

ComWinTop CWT_IO protocol

V1.405

一、 Introduction

1. History

2012/08/08: Add ainalog status 3

2012/11/30: Add package Device Registry

2013/02/19: modify counter value dump

2013/04/03: modify counter daily report

2014/9/17: Add package humidity

2015/04/07: add GPS report type char description list

2015/04/19: add wireless sensor events

2015/09/10: add extend humidity sensor package

2016/03/01: modify GSP package

2016/06/03: Add "GP66" package

2017/05/21: Add AD53、AD55、AD56、EH58、EH59、EH60、ET42、ET44、ET49、TM50

2017/9/21: Modify GP35、LB35

2017/10/11: Add RG76、RG78、RG80、RG81、RG82

2018/2/5: Add RD86



2. Data type description

| Type | Description | Remark |
|-----------------------------|--|-----------------------|
| \$%%CMD\$ | Remote command | From server to device |
| \$CMD MSG\$ | Device reply of "Remote command" | |
| \$/D\$ | ID data package | |
| AD10 | Single analog input channel | |
| AD22 | All analog input | |
| AD52 | | not used |
| AD53 | Analog parameters: high value, low value, alarm range lag, physical scale high, physical scale low | |
| AD55 | Single analog name | |
| AD56 | Single analog unit | |
| CK83 | Synchronize time | |
| DC03 | Power supply | |
| DI02 | Single digital input channel | |
| DI20 | All digital input | |
| DI64 | Single digital input name | |
| DO21 | All digital output channel | |
| DO65 | Single digital output name | |
| EH36 | All external humidity | |
| EH57 | Single external humidity | |
| EH58 | External humidity log data | |
| EH59 | External humidity parameters | |
| EH60 | External humidity name | |
| ET33 | All external temperature | |
| ET43 | Single external temperature | |
| ET47 | External temperature log data | |
| ET44 | External temperature parameters | |
| ET42 | External temperature name | |
| ET49 | External temperature unit | |
| GM67 | Device GSM (network) info: IMEI, SIMID, OPERATOR, GPRSIP, MCC, MNC, SIGNAL, GPRSTX, GPRSRX, LIVED | |
| GP35 | GPS info: longitude, latitude, speed | |
| GY18 | | not used |
| HT99 | Heart beat (keep alive) | |
| HU34 | | |
| IC32 | Multi counter | |
| IC46 | | not used |
| IC48 | | not used |
| IC61 | | not used |
| IC62 | | not used |



| | | |
|----------------------|--|-----------------------------|
| IC63 | | not used |
| CT19 | | not used |
| LB35 | Location Based Service: longitude, latitude, speed | |
| LB66 | Network location: MCC, MNC, LAC, CID | |
| RG31 | Single modbus register channel | Register count 32/64 |
| RG41 | Multi modbus register channel | |
| RG76 | Modbus register name | Only register 00-31 |
| RG78 | Modbus register high/low/lag value | Register count 32/64 |
| RG80 | Single modbus register channel | Register count >100 |
| RG81 | Multi modbus register channel | Register count >100 |
| RG82 | Modbus register high/low/lag value | Register count >100 |
| PW00 | Power up package: arm status, power supply, signal, inner temperature, all digital inputs, all digital outputs | |
| RP01 | | |
| RS45 | Device hardware version info: IMIE, SIMID, OPERATOR, PVer, FVer, HdVer, DevTp | |
| SC83 | Server reply the "synchronize time" | From server to device |
| SS70 | Device description info | |
| TM11 | Inner temperature | |
| TM50 | Inner temperature unit | |
| WS01 | | not used |
| DF04 | | not used |
| PC66 | | not used |
| RD86 | Server redirection | From server to device (dtu) |

3. Format specification

Example:

| | | | | | | |
|------|----------|------|----------------|---------|--------------------------|-----|
| \$ | 00000001 | ET43 | 16/06/03 09:54 | 00 | 125.88 | \$ |
| Head | Unit ID | Type | Time | Channel | Temperature status/Value | End |

The fixed characters with **black font**, eg: \$ (Head or End)

Variables with different color font, eg: **125.88** (1 is Temperature status(Alarm), **25.88** is temperature value)

Note: All data package are no have any "Enter" character



二、Type data detail

1 Power Up Package (PW00/ RP01)

| | | | | |
|------|----------|------|----------------|---------|
| \$ | 00000001 | PW00 | 12/10/11 11:12 | 00 |
| | | RP01 | | |
| head | unit ID | type | time | channel |

| | | | | | | |
|------------|--------------|--------|-------------------|--------------|---------------|-----|
| 0 | 0 | 0287 | 31.50 | 0100 | 0010 | \$ |
| 1 | 1 | | | | | |
| arm status | power status | signal | inner temperature | input status | output status | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type (PW00=RP01)

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: Fixed value 00

arm status: 0/1; 0 means disarm status; 1 means arm status

power status: 0/1; 0 means powered by DC; 1 means powered by battery

signal: 4 numbers, value minus 256 is the real signal (eg: 0287-256=31) , signal range is 0-31

inner temperature: two decimal value (eg: 30.15)

input status: form left to right, each number represents status of each DI channel (DI0-DIx), 0 means normal status, 1 means alarm status

eg: 0100, means DI0, DI2, DI3 in normal status, DI1 in alarm status

output status: deprecation. all outputs status (from left to right), one output with one number (0 means open, 1 means close), so length may not count 4

form left to right, each number represents status of each DO channel (DO0-DOx), 0 means open, 1 means close

eg: 0010, means DO0, DO1, DO3 is opening, DO2 is closing

Note: this package is sent when power up or by timer

Example: \$00000001RP0112/10/11 11:120010028731.5001000010\$

(the device ID is 1 (00000001),arm status, DC powered, signal is 31 (0287-256), inner temperature is 31.50, input and ouput status is 01000010)

2 Analog input

2.1 Single analog input (AD10)

| | | | | | | |
|------|----------|------|----------------|---------|--------------|-----|
| \$ | 00000001 | AD10 | 16/06/03 09:54 | 00 | 129.28 | \$ |
| head | unit id | type | time | channel | status value | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, analog input index, start with 00

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: analog value (float, eg: 29.28)

end: Fixed format (\$)

Note: this package is sent when analog input alarm/recover or "upload step" is triggered

Example: \$00000001AD1016/06/03 09:5401129.28\$ (Analog input 01 alarm, value is 29.28)

2.2 All analog input(AD22)

| | | | | | | |
|------|----------|------|----------------|---------|------------------------------|-----|
| \$ | 00000001 | AD22 | 16/06/03 09:54 | 04 | 129.28,139.28,049.28,059.28, | \$ |
| head | unit id | type | time | channel | status value, ... | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, all analog input count (eg: 04 means there are 4 analog inputs)

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: analog value (float, eg: 29.28)

Each analog input is separated by commas (eg: 129.28,139.28,049.28,059.28,)

end: Fixed format (\$)

Note: this package is sent when power up or by timer

Example: \$00000001AD2212/05/12 09:5404129.28,139.28,049.28,059.28,\$

(4 analog inputs: AI0 value is 29.28, alarm status; AI1 value is 39.28, alarm status; AI2 value is 49.28, normal status; AI3 value is 59.28, normal status)

2.3 Analog input parameter(AD53)

| | | | | | | |
|------|----------|------|----------------|---------|--------------------------|-----|
| \$ | 00000001 | AD53 | 16/06/03 09:54 | 00 | 2.00,1.00,3.00,4.00,5.00 | \$ |
| head | unit id | type | time | channel | parameters | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, analog input index, start with 00

parameters: high value, low value, alarm range lag, physical scale high, physical scale low;

separated by comma

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example: \$00000001AD5317/05/19 09:52002.00,1.00,5.00,7.00,6.00\$

(Analog input 00: high value is 2, low value is 1, alarm range lag is 5, physical scale high is 7, physical scale low is 6)

2.4 Single analog name(AD55)

| | | | | | | | |
|----|----------|------|----------------|----|------|------|----|
| \$ | 00000001 | AD55 | 16/06/03 09:54 | 00 | ASC: | Ain0 | \$ |
|----|----------|------|----------------|----|------|------|----|

| head | unit id | type | time | channel | character type | name | end |
|------|---------|------|------|---------|----------------|------|-----|
|------|---------|------|------|---------|----------------|------|-----|

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, analog input index, start with 00

character type: "ASC:" means "name" is ASCII code

name: analog input name

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example: \$00000002AD5517/05/19 14:4401ASC:Voltage\$

(Analog input 01 name is "Voltage")

2.5 Single analog input unit (AD56)

| \$ | 00000001 | AD56 | 16/06/03 09:54 | 00 | ASC: | V | \$ |
|------|----------|------|----------------|---------|----------------|------|-----|
| head | unit id | type | time | channel | character type | unit | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, analog input index, start with 00

character type: "ASC:" means "unit" is ASCII code

unit: analog input unit

end: Fixed format (\$)



Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example: \$00000001AD5617/05/19 16:2100ASC:V\$

(Analog input unit is "V")

3 Digital input and output

3.1 Single digital input (DI02)

| | | | | | | |
|------|----------|------|----------------|---------|--------|-----|
| \$ | 00000001 | DI02 | 16/06/03 09:54 | 00 | 0 | \$ |
| | | | | | 1 | |
| head | unit id | type | time | channel | status | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, analog input index, start with 00

status: 0/1, 0 means normal; 1 means alarm

end: Fixed format (\$)

Note: this package is sent when digital input alarm/recover

Example: \$00000001DI0216/06/03 09:54001\$ (Digital input 0 is alarm)

3.2 All digital inputs (DI20)

| | | | | | | |
|------|----------|------|----------------|---------|-----------|-----|
| \$ | 00000001 | DI20 | 16/06/03 09:54 | 04 | 1110 | \$ |
| head | unit id | type | time | channel | status... | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type



time: 14 lengths time format “YY/MM/DD HH:MM” (year/month/day hour:minute),there is a space between “DD HH”

channel: 2 lengths, all digital input count (eg: 04 means there are 4 digital inputs)

status: length is equal to count (channel), each one is 0/1; 0 means normal, 1 means alarm

end: Fixed format (\$)

Note: this package is sent when power up or by timer

Example: \$00000001DI2016/06/03 09:54041110\$

(4 digital inputs; DI0, DI1,DI2 are all alarm; DI3 is normal)

3.3 Single digital input name (DI64)

| | | | | | | | |
|------|----------|------|----------------|---------|----------------|-------------|-----|
| \$ | 00000001 | DI64 | 16/06/03 09:54 | 01 | ASC: | Overvoltage | \$ |
| head | unit id | type | time | channel | character type | name | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format “YY/MM/DD HH:MM” (year/month/day hour:minute),there is a space between “DD HH”

channel: 2 lengths, digital input index, start with 00

character type: “ASC:” means “name” is ASCII code

name: digital input name

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software “Data Transimission->Protocols->Sync settings on power-up”)

Example: \$00000001DI6417/05/19 14:4401ASC:Overvoltage\$

(Digital input 1 name is “Overvoltage”)



3.4 All digital output (DO21)

| | | | | | | |
|------|----------|------|----------------|---------|-----------|-----|
| \$ | 00000001 | DO21 | 16/06/03 09:54 | 04 | 1110 | \$ |
| head | unit id | type | time | channel | status... | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute), there is a space between "DD HH"

channel: 2 lengths, all digital output count (eg: 04 means there are 4 digital outputs)

status: length is equal to count (channel), each one is 0/1; 0 means off, 1 means on

end: Fixed format (\$)

Note: this package is sent when power up or by timer or digital output status change

Example: \$00000001DO2116/06/03 09:54041110\$

(4 digital outputs; DO0, DO1, DO2 are all on, DO3 is off)

3.5 Single digital output name (DO65)

| | | | | | | | |
|------|----------|------|----------------|---------|----------------|------|-----|
| \$ | 00000001 | DO65 | 16/06/03 09:54 | 01 | ASC: | Door | \$ |
| head | unit id | type | time | channel | character type | name | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute), there is a space between "DD HH"

channel: 2 lengths, digital output index, start with 00

character type: "ASC:" means "name" is ASCII code

name: digital output name

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example: \$00000001DO6517/05/19 14:4500ASC:Door\$

(Digital output 0 name is "Door")

4 GPS information (GP35)

| | | | | | |
|------|----------|------|----------------|---------------|-------|
| \$ | 00000001 | GP35 | 16/06/03 09:54 | 00 | : |
| head | unit id | type | time | location type | colon |

| | | | |
|---------------|--------------|-------|-----|
| E 113.921567, | N 22.574423, | 20.88 | \$ |
| longitude | latitude | speed | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

location type: 00/01, 00 means GOOGLE_EARTH location, 01 means GOOGLE_MAP location

colon: Fixed format (:)

longitude: E/W and one space and number (W is negative) and comma (,)

eg: W -100.123456, (West longitude 100.123456); E 23.123456, (East longitude 23.123456)

latitude: N/S and one space and number (S is negative) and comma (,)

eg: S -22.654321, (South latitude 22.654321) , N 19.654321, (North latitude 19.654321)

speed: float value, unit of km/h

end: Fixed format (\$)

Note: this package is sent when power up or by timer (If the device support GPS)

Example: \$10000002GP3516/06/03 09:5400:E 113.921567,N 22.574423,0.23\$

(GPS: East longitude 113.921567, North latitude 22.574423, speed is 0.23 km/h)

\$10000001GP3516/06/03 09:5500:W -70.745167,S -33.395057,0.41\$



(GPS: West longitude 70.745167, South latitude 33.395057, speed is 0.41km/h)

5 Location Based Service (LB35)

| | | | | | |
|------|----------|------|----------------|---------------|-------|
| \$ | 00000001 | LB35 | 16/06/03 09:54 | 01 | : |
| head | unit id | type | time | location type | colon |

| | | | |
|-------------|-------------|-------|-----|
| 113.234567, | -23.123456, | 20.88 | \$ |
| longitude | latitude | speed | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute), there is a space between "DD HH"

location type: 00/01, 00 means GOOGLE_EARTH location, 01 means GOOGLE_MAP location

colon: Fixed format (:)

longitude: two space and number (W is negative) and comma (,)

eg: -100.123456, (West longitude 100.123456); 23.123456, (East longitude 23.123456)

latitude: two space and number (S is negative) and comma (,)

eg: -22.654321, (South latitude 22.654321), 19.654321, (North latitude 19.654321)

speed: float value, unit of km/h

end: Fixed format (\$)

Note: this package is sent when power up or by timer (If the device support LBS)

Example: \$10000010LB3516/06/23 04:0601: 113.918167, 22.569696,0.00\$

(LBS: East longitude 113.918167, North latitude 22.569696, speed is 0.00 km/h)

\$10000007LB3516/10/11 09:5301: -62.563354, -19.339565,0.00\$

(LBS: West longitude 62.563354, South latitude 19.339565, speed is 0.00 km/h)



6 Inner temperature

6.1 Inner temperature (TM11)

| | | | | | | |
|------|----------|------|----------------|---------|--------------|-----|
| \$ | 00000001 | TM11 | 16/06/03 09:54 | 00 | 129.28 | \$ |
| head | unit id | type | time | channel | status value | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: Fixed value (00)

status|value:

status: 0/1, 0 means normal; 1 means alarm; 2 means disconnect

value: temperature value (float, eg: 29.28)

end: Fixed format (\$)

Note: this package is sent when power up or by timer or alarm/recover

Example: \$00000001TM1116/06/03 09:5400133.80\$

(Inner temperature alarm, value is 33.80)

6.2 Inner temperature unit (TM50)

| | | | | | | |
|------|----------|------|----------------|---------|--|-----|
| \$ | 00000001 | TM50 | 16/06/03 09:54 | 00 | | \$ |
| head | unit id | type | time | channel | | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: Fixed value (00)

character type: "ASC:" means "unit" is ASCII code



unit: Inner temperature unit

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example:

7 Modbus register

7.1 Single register channel(RG31)

| | | | | | |
|------|----------|------|----------------|-------------------------|-----|
| \$ | 00000001 | RG31 | 16/06/03 09:54 | 00C01 | \$ |
| | | | | 00R019.88 | |
| head | unit ID | type | time | index type status value | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

index|type|status|value:

index: two numbers, it is register index, start with 00

type: R/C, R means Keep or Input register; C means Coil or Instate register

status: 0/1/2, 0 means normal; 1 means alarm; 2 means disconnect

value: if **type** is R, value is float (eg:19.88); if **type** is C, value only have 0 or 1

end: Fixed format (\$)

Note: this package is sent when register alarm/recover or "upload step" is triggered

Example: \$10000002RG3116/06/03 09:5400C11\$

(Register 00 is Coil/Instate type, alarm status, value is 1)

7.2 Multi modbus register (RG41)

| | | | | | | |
|----|----------|------|----------------|----|---------------------|----|
| \$ | 00000001 | RG41 | 12/05/13 09:54 | 12 | 00R019.88,01C11,... | \$ |
|----|----------|------|----------------|----|---------------------|----|

| | | | | | | | | | |
|------|---------|------|------|-------|-------|------|--------|-------|-----|
| head | unit ID | type | time | count | index | type | status | value | end |
|------|---------|------|------|-------|-------|------|--------|-------|-----|

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

count: two numbers, eg 12 means this package have 12 register values

index|type|status|value:

index: 2 numbers, it is register index, start with 00

type: R/C, R means Keep or Input register; C means Coil or Instate register

status: 0/1/2, 0 means normal; 1 means alarm; 2 means disconnect

value: if **type** is R, value is float (eg:19.88); if **type** is C, value only have 0 or 1

End with comma (,) of every register value

end: Fixed format (\$)

Note: this package is sent when power up or by timer

Example: \$00000001RG4116/06/03

09:511200R019.88,01C11,02R20.00,03R20.00,04R20.00,05R20.00,06R20.00,07R20.00,08R20.00,09R20.00,10R20.00,11R20.00,\$

(12 registers: register 00 is Keep or Input type, normal status, value is 19.88; register 01 is Coil or Instate type, alarm status, value is 1, other registers are disconnected)

7.3 Register name (RG76)

| | | | | | | | |
|------|----------|------|----------------|---------|-------|-------------|-----|
| \$ | 00000001 | RG76 | 16/06/03 09:54 | 00 | ASC: | Temperature | \$ |
| head | unit ID | type | time | channel | ASCII | name | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, register index, start with 00

ASCII: name is ASCII code (english)

name: register name



end: Fixed format (\$)

Note: this data is sent when power up (If "Sync settings on power-up" is checked at "Data Transmission-Protocols" in the config tool)

Example: \$00000000RG7616/06/03 09:5400ASC:Voltage\$

(Register 00 name is Voltate)

7.4 Register parameters (RG78)

| | | | | | | |
|------|----------|------|----------------|---------|----------------------------|-----|
| \$ | 00000001 | RG78 | 16/06/03 09:54 | 00 | 25.00,10.00,2.00,0.00,0.00 | \$ |
| head | unit ID | type | time | channel | parameters | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, register index, start with 00

parameters: Five values, from left to right: high value, low value, alarmlag, the highest range (unused), the lowest range (unused), they are separated by commas

end: Fixed format (\$)

Note: Device start will send this data (If "Sync settings on power-up" is checked at "Data Transmission-Protocols" in the config tool)

Example: \$10000006RG7816/06/03 09:540130.00,17.00,2.00,0.00,0.00\$

(Register 01 high value is 30, low value is 17, alarm lag is 2)

7.5 Single register channel-3 lengths index (RG80)

| | | | | | |
|------|----------|------|----------------|-------------------------|-----|
| \$ | 00000001 | RG80 | 17/10/11 09:54 | 000C01 | \$ |
| | | | | 000R019.88 | |
| head | unit ID | type | time | index type status value | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type



time: 14 lengths time format “YY/MM/DD HH:MM” (year/month/day hour:minute),there is a space between “DD HH”

index|type|status|value:

index: 3 numbers, it is register index, start with 000

type: R/C, R means Keep or Input register; C means Coil or Instate register

status: 0/1/2, 0 means normal; 1 means alarm; 2 means disconnect

value: if **type** is R, value is float (eg:19.88); if **type** is C, value only have 0 or 1

end: Fixed format (\$)

Note: this package is sent when register alarm/recover or “upload step” is triggered

Example 1: \$00000001RG8017/10/11 09:53000C11\$

(Register 000 is Coil/Instate type, alarm status, value is 1)

Example 2: \$00000001RG8017/10/11 09:53001R03.00\$

(Register 001 is Keep/Input type, normal status, value is 3.00)

7.6 Multi modbus register-3 lengths index(RG81)

| | | | | | | |
|------|----------|------|----------------|-------|-------------------------|-----|
| \$ | 00000001 | RG81 | 17/10/11 09:54 | 12 | 000R019.88,001C11,... | \$ |
| head | unit ID | type | time | count | index type status value | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format “YY/MM/DD HH:MM” (year/month/day hour:minute),there is a space between “DD HH”

count: two numbers, eg 12 means this package have 12 register values

index|type|status|value:

index: 3 numbers, it is register index, start with 000

type: R/C, R means Keep or Input register; C means Coil or Instate register

status: 0/1/2, 0 means normal; 1 means alarm; 2 means disconnect

value: if **type** is R, value is float (eg:19.88); if **type** is C, value only have 0 or 1

End with comma (,) of every register value

end: Fixed format (\$)

Note: this data is sent when power up or by timer



Example: \$00000001RG8117/10/11

09:5112000C00,001R01.00,002R20.00,003R20.00,004R20.00,005R20.00,006R20.00,007R20.00,008R20.00,009R20.00,010R20.00,011R20.00,\$

(12 registers: register 000 is Coil or Instate type, normal status, value is 0; register 001 is Keep or Input type, normal status, value is 1.00; register 002-011 are all disconnected)

7.7 Register parameters-3 lengths index (RG82)

| | | | | | | |
|------|----------|------|----------------|---------|----------------------------|-----|
| \$ | 00000001 | RG82 | 16/06/03 09:54 | 000 | 25.00,10.00,2.00,0.00,0.00 | \$ |
| head | unit ID | type | time | channel | parameters | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 3 lengths, register index, start with 000

parameters: Five values, from left to right, from left to right: high value, low value, alarmlag, the highest range (unused), the lowest range (unused), they are separated by commas

end: Fixed format (\$)

Note: this data is sent when power up (If "Sync settings on power-up" is checked at "Data Transmission-Protocols" in the config tool)

Example: \$10000006RG8216/06/03 09:5400130.00,17.00,2.00,0.00,0.00\$

(Register 001 high value is 30, low value is 17, alarm lag is 2)

8 Heart (HT99)

| | | | | | |
|------|----------|------|----------------|--------------|-----|
| \$ | 00000001 | HT99 | 03/15/12 11:12 | Heart Packet | \$ |
| head | unit ID | type | time | content | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type



time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

content: heart content (nullable)

end: Fixed format (\$)

Note: Only version 2.51 or later can send this data (If "Heart timer (sec)" is not 0 at "Data Transmission-GPRS Setup" in the config tool)

Example: \$00000001HT9916/06/03 16:16999\$ (heart content "999")

9 Remote control

Remote control data package (From server to device)

| | | |
|------|-------------|-----|
| \$%% | IOOH0 | \$ |
| | IOOH0#IOOH1 | |
| head | command | end |

head: Fixed format (\$%%)

command: device command which send from server to device, used for set parameter or control device. Multiple command split with #

end: Fixed format (\$)

Note: device must connected to server by TCP

Example: \$%%IOOH0#IOOH1\$ (server send IOOH0#IOOH1 to device control output close)

device reply server command

| | | | | | | |
|------|----------|-------------------|----------------------|----------------|--------------------------|-----|
| \$ | 00000001 | ,CMD_RESULT: 0 | ,CMD_MSG:" %CMD " | ,CMD_ECHO : | ,TIME:YY/MM/D D HH:MM | \$ |
| | | ,CMD_RESULT: 1 | | "REPLY" | | |
| head | unit ID | result code | command | query reply | time | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

result code: ,CMD_RESULT:0 means command execute error; ,CMD_RESULT:1 means command execute success

command: foramt is ,CMD_MSG:" %CMD" , the CMD is command which server send to device



query reply: only query command have this content, format is ,**CMD_ECHO:" REPLY"** , the **REPLY** is device reply the the server command (maybe not intact)

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

end: Fixed format (\$)

Example: \$00000001,CMD_RESULT:1,CMD_MSG:"%IOOH0#IOOH1",TIME:16/06/03 16:47\$

(device reply server command (\$%%IOOH0#IOOH1\$), command execute success)

\$00000001,CMD_RESULT:1,CMD_MSG:"%IOIS",CMD_ECHO:"input status:

00",TIME:16/11/16 11:07\$

(device reply server command (\$%%IOIS\$), reply content is input status: 00

10 Power supply (DC03)

| | | | | | | |
|------|----------|------|----------------|---------|--------|-----|
| \$ | 00000001 | DC03 | 03/15/12 11:12 | 00 | 0 1 | \$ |
| head | unit ID | type | time | channel | status | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: Fixed value 00

status: 0/1; 0 means DC powered, 1 means battery powered (DC is not powered)

end: Fixed format (\$)

Note: Device start, or status change will send this data

Example: \$10000010DC0316/06/23 04:08001\$ (battery powered)

11 Device description info (SS70)

| | | | | | | |
|------|----------|------|----------------|----------------|---------------|-----|
| \$ | 00000001 | SS70 | 03/15/12 11:12 | MEMOASC: | Computer room | \$ |
| head | unit id | type | time | character type | description | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format “YY/MM/DD HH:MM” (year/month/day hour:minute),there is a space between “DD HH”

character type: “MEMOASC:” means “**description**” is ASCII code

description: device description information

end: Fixed format (\$)

Note: this package is sent when power up

Example: \$00000002SS7016/10/27 13:51MEMOASC:Computer room\$
 (Device description information is “Computer room”)

12 External humidity

12.1 Single external humidity (EH57)

| | | | | | | |
|------|----------|------|----------------|---------|--------------|-----|
| \$ | 00000001 | EH57 | 16/06/03 09:54 | 00 | 125.88 | \$ |
| head | unit id | type | time | channel | status value | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format “YY/MM/DD HH:MM” (year/month/day hour:minute),there is a space between “DD HH”

channel: 2 lengths, external humidity index, start with 00

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: humidity value (float, eg: 45.28)

end: Fixed format (\$)

Note: this package is sent when humidity alarm/recover or “upload step” is triggered

Example: \$00000001EH5716/06/03 09:5400150.90\$ (humidity 00 alarm, value is 50.90)



12.2 All external humidity (EH36)

| | | | | | | |
|------|----------|------|----------------|---------|------------------|-----|
| \$ | 00000001 | EH36 | 16/06/03 09:54 | 02 | 125.88,115.82 | \$ |
| head | unit id | type | time | channel | status value,... | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, all external humidity count (eg: 04 means there are 4 external humidity)

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: humidity value (float, eg: 49.28)

Each humidity is separated by commas (eg: 129.28,139.28,049.28,059.28)

end: Fixed format (\$)

Note: this package is sent when power up or by timer

Example: \$00000001EH3616/06/03 09:1302145.30,20.00\$

(2 humidity: humidity 0 value is 45.30, alarm status; humidity 1 is disconnected)

12.3 External humidity log data (EH58)

| | | | | | | |
|------|----------|------|----------------|-------------------|---------|--------------|
| \$ | 00000001 | EH58 | 16/06/03 09:54 | 16/06/03 08:55:33 | 00 | 155.88,... |
| head | unit id | type | upload time | record time | channel | status value |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

upload time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

record time: 17 lengths time format "YY/MM/DD HH:MM:SS" (year/month/day hour:minute:second),there is a space between "DD HH"

channel: 2 lengths, upload humidity count (eg: 02 means upload 2 humidity data)

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: humidity value (float, eg: 49.28)

Each humidity is separated by commas (eg: 129.28,139.28,049.28,059.28)

end: Fixed format (\$)

Note: this package is sent by timer

Example: \$00000001EH5817/05/20 10:2917/05/20 10:28:540220.00,155.75\$

(2 log humidity: humidity 0 is disconnected; humidity 1 is alarm status, value is 55.75)

12.4 External humidity parameters (EH59)

| | | | | | | |
|------|----------|------|----------------|---------|-------------------------------|-----|
| \$ | 00000001 | EH59 | 16/06/03 09:54 | 00 | 2.00,1.00,4.00,200.00,-200.00 | \$ |
| head | unit id | type | time | channel | parameters | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, external humidity index, start with 00

parameters: Five values, from left to right: high value, low value, alarmlag, the highest range (unused), the lowest range (unused), they are separated by commas

end: Fixed format (\$)

Note: Device start will send this data (If "Sync settings on power-up" is checked at "Data Transmission-Protocols" in the config tool)

Example: \$00000001EH5917/05/22 08:580055.00,30.00,4.00,100.00,0.00\$

(External humidity 0 high value is 55, low value is 30, alarmlag is 4, the highest range is 100,

the lowest range is 0)

12.5 External humidity name (EH60)

| | | | | | | | |
|------|----------|------|----------------|---------|----------------|------------|-----|
| \$ | 00000001 | EH60 | 16/06/03 09:54 | 00 | ASC: | humidity x | \$ |
| head | unit id | type | time | channel | character type | name | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, external humidity index, start with 00

character type: "ASC:" means "name" is ASCII code

name: external humidity name

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example: \$00000001EH6017/05/22 08:5800ASC:Room humidity\$

(External humidity 0 name is "Room humidity")

13 External temperature

13.1 Single external temperature (ET43)

| | | | | | | |
|------|----------|------|----------------|---------|--------------|-----|
| \$ | 00000001 | ET43 | 16/06/03 09:54 | 00 | 125.88 | \$ |
| head | unit id | type | time | channel | status value | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, external temperature index, start with 00

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: temperature value (float, eg: 25.88)

end: Fixed format (\$)



Note: this package is sent when temperature alarm/recover or "upload step" is triggered

Example: \$00000001ET4316/06/03 09:5400125.88\$ (temperature 0 alarm, value is 25.88)

13.2 All external temperature (ET33)

| | | | | | | |
|------|----------|------|----------------|---------|-------------------|-----|
| \$ | 00000001 | ET33 | 16/06/03 09:54 | 02 | 125.88,115.82 | \$ |
| head | unit id | type | time | channel | status value, ... | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute), there is a space between "DD HH"

channel: 2 lengths, all external temperature count (eg: 04 means there are 4 external temperature)

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: temperature value (float, eg: 49.28)

Each temperature is separated by commas (eg: 129.28,139.28,049.28,059.28)

end: Fixed format (\$)

Note: this package is sent when power up or by timer

Example: \$00000001ET3316/06/03 09:1302129.69,20.00\$

(2 temperature: temperature 0 value is 29.69, alarm status; temperature 1 is disconnected)

13.3 External temperature log data (ET47)

| | | | | | | |
|------|----------|------|----------------|-------------------|---------|--------------|
| \$ | 00000001 | ET47 | 16/06/03 09:54 | 16/06/03 08:55:33 | 02 | 125.88,... |
| head | unit id | type | upload time | record time | channel | status value |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

upload time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

record time: 17 lengths time format "YY/MM/DD HH:MM:SS" (year/month/day hour:minute:second),there is a space between "DD HH"

channel: 2 lengths, upload temperature count (eg: 02 means upload 2 temperature data)

status|value:

status: 0/1/2/3, 0 means normal; 1 means alarm; 2 means disconnect; 3 means out of range

value: temperature value (float, eg: 49.28)

Each temperature is separated by commas (eg: 129.28,139.28,049.28,059.28)

end: Fixed format (\$)

Note: this package is sent by timer

Example: \$00000001ET4717/05/20 10:2917/05/20 10:28:540220.00,024.75\$

(2 log temperature: temperature 0 is disconnected; temperature 1 is normal status, value is 24.75)

13.4 External temperature parameters (ET44)

| | | | | | | |
|------|----------|------|----------------|---------|-------------------------------|-----|
| \$ | 00000001 | ET44 | 16/06/03 09:54 | 00 | 2.00,1.00,4.00,200.00,-200.00 | \$ |
| head | unit id | type | time | channel | parameters | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, external temperature index, start with 00

parameters: Five values, from left to right: high value, low value, alarmlag, the highest range (unused), the lowest range (unused), they are separated by commas

end: Fixed format (\$)



Note: Device start will send this data (If "Sync settings on power-up" is checked at "Data Transmission-Protocols" in the config tool)

Example: \$00000001ET4417/05/22 08:580030.00,17.00,2.00,80.00,-45.00\$

(External temperature 0 high value is 30, low value is 17, alarmlag is 2, the highest range is 80, the lowest range is -45)

13.5 External temperature name (ET42)

| | | | | | | | |
|------|----------|------|----------------|---------|----------------|---------------|-----|
| \$ | 00000001 | ET42 | 16/06/03 09:54 | 00 | ASC: | temperature x | \$ |
| head | unit id | type | time | channel | character type | name | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute),there is a space between "DD HH"

channel: 2 lengths, external temperature index, start with 00

character type: "ASC:" means "name" is ASCII code

name: external temperature name

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example: \$00000001ET4217/05/22 08:5800ASC:Room temperature\$

(External temperature 0 name is "Room temperature")

13.6 External temperature unit (ET49)

| | | | | | | | |
|------|----------|------|----------------|---------|----------------|------|-----|
| \$ | 00000001 | ET49 | 16/06/03 09:54 | 00 | ASC: | F | \$ |
| head | unit id | type | time | channel | character type | unit | end |

head: Fixed format (\$)

unit ID: 8 character string (Letter or number)

type: 2 letters and 2 numbers, distinguish different data type

time: 14 lengths time format "YY/MM/DD HH:MM" (year/month/day hour:minute), there is a space between "DD HH"

channel: 2 lengths, external temperature index, start with 00

character type: "ASC:" means "unit" is ASCII code

unit: external temperature unit

end: Fixed format (\$)

Note: this package is sent when power up (It is not sent by default after version 2.90, It can be configured in software "Data Transimission->Protocols->Sync settings on power-up")

Example: \$00000001ET4917/05/22 08:5800ASC:F\$

(External temperature 0 unit is "F" (Fahrenheit))

