

# 集中器上行通讯内部规约

版本号：V13



编制人： 周游

审核人： 李浩颖

批准人： 吴曙刚

发布日期:2018 年 12 月 1 日 实施日期:2018 年 12 月 1 日

## 第二部分 集中器上行通讯内部规约

### 一. 前言

本数据通讯协议集中器采用 TCP/IP 连接方式和服务器后台软件进行通讯，默认通讯端口 5002（可在服务器配置文件更改），采用 CRC-CCITT（循环冗余码）检验技术，保证通讯数据准确及可靠。

### 二、通讯规约

#### 1.1 数据格式

通讯用数据包交换信息。数据包以“数据包头”开头，“数据包尾”结尾。

#### 1.2 数据包格式

发送数据包：数据方向 *服务器后台软件* → *集中器*

(TCP/IP 识别包+发送数据包头)	(有效数据)	(CRC 校验码)	(数据包尾)
25 字节		2 字节	1 字节

返回数据包：数据方向 *集中器* → *服务器后台软件*

(TCP/IP 识别包+返回数据包头)	(有效数据)	(CRC 校验码)	(数据包尾)
25 字节		2 字节	1 字节

备注：CRC 校验码为有效数据的校验码。

#### 1.3 数据包

##### (1) TCP/IP 识别包

0x7B 01 00 16H 长度-4 字节 数据识别头

11 位设备号码 (ASCII) 长度-11 字节

##### (2) 发送数据包头—服务器后台软件发至集中器

数据“BBBBBBBBBS” - (0x42 42 42 42 42 42 42 42 42 53H) 长度-10 字节：发送数据包头

##### (3) 返回数据包头—集中器发至服务器后台软件

数据“SSSSSSSSB” - (0x53 53 53 53 53 53 53 53 53 42H) 长度-10 字节：返回

## 数据包头

### (4) 数据包尾

数据“E” - (0x45H) 长度-1 字节：数据包尾

## 2.1 数据术语及格式

### ★ 指令

#### (1) 读取指令（数据提取、读编号列表、读设备信息）

数据“RRR” - (0x52 52 52H) 长度-3 字节

#### (2) 采集指令（数据集中、数据刷新） 数据

“MMM” - (0x4D 4D 4DH) 长度-3 字节

#### (3) 设置系统时钟

数据“TTT” - (0x54 54 54H) 长度-3 字节

#### (4) 设置定时采集

数据“PPP” - (0x50 50 50H) 长度-3 字节

#### (5) 下载编号列表指令（页方式）（下载编号列表）

数据“WWW” - (0x57 57 57H) 长度-3 字节

#### (6) 写备注信息指令（设置集中器地址） 数据

“NNN” - (0x4E 4E 4EH) 长度-3 字节

#### (7) 写 MODEM 配置指令（GPRS 通讯方式配置集中器的目的 IP 和端口）

数据“AAA” - (0x41 41 41H) 长度-3 字节

#### (8) 采集通道打开指令

数据“000” - (0x4F 4F 4FH) 长度-3 字节

#### (9) 表阀门打开指令

数据“KKK” - (0x4B 4B 4BH) 长度-3 字节

#### (10) 表阀门关闭指令

数据“GGG” - (0x47 47 47H) 长度-3 字节

#### (11) 采集通道关闭指令

数据“CCC” - (0x43 43 43H) 长度-3 字节

#### (12) 成功返回

数据“UUU” - (0x55 55 55H) 长度-3 字节

★ 数据 内

存结构:

页方式的参数: 页号-1 字节。页方式下有效数据的长度为 384 字节。页方式下有效数据起始地址由页号给出。

页面容量为 384 字节, 页面数根据集中器容量确定。若为单倍容量的集中器, 页面数为 85 (相应表数目为 2656);

(5) 第 0 页, 设备配置信息

Char At[32]; //AT 命令, 对无线集中器无用

Char St[16]; //定时抄表, 数据格式为 PT1 (Y/N) dd2 (Y/N) ddhhmmss

(数据格式见第 5 条命令: 设置定时采集 PPP)

Char Ct[16]; //设备时钟, 数据格式为 CTyyyyymmddhhmmss

(数据格式见第 4 条命令: 设置系统时钟 TTT)

Char Ht[12][16]; //历史记录, A\*yyyyymmddhhmmss 或 A-yyyyymmddhhmmss  
12 个记录, 第一位从 A 到 T

(每个记录第一个字符为 A 到 T 表示序号, 第二个字母为当前已抄表项则用\*表示, 其他项用-表示, 后面为年月日时分秒字母)

Char Note[128]; //备注信息, 字符串, 最大长度 127

(备注信息用于描述集中器的一般信息, 为汉字或纯 ASCII 字符组成, 无特殊含义, 只供查看和阅读, 设置什么就保存什么, 电脑读取什么就显示什么即可, 不处理也不影响使用)。

(6) 第 1 页, 采集器编号

Char NodeAddress[64][6] 可保存 64 个采集器编号 如:

采集器编号为 123456789012, 数据表示: 0x129078563412H

(7) 第 2~84 页, 抄表数据

typedef struct {

```

Char NodeIndex; //采集器序号
Char Address[6]; //表地址
Char Value[4]; //表读数
Char CommState; //通讯状态
} MeterDataType;
数据- (0x@@###$%$%AABBH) 长度-6 字节：表地址 数据
- (0x@@###$%$%H) 长度-4 字节：表读数 如：上次读数为 123456.78，数据表示：0x78563412H

```

(8) 通讯状态

- 0 - 正常
- M - 读表失败
- J - 采集器失败

(9) CRC 校验码

数据- (0x@@%H) 长度-2 字节：CRC 校验数据

CRC-CCITT 是一个 17 位生成多项式  $G=[1\ 0001\ 0000\ 0010\ 0001]$ ，用多项式形式表示为  $G(x) = x^{16} + x^{12} + x^5 + 1$ ，由它产生的检验码  $R$  的二进制位数是 16 位 (2 字节)

3.1 操作方式

(1) 集中器心跳包：

数据方向 *集中器*→ *服务器后台软件*

(TCPIP 识别包)
15 字节
0x7B 01 00 16H + 11 位设备号码 (ASCII)

(2) 读取：读取集中器数据 (数据提取或读编号列表) RRR

数据方向 *服务器后台软件*→ *集中器*

(TCPIP 识别包+发送数据包头)	(指令字+页面号)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+1 字节	2 字节	1 字节

TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53H	0x52 52 52H + XX	XXXX	0x45H
---	------------------	------	-------

数据方向 集中器—) 服务器后台软件

(TCPIP 识别包+返回数据包头)	(指令字+页面号+页面 数据)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+1 字节+384 字节	2 字节	1 字节
TCPIP 识别包+0x53 53 53 53 53 53 53 53 53 42H	0x52 52 52H + XX + XXX……		

备注：页面编号 0 表示为 0x00H，1 表示为 0x01H，类推如 TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 42 53 52 52 52 3F 40 55 45 H 即为读取第 63(0x3F)页。页面数据固定长度为 384 字节。

(3) 采集：采集表数据（数据集中、数据刷新、不提取数据）MMM

数据方向 服务器后台软件—) 集中器

(TCPIP 识别包+发送数据包头)	(指令字+采集器地址+ 表地址)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+6 字节+6 字节	2 字节	1 字节
TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53H	0x4D 4D 4DH + XXXXXXXXXXXXX + XXXXXXXXXXXXX		

数据方向 集中器—) 服务器后台软件

(TCPIP 识别包+返回数据包头)	(指令字+采集器地址+ 表地址)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+6 字节+6 字节	2 字节	1 字节
TCPIP 识别包+0x53 53 53 53 53 53 53 53 53 42H			

备注：采集所有表时，采集器地址和表地址可输入 0xFFFFFFFFFFFFH，即可读取表数据。

如 TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53 4D 4D 4D 02 45 76 43 12 34 FF FF FF FF FF H 表示读取采集器编号为 341243764502 的采集器下的所有表

返回数据有两种： 1. 开始采集返回数据（指令字为“MMM”-0x4D 4D 4DH）；

如：TCPIP 识别包+535353535353535353535353424D4D4DFFFFFFFFFFFFFFFFFFFFFFFF0E7F45。

2. 采集成功返回数据（指令字为“UUU”-0x55 55 55H）。

如：TCPIP 识别包+53535353535353535353535342555555FFFFFFFFFFFFFFFFFFFFFFFFFC8DA45。

(4) 设置系统时钟 TTT

数据方向 服务器后台软件—> 集中器

(TCPIP 识别包+发送数据包头)	(指令字+数据)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+16 字节	2 字节	1 字节

数据方向集中器—> 服务器后台软件

(TCPIP 识别包+返回数据包头)	(指令字)	(CRC 校验码)	(数据包尾)
25 字节	3 字节	2 字节	1 字节

备注：数据格式为 CTyyyyymmddhhmmss。

同步时间举例

设置时间为 2005/11/09 17:17:03

发送：TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53

54 54 54

43 54 32 30 30 35 31 31 30 39 31 37 31 37 30 33

A5 27 45 H

返回：TCPIP 识别包+0x53 53 53 53 53 53 53 53 53 42

54 54 54

4E 25 45 H

(5) 设置定时采集 PPP

数据方向 服务器后台软件—> 集中器

(TCPIP 识别包+发送数据包头)	(指令字+数据)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+16 字节	2 字节	1 字节

数据方向集中器—> 服务器后台软件

(TCPIP 识别包+返回数据包头)	(指令字)	(CRC 校验码)	(数据包尾)
25 字节	3 字节	2 字节	1 字节

备注：数据格式为 PT1 (Y/N) dd2 (Y/N) ddhhmmss。

定时采集举例

每月 10 号和 30 号 8:30 采集

发送：TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53

50 50 50 50 54 31 59 31 30 32 59 33 30 30 38 33 30 30 30

FD B8 45 H

返回：TCPIP 识别包+0x53 53 53 53 53 53 53 53 53 42

50 50 50

0A A5 45 H

每月 15 号 9:25 采集

发送：TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53

50 50 50 50 54 31 59 31 35 32 4E 33 30 30 39 32 35 30 30

F0 EF 45 H

2N 后的日期是上次设置的日期，实际不起作用

(6) 写编号：(页方式)(下载编号列表) WWW

数据方向 服务器后台软件—> 集中器

(TCPIP 识别包+发送数据包头)	(指令字+页面号+设备地址数组)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+1 字节+384 字节	2 字节	1 字节

数据方向集中器—> 服务器后台软件

(TCPIP 识别包+返回数据包头)	(指令字)	(CRC 校验码)	(数据包尾)
25 字节	3 字节	2 字节	1 字节

先写采集器编号页，再写表数据页。

采集器编号页长度 384 字节，每采集器地址 6 个字节，采集器页实际可以用数组 Array[64][6] 表示，从 Array[0] 开始，依次写入采集器地址，如采集器地址为 123425064801 写入 Array[0] 为 [0x01][0x48][0x06][0x25] [0x34] [0x12]，空余地址用 0x00 填充构建采集器编号页。

发送：TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53 (发送数据包头)

57 57 57 (指令字)

01 (页面号)









(12) 采集通道关闭

数据方向 服务器后台软件—> 集中器

(TCPIP 识别包+发送数据包头)	(指令字+采集器地址)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+6 字节	2 字节	1 字节
TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53H	0x43 43 43 H + XXXXXXXXXXXX		

数据方向 集中器—> 服务器后台软件

(TCPIP 识别包+返回数据包头)	(指令字+采集器地址)	(CRC 校验码)	(数据包尾)
25 字节	3 字节+6 字节	2 字节	1 字节
TCPIP 识别包+0x53 53 53 53 53 53 53 53 53 42H			

备注：当要关闭集中器的输出 1 通道，采集器地址输入：0x000000000001H，即可关闭采集通道。如 TCPIP 识别包+0x42 42 42 42 42 42 42 42 42 53 43 43 43 01 00 00 00 00 00 H 表示关闭集中器的输出 1 通道(即采集通道关闭)

发送数据：集中器采集通道关闭发送数据（指令字为“CCC”-0x43 43 43H）；

如 TCPIP 识别包+0x4242424242424242425343434301000000000289045

返回数据：集中器采集通道关闭返回数据（指令字为“CCC”-0x43 43 43H）；

如：TCPIP 识别包+0x5353535353535353534243434301000000000289045

指令汇总：

(一) 下载表编号列表指令： 页方式：多次

用指令 WWW

(二) 读编号列表指令： 页方式：多次用指

令 RRR

(三) 数据采集：

1、数据集中、数据刷新及数据提取：先用指令 MMM、刷新完毕再多次用指令 RRR

2、只提取（不集中、不刷新）：多次用指令 RRR

集中是指读取表实际数据（集中时间较长但表数据与表盘数据一致），刷新是指将数据写到集中器定时抄表数据库(存储器)，提取是指将数据返回给电脑系统软件。刷新将影响集中器定时抄表数据库，不刷新将不影响集中器定时抄表数据库。

产生 CRC 校验码举例：

采集

数据方向 服务器后台软件→ 集中器

TCPIP 识别包	0x4D 4D 4DH + 0xFF FF FF FF FF FFH	0x@@ @@H	0x45H
+0x42 42 4242 42	+ 0xFF FF FF FF FF FFH		
42 42 42 42 53H			

//--说明--

--有效数据(0x4D 4D 4DH + 0xFF FF FF FF FF FFH + 0xFF FF FF FF FF FFH) 15 字节长度

//j - 有效数据长度

//s[0]-s[j-1] --有效数据(0x4D 4D 4DH + 0xFF FF FF FF FF FFH + 0xFF FF FF FF FF FFH)

//s[j]-s[j+1] --CRC 校验码(0x@@@H)

//i, CrcA, CrcB, CrcC --临时变量

//--程序--

CrcA = s[0];

CrcB = s[1];

CrcC = s[2];

for (i=2;i<j;i++) {

    CRC(&CrcA, &CrcB, &CrcC);

    CrcC = s[i+1];

}

s[j] = CrcA;

s[j+1] = CrcB;

//\*\*\*\*\*

//这里以 CRC-CCITT 标准生成多项式。CRC-CCITT 是一个 17 位生成多项式

//G=[1 0001 0000 0010 0001]，用多项式形式表示为  $G(x) = x^{16} + x^{12}$

//+x<sup>5</sup>+1，由它产生的检验码 R 的二进制位数是 16 位（2 字节）。

const unsigned char ha00[256] = {

    0x00, 0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70, 0x81, 0x91, 0xA1, 0xB1, 0xC1, 0xD1, 0xE1, 0xF1,

    0x12, 0x02, 0x32, 0x22, 0x52, 0x42, 0x72, 0x62, 0x93, 0x83, 0xB3, 0xA3, 0xD3, 0xC3, 0xF3, 0xE3,

```
0x24, 0x34, 0x04, 0x14, 0x64, 0x74, 0x44, 0x54, 0xA5, 0xB5, 0x85, 0x95, 0xE5, 0xF5, 0xC5, 0xD5,
0x36, 0x26, 0x16, 0x06, 0x76, 0x66, 0x56, 0x46, 0xB7, 0xA7, 0x97, 0x87, 0xF7, 0xE7, 0xD7, 0xC7,
0x48, 0x58, 0x68, 0x78, 0x08, 0x18, 0x28, 0x38, 0xC9, 0xD9, 0xE9, 0xF9, 0x89, 0x99, 0xA9, 0xB9,
0x5A, 0x4A, 0x7A, 0x6A, 0x1A, 0x0A, 0x3A, 0x2A, 0xDB, 0xCB, 0xFB, 0xEB, 0x9B, 0x8B, 0xBB, 0xAB,
0x6C, 0x7C, 0x4C, 0x5C, 0x2C, 0x3C, 0x0C, 0x1C, 0xED, 0xFD, 0xCD, 0xDD, 0xAD, 0xBD, 0x8D, 0x9D,
0x7E, 0x6E, 0x5E, 0x4E, 0x3E, 0x2E, 0x1E, 0x0E, 0xFF, 0xEF, 0xDF, 0xCF, 0xBF, 0xAF, 0x9F, 0x8F,
0x91, 0x81, 0xB1, 0xA1, 0xD1, 0xC1, 0xF1, 0xE1, 0x10, 0x00, 0x30, 0x20, 0x50, 0x40, 0x70, 0x60,
0x83, 0x93, 0xA3, 0xB3, 0xC3, 0xD3, 0xE3, 0xF3, 0x02, 0x12, 0x22, 0x32, 0x42, 0x52, 0x62, 0x72,
0xB5, 0xA5, 0x95, 0x85, 0xF5, 0xE5, 0xD5, 0xC5, 0x34, 0x24, 0x14, 0x04, 0x74, 0x64, 0x54, 0x44,
0xA7, 0xB7, 0x87, 0x97, 0xE7, 0xF7, 0xC7, 0xD7, 0x26, 0x36, 0x06, 0x16, 0x66, 0x76, 0x46, 0x56,
0xD9, 0xC9, 0xF9, 0xE9, 0x99, 0x89, 0xB9, 0xA9, 0x58, 0x48, 0x78, 0x68, 0x18, 0x08, 0x38, 0x28,
0xCB, 0xDB, 0xEB, 0xFB, 0x8B, 0x9B, 0xAB, 0xBB, 0x4A, 0x5A, 0x6A, 0x7A, 0x0A, 0x1A, 0x2A, 0x3A,
0xFD, 0xED, 0xDD, 0xCD, 0xBD, 0xAD, 0x9D, 0x8D, 0x7C, 0x6C, 0x5C, 0x4C, 0x3C, 0x2C, 0x1C, 0x0C,
0xEF, 0xFF, 0xCF, 0xDF, 0xAF, 0xBF, 0x8F, 0x9F, 0x6E, 0x7E, 0x4E, 0x5E, 0x2E, 0x3E, 0x0E, 0x1E
};
```

```
const unsigned char la00[256] = {
0x00, 0x21, 0x42, 0x63, 0x84, 0xA5, 0xC6, 0xE7, 0x08, 0x29, 0x4A, 0x6B, 0x8C, 0xAD, 0xCE, 0xEF,
0x31, 0x10, 0x73, 0x52, 0xB5, 0x94, 0xF7, 0xD6, 0x39, 0x18, 0x7B, 0x5A, 0xBD, 0x9C, 0xFF, 0xDE,
0x62, 0x43, 0x20, 0x01, 0xE6, 0xC7, 0xA4, 0x85, 0x6A, 0x4B, 0x28, 0x09, 0xEE, 0xCF, 0xAC, 0x8D,
0x53, 0x72, 0x11, 0x30, 0xD7, 0xF6, 0x95, 0xB4, 0x5B, 0x7A, 0x19, 0x38, 0xDF, 0xFE, 0x9D, 0xBC,
0xC4, 0xE5, 0x86, 0xA7, 0x40, 0x61, 0x02, 0x23, 0xCC, 0xED, 0x8E, 0xAF, 0x48, 0x69, 0x0A, 0x2B,
0xF5, 0xD4, 0xB7, 0x96, 0x71, 0x50, 0x33, 0x12, 0xFD, 0xDC, 0xBF, 0x9E, 0x79, 0x58, 0x3B, 0x1A,
0xA6, 0x87, 0xE4, 0xC5, 0x22, 0x03, 0x60, 0x41, 0xAE, 0x8F, 0xEC, 0xCD, 0x2A, 0x0B, 0x68, 0x49,
0x97, 0xB6, 0xD5, 0xF4, 0x13, 0x32, 0x51, 0x70, 0x9F, 0xBE, 0xDD, 0xFC, 0x1B, 0x3A, 0x59, 0x78,
0x88, 0xA9, 0xCA, 0xEB, 0x0C, 0x2D, 0x4E, 0x6F, 0x80, 0xA1, 0xC2, 0xE3, 0x04, 0x25, 0x46, 0x67,
0xB9, 0x98, 0xFB, 0xDA, 0x3D, 0x1C, 0x7F, 0x5E, 0xB1, 0x90, 0xF3, 0xD2, 0x35, 0x14, 0x77, 0x56,
0xEA, 0xCB, 0xA8, 0x89, 0x6E, 0x4F, 0x2C, 0x0D, 0xE2, 0xC3, 0xA0, 0x81, 0x66, 0x47, 0x24, 0x05,
0xDB, 0xFA, 0x99, 0xB8, 0x5F, 0x7E, 0x1D, 0x3C, 0xD3, 0xF2, 0x91, 0xB0, 0x57, 0x76, 0x15, 0x34,
```

```
0x4C, 0x6D, 0x0E, 0x2F, 0xC8, 0xE9, 0x8A, 0xAB, 0x44, 0x65, 0x06, 0x27, 0xC0, 0xE1, 0x82, 0xA3,  
0x7D, 0x5C, 0x3F, 0x1E, 0xF9, 0xD8, 0xBB, 0x9A, 0x75, 0x54, 0x37, 0x16, 0xF1, 0xD0, 0xB3, 0x92,  
0x2E, 0x0F, 0x6C, 0x4D, 0xAA, 0x8B, 0xE8, 0xC9, 0x26, 0x07, 0x64, 0x45, 0xA2, 0x83, 0xE0, 0xC1,  
0x1F, 0x3E, 0x5D, 0x7C, 0x9B, 0xBA, 0xD9, 0xF8, 0x17, 0x36, 0x55, 0x74, 0x93, 0xB2, 0xD1, 0xF0  
};  
void CRC(uchar *a, uchar *b, uchar *c) {  
    *c = la00[*a] ^ *c;  
    *a = ha00[*a] ^ *b;  
    *b = *c;  
}
```