC LSB** | **CRC MSB** |
| 0x04 | 0x03 | 0x00 | 0x80 | 0x00 | 0x02 | 0xC5 | 0xB6 |

回应

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **Bytes数量** | **内容1 MSB** | **内容1 LSB** | **内容2 MSB** | **内容2 LSB** | **CRC LSB** | **CRC MSB** |
| 0x04 | 0x03 | 0x04 | 0x12 | 0x34 | 0x56 | 0x78 | 0xD4 | 0x07 |

# 写单个寄存器功能码0x06

写单个寄存器功能码0x06查询包和响应包定义如下例，例子中地址为0x80，查询数据为12345，对应16进制数为0x3039；

查询

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **地址MSB** | **地址LSB** | **内容MSB** | **内容LSB** | **CRC LSB** | **CRC MSB** |
| 0x04 | 0x06 | 0x00 | 0x80 | 0x30 | 0x39 | 0x5C | 0x65 |

回应

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **地址MSB** | **地址LSB** | **内容MSB** | **内容LSB** | **CRC LSB** | **CRC MSB** |
| 0x04 | 0x06 | 0x00 | 0x80 | 0x30 | 0x39 | 0x5C | 0x65 |

# 写多个寄存器功能码0x10

写多个寄存器功能码0x10查询包和响应包定义如下例，例子中起始地址为0x80，连续写入数据为0x1234, 0x5678，注意字节数量为寄存器个数\*2；

查询

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **地址MSB** | **地址LSB** | **寄存器个数MSB** | **寄存器个数LSB** | **字节数量** |
| 0x04 | 0x06 | 0x00 | 0x80 | 0x00 | 0x02 | 0x04 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **寄存器1 MSB** | **寄存器1 LSB** | **寄存器2 MSB** | **寄存器2 LSB** | **CRC LSB** | **CRC MSB** |
| 0x12 | 0x34 | 0x56 | 0x78 | 0xF8 | 0x8E |

回应

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **地址** | **功能码** | **地址MSB** | **地址LSB** | **寄存器个数MSB** | **寄存器个数LSB** | **CRC LSB** | **CRC MSB** |
| 0x04 | 0x06 | 0x00 | 0x80 | 0x00 | 0x02 | 0x09 | 0xB6 |

# 数据格式

命令当中包含3种格式，分别是8位整数，16位整数，32位整数和32位浮点数。由于modbus传输数据至少是16位，因此，如果是8位数据，则按照如下格式发送或者接收，注意8位整数包括bool类型，bool类型不是1就是0。

|  |  |
| --- | --- |
| **寄存器1 MSB** | **寄存器1 LSB** |
| 0x0 | 8bit data |

同样，如果是16bit数据，则按照如下格式接收或者发送数据：

|  |  |
| --- | --- |
| **寄存器1 MSB** | **寄存器1 LSB** |
| Data MSB | Data LSB |

如果是32bit数据，则不管是整数，还是浮点数，均按照如下格式接收或这发送数据，注意格式并不是电脑中传统的32bit数据的Big Endian和Little Endian格式：

|  |  |  |  |
| --- | --- | --- | --- |
| **寄存器1 MSB** | **寄存器1 LSB** | **寄存器2 MSB** | **寄存器2 LSB** |
| 低16bit MSB | 低16bit LSB | 高16bit MSB | 高16bit LSB |

# CRC生成代码

uint16\_t **Checksum\_computeChecksum**(**const** uint8\_t \*buffer, uint16\_t length)

{

uint16\_t checksum = 0;

uint16\_t i;

uint16\_t crc = 0xFFFF;

**if** (length == 0) length = 1;

**while** (length--) {

crc ^= \*buffer;

**for** (i = 0; i<8; i++)

{

**if** (crc & 1) {

crc >>= 1;

crc ^= 0xA001;

} **else** {

crc >>= 1;

}

}

buffer++;

}

**return** crc;

}

# 设备命令

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **命令** | **地址** | **支持功能码** | **数据类型** | **描述** |
|  | GAP\_PLS\_ADC\_START | 0x80 | 0x03,0x06 | Uint16 |  |
|  | NUM\_PLS | 0x81 | 0x03,0x06 | Uint16 |  |
|  | GAP\_UPS\_DNS | 0x82 | 0x03,0x06 | Uint16 |  |
|  | GAP\_UPS\_UPS | 0x83 | 0x03,0x06 | Uint16 |  |
|  | GAIN\_CONTROL | 0x84 | 0x03,0x06 | Uint8 |  |
|  | METER\_CONSTANT | 0x85 | 0x03,0x10 | Float |  |
|  | XT2\_FREQ | 0x86 | 0x03,0x06 | Uint16 |  |
|  | ADC\_SAMP\_FREQ | 0x87 | 0x03,0x06 | Uint16 |  |
|  | SIG\_SAMP\_FREQ | 0x88 | 0x03,0x06 | Uint16 |  |
|  | ADC\_OVERSAMPLING | 0x89 | 0x03,0x06 | Uint8 |  |
|  | DELTA\_TOF\_OFFSET | 0x8A | 0x03,0x10 | Int32 |  |
|  | ABS\_TOF\_ADDITIONAL | 0x8B | 0x03,0x10 | Int32 |  |
|  | CAPTURE\_DURATION | 0x8C | 0x03,0x06 | Int16 |  |
|  | PARAM1 | 0x8D | 0x03,0x10 | Int32 |  |
| u | PARAM2 | 0x8E | 0x03,0x10 | Int32 |  |
|  | PARAM3 | 0x8F | 0x03,0x10 | Int32 |  |
|  | PARAM4 | 0x90 | 0x03,0x10 | Int32 |  |
|  | PARAM5 | 0x9E | 0x03,0x10 | Int32 |  |
|  | PARAM6 | 0x9F | 0x03,0x10 | Int32 |  |
|  | PARAM7 | 0x91 | 0x03,0x10 | Int32 |  |
|  | PARAM8 | 0x92 | 0x03,0x10 | Int32 |  |
|  | PARAM9 | 0x93 | 0x03,0x10 | Int32 |  |
|  | PARAM10 | 0x94 | 0x03,0x10 | Int32 |  |
|  | APP\_UPDATE\_REQ | 0x96 | 0x03,0x10 | bool |  |
|  | APP\_REQ\_CAPTURES1 | 0x98 | 0x03 | Int16\*N | N<=125 |
|  | APP\_VERSION | 0x99 | 0x03 | bool |  |
|  | F1\_F2 | 0x9B | 0x03,0x10 | Uint16\*2 |  |
|  | AUTO\_GAIN\_ENABLE | 0xE0 | 0x03,0x06 | bool |  |
|  | FlOW\_RANGE | 0xE1 | 0x03,0x10 | float\*10 |  |
|  | RUNNING\_AVG | 0xE2 | 0x03,0x06 | Uint8 | Filter\_type |
|  | SET\_DAC | 0xE3 | 0x03,0x10 | Uint8+float | Dac\_force+dac\_data |
|  | SET\_KF | 0xE4 | 0x03,0x10 | Float | K\_factor |
|  | SET\_KF1 | 0xE5 | 0x03,0x10 | Float\*19 | measureratio |
|  | SET\_KF2 | 0xE6 | 0x03,0x10 | Float\*19 | calipoint |
|  | SET\_SNSR\_TYP | 0xE7 |  |  |  |
|  | LNCAL\_EN | 0xE8 | 0x03,0x06 | Bool | Line\_cal |
|  | STA\_UP | 0xE9 | 0x06 | Bool | Sta\_up |
|  | RESET | 0xEA | 0x06 | Bool | reset |
|  | ZERO\_CAL | 0xEB | 0x06 | Float\*2 | Zero\_cal + zero\_cal\_value |
|  | AUTO\_GAIN\_UP | 0xEC | 0x3,0x6 | Int8 | Gain\_control |
|  | ADDR | 0xED | 0x3,0x6 | Int8 | Addr |
| 6.42 | APP\_REQ\_CAPTURES2 | 0xEF | 0x03 | Int16\*N | N<=125 |
| 6.43 | APP\_REQ\_CAPTURES3 | 0xF0 | 0x03 | Int16\*N | N<=125 |
| 6.44 | APP\_REQ\_CAPTURES4 | 0xF1 | 0x03 | Int16\*N | N<=125 |
| 6.45 | APP\_REQ\_DATA | 0x97 | 0x03 | Float\*3+int32+Float\*N | N defined by sample per second parameter. |

# GAP\_PLS\_ADC\_START (0x80)

设置pulse起始位置和ADC采样之间的时间间隔，单位是uS，该参数写入主板flash，掉电后自动保存。16bit无符号整数。数据范围3~9000；

# NUM\_PLS(0x81)

设置pulse个数。该参数写入主板flash，掉电后自动保存。16bit无符号整数。数据范围0~63；

# GAP\_UPS\_DNS(0x82)

设置单次采样中上行和下行采样超声波的时间间隔，单位us，掉电自动保存。16bit无符号整数。数据范围100~16000；

# GAP\_UPS\_UPS(0x83)

设置连续两次采样中超声波的时间间隔，单位是mS，掉电自动保存。16bit无符号整数。注意此参数目前没有用处，系统采样间隔为固定10mS。

# GAIN\_CONTROL(0x84)

波形增益，无单位，掉电自动保存，数据范围为17~63，对应信号放大分贝如下表所示。8bit无符号整数。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **数值** | **dB** | **数值** | **dB** | **数值** | **dB** |
| 17 | -6.5 | 33 | 6.8 | 49 | 19.6 |
| 18 | -5.5 | 34 | 7.7 | 50 | 20.5 |
| 19 | -4.6 | 35 | 8.7 | 51 | 21.5 |
| 20 | -4.1 | 36 | 9.0 | 52 | 22.0 |
| 21 | -3.3 | 37 | 9.8 | 53 | 22.8 |
| 22 | -23 | 38 | 10.7 | 54 | 23.6 |
| 23 | -1.4 | 39 | 11.7 | 55 | 24.6 |
| 24 | -0.8 | 40 | 12.2 | 56 | 25.0 |
| 25 | 0.1 | 41 | 13.0 | 57 | 25.8 |
| 26 | 1.0 | 42 | 13.9 | 58 | 26.7 |
| 27 | 1.9 | 43 | 14.9 | 59 | 27.7 |
| 28 | 2.6 | 44 | 15.5 | 60 | 28.1 |
| 29 | 3.5 | 45 | 16.3 | 61 | 28.9 |
| 30 | 4.4 | 46 | 17.2 | 62 | 29.8 |
| 31 | 5.2 | 47 | 18.2 | 63 | 30.8 |
| 32 | 6.0 | 48 | 18.8 |  |  |

# METER\_CONSTANT(0x85)

设备常数，根据管径、材料等参数计算，为32bit浮点数。

# XT2\_FREQ(0x86)

Xtal频率，单位Khz，数据范围4000~16000，16bit无符号整数。。

# ADC\_SAMP\_FREQ(0x87)

此参数暂时无用。

# SIG\_SAMP\_FREQ(0x88)

信号采样频率，单位kHz，数据范围1000~8000，16bit无符号整数。

# ADC\_OVERSAMPLING(0x89)

过采样倍数，10，20，40，80和160可选，无单位，8bit无符号整数。注意68Mhz < SIG\_SAMP\_FREQ\*ADC\_OVERSAMPLING <80Mhz.

# DELTA\_TOF\_OFFSET(0x8A)

DC offset，单位ps，32位无符号整数

# ABS\_TOF\_ADDITIONAL(0x8B)

发射pulse和capture数据间额外延时，单位ns，32位浮点数。

# CAPTURE\_DURATION(0x8C)

采样窗口时长，单位us，16bit无符号整数，数据范围4~400；采样点数：

num\_samples = capture­\_duration\*sig\_samp\_freq )/1000;此点数应不超过372.

# PARAM1(0x8D)

Abstof计算模式，0为lobe方式，1为希尔伯特变换方式。32bit整数。

# PARAM2(0x8E)

Ulp bias delay, 此参数定义发起采样过程至获取数据之间的时间间隔，保证获取数据是在PGA稳定之后。数据范围0~3，单位uS，32bit整数。

# PARAM3(0x8F)

Start PPG Count，数据范围0~10E6，单位是nS，32bit整数。

# PARAM4(0x90)

Turn On ADC Count, 数据范围0~10E6，单位是nS，32bit整数。

# PARAM5(0x9E)

Start PGA and INBias Count, 数据范围0~10E6，单位是nS，32bit整数。

# PARAM6(0x9F)

# PARAM7(0x91)

USS XTAL settling count，数据范围0~10000，32bit整数。

# PARAM8(0x92)

Envelop crossing Threshold, 数据范围1~100，代表threshold占有效信号的百分比。32bit整数。

# PARAM9(0x93)

Negative Ring Tracking，数据范围0~32768，32bit整数

# PARAM10(0x94)

Positive Ring Tracking，数据范围0~32768，32bit整数。

# APP\_UPDATE\_REQ(0x96)

更新参数命令，所有更新的参数传递完毕之后，如没有错误，需要发送此命令通知device重新配置系统和存储参数到本地。数据类型为bool变量，通过16bit数据发送数据0x01出发参数写入。

# APP\_REQ\_CAPTURES1(0x98)

此功能用于回传采样数据，支持功能码0x10，上行或者下行数据的数量由4.13中num\_samples确定，由于num\_samples最大可能是200, 如果num\_sample在1和 125之间，上行采样数据使用APP\_REQ\_CAPTURES1命令，下行采样数据使用APP\_REQ\_CAPTURES3命令，如果多余125个数，则126~200之间的数据，上行使用APP\_REQ\_CAPTURES2命令，下行使用APP\_REQ\_CAPTURE4命令。如果数据多余125，建议将数据平均分配到两个数据包中传送。每一个sample使用16bit无符号数传送。

# APP\_VERSION(0x99)

此功能用于回读设备应用、设备、硬件以及固件ID，32bit数据，内容待定，支持功能码0x03。

# F1\_F2(0x9B)

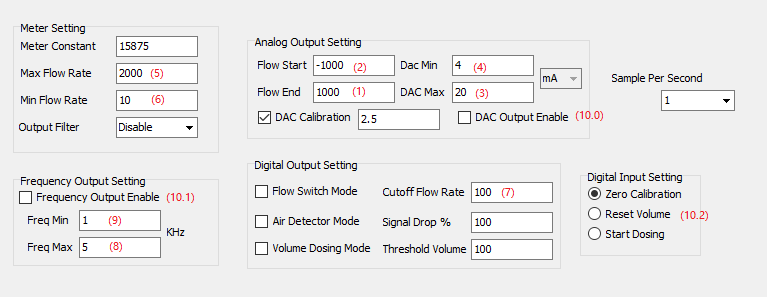
此功能用于配置扫描频率，单位kHz，支持0x06功能码，连续写两个寄存器，分别为F1, F2，F1代表频率扫描起始频率，F2的功能待定。

# AUTO\_GAIN\_ENABLE(0xE0)

允许自动gain调整，bool类型，支持0x03和0x06单次寄存器读写。写0x0001表示允许自动gain调整，0x0000表示禁止。

# FlOW\_RANGE(0xE1)

连续9个float数据，支持功能码0x03, 0x06，数据依次是：flow min，flow max，flow meter min, flow meter max, flow rate min, flow rate max, flow rate cutoff, flow frequency min, flow frequency max. 紧接着1个32位整数，其中0byte代表DAC output Enable, 1 byte 代表frequency output enable，2byte代表Digital input setting, 此byte为0代表数字输入配置为zero calibration，为1代表配置为reset volume，为2代表配置为start dosing。参数顺序如下图所示：



# RUNNING\_AVG(0xE2)

设置filter参数，支持0x03和0x06单次寄存器读写。0x0000表示不进行滤波，0x0001表示cutoff 50hz，0x0002表示cutoff 10hz，0x0003表示cutoff 1hz，0x0004表示cutoff 0.1hz

# SET\_DAC(0xE3)

设置DAC输出，支持0x03和0x10功能码，连续读写两个寄存器，寄存器1如果是0x0000表示DAC由系统控制。如果是0x0001表示DAC由寄存器2写入值控制。寄存器2值中【11：0】bit有效，代表12位DAC的配置值。

# SET\_KF(0xE4)

设置k\_factor参数。32bit数据，采用功能码0x03和0x10读写。

# SET\_KF1(0xE5)

设置measureratio数组，连续19个浮点数，代表不同流量段应乘的系数。占用38个寄存器地址。使用功能码0x03, 0x10读写。

# SET\_KF2(0xE6)

设置calipoint数组，连续19个浮点数，代表不同标定的流量段。占用38个寄存器地址。使用功能码0x03和0x10读写。

# SET\_SNSR\_TYP(0xE7)

功能未定。

# LNCAL\_EN(0xE8)

使能流量段标定功能。Bool变量，0x0001代表使能，0x0000代表禁止。

# STA\_UP(0xE9)

更新系统状态，暂未定义。

# RESET(0xEA)

清除系统buffer，流量累计等参数，重新开始计数。Bool变量，0x0001代表使能，每次发送该命令系统reset一次并自清0。

# ZERO\_CAL(0xEB)

校准0点，写入2个浮点数，其中zero\_cal单位为s，代表用这段时间内的采样数据计算0点。Zero\_cal\_value则在写入操作时被系统忽略，可以写入0。而在zero calibration过程结束可在此位置读回最新的计算0点offset值。

# AUTO\_GAIN\_UP(0xEC)

自动gain值更新，该功能暂未定义。

# ADDR(0xED)

写入device地址。8bit数据，范围从1~4，及一条总线上可以连结4个device。

# APP\_REQ\_CAPTURES2(0xEF)

此功能用于回传上行采样数据，见4.25节定义。

# APP\_REQ\_CAPTURES3(0xF0)

此功能用于回传下行采样数据，见4.25节定义。

# APP\_REQ\_CAPTURES4(0xF1)

此功能用于回传采样数据，见4.25节定义。

# APP\_REQ\_DATA(0x97)

此功能仅用于从系统读回流量累计量(float)，系统时间(int32)，上行ATOF(float)，下行ATOF(float)，DTOF, 以及可能的1，2，5，10，20，50个flow\_rate(float)数据。其中各位数据按照上述顺序排列。