



众山DTU脚本编程手册

众联万物 智慧未来
我们用心创造

目录

前 言.....	4
版权声明	4
版本信息.....	4
相关文档.....	4
一、众山 DTU 自动采集脚本简介.....	6
二、脚本详解.....	6
2.1 脚本格式.....	6
2.2 脚本指令详解.....	7
2.2.1 @E 命令回传指令.....	7
2.2.2 @Q 无限循环指令.....	7
2.2.3 @C 执行命令指令.....	7
2.2.4@CS 执行命令指令.....	8
2.2.5 @O 采集命令指令.....	8
2.2.6 @OS 采集命令指令.....	9
2.2.7 @V 插入校验指令.....	9
2.2.8@D 延时控制指令.....	9
2.2.9 @A 接收应答指令.....	10
2.2.10 @S 立即上报指令.....	10
2.2.11 @H 数据标识指令.....	11
2.2.12 @STR 数据标识指令.....	11
2.2.13 @B 波特率控制指令.....	11
2.2.14 @T 上报周期指令.....	12
2.2.15 @F 接收数据格式指令.....	12
2.2.16 @P 包序号指令.....	12
2.2.17 @BV 获取电池电压指令.....	13
2.2.18 @CSQ 获取信号强度指令.....	13
2.2.19 @CID 获取云 ID 指令.....	14
2.2.20 @DID 获取设备 ID 指令.....	14
2.2.21 @IMEI 获取设备 IMEI 码指令.....	14
2.2.22 @IMSI 获取设备 IMSI 码指令.....	14
2.2.23 @ICCID 获取设备 SIM 卡 ICCID 指令.....	14
2.2.24 @SSW 透传开关指令.....	15
2.2.25 @CUT 接收数据截取指令.....	15
2.2.26 @M 标记指令.....	15
2.2.27 @L 循环控制指令.....	16

2.2.28 @DOx 继电器开关控制指令.....	16
2.2.29 @DIx 触发采集、状态上报指令.....	16
2.2.30 @AIx 值上报指令.....	18
2.2.31 @IF @ELSE @ENDIF 条件执行指令.....	18
三、应用案例.....	20
案例 1.....	20
案例 2.....	21

前 言

感谢您使用成都众山科技有限公司提供的DTU产品。

本手册主要介绍众山 DTU 本地脚本编程指令。

适用型号：ZSDxxxx、ZSNxxx、ZSLxxx、ZSExxx、ZSDRxxxx、ZSNRxxx、ZSRxxx、ZSLRxxx

版权声明

本手册版权属于成都众山科技有限公司，任何人未经我公司书面同意复制将承担相应法律责任。

版本信息

文档名称：众山DTU脚本编程手册

版本：1.12

修改日期：2019年5月23日

相关文档

- 1、《ZSDxxxx DTU Easy 控件接口说明》
- 2、《众山 DTU Modbus 协议手册》
- 3、《串口及远程控制协议》

更新记录

日期：2018-12-17

版本：ZSDxx1x 2/3/4G DTU(5.22及以上)

ZSNx11 NB-IoT DTU(1.34及以上)

ZSNx31 NB-IoT DTU(1.39及以上)

ZSL LoRa DTU(2.18及以上)

ZSR2184 RTU(1.12及以上)

- 1、增加@CSQ 指令获取信号强度
- 2、增加@BV 指令获取电池电压
- 3、增加@STR 指令插入字符格式的数据
- 4、增加@CS 和@OS 指令插入字符格式的命令
- 5、增加@CID 指令插入云 ID
- 6、增加@DID 指令插入设备 ID
- 7、增加@IMEI 指令插入设备的 IMEI 码
- 8、增加@IMSI 指令插入设备的 IMSI 码
- 9、增加@ICCID 指令插入设备 SIM 卡的 ICCID 码
- 10、增加?1 或?(兼容以前的格式)表示返回 HEX 格式的数据，?2 表示返回可见文本格式的数据
- 11、增加~1 或~表示返回 DI/DO 状态的 HEX 格式反码，~2 表示返回 DI/DO 状态的可见文本格式反码

更新记录

日期：2019-05-23

版本：ZSDxx1x 2/3/4G DTU(5.27及以上)

ZSNx11 NB-IoT DTU(1.35及以上)

ZSNx31 NB-IoT DTU(1.41及以上)

ZSL LoRa DTU(2.20及以上)

ZSR2184 RTU(1.15及以上)

- 1、增加@IF @ELSE @ENDIF条件控制脚本

一、众山DTU自动采集脚本简介

在很多应用中，用户设备是被动仪表如 Modbus 设备，需要发送命令进行数据采集，众山 DTU 针对这类应用开发了一种脚本语言，用户可以使用脚本语言自定义用户设备的采集行为。

众山 DTU 支持用户自定义脚本实现自动采集外接传感器、仪表数据，无需用户再单独增加控制器。通过配置 DTU 的脚本指令实现 DTU 定时自动采集，用户只需知道外接仪表、传感器的采集流程，然后通过编写脚本指令即可让 DTU 按照用户的流程自动采集、上传数据。脚本实现了基本的开关控制、模拟量数字量及继电器状态上报、延时控制、指令下发、循环控制、上报控制、插入自定义报文头、是否接收仪表应答、波特率改变、数据接收截取等功能，能满足大部分仪表的采集需求。脚本可以周期执行，也可以 DI 触发执行，或者两者同时存在。

此文档定义的脚本指令主要是为了实现 DTU 自动采集功能，在某种情况下，用户可能需要数据中心远程控制采集或者干预一些开关动作，众山 DTU 也支持用户的数据中心通过 SHELL:+脚本的形式发送给 DTU，DTU 收到此数据会临时执行一次数据中心下发的脚本来实现远程采集和控制。如数据中心下发：
SHELL:@DO1=1@C=0102@DO1=0 DTU 收到 SHELL:开始的数据不会透传给 DTU 串口，而是执行一次脚本。

数据中心可以通过 STARTSHELL 指令让 DTU 立即启动本地脚本的执行。远程控制 AA55 协议中也有完成相同功能的命令用来远程启动一次脚本执行。

众山 DTU 具有的所有控制功能和 Modbus 协议都可以用在脚本的 @C 命令中，让脚本不仅实现采集外接仪表的功能，还能实现控制 DTU 自己。

脚本执行和程序一样，从开始顺序执行，有循环指令时才会改变执行顺序，直到脚本执行完毕退出。默认情况下，在 1 次脚本执行中，收到的 DTU 响应或外接串口仪表响应数据时在脚本执行完毕才会打包上传，当然用户可以用 @T 定义多个采集周期才上传或者使用 @S=1 在脚本执行中用户需要的地方立即上传当前接收到的数据。

二、脚本详解

2.1 脚本格式

@cmd=value

@: 脚本头，每一条脚本指令都使用 @ 开始。

cmd: 指令，为 1 或多个字符、数字组成的字符串，不区分大小写，下面详解每一条脚本指令的含义。

value: 指令动作，指定指令需要执行的值。

2.2 脚本指令详解

2.2.1 @E 命令回传指令

指令	名称	格式	含义	默认值
@E	命令回传	@E=1 @E=0	控制@C 和@O 定义的命令是否需要回传中心 @E=1 表示回传 @E=0 表示不回传	@E=0

E: 命令回传, 此脚本指令控制@C 和@O 脚本指令定义的采集命令是否需要回传给中心, value 为 1 表示需要回传, value 为 0 表示不回传。不使用此指令时默认为不回传。此指令为整个脚本全局生效指令, 直到脚本结束。如果有些采集指令需要回传有些不需要回传只需要在相应的采集指令前加入此指令做相应的控制即可。

示例: @E=1 回传采集命令

@E=0 不回传采集命令

2.2.2 @Q 无限循环指令

指令	名称	格式	含义	默认值
@Q	无限循环执行	@Q=1 @Q=0	脚本循环执行永不退出 @Q=1 循环执行 @Q=0 不循环执行	@Q=0

Q: 无限循环执行, 在某些需要快速不间断采集的场合, 脚本不需要退出等待下一个周期时使用 @Q=1 让脚本一直循环执行, 不使用此指令时默认为不循环执行。此指令只需要在脚本任意位置定义 1 次即可。

示例: @Q=1 循环执行

@Q=0 执行完毕退出, 等待下一个周期

2.2.3 @C 执行命令指令 (HEX 格式)

指令	名称	格式	含义
@C	执行命令	@C=HEX @C=HEX+V1 @C=HEX+V2	定义执行命令, 等效于中心下发数据, 命令输入采用 HEX 格式

C: 执行命令, @C 定义的命令 DTU 首先会判断是不是 AA55 开始的控制自己的指令 (参考控制协议手册) 或者 Modbus 地址为自己的指令 (参考 DTU 的 Modbus 手册), 如是, 则执行命令的功能, DTU

响应的数据会进入发送队列，如不是，DTU 则会输出命令到串口作为外接仪表的采集命令。当命令为众山 DTU 控制协议时，可以在命令最后使用 V2 让 DTU 自动计算累加和校验并跟在命令的结尾，当命令为 Modbus 指令时，可以在命令最后使用 V1 让 DTU 自动计算 CRC 校验并跟在命令的结尾，省去了用户需要自己计算校验的麻烦。此脚本指令定义的采集命令内容为可见的 HEX 字符。

示例：Modbus 采集指令：01 03 00 01 00 01 12 34

脚本表示为：@C=0103000100011234

定义取得 DTU 信号值的指令并自动计算 ACC 校验：AA 55 00 04 E0 23

脚本表示为：@C=AA550004E023V2

定义取得 DTU 的 DI2 状态的 Modbus 指令并自动计算 CRC 校验：64 03 00 11 00 01

脚本表示为：@C=640300110001V1

命令执行或者下发后，DTU 自己的响应命令或者仪表返回的数据，DTU 放入发送队列，根据定义的发送控制适当的时候进行上传。

2.2.4 @CS 执行命令指令（字符格式）

指令	名称	格式	含义
@CS	执行命令	@CS=STR	定义执行命令，等效于中心下发数据，字符格式，功能同 @C，不支持自动校验。字符内容不能出现 @ 字符，其他字符都支持

CS： 执行命令，@CS 定义的命令 DTU 会首先判断是不是控制 DTU 自己的命令，如是，则执行命令的功能，DTU 响应的数据会进入发送队列，如不是，DTU 则会输出命令到串口作为外接仪表的采集命令。此指令功能和 @C 相同，@C 命令输入的是 HEX 格式命令，@CS 输入的命令采用文本格式，适合于用户仪表为字符通讯协议的应用，免去用户转换成 HEX 格式命令的麻烦。

2.2.5 @O 采集命令指令（HEX 格式）

指令	名称	格式	含义
@O	采集命令	@O=HEX @O=HEX+V1 @O=HEX+V2	同 @C 命令，@O 指令定义的命令 DTU 不进行判断直接输出到串口作为仪表采集命令，命令输入采用 HEX 格式

O: 采集命令, @O定义的命令格式和@C完全一样, 区别仅在于DTU不会识别是否为控制DTU自己的命令, 都作为仪表采集指令下发到串口, 即使用@O指令不能控制DTU自己。

2.2.6 @OS 采集命令指令 (字符格式)

指令	名称	格式	含义
@OS	采集命令	@OS=STR	同@O 命令, 采用字符格式输入命令, 不支持自动校验。字符内容不能出现@字符, 其他字符都支持。

OS: 采集命令, @O采用HEX格式输入命令, @OS采用字符格式输入命令, 适合于用户仪表为字符通讯协议的应用, 免去用户转换成HEX格式命令的麻烦。

2.2.7 @V 插入校验指令

指令	名称	格式	含义
@V	插入校验	@V=1 @V=2	计算接收数据的校验并在当前位置插入

V: 插入校验, @V=1 在当前位置插入此前接收到的所有数据的 CRC 校验, @V=2 在当前位置插入此前接收到的所有数据的累加和校验, 此指令不同于@C 和@O 命令后面的 V1 和 V2, 命令后面紧跟的 V1 和 V2 表示计算@C@O 定义的命令的校验, 此处单独的@V 指令 表示在此处插入前面接收数据的校验, 便于用户中心接收数据进行合法性校验。

2.2.8 @D 延时控制指令

指令	名称	格式	含义
@D	延时控制	@D=XY	在此处 DTU 会等待指令所定义的延时, 然后再继续执行 X 表示数值, Y 表示单位 支持的单位有: 'S': 秒 'M': 分钟 'H': 小时 'm': 毫秒

D: 延时控制, 此脚本指令用于控制延时, DTU 开关仪表电源后等待仪表稳定的延时或者发送采集命令后等待应答的延时控制。value 为数字+单位的格式, 单位有 S(秒)、M(分钟)、H(小时), m(毫秒), DTU 遇到此指令后等待相应的延时值再继续执行脚本。当单位为毫秒时, 由于 DTU 的逻辑定时以 0.01 秒为最小分辨率, 所以不能被 10 整除的 0.001 秒数值会被增加到一个 0.01 秒, 比如 11 毫秒实际 DTU 延

时 20 毫秒，由于 DTU 的逻辑定时器的最小分辨率为 10 毫秒，所以尽量不要使用低于或等于 10 毫秒的值，即延时 10 毫秒 DTU 实际延时可能会在 0~10 毫秒，在 0 时导致实际没有延时。

示例：@D=10S （等待 10 秒）

@D=2M （等待 2 分钟）

@D=1H （等待 1 小时）

@D=100m (等待 100 毫秒)

2.2.9 @A 接收应答指令

指令	名称	格式	含义	默认值
@A	接收应答	@A=1 @A=0	在某些命令用户不需要关心仪表的返回值时可以使用@A=0 关闭接收仪表响应，在需要接收的时候再用@A=1 打开	@A=1

A: 接收应答，此脚本指令用于控制 DTU 是否接收仪表的应答，value 为 1 表示接收应答，value 为 0 表示不接收应答。不使用此指令时默认为接收应答。此指令为整个脚本全局生效指令，直到脚本结束，如果有些采集指令需要接收应答有些不需要接收应答只需在相应的采集命令前加入此指令做相应的控制即可。

示例：@A=0 不接收后面仪表的应答数据

小提示：在有些需要开关外接传感器电源的应用场合，此指令的灵活使用可以避免传感器开电期间串口数据的不稳定性，如传感器在开电期间可能会产生无用的数据或乱码输出给DTU，此时，用户可以在开电之前使用@A=0关闭DTU接收传感器数据，然后开电，等待传感器稳定，然后再@A=1打开接收，最后发送命令，这样可以过滤掉传感器上电期间串口不稳定性导致的多余数据。比如：

```
@A=0@DO1=1@D=2S@A=1@C=xxxxxx@D=2S
```

2.2.10 @S 立即上报指令

指令	名称	格式	含义	默认值
@S	立即上报	@S=1	在进行多次采集或者多仪表采集时，可以使用@S=1 命令强制 DTU 进行一次当前接收数据的发送，如果不使用@S=1 DTU 默认会脚本执行完毕才进行发送	@S=0

S: 立即上报，此脚本指令用于控制 DTU 立即上报仪表数据。value 为 1 表示立即上报，默认情况下 DTU 执行完 T 指令控制的采集次数后上报（T 默认为 1，每次脚本执行结束都上报），用户在需要分

开上报多个仪表数据时或者 1 个仪表的多条数据时，可以在发送采集命令后，DTU 已经收到仪表应答时插入@S=1 来立即上报之前的数据。

示例：@S=1 立即上报

2.2.11 @H 数据标识指令（HEX 格式）

指令	名称	格式	含义
@H	数据标识	@H=HEX	定义固定上传的数据，用于用户在不同的数据前使用@H 指令插入固定的标识数据，DTU 遇到@H 指令时直接向发送缓冲区的当前位置插入@H 后面定义的数据，数据输入格式为 HEX 格式

H： 数据标识，此脚本指令用于在采集的仪表数据前加入用户自定义的数据标识或用户便于解析的特征数据等，比如在进行多个仪表数据采集时加入分隔符便于区分不同的仪表数据，value 为可见的 HEX 字符，比如 123 表示为 313233。如需要多个标识数据需要在需要的位置使用@H 插入。

示例：@H=DD01 @H=DD02

2.2.12 @STR 数据标识指令（字符格式）

指令	名称	格式	含义
@STR	数据标识	@STR=STR	定义固定上传的数据，用于用户在不同的数据前使用@STR 指令插入固定的标识数据，DTU 遇到@STR 指令时直接向发送缓冲区的当前位置插入@STR 后面定义的数据，数据格式采用字符格式输入，字符内容不能出现@ 字符，其他字符都支持

STR： 数据标识，此脚本指令用于在采集的仪表数据前加入用户自定义的数据标识或用户便于解析的特征数据等，比如在进行多个仪表数据采集时加入分隔符便于区分不同的仪表数据，value 为可见的文本字符。如需要多个标识数据需要在需要的位置使用@STR 插入。

示例：@STR=123 @STR=VCC=

2.2.13 @B 波特率及串口配置指令

指令	名称	格式	含义
@B	串口配置 波特率改变	@B=38400,8E1 @B=9600 @B=,8O1	当采集多个仪表，仪表的波特率或串口配置不一致时，用于改变 DTU 串口的波特率和配置，脚本执行完毕自动回到 DTU 默认参数配置的串口波特率和配置。

B: 波特率控制, 此脚本指令用于外接多个不同波特率或不同串口配置的仪表或者仪表波特率与 DTU 串口波特率不相同临时改变 DTU 的串口波特率。value 为波特率值和串口配置, 需要在发送采集命令前通过此指令改变 DTU 波特率与仪表波特率一致。当脚本执行结束, 会自动回到 DTU 默认的串口波特率。当只需要改变其中一项时, 可以省略另一项, 当省略波特率时, 逗号不能省略, 串口配置允许的值为 8E1,8O1,8N1,7E1,7O1,7N1,8E2,8O2,8N2,7E2,7O2,7N2

2.2.14 @T 上报周期指令

指令	名称	格式	含义	默认值
@T	上报周期	@T=3	定义执行多少次脚本后上报数据	@T=1

T: 上报周期, 默认情况下, 在用户没有插入强制发送@S=1 指令时, DTU 脚本执行结束才会发送采集到的数据, 即默认@T=1 脚本执行一次就发送, 某些情况下, 用户希望节约流量或功耗, 采集几个周期以后才进行一次发送, 比如 5 分钟采一次, 30 分钟发送一次数据, 此时就可以使用@T=6 让 DTU 采集 6 次以后再进行发送。由于 DTU 在脚本执行完毕后会回到透传模式, 在透传模式下, DTU 是根据串口分包时间间隔进行打包发送数据的, 所以如果用户使用多次采集一次发送时, 需要在非采集期间用户仪表没有主动数据输出, 否则 DTU 在透传期间如果收到了数据, 会立即打包发送, 幸运的是针对这种情况, 我们提供了另一条指令@SSW=0 用来关闭透传功能, 用户只要在脚本中任何地方插入@SSW=0, DTU 只有在脚本执行期间接收串口数据, 其他时候不理睬串口数据。此指令为整个脚本全局生效指令, 直到脚本结束, 并且只允许出现一次此指令, 如果此指令出现多次时, 以最后一次的值为有效。注意: 此指令控制的上报次数以一次脚本完全执行结束为一个周期, 与 L 指令控制的循环执行不产生影响。

示例: @T=3 表示 3 个采集周期结束后才上报数据。

2.2.15 @F 接收数据格式指令

指令	名称	格式	含义	默认值
@F	接收数据格式	@F=0 @F=1	定义接收原始的仪表数据或进行 HEX 转可见字符	@F=0

F: 接收数据格式, 默认情况下@F=0 DTU 接收仪表的原始数据, 在某些应用中, 比如用户中心希望接收到可见的 ASCII 字符, 使用@F=1, DTU 会自动把接收到的数据 1 个字节转换成 2 个可见字符, 如: 仪表数据位 01 03 02 00 01 AC 8F (7 个 HEX 数据), DTU 转换后变成 “0103020001AC8F” (14 个可见字符)。

示例: @F=1 上报 1 个字节转换成 2 个可见的 HEX 字符。

2.2.16 @P 包序号指令

指令	名称	格式	含义
@P	包序号	@P=? @P=?1 @P=?2	在当前位置插入包序号,?和?1 采用 4 字节的 HEX 格式插入包序号, ?2 采用可见十进制数字插入包序号

P: 包序号, 包序号从 0 开始, 每次增加 1, 当 DTU 碰到@P=?指令会立即向发送缓冲区放置一个包序号, 然后递增包序号。

2.2.17 @BV 获取电池电压指令

指令	名称	格式	含义
@BV	电池电压	@BV=? @BV=?1 @BV=?2	在当前位置插入电池电压, ?和?1 采用 2 字节的 HEX 格式插入电池电压, ?2 采用可见十进制数字插入电池电压。

BV: 电池电压, 仅有硬件支持测电池电压的 DTU 型号才支持此指令。?和?1 获取的电池电压采用 2 字节 HEX 格式, 实际电压需要把 HEX 转换成十进制数据再除以 100, 得到实际电压值。?2 获取的电压值即为文本格式的电压, 比如 3.3。

示例: @STR=VCC=@BV=?2 如果电压为 3.3V, DTU 上报的数据为: VCC=3.3

@BV=?1 如果电压为 3.3V, DTU 上报的为 2 个数据: 0x01 0x4A

2.2.18 @CSQ 获取信号强度指令

指令	名称	格式	含义
@CSQ	信号强度	@CSQ=? @CSQ=?1 @CSQ=?2	在当前位置插入信号强度, ?和?1 采用 HEX 格式插入信号强度, ?2 采用可见文本格式插入信号强度。

CSQ: 信号强度。在 234G 和 NB-IoT DTU 型号中?和?1 获取的信号强度采用 1 字节 HEX 格式, ?2 获取的信号强度为文本格式的信号强度。在 LoRa DTU 型号中, 信号强度表示和最近通讯的 LoRa 之间的信号强度, 所以?和?1 获取的信号强度采用 2 字节 ID 号+1 字节信号强度的 HEX 格式, ?2 获取的信号强度为 id:value 的格式。

示例: 234G 和 NB IoT 型号

@STR=CSQ=@CSQ=?2 如果信号强度为 27, DTU 上报的数据为: CSQ=27

@CSQ=?1 如果信号强度为 27, DTU 上报的为 1 个数据: 0x1B

示例: LoRa 型号

@STR=CSQ=@CSQ=?2 如果最近一次和 LoRa DTU 通讯的 ID 号为 18 号，信号强度为 27，DTU 上报的数据为：CSQ=18:27

@CSQ=?1 如果最近一次和 LoRa DTU 通讯的 ID 号为 18 号，信号强度为 27，DTU 上报的为 3 个数据：0x00 0x12 0x1B

2.2.19 @CID 获取云 ID 指令

指令	名称	格式	含义
@CID	云 ID	@CID=? @CID=?1 @CID=?2	在当前位置插入云 ID,?和?1 采用 8 字节的 HEX 格式插入云 ID, ?2 采用 16 字节可见 HEX 字符插入云 ID

CID: 云 ID, DTU 的云 ID 为 8 字节 BCD 码格式, ?和?1 采用 8 字节 HEX 源码插入云 ID, ?2 采用 16 字节可见 HEX 字符插入云 ID。此指令可以让用户在脚本上传的数据包中加入云 ID, 便于中心识别不同的 DTU。

2.2.20 @DID 获取设备 ID 指令

指令	名称	格式	含义
@DID	设备 ID	@DID=?	在当前位置插入设备 ID, 设备 ID 为 8 位字符, 所以均采用 8 字符源码格式

DID: 设备 ID, 8 位字符, 此指令可以让用户在脚本上传的数据包中加入设备 ID, 便于中心识别不同的 DTU。LoRa DTU 系列不支持此指令。

2.2.21 @IMEI 获取设备 IMEI 码指令

指令	名称	格式	含义
@IMEI	IMEI 码	@IMEI=?	在当前位置插入 IMEI 码, IMEI 为字符格式, 所以均采用字符源码格式

IMEI: 设备 IMEI 码, 此指令可以让用户在脚本上传的数据包中加入设备 IMEI 码。LoRa DTU 系列不支持此指令。

2.2.22 @IMSI 获取设备 IMSI 码指令

指令	名称	格式	含义
@IMSI	IMSI 码	@IMSI=?	在当前位置插入 IMSI 码, IMSI 为字符格式, 所以均采用字符源码格式

IMSI: 设备 IMSI 码, 此指令可以让用户在脚本上传的数据包中加入设备 IMSI 码。LoRa DTU 系列不支持此指令。

2.2.23 @ICCID 获取设备 SIM 卡的 ICCID 指令

指令	名称	格式	含义
@ICCID	ICCID 码	@ICCID=?	在当前位置插入 SIM 卡的 ICCID 码, ICCID 为字符格式, 所以均采用字符源码格式

ICCID: 设备 SIM 卡的 ICCID 码, 此指令可以让用户在脚本上传的数据包中加入设备 SIM 卡的 ICCID 码。LoRa DTU 系列不支持此指令。

2.2.24 @SSW 透传开关指令

指令	名称	格式	含义	默认值
@SSW	透传开关	@SSW=0 @SSW=1	关闭或开启 DTU 的透传功能	@SSW=1

SSW: 透传开关, 关闭 DTU 的透传功能, 如果用户只专注于脚本采集期间仪表返回的数据, 则可用 @SSW=0 关闭 DTU 在非脚本采集期间收到的串口数据。默认情况下, DTU 在非脚本采集期间会回到透传功能, 即串口有数据会立即上报。

小提示: 某些传感器可能在非采集时间也会有些主动输出的数据, 如果用户不希望接收到这些数据, 可以使用此指令关闭透传功能, 只在脚本执行期间接收传感器数据

2.2.25 @CUT 接收数据截取指令

指令	名称	格式	含义
@CUT	接收数据截取	@CUT=X,Y	截取仪表返回数据的一部分进行上报

CUT: 接收数据截取, 当用户仪表返回数据较多, 并且用户不想关心一些无用的信息, 可以使用 @CUT 指令截取仪表返回数据中的有用信息, X 定义数据位置, 从 1 开始, Y 定义需要的数据个数。比如仪表返回 1234567890, 用户只需要 678 数据, 则可以在 @C 命令发送之前使用 @CUT=6,3 进行有选择的接收, CUT 指令一直生效直到碰到下一个 CUT 指令, 当用户需要取消 CUT 功能进行全接收模式时, 使用 @CUT=, 可以让 DTU 回到接收所有数据。

2.2.26 @M 标记指令

指令	名称	格式	含义
----	----	----	----

@M	标记	@M=1 @M=2 @M=3	产生标记点
----	----	----------------------	-------

M: 标记，此脚本指令无实际动作，用于产生一个标记，和 L 指令配合完成 LOOP 循环控制。value 为 1-3，支持 3 个标记，循环嵌套时需要使用不同的标记，当退出循环后，又可以使用相同的标记序号或在非循环嵌套可以重复使用相同的标记序号。

示例：@M=1 产生 M1 标记点

@M=2 产生 M2 标记点

2.2.27 @L 循环控制指令

指令	名称	格式	含义
@L	循环控制	@L=X,Y	和@M 配合进行循环控制

L: 循环控制，此脚本指令控制循环到@M 标记点的次数，用于需要对仪表进行多次重复数据采集的场合，比如需要对仪表连续采集 3 次数据。value 为 X,Y 的格式，X 表示标记点，值为 1-3 表示@M 定义的标记点，Y 为循环次数。3 个标记点允许实现 3 层循环的嵌套，不嵌套的循环无限制，可以重复使用标记点。脚本执行到 L 指令时立即跳转到指定的 M 标记点，直到达到循环指定的次数后，继续往后面执行。

示例：@L=1,3 表示跳转到 M1 标记，循环执行 3 次

@L=2,2 表示跳转到 M2 标记，循环执行 2 次

2.2.28 @DOx 继电器开关控制指令

指令	名称	格式	含义
@DO1 @DO2 @DO3 @DO4	DO 开关控制 DO 状态上报	@DO1=0 @DO1=1 @DO1=? @DO1=?1 @DO1=?2 @DO1=~ @DO1=~1	此指令用户控制 DTU 的 DO1~DO4 继电器，或者上报 DO1~DO4 的状态 ?和?1 为上报 DO 的 HEX 格式的状态，0x00 或者 0x01 ?2 为上报 DO 的字符格式的状态，‘0’ 或者 ‘1’ ~和~1 为上报 DO 状态反码的 HEX 格式，0x00 或者 0x01 ~2 为上报 DO 状态反码的字符格式，‘0’ 或者 ‘1’

		@DO1=~2	
--	--	---------	--

DO: DO 开关控制，此脚本指令用于控制 DTU 的 DO1~DO4。value 为 1 表示开，value 为 0 表示关。只有带有 DI DO 的 DTU 才具有此功能，并且不同的型号 DO 个数可能不一样，用户可以查看产品手册了解该型号支持的 DO 个数。@DO1=0 表示让 DTU 断开 DO1 继电器，@DO1=1 表示让 DTU 闭合 DO1 继电器，@DO1=?表示让 DTU 上报 DO1 继电器的状态，状态为 1 个字节，?和?1 格式时 00 表示断开，01 表示闭合。?2 格式时 ‘0’ 表示断开，‘1’ 表示闭合。~和~1 格式时 01 表示断开，00 表示闭合。~2 格式时 ‘1’ 表示断开，‘0’ 表示闭合。此指令经常用于用户在采集仪表数据时给仪表上电和关电的动作。

注：使用此指令上报时仅仅会上报一个字节的值，用户也可以用@C 指令使用 Modbus 协议控制或读取 DO，这样上报的就是符合 Modbus 协议的 DO 状态。详细信息请参考 Modbus 协议手册。

示例：@DO1=1 打开开关 1

@DO1=0 关闭开关 1

2.2.29 @DIx 触发采集、状态上报指令

指令	名称	格式	含义
@DI1 @DI2 @DI3 @DI4	DI 触发采集 DI 状态上报	@DI1=0 @DI1=1 @DI1=2 @DI1=? @DI1=?1 @DI1=?2 @DI1=~ @DI1=~1 @DI1=~2	此指令用于配置使用 DI1~DI4 进行触发采集动作或上报 DI1~DI4 的状态 ?和?1 为上报 DI 的 HEX 格式的状态，0x00 或者 0x01 ?2 为上报 DI 的字符格式的状态，‘0’ 或者 ‘1’ ~和~1 为上报 DI 状态反码的 HEX 格式，0x00 或者 0x01 ~2 为上报 DI 状态反码的字符格式，‘0’ 或者 ‘1’

DI: DI 触发采集，状态上报，此脚本指令用于配置 DTU 使用 DI1~DI4 进行触发采集动作。value 的值可为：?（报告状态）、0（下降沿触发采集）、1（上升沿触发采集）、2（下降沿和上升沿均触发采集）、其他（不触发采集）。默认 DI1 不触发采集。@DI1=0 或 1 或 2 时只进行 DI 触发采集配置，不会产生实际动作，如需要上报 DI1 状态必须在需要的位置插入@DI1=?DTU 才会上报 DI1 的状态。状态为 1 个字节，?和?1 格式时 00 表示低电平，01 表示高电平。?2 格式时 ‘0’ 表示低电平，‘1’ 表示高电平。~和~1 格式时 01 表示低电平，00 表示高电平。~2 格式时 ‘1’ 表示低电平，‘0’ 表示高电平。

注：使用此指令上报时仅仅会上报一个字节的值，用户也可以用@C 指令使用 Modbus 协议控制或读取 DI，这样上报的就是符合 Modbus 协议的 DI 状态。详细信息请参考 Modbus 协议手册。

示例：@DI1=? 采集 DI1 状态

@DI1=0 DI1 下降沿触发采集

@DI1=1 DI1 上升沿触发采集

@DI1=2 DI1 上升沿和下降沿触发采集

@DI1=其他 DI1 不触发采集（默认，可省略），即使不触发采集也可在脚本中插入@I1=?让 DTU 定时采集时报告 DI1 的状态。

2.2.30 @AIx 值上报指令

指令	名称	格式	含义
@AI1 @AI2 @AI3 @AI4 @AI5 @AI6 @AI7 @AI8	AI 值上报	@AI1=? @AI1=?1 @AI1=?2	上报 AI1~AI8 的值 ?和?1 上报 4 字节的 IEEE754 标准格式的浮点数 ?2 上报字符格式的实际值，如电流为 12.3mA 上报的数据为 12.3

AI: AI 值上报，此指令用于上报模拟量 AI1~AI8 的值，只有带有模拟量接口的设备才支持如 ZSR2184。

注：使用此指令上报时仅仅会上报模拟量的值，用户也可以用@C 指令使用 Modbus 协议控制或读取 AI，这样上报的就是符合 Modbus 协议的 AI 值。详细信息请参考 Modbus 协议手册。

示例：@AI1=?@AI2=? 上报 AI1 和 AI2 的值（浮点数格式）。

@STR=AI1=@AI1=?2 如 AI1 实际值为 8.93 毫安，则 DTU 实际上报数据为：AI1=8.93

2.2.31 @IF @ELSE @ENDIF 条件执行指令

指令	名称	格式	含义
@IF @ELSE @ENDIF	条件执行指令	@IF=X @ELSE @ENDIF	使用此指令可以在脚本中根据不同的条件执行不同的脚本，当条件为真时执行@IF 和@ELSE 之间的脚本，然后继续从@ENDIF 后开始执行；当条件为假时执行@ELSE 和@ENDIF 之间的脚本，然后继续从@ENDIF 后开始执行；

		<p>当不需要执行条件为假时的脚本时省略@ELSE,变成@IF=X @ENDIF</p> <p>@IF=X, X 为条件, 支持的条件如下 ACK: 当传感器响应了数据时为真 NACK: 当传感器未响应数据时为真</p> <p>此条件用在@C@CS@O@OS 命令后判断传感器是否响应了脚本发送的命令, 当下一个@C@CS@O@OS 命令出现时, 脚本会自动清除响应标记, 以便脚本中可以出现多个命令是否响应的条件</p> <p>DI: 当脚本是由任一 DI 触发执行时为真 DIU: 当脚本是由任一 DI 上升沿触发执行时为真 DID: 当脚本是由任一 DI 下降沿触发执行时为真 DI1: 当脚本是由 DI1 触发执行时为真 DI1U: 当脚本是由 DI1 上升沿触发执行时为真 DI1D: 当脚本是由 DI1 下降沿触发执行时为真 DI2: 当脚本是由 DI2 触发执行时为真 DI2U: 当脚本是由 DI2 上升沿触发执行时为真 DI2D: 当脚本是由 DI2 下降沿触发执行时为真 DIH: 当此时任一 DI 为高时为真 DIL: 当此时任一 DI 为低时为真 DI1H: 当此时 DI1 为高时为真 DI1L: 当此时 DI1 为低时为真 DI2H: 当此时 DI2 为高时为真 DI2L: 当此时 DI2 为低时为真 DO1H: 当此时 DO1 为高时为真 DO1L: 当此时 DO1 为低时为真 DO2H: 当此时 DO2 为高时为真 DO2L: 当此时 DO2 为低时为真 DO3H: 当此时 DO3 为高时为真 DO3L: 当此时 DO3 为低时为真</p> <p>条件为 DI 沿触发时是指引起脚本执行时 DI 的状态 条件为 DI/DO 高或低时是指脚本执行到当时, 此时此刻 DI/DO 当前的状态。</p> <p>有些具有更多 DI/DO 的型号, 增加相应的数字代表更多的 DI/DO 的条件。</p>
--	--	---

此条件执行指令主要用于用户可以在脚本中可以根据当时不同的条件来执行不同的操作, 让用户应对更复杂的采集需求

示例: @IF=DI1H@CS=123@ELSE@CS=456@ENDIF

当 DI1 为高时, 发送采集命令 123, 否则发送命令 456

```
@CS=123@D=2S@IF=NACK@STR=NODATA@ENDIF
```

发送命令 123，等待 2 秒，假如传感器没响应上报 NODATA，传感器应答了会上报应答的数据

三、应用案例

案例 1:

在众山 DTU 的串口外接 2 个 RS485 接口的仪表，2 个仪表使用 DO1 开关控制电源，打开电源需要 10 秒时间让仪表稳定，仪表为标准的 Modbus 协议，Modbus 地址分别为 01 和 02。每个仪表需要采集 2 个寄存器数据，寄存器地址为 0000 和 0001，仪表 1 的 2 个寄存器采集指令为：01 03 00 00 00 01 和 01 03 00 01 00 01。仪表 2 的 2 个寄存器采集指令为：02 03 00 00 00 01 和 02 03 00 01 00 01。一次采集需要连续采集 3 次数据。仪表 1 的波特率为 4800，仪表 2 的波特率为 38400。

用户需求为每 1 小时采集一次数据，一次数据采集需要采集到 2 个仪表的 2 个寄存器数据，并且需要采集 3 次，上报的数据需要回显采集命令，在每条数据前加入 AA55 以便于用户解析数据，每条采集指令之间延时 3 秒，都需要接收仪表应答。

首先需要设置 DTU 的仪表自动采集周期参数为 3600 秒。

脚本设计如下，和默认值一致的脚本指令可省略

1: 每次采集周期结束立即上报

```
@E=1@T=1@DO1=1@D=10S@M=1@B=4800@H=AA55@C=010300000001V1@D=3S@H=AA55@C=010300010001V1@D=3S@B=38400@H=AA55@C=020300000001V1@D=3S@H=AA55@C=020300010001V1@D=3S@L=1,3@DO1=0
```

注：@T=1 可省略，默认即为 1 次采集周期结束立即上报。

2: 每 3 小时上报一次数据

```
@E=1@T=3@DO1=1@D=10S@M=1@B=4800@H=AA55@C=010300000001V1@D=3S@H=AA55@C=010300010001V1@D=3S@B=38400@H=AA55@C=020300000001V1@D=3S@H=AA55@C=020300010001V1@D=3S@L=1,3@DO1=0
```

注：3 次采集周期结束才上报一次数据，所以 @T=3

3: 每采集到 1 个仪表的 2 个寄存器数据时分开上报

```
@E=1@DO1=1@D=10S@M=1@B=4800@H=AA55@C=010300000001V1@D=3S@H=AA55@C=010300010001V1@D=3S@S=1@B=38400@H=AA55@C=020300000001V1@D=3S@H=AA55@C=020300010001V1@D=3S@S=1@L=1,3@DO1=0
```

注: 采集完一个仪表的 2 个数据后用@S=1 立即上报

4: 每采集到 1 个仪表的 1 个寄存器数据时分开上报

```
@E=1@DO1=1@D=10S@M=1@B=4800@H=AA55@C=010300000001V1@D=3S@S=1@H=AA55@C=010300010001V1@D=3S@S=1@B=38400@H=AA55@C=020300000001V1@D=3S@S=1@H=AA55@C=020300010001V1@D=3S@S=1@L=1,3@DO1=0
```

注: 每采集一个数据后用@S=1 立即上报

案例 2:

使用众山 DTU 每 5 分钟读取一次 DTU 的信号强度并且通过 Modbus 协议上报 DI1 和 DI2 的状态。

首先设置采集周期参数为 300 秒。

脚本设计如下:

1: 信号强度和 DI 状态在一个包上报

```
@C=AA550004E023V2@C=640300100002V1
```

获取信号强度和读取 DI 值的协议请参考控制协议手册和 Modbus 协议手册, 需要注意的是要通过 Modbus 读取 DTU 自己的寄存器时 Modbus 地址必须为 DTU 的 Modbus 地址, 否则 DTU 会透传命令到串口。

注: 此两条命令均为 DTU 响应的命令, 所以不需要插入延时, DTU 会立即响应

2: 信号强度和 DI 状态分 2 次分开发送

```
@C=AA550004E023V2@S=1@C=640300100002V1
```

小提示:

众山DTU提供的脚本自动采集功能和串口透传功能同时存在的，并且共用了一个串口缓存，串口透传采用分包间隔时间参数进行打包发送，脚本使用@S=1或者@T控制分包发送。所以存在当刚好DTU非脚本期间收到了串口数据启动定时器打包发送时，执行脚本的时间到来，此时如果脚本马上执行势必会造成脚本执行期间收到的靠前的数据和以前未打包的串口数据一起被打包提前发送，造成脚本数据上报不完整，如果DTU一旦启动脚本就失效前面的串口数据打包机制也会造成以前的数据和脚本数据连接在一起。针对这种现象，建议用户在脚本启动开始，使用@D插入一个超过20ms的延时(因为DTU默认串口分包间隔时间为20ms)，让脚本执行还未产生数据之前，以前的串口数据得以打包发送，这样就不会出现脚本数据不完整或者脚本期间接收的数据和非脚本期间的透传数据连接在一起。