

## USR-GPRS232-7S3 User Manual

File Version: V1.0



## Contents

1. Quick Start.....	5
1.1. Testing environment.....	5
1.2. Transmission.....	5
2. Product Introduction.....	7
2.1. Short Description.....	7
2.2. Features.....	7
2.3. Parameters.....	8
2.4. Hardware.....	9
2.5. Dimensions.....	9
3. Function.....	10
3.1. Setup Process.....	10
3.2. Work mode.....	11
3.2.1. Network Transmission Mode.....	11
3.2.1.1. Description.....	11
3.2.1.2. Setup Steps.....	12
3.2.2. HTTPD mode.....	12
3.2.2.1. Description.....	12
3.2.2.2. Setup steps.....	12
3.2.3. SMS Transmission Mode.....	14
3.2.3.1. Description.....	14
3.2.3.2. Setup steps.....	14
3.3. Register Package.....	14
3.3.1. Description.....	14
3.3.2. Setup steps.....	14
3.4. Heart Beat Package.....	15
3.4.1. Description.....	15
3.4.2. Setup steps.....	15
3.5. USR CLOUD.....	16
3.5.1. Description.....	16
3.5.2. Setup Steps.....	16
3.6. UART Function.....	16
3.6.1. RS485.....	16
3.6.2. Similar RFC2217.....	16
3.7. Indicator Status.....	17
4. Parameters Setting.....	19
4.1. AT commands Set.....	19
4.1.1. AT command Mode.....	19
4.1.2. Serial AT command.....	20
4.1.3. Network AT command.....	20
4.1.4. SMS AT Command.....	21
4.2. AT Command Set.....	22
4.2.1. AT Command Set.....	24
4.2.1.1. AT+H.....	25

4.2.1.2. AT+Z.....	26
4.2.1.3. AT+E.....	26
4.2.1.4. AT+ENTM.....	26
4.2.1.5. AT+WKMOD.....	26
4.2.1.6. AT+CALEN.....	27
4.2.1.7. AT+NATEN.....	27
4.2.1.8. AT+UATEN.....	28
4.2.1.9. AT+CMDPW.....	28
4.2.1.10. AT+CACHEN.....	28
4.2.1.11. AT+STMSG.....	29
4.2.1.12. AT+RSTIM.....	29
4.2.1.13. AT+S.....	30
4.2.1.14. AT+RELD.....	30
4.2.1.15. AT+CLEAR.....	30
4.2.1.16. AT+CFGTF.....	30
4.2.1.17. AT+VER.....	31
4.2.1.18. AT+SN.....	31
4.2.1.19. AT+ICCID.....	31
4.2.1.20. AT+IMEI.....	31
4.2.1.21. AT+CNUM.....	32
4.2.1.22. AT+UART.....	32
4.2.1.23. AT+RFCEN.....	33
4.2.1.24. AT+APN.....	33
4.2.1.25. AT+SOCKA.....	33
4.2.1.26. AT+SOCKB.....	34
4.2.1.27. AT+SOCKAEN.....	34
4.2.1.28. AT+SOCKBEN.....	35
4.2.1.29. AT+SOCKASL.....	35
4.2.1.30. AT+SOCKBSL.....	36
4.2.1.31. AT+SOCKALK.....	36
4.2.1.32. AT+SOCKBLK.....	36
4.2.1.33. AT+SOCKIDEN.....	37
4.2.1.34. AT+REGEN.....	37
4.2.1.35. AT+REGTP.....	38
4.2.1.36. AT+REGID.....	38
4.2.1.37. AT+REGDT.....	38
4.2.1.38. AT+REGSND.....	39
4.2.1.39. AT+HEARTEN.....	39
4.2.1.40. AT+HEARTDT.....	40
4.2.1.41. AT+HEARTTP.....	40
4.2.1.42. AT+HEARTTM.....	41
4.2.1.43. AT+HTPTP.....	41
4.2.1.44. AT+HTPURL.....	41
4.2.1.45. AT+HTPSV.....	42

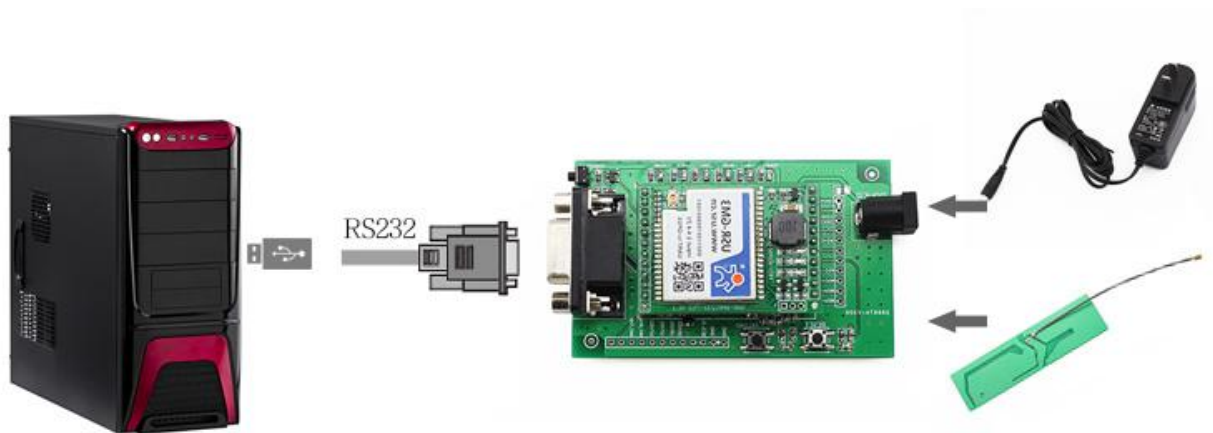
4.2.1.46. AT+HTPHD.....	42
4.2.1.47. AT+DSTNUM.....	43
4.2.1.48. AT+SMSSEND.....	43
4.2.1.49. AT+CLOUDEN.....	43
4.2.1.50. AT+CLOUDID.....	44
4.2.1.51. AT+CLOUDPA.....	44
5. Contact Us.....	45
6. Disclaimer.....	45
7. Update History.....	45

## 1. Quick Start

This chapter is a quick start guide for using USR-7S3 module. If it is the first time to use, you should read this chapter carefully, and then follow the tips, try to do some tests by yourself. It will help you to know USR-7S3 generally.

All the software used in this document can be obtained from our website [www.usriot.com](http://www.usriot.com).  
For any technical questions, please submit to customer support center: <http://h.usriot.com>.

### 1.1. Testing environment



Hardware connection see above diagram.

<Note>

- If it is the first time to use our module for you, please use our evaluation kit , it may save you lot of trouble and time.
- keep default setting in this test.
- Put SIM card into the socket of module.

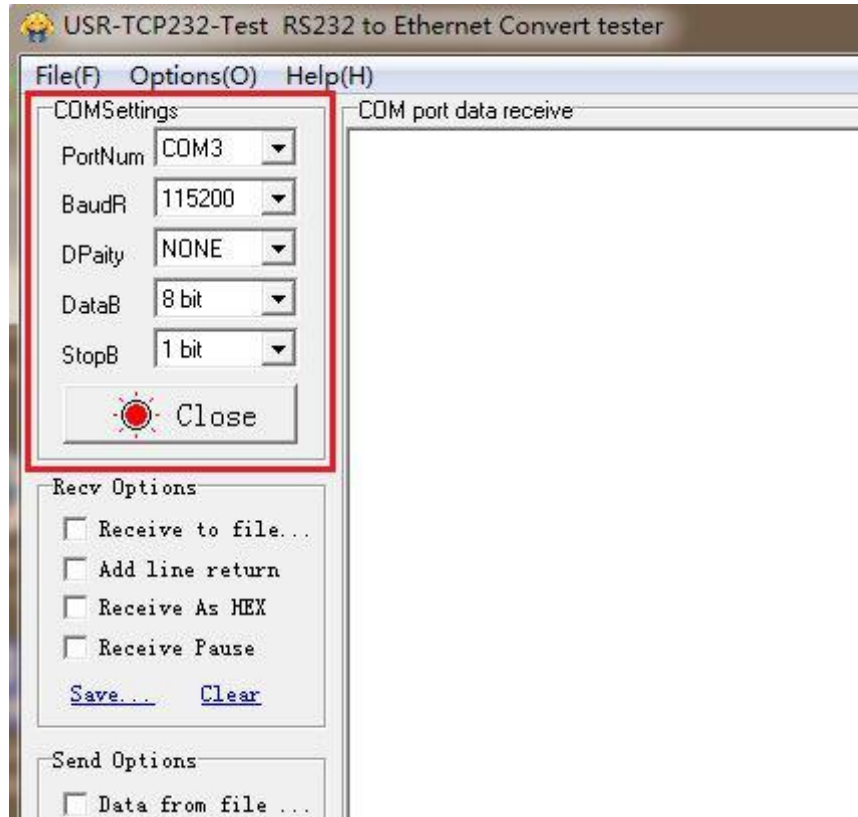
### 1.2. Transmission

#### Module default settings:

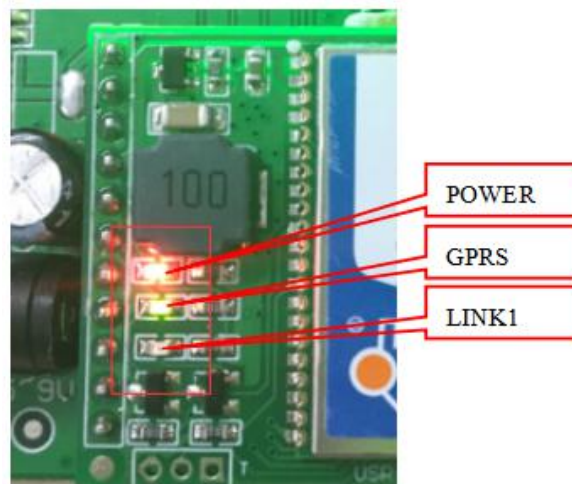
- Work mode: data transparent transmission mode
- Server Address: test.usr.cn Port: 2317 Type: TCP long connection
- UART setting parameters: 115200, 8, 1 None
- Enable heartbeat packet: www.usr.cn, time: 30s

**Test Steps:**

1. Open setup software, select the UART parameters, such as serial port number, baud rate and open serial port, see below image:



2. Power on, then the POWER led will be on, several seconds later, the GPRS led will be on, and then LINK1 led will be on.



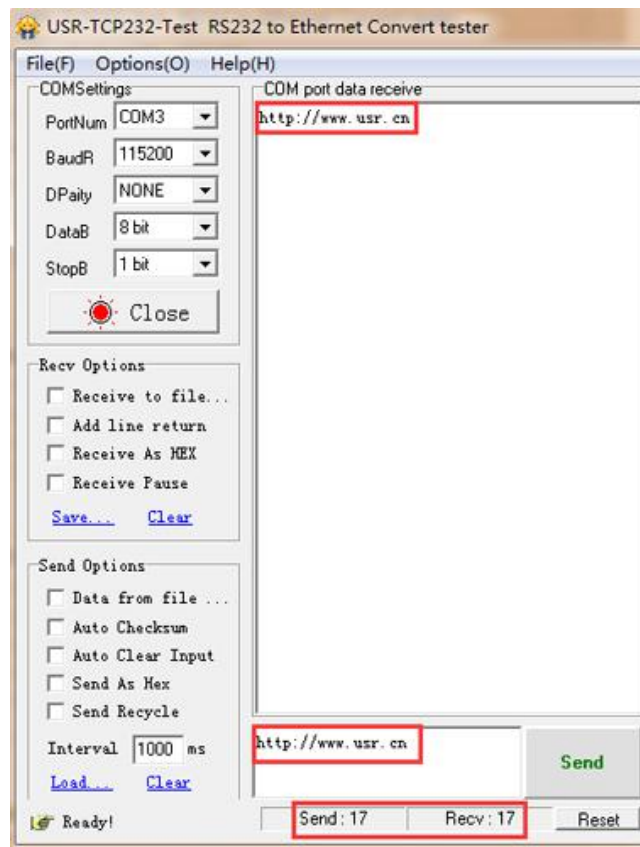
<Note>

Do keep module default settings in this process.

**Meaning of LEDs**

POWER	“on” means normal power supply, “off” means power supply failure
GPRS	“on” means attached, “off” means attach failure
LINK1	“on” means socket A has connected to server, “off” means not connect

3. Wait for LINKA led on, send data to module via serial port.  
For example, if you send “http://www.usr.cn”, at the same time, the DATA led will be on for a while.  
Then, you will get “http://www.usr.cn” in the receiving window from our test server.



## 2. Product Introduction

### 2.1. Short Description

USR-GPRS232-7S3 is new listing at 2015, it is one embedded GSM module. It is designed for embedded system, small and convenient that you can integrate it to your system easily.

### 2.2. Features

- Small and completed function GPRS DTU, transmission between serial and GPRS
- Support GSM850/900,DCS1800/1900
- Support GSM/GPRS/EDGE network and 2G/3G/4G SIM card, but 2G traffic only
- Support 2 sockets at the same time, Support TCP and UDP
- 4KB cache for each of sockets, user can choose to cache data or not
- Support register package and heart beat package

- Support setting parameters via SMS message
- Support 3 work modes: SMS message transmission mode, network transmission mode and HTTPD mode
- Support basic and extend AT command sets
- Support similar RFC2217
- Support flow control: RTS/CTS
- Support RS485

## 2.3. Parameters

		Parameters
	Wireless standard	GSM / GPRS
	Frequency range	850/900/1800/1900MHz
	Transmit power	GSM900 class4 (2W)
		DCS1800 class1 (1W)
	GPRS Terminal Device Class	Class B
	GPRS Multi-slot Class	GPRS Class 10
	GPRS Coding Schemes	CS1 ~ CS4
Antenna interface	I-PEX interface	
<b>Hardware</b>	Data interface	UART: 2400bps - 921600bps
	Working voltage	DC 5V~16V
	Working current	Average 35mA-48mA Max: 394mA(12V)
	Working temperature	-25- 85 degree
	Storage temperature	-40- 125 degree
	Dimensions	31.5×44.2
<b>Software</b>	Wireless type	GSM / GPRS
	Work mode	Transparent transmission mode SMS transparent transmission mode HTTPD mode
	Setup command	AT+ command
	Network protocol	TCP /UDP/ DNS/HTTP
	Max TCP connection	2
	Config	UART AT command Network AT command SMS ATcommand
	Application software	Customize for VIP customers

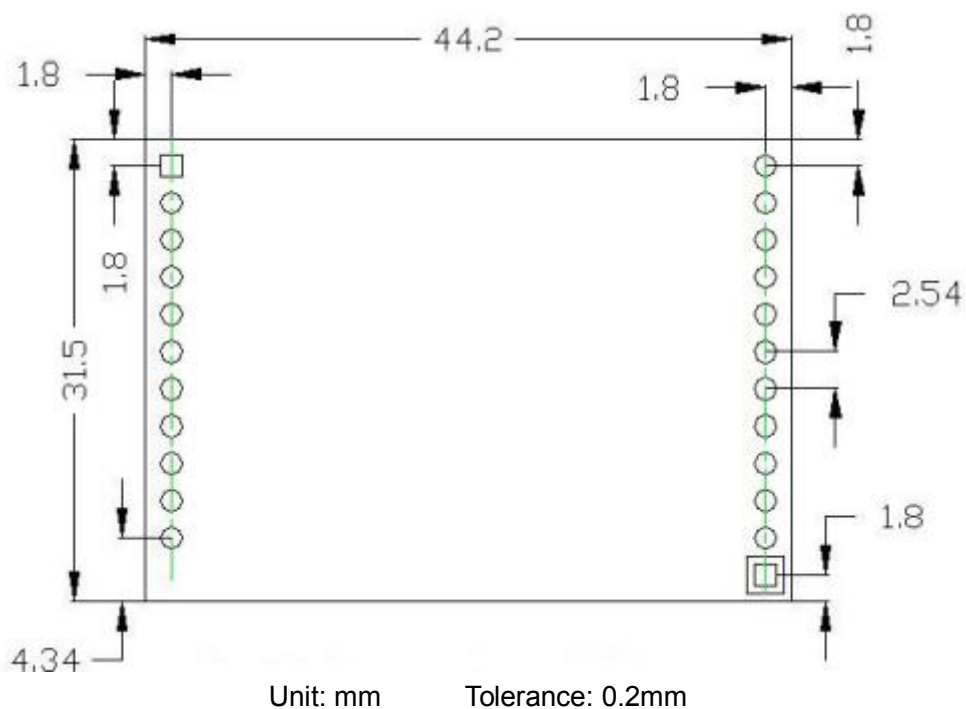


## 2.4. Hardware

### Pin Defination

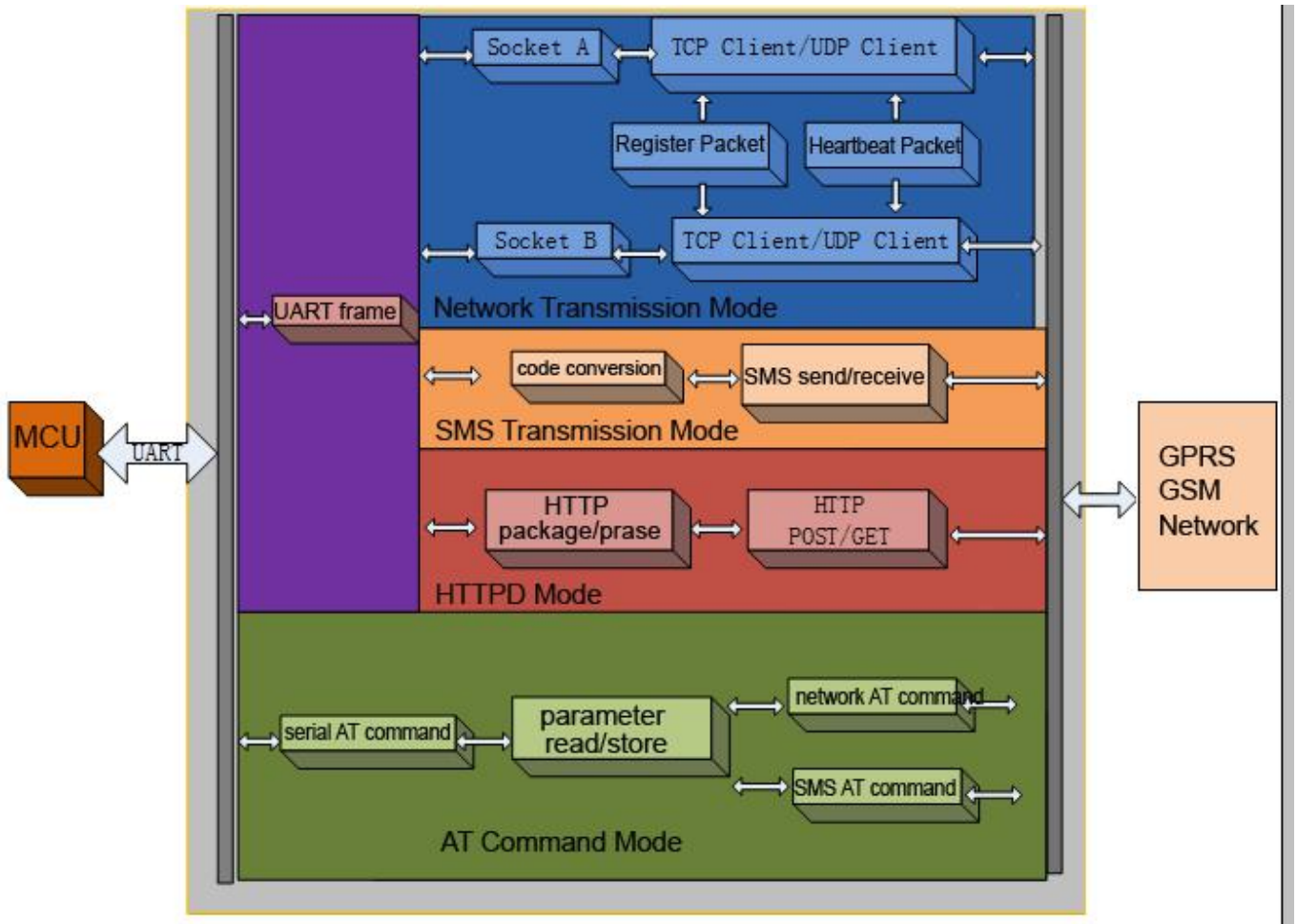


## 2.5. Dimensions



### 3. Function

Below diagram shows the module function



#### 3.1. Setup Process

USR-GPRS232-7S3 will automatically connect to GPRS network and server, get into correct work mode and open serial port according to the pre-set parameters.

##### Pre-set parameters

- ❖ Work mode
  - Network transparent transmission mode
  - HTTPD mode
  - SMS transparent transmission mode
- ❖ Default TCP/UDP connection
  - Protocol
  - Connection (Long/Short)
  - Destination port

- Destination IP
- ❖ UART parameters
  - Baud rate
  - Data bit
  - Parity bit
  - Stop bit
  - Hardware flow control

Setup and restart, module will work according to pre-set parameters.

## 3.2. Work mode

There are 3 work modes in total

- Network transparent transmission mode
- HTTPD mode
- SMS transparent transmission mode

### Switching Work Mode

Customers can change work mode by sending AT+WKMOD

Work mode	Command
SMS message transmission mode	AT+WKMOD="SMS"{CR}
Network transmission mode	AT+WKMOD="NET"{CR}
HTTPD mode	AT+WKMOD="HTTPD"{CR}

<Note>

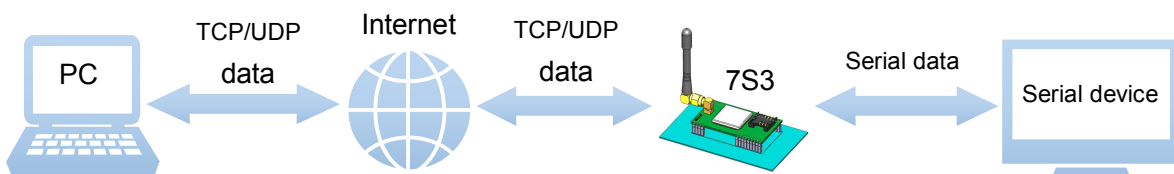
After setting, user need to restart module.

{CR} means carriage return, hex number is 0x0D.

More details of AT commands, please refer to chapter 4.1.

### 3.2.1. Network Transmission Mode

#### 3.2.1.1. Description



Under this mode, customer's serial device can send data to TCP or UDP server via this module, Module can also receive data from server, and transmit the data to customer's serial device.

Customers don't need to know the detail of communication, just need some simple setting.

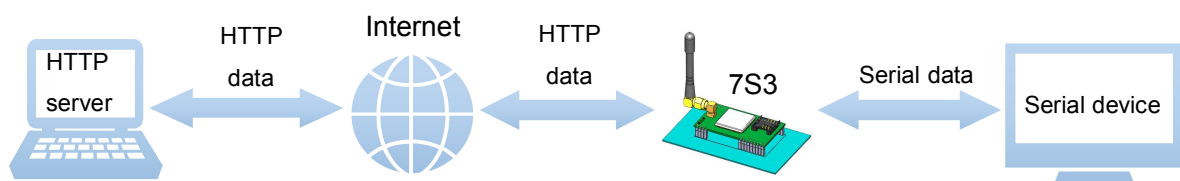
USR-GPRS232-7S3 support 2 sockets, socket A and socket B, they are standalone with each other. This module only supports TCP Client and UDP Client.

### 3.2.1.2. Setup Steps

- 1) Set work mode as network transmission mode  
**AT+WKMOD="NET"**
- 2) Enable socket A  
**AT+SOCKAEN="on"**
- 3) Set socket A as TCP client, server address is test.usr.cn(domain), server port is 10000.  
**AT+SOCKA="TCP","test.usr.cn",10000**
- 4) Restart USR-GPRS232-7S3, wait the connection of socket A has been established, send data to module via serial port, then you will get data at network server software. And send data to module via network, then the data will be got at module serial port.

## 3.2.2. HTTPD mode

### 3.2.2.1. Description



Under this mode, customer's serial device can request data from HTTP server via this module, Module can also receive data from HTTP server, analysis the data and transmit the data to customer's serial device.

### 3.2.2.2. Setup steps

- 1) Set work mode as HTTPD  
**AT+WKMOD="HTTPD"**
- 2) Set request type as GET  
**AT+HTPTP="get"**
- 3) Set the address of HTTP server as "www.usr.cn", and set server port as 80  
**AT+HTPSV="www.usr.cn",80**
- 4) Set the request URL as "/1.php?"  
**AT+HTPURL="/1.php?"**
- 5) Set request head string.  
**AT+HTPHD="Accept:text/html[0D][0A]Accept-Language:zh-CN[0D][0A]User-Agent: Mozilla/5.0[0D][0A]Connection: Keep-Alive[0D][0A]"**

<Note>

[0D] means carriage return, it is 0x0D in hex number. [0A] means line feed, it is 0x0A in hex. Because this kind of data is not able to be put into AT commands string, so it is tropical. The rule is two bytes in [] means one byte in hex form.

For example,

“[0D]” means carriage return, 0x0D in hex.

“[20]” means space, 0x20 in hex.

- Restart module, send “data=http://www.usr.cn” to HTTP server, will receive [http://www.usr.cn] from server.

<Note>

The server “www.usr.cn” is USR’s HTTP test server, if you send “data=http://www.usr.cn” to it, it will feed back the data after “=” with “[ ]” including.

In step 1, serial device send “data=http://www.usr.cn” to USR-GPRS232-7S3, then module prepares HTTP package with head information set before, so the package data is:

```
GET /1.php?data=http://www.usr.cn HTTP/1.1  
Host: www.usr.cn  
Accept:text/html  
Accept-Language:zh-CN  
User-Agent: Mozilla/5.0  
Connection: Keep-Alive
```

After sending data, USR-GPRS232-7S3 receives data:

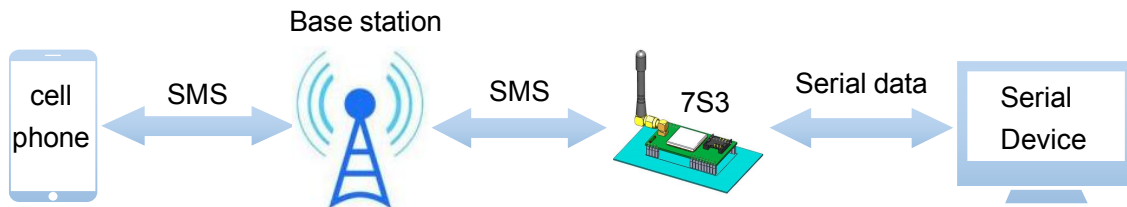
```
HTTP/1.1 200 OK  
Date: Wed, 24 Jun 2015 04:08:35 GMT  
Server: Apache/2.0.63 (Win32) mod_ssl/2.0.65 OpenSSL/0.9.8o PHP/5.2.14  
X-Powered-By: PHP/5.2.14  
Cache-Control: max-age=600  
Expires: Wed, 24 Jun 2015 04:18:35 GMT  
Content-Length: 19  
Keep-Alive: timeout=15, max=99  
Connection: Keep-Alive  
Content-Type: text/html
```

```
[http://www.usr.cn]
```

After USR-GPRS232-7S3 receives the above data, it will analysis the information, and send the result to serial device.

### 3.2.3. SMS Transmission Mode

#### 3.2.3.1. Description



Under this mode, customer's serial device can send SMS message to a cell phone via this module, Module can also receive SMS message from cell phone, and transmit the message in serial data to serial device.

#### 3.2.3.2. Setup steps

- 1) Set the destination phone number.  
**AT+DSTNUM="phone number"**
- 2) Set work mode as SMS message transmission.  
**AT+WKMOD="SMS"**
- 3) Restart module, and send message to module via serial port, then you will get the message on your cellphone. Then send a message back to module, then you will get the message from module's serial port.

Note: every command string should terminate with carriage return. Here we use [0D].

## 3.3. Register Package

### 3.3.1. Description

Under network transmission mode, customers can send Register Package, Register Package is used to make server know what device is sending data, or as a password. Customers can choose to send register package to network server, or serial device. Register Package could be sent when network connection is established, or as the head of every TCP/UDP package. The type of register package could be ICCID ,IMEI, or user-defined data.

### 3.3.2. Setup steps

- 1) Enable sending register package.

**AT+REGEN="on"**

- 2) Set the type of register package data. for instance, using user-defined data.

**AT+REGTP="REGDT"**

- 3) Set user-defined register data, for instance, set data as string "www.usr.cn", you need to convert it to hex string "7777772E7573722E636E".

**AT+REGDT="7777772E7573722E636E"**

- 4) Set sending type, for instance, send data when connection is established.

**AT+REGSND="link"**

- 5) Besides, customers need to configure socket A or socket B for network connection, refer to chapter 4.1. After setting successful, restart USR-GPRS232-7S3. When connection is establishing, module will send string "www.usr.cn" to server via socket A and socket B.

## 3.4. Heart Beat Package

### 3.4.1. Description

Under network transmission mode, customers can send Heart Beat Package to network server or serial device.

Sending heart beat data to network server, is used to check if connection is alive. If USR-GPRS232-7S3 sends data unsuccessful for 3 times, it will try to reconnect to server.

Sending heart beat data to serial device, is usually used to query information from serial device instead of sending query command from server, To do this can cut some communication flow.

### 3.4.2. Setup steps

- 1) Enable sending heart beat package.

**AT+HEARTEN="on"**

- 2) Set the destination of heart beat package data. for instance, to network server.

**AT+HEARTTP="NET"**

- 3) Set heart beat data, for instance, set data as string "www.usr.cn", customer need to convert it to hex string "7777772E7573722E636E".

**AT+HEARTDT="7777772E7573722E636E"**

- 4) Set sending period, for instance, 30 seconds.

**AT+HEARTTM=30**

Besides, customers need to configure socket A or socket B for network connection, refer to chapter 4.1. After setting successful, restart USR-GPRS232-7S3. When connection is establishing, If device is idle for 30 seconds, USR-GPRS232-7S3 will send string "www.usr.cn" to server via socket A and socket B.

## 3.5. USR CLOUD

### 3.5.1. Description

USR CLOUD is a platform for communication between machine-to-machine, machine-to-upper computer (including Android, IOS, PC). It is used for data transparent transmission, device can realize remote communication with little settings. USR CLOUD application field: remote monitor and control, internet of things, smart home and so on. USR-GPRS232-7S3 can also join in USR CLOUD.

### 3.5.2. Setup Steps

- 1) Enable USR CLOUD function  
**AT+CLOUDEDN="ON"**
- 2) Set 20 bits device ID and 8 bit password.  
**AT+CLOUDID="10011111100000111111"**  
**AT+CLOUDPA="0000test"**
- 3) Set USR CLOUD server address and enable. Now our address is cloud.usr.cn, port 15000  
**AT+SOCKAEN="on"**  
**AT+SOCKA="TCP","cloud.usr.cn",15000**
- 4) Click to save all parameters. After restart, module will connect the network and reply `usr_cloud_code_0`, then connect USR CLOUD succeed.

## 3.6. UART Function

### 3.6.1. RS485

USR-GPRS232-7S3 module supports RS485 function. RS485 is half duplex, the direction of transmitting data should be controlled manually. Refer to command AT+UART

### 3.6.2. Similar RFC2217

This function is similar to RFC2217. Customer can change the parameters of serial port dynamically via network. This modification is temporary, when module restarts, the parameters changed will return back to previous values.

Definition of similar RFC2217

This protocol contains 8 bytes, details as the following form



Item	Head	Baud Rate	Bits Parameters	Check Sum
bytes	3	3	1	1
interpretation	Fixed data	MSB	Refer to the following table	Sum of Baud Rate and Bits Parameters
Exmample1 (115200,N,8,1)	55 AA 55	01 C2 00	83	46
Exmample2 (9600,N,8,1)	55 AA 55	00 25 80	83	28

### Interpretation of bits parameters

Bits	Interpretation	Value	Description
1:0	Data bits	00	5
		01	6
		10	7
		11	8
2	Stop bits	0	1
		1	2
3	Enable/disable parity	0	disable
		1	enable
5:4	Parity type	00	ODD
		01	EVEN
		10	Mark
7:6	No defined	00	write 0

## 3.7. Indicator Status

### Indicator status meaning

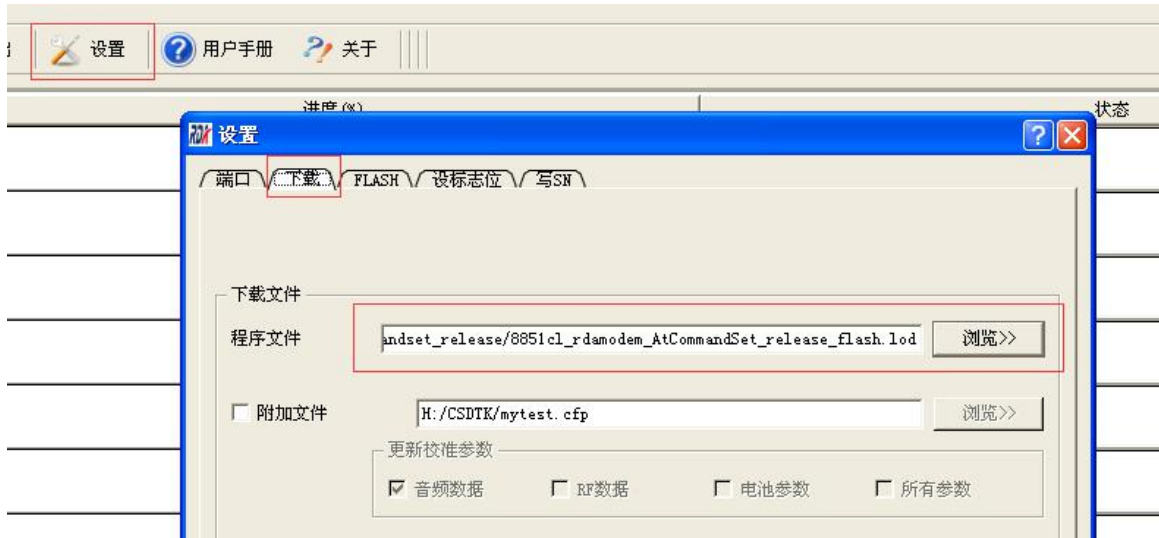
POWER	“on” means normal power supply, “off” means power supply failure
GPRS	“on” means attached, “off” means attach failure
LINK1	“on” means socket A has connected to server, “off” means not connect

### Firmware Upgrade

USR-GPRS232-7S3 can be upgrade by UART port. The upgrade port is reserved on PCB, a USB to TTL cable is needed. Wiring hole is GND, RXD, TXD from left to right.



Open the update software we supply, click Setup, select correct firmware to load.



Select port number



Get back to home and click upgrade



Wait until progress bar finished, upgrade succeed.

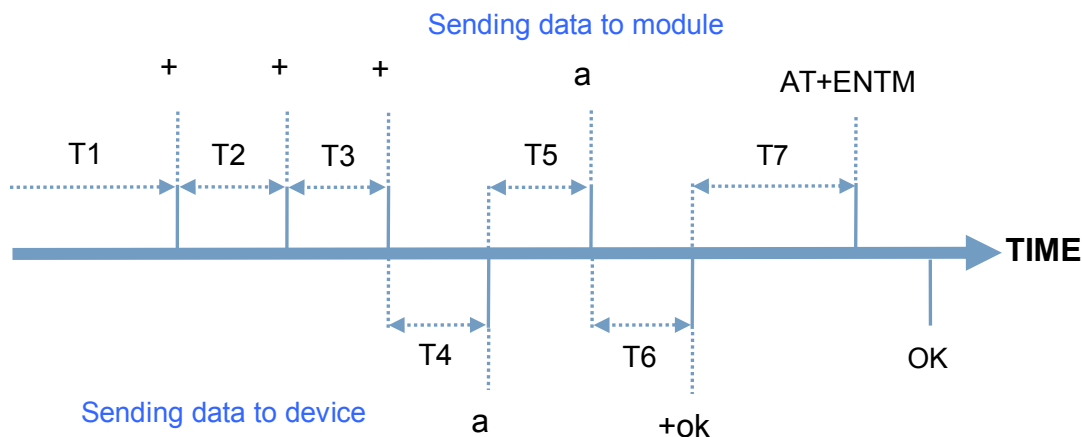


## 4. Parameters Setting

### 4.1. AT commands Set

#### 4.1.1. AT command Mode

When USR-GPRS232-7S3 works in one of the three modes: SMS message transmission, network transmission mode or HTTPD mode. You can get module into “command mode” by sending specific timing data to it, then send commands to change module’s running parameters. If changing is done, get module back to previous work mode.



In the above figure, abscissa axis means time, the data above time axis is sent to module from device, and the data below time axis is sent to device from module.

#### Time requirement

T1 > 200ms

T2 < 50ms

T3 < 50ms

T5 < 3s

#### Get into command mode:

- 1) Module is in SMS message transmission, network transmission or HTTPD mode.
- 2) Device send “+++” to module, then module send ‘a’ to device.
- 3) If device receives ‘a’, it must send ‘a’ to module within 3 seconds.
- 4) Module receives ‘a’, and sends “+ok” to device.
- 5) Device receives “+ok”, it means module has gone to temporary command mode, so device can send AT commands to module.

#### Back to previous mode:

- 1) Device send command “AT+ENTM” to module.

- 2) Module receives "AT+ENTM", and send "OK" to device.
- 3) Device receives "OK", it means module has gone back to previous mode.

### 4.1.2. Serial AT command

#### Steps

- 1) Set work mode as network transmission.

**AT+WKMOD="NET"[0D]**

- 2) Enable serial command function.

**AT+UATEN="on"[0D]**

- 3) Query command password.

**AT+CMDPW[0D]**

We suppose the command password is "usr.cn" here.

After setting successful, restart USR-GPRS232-7S3, then send "usr.cn#AT+VER" via serial.

Then USR-GPRS232-7S3 will respond the firmware version.



### 4.1.3. Network AT command

#### Steps

- 1) Enable network command function.

**AT+NATEN="on"**

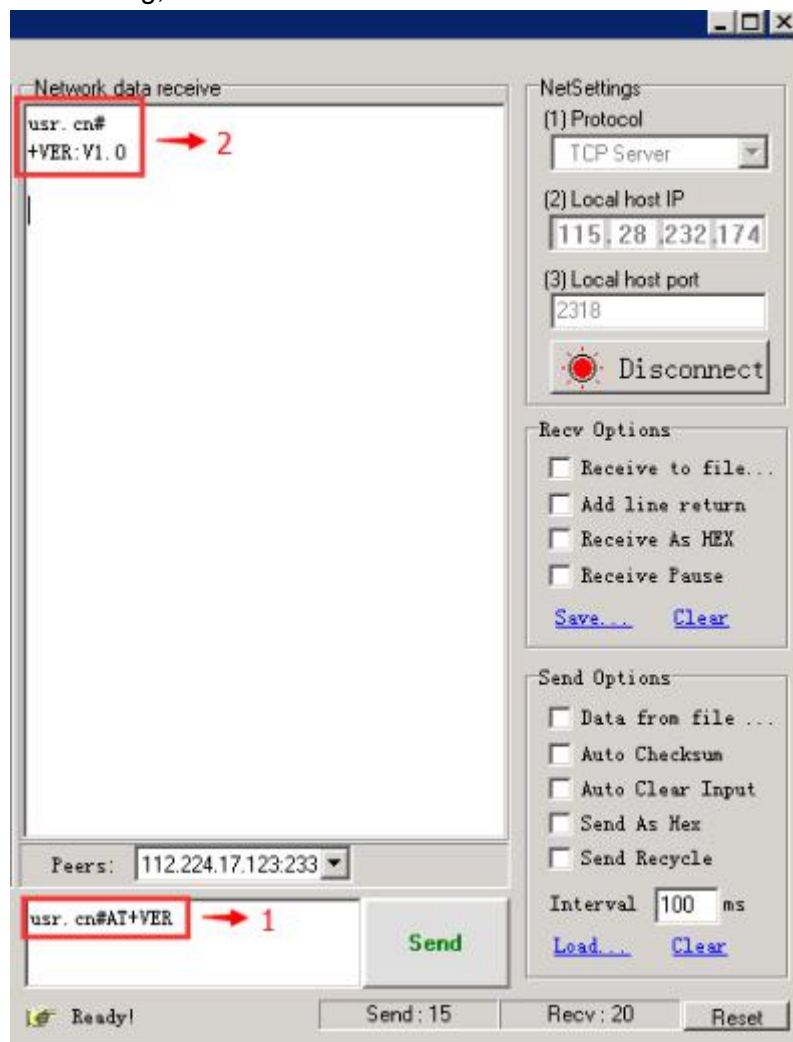
- 2) Query command password.

**AT+CMDPW**

We suppose the command password is "usr.cn" here.

Besides, customers need to configure socket A or socket B for network connection, refer to chapter 4.1. After setting successful, restart USR-GPRS232-7S3.

When connection is establishing, send "usr.cn#AT+VER" from network.



#### 4.1.4. SMS AT Command

<Note>

every command string should terminate with carriage return. Here we use [0D].

##### Steps

- 1) Query command password.

**AT+CMDPW**

We suppose the command password is "usr.cn" here.

Besides, customers need to configure SMS message transmission, refer to chapter 4.1. After setting successful, restart USR-GPRS232-7S3. send "usr.cn#AT+VER" to USR-GPRS232-7S3 from cellphone, then then you will get response from USR-GPRS232-7S3.



## 4.2. AT Command Set

AT commands work in query-reply mode, “query” means device sends commands to module, “reply” means module responds to device.

<Note>

Characters in command is case insensitive.

Symbol	Description
<>	content inside is necessary
[]	content inside is not necessary
{ }	content inside is with particular meaning
~	range, for example, A~B means the range is from A to B
CMD	command code
OP	operator
PARA	parameter
CR	carriage return in ASCII, 0x0D in hex
LF	line feed in ASCII, 0x0A in hex

## The format of query

Command string: <AT+>[CMD][OP][PARA]<CR>

Code	Meaning	Necessary
AT+	AT command head	Yes
CMD	function of command	Yes
OP	operator, example: =, ?, =?	No
PARA	parameters	No
CR	carriage return, end of command	Yes

## Type of command

Type	Command string	Description
0	AT+{CMD}?{CR}	execute the command or query current parameter value
1	AT+{CMD}=?{CR}	query the value range or type
2	AT+{CMD}{CR}	execute the command or query current parameter value
3	AT+{CMD}={PARAM}{CR}	set the value of parameter

## The format of reply

There are two kinds of response, with or without echo.  
 All the followings are using without echo as examples.

Command string: [CR][LF][+CMD][OP][PARA][CR][LF]<CR><LF>[OK]<CR><LF>

Code	Meaning	Necessary
CR	carriage return	No
LF	line feed	No
+CMD	response head	No
OP	operator	No
PARA	parameter	No
CR	carriage return	No
LF	line feed	No
CR	carriage return	Yes
LF	line feed	Yes
OK	operating successful	No
CR	carriage return	Yes
LF	line feed	Yes

## Type of command

Type	Command string	Description
0	<CR><LF><OK><CR><LF>	successful
1	<CR><LF><+CMD:><PARA><CR><LF><CR>><LF><OK><CR><LF>	return current parameter

#### 4.2.1. AT Command Set

NO.	Command	Description
<b>Management commands</b>		
1	H	help message
2	Z	restart
3	E	query/set enable/disable echo
4	ENTM	quit command mode
5	WKMOD	query/set work mode
6	CALEN	query/set if allow to call
7	NATEN	query/set enable/disable network command
8	UATEN	query/set enable/disable serial port command
9	CMDPW	query/set command password
10	CACHEN	query/set enable/disable data cache
11	STMSG	query/set welcome message
12	RSTIM	query/set restart time
<b>Configuration commands</b>		
13	S	save current settings
14	RELD	restore to user default settings
15	CLEAR	restore to factory settings
16	CFGTF	save current setting as default setting
<b>Query information commands</b>		
17	VER	query firmware version
18	SN	query SN code
19	ICCID	query CCID code
20	IMEI	query IMEI code
21	CNUM	query the current phone number
<b>UART parameters commands</b>		
22	UART	query/set serial port parameters
23	RFCEN	query/set enable/disable similar RFC2117
<b>Socket parameters commands</b>		
24	APN	query/set APN code
25	SOCKA	query/set socket A parameters
26	SOCKB	query/set socket B parameters
27	SOCKAEN	query/set enable/disable socket A
28	SOCKBEN	query/set enable/disable socket B



29	SOCKASL	query/set socket A long or short connection
30	SOCKBSL	query/set socket B long or short connection
31	SOCKALK	query socket A connection status
32	SOCKBLK	query socket B connection status
33	SOCKIDEN	query/set wether show socket ID function
<b>Register package parameters commands</b>		
34	REGEN	query/set register package type
35	REGTP	query/set register package data type
36	REGID	query/set register ID (used for D2D, no longer supprt now)
37	REGDT	query/set register user-defined data
38	REGSND	query/set register package sending type
<b>Heartbeat package parameters commands</b>		
39	HEARTEN	query/set enable/disable heartbeat
40	HEARTDT	query/set heartbeat data
41	HEARTTP	query/set heartbeat package sending type
42	HEARTTM	query/set heartbeat time
<b>HTTPD parameters commands</b>		
43	HTPTP	query/set HTTP request type
44	HTPURL	query/set HTTP URL
45	HTPSV	query/set HTTP server address and port
46	HTPHD	query/set HTTP request head
47	HTPPK	query/set whether open HEAD filtering function
<b>SMS message parameters commands</b>		
48	DSTNUM	query/set destination phone number
49	SMSSEND	send SMS message
<b>USR Cloud function</b>		
50	CLOUDEN	Set cloud enable
51	CLOUDID	Set 20 bits device ID in cloud
52	CLOUDPA	Set 8 bit communication password in cloud

#### 4.2.1.1. AT+H

function : help message

format :

query : AT+H{CR}{CR}{LF}help message{CR}{LF}{CR}{LF}OK{CR}{LF}

parameter :

help message : commands introduction

#### 4.2.1.2. AT+Z

function : restart

format : AT+Z{CR}{CR}{LF}OK{CR}{LF}

#### 4.2.1.3. AT+E

function : query/set enable/disable echo

format :

Query parameter introductions :

AT+E=?{CR}{CR}{LF}+E:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}

query current value of parameter:

AT+E{CR} or AT+E?{CR}

{CR}{LF}+E:status{CR}{LF}{CR}{LF}OK{CR}{LF}

set parameter:

AT+E=status{CR}

{CR}{LF}OK{CR}{LF}

parameter :

status : enable or disable echo

“on” : enable

“off” : disable

#### 4.2.1.4. AT+ENTM

function : quit from command mode

format :

setting: AT+ENTM{CR}{CR}{LF}OK{CR}{LF}

#### 4.2.1.5. AT+WKMOD

function : query/set work mode

format :

query parameter introductions:

AT+WKMOD=?{CR}

{CR}{LF}+WKMOD:<"CMD","SMS","NET","HTTPD">{CR}{LF}{CR}{LF}OK{CR}{LF}

query current value of parameter:

AT+WKMOD{CR} or AT+WKMOD?{CR}

{CR}{LF}+WKMOD:mode{CR}{LF}{CR}{LF}OK{CR}{LF}

set parameter:

AT+WKMOD=mode{CR}

{CR}{LF}OK{CR}{LF}

parameter :

mode : work mode

“CMD” : AT command mode

“SMS” : SMS message transmission mode

“NET” : network transmission mode

“HTTPD” : HTTPD mode

#### 4.2.1.6. AT+CALEN

function : query/set if allow to call.

format :

query parameter introductions:

AT+CALEN=?{CR}

{CR}{LF}+CALEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}

query current value of parameter:

AT+CALEN{CR} or AT+CALEN?{CR}

{CR}{LF}+CALEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}

set parameter:

AT+CALEN=status{CR}

{CR}{LF}OK{CR}{LF}

parameter :

status : enable or disable call

“on” : enable

“off” : disable

#### 4.2.1.7. AT+NATEN

function : query/set enable/disable network command.

format :

query parameter introductions:

AT+NATEN=?{CR}

{CR}{LF}+NATEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}

query current value of parameter:

AT+NATEN{CR} or AT+NATEN?{CR}

{CR}{LF}+NATEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}

set parameter:

AT+NATEN=status{CR}

{CR}{LF}OK{CR}{LF}

parameter :

status : enable or disable network command

“on” : enable

“off” : disable

#### 4.2.1.8. AT+UATEN

function : query/set enable/disable serial port command

format :

query parameter introductions:

```
AT+UATEN=?{CR}
{CR}{LF}+UATEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+UATEN{CR} or AT+UATEN?{CR}
{CR}{LF}+UATEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+UATEN=status{CR}
{CR}{LF}OK{CR}{LF}
```

parameter:

status: enable or disable serial port command.

“on”: enable

“off”: disable

#### 4.2.1.9. AT+CMDPW

function : query/set command password

format :

query parameter introductions:

```
AT+CMDPW=?{CR}
{CR}{LF}+CMDPW:<"password">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+CMDPW{CR} or AT+CMDPW?{CR}
{CR}{LF}+CMDPW:password{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+CMDPW=password{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

password : command password

#### 4.2.1.10. AT+CACHEN

function : query/set enable/disable data cache.

format :

query parameter introductions:

```
AT+CACHEN=?{CR}
{CR}{LF}+CACHEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+CACHEN{CR} or AT+CACHEN?{CR}
{CR}{LF}+CACHEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+CACHEN=status{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

status : enable/disable data cache.

“on” : enable

“off” : disable

#### 4.2.1.11. AT+STMSG

function : query/set welcome message.

format :

query parameter introductions:

```
AT+STMSG=?{CR}
{CR}{LF}+STMSG:<"welcome message">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+STMSG{CR} or AT+STMSG?{CR}
{CR}{LF}+STMSG:message{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+STMSG=message{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

message : welcome message, when module powers up, it will show them, this can be used to check if module starts up normally.

#### 4.2.1.12. AT+RSTIM

function: query/set automatic restart time. Module will restart if no data received in serial and net port within this time.

format:

query parameter introductions:

query current value of parameter:

```
AT+RSTIM{CR}orAT+RSTIM?{CR}
{CR}{LF}+RSTIM:time{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+RSTIM=time{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

time: automatic restart time, unit: s. Disable this function when set to be 0.

#### 4.2.1.13. AT+S

function : save current settings.

format :

setting:

```
AT+S{CR}  
{CR}{LF}OK{CR}{LF}
```

#### 4.2.1.14. AT+RELD

function : restore user default settings.

format :

setting:

```
AT+RELD{CR}  
{CR}{LF}OK{CR}{LF}
```

#### 4.2.1.15. AT+CLEAR

function : restore factory settings.

format :

setting:

```
AT+CLEAR{CR}  
{CR}{LF}OK{CR}{LF}
```

#### 4.2.1.16. AT+CFGTF

function : save current setting as default setting.

format :

setting:

```
AT+CFGTF{CR}  
{CR}{LF}OK{CR}{LF}
```

#### 4.2.1.17. AT+VER

function : query version.

format :

query current value of parameter:

```
AT+VER{CR} or AT+VER?{CR}
{CR}{LF}+VER:version{CR}{LF}{CR}{LF}OK{CR}{LF}
```

parameter :

version : firmware version

#### 4.2.1.18. AT+SN

function : query SN code.

format :

query current value of parameter:

```
AT+SN{CR} or AT+SN?{CR}
{CR}{LF}+SN:code{CR}{LF}{CR}{LF}OK{CR}{LF}
```

parameter :

code : SN code

#### 4.2.1.19. AT+ICCID

function : query ICCID code.

format :

query current value of parameter:

```
AT+ICCID{CR} or AT+ICCID?{CR}
{CR}{LF}+ICCID:code{CR}{LF}{CR}{LF}OK{CR}{LF}
```

parameter :

code : ICCID code

#### 4.2.1.20. AT+IMEI

function : query IMEI code.

format :

query current value of parameter:

```
AT+IMEI{CR} or AT+IMEI?{CR}
{CR}{LF}+IMEI:code{CR}{LF}{CR}{LF}OK{CR}{LF}
```

parameter :

code : IMEI code

#### 4.2.1.21. AT+CNUM

function : query the MS ISDN related to the subscriber.

format :

query current value of parameter:

```
AT+CNUM{CR} or AT+CNUM?{CR}
{CR}{LF}+CNUM:phone number{CR}{LF}{CR}{LF}OK{CR}{LF}
```

parameter :

phone number : phone number

#### 4.2.1.22. AT+UART

function : query/set serial port parameters.

format :

query parameter introductions:

```
AT+UART=?{CR}
{CR}{LF}+UART:<2400,4800,9600,14400,19200,28800,33600,38400,57600,115200,230400,460800,
921600>,<"NONE","ODD","EVEN">,<7,8>,<1,2>,<"NONE","CRTS","RS485">{CR}{LF}{CR}{LF}OK{
CR}{LF}
```

query current value of parameter:

```
AT+UART{CR} or AT+UART?{CR}
{CR}{LF}+UART:baud,parity,data bit,stop bit,flow control{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+UART=baud,parity,data bit,stop bit,flow control{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

baud : baud rate

2400,4800,9600,14400,19200,28800,33600,38400,57600,115200,230400,460800,921600

parity :

“NONE” : no parity

“ODD” : odd parity

“EVEN” : even parity

data bit :

7 : 7 data bits

8 : 8 data bits

flow control :

“NONE” : no flow control

“CRTS” : use RTS/CTS for flow control

“RS485” : enable RS485 function



#### 4.2.1.23. AT+RFCEN

function : query/set enable/disable similar RFC2117.

format :

query parameter introductions:

```
AT+RFCEN=?{CR}
{CR}{LF}+RFCEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+RFCEN{CR} or AT+RFCEN?{CR}
{CR}{LF}+RFCEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+RFCEN=status{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

status : enable/disable similar RFC2117

“on” : enable

“off” : disable

#### 4.2.1.24. AT+APN

function : query/set APN code.

format :

query parameter introductions:

```
AT+APN=?{CR}
{CR}{LF}+APN:<"code">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+APN{CR} or AT+APN?{CR}
{CR}{LF}+APN:code{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+APN=code{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

code : APN code

#### 4.2.1.25. AT+SOCKA

function : query/set socket A parameters.

format :

query parameter introductions:

```
AT+SOCKA=?{CR}
{CR}{LF}+SOCKA:<"TCP","UDP">,<"address">,<port>{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+SOCKA{CR} or AT+SOCKA?{CR}
{CR}{LF}+SOCKA:protocol,address,port{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+SOCKA=protocol,address,port{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

protocol : communication protocol

“TCP” : TCP

“UDP” : UDP

address : server address, IP or domain name

port : server port, 1~65535

#### 4.2.1.26. AT+SOCKB

function : query/set socket B parameters.

format :

query parameter introductions:

```
AT+SOCKB=?{CR}
{CR}{LF}+SOCKB:<"TCP","UDP">,<"address">,<port>{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+SOCKB{CR} or AT+SOCKB?{CR}
{CR}{LF}+SOCKB:protocol,address,port{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+SOCKB=protocol,address,port{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

protocol : communication protocol

“TCP” : TCP

“UDP” : UDP

address : server address, IP or domain name

port : server port, 1~65535

#### 4.2.1.27. AT+SOCKAEN

function : query/set enable/disable socket A.

format :

query parameter introductions:

```
AT+SOCKAEN=?{CR}
{CR}{LF}+SOCKAEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+SOCKAEN{CR} or AT+SOCKAEN?{CR}
```

```
{CR}{LF}+SOCKAEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+SOCKAEN=status{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

status : enable or disable socket A

“on” : enable

“off” : disable

#### 4.2.1.28. AT+SOCKBEN

function : query/set enable/disable socket B.

format :

query parameter introductions:

```
AT+SOCKBEN=?{CR}
```

```
{CR}{LF}+SOCKBEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+SOCKBEN{CR} or AT+SOCKBEN?{CR}
```

```
{CR}{LF}+SOCKBEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+SOCKBEN=status{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

status : enable or disable socket B.

“on” : enable

“off” : disable

#### 4.2.1.29. AT+SOCKASL

function : query/set socket A long or short connection.

format :

query parameter introductions:

```
AT+SOCKASL=?{CR}
```

```
{CR}{LF}+SOCKASL:<"short","long">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+SOCKASL{CR} or AT+SOCKASL?{CR}
```

```
{CR}{LF}+SOCKASL:type{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+SOCKASL=type{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

type : connection style

“short” : short connection

“long” : long connection

#### 4.2.1.30. AT+SOCKBSL

function : query/set socket B long or short connection.

format :

query parameter introductions:

AT+SOCKBSL=?{CR}

{CR}{LF}+SOCKBSL:<"short","long">{CR}{LF}{CR}{LF}OK{CR}{LF}

query current value of parameter:

AT+SOCKBSL{CR} or AT+SOCKBSL?{CR}

{CR}{LF}+SOCKBSL:type{CR}{LF}{CR}{LF}OK{CR}{LF}

set parameter:

AT+SOCKBSL=type{CR}

{CR}{LF}OK{CR}{LF}

parameter :

type : connection style

“short” : short connection

“long” : long connection

#### 4.2.1.31. AT+SOCKALK

function : query socket A connection status.

format :

query current value of parameter:

AT+SOCKALK{CR} or AT+SOCKALK?{CR}

{CR}{LF}+SOCKALK:status{CR}{LF}{CR}{LF}OK{CR}{LF}

parameter :

status : socket A connection status.

“connected” : connected

“disconnected” : disconnected

#### 4.2.1.32. AT+SOCKBLK

function : query socket B connection status.

format :

query current value of parameter:

AT+SOCKBLK{CR} or AT+SOCKBLK?{CR}

{CR}{LF}+SOCKBLK:status{CR}{LF}{CR}{LF}OK{CR}{LF}

parameter :

status : socket B connection status.

“connected” : connected

“disconnected” : disconnected

#### 4.2.1.33. AT+SOCKIDEN

function: query/set whether show data come from which socket

format

query parameter introductions:

```
AT+ SOCKIDEN =?{CR}
```

```
{CR}{LF}+ SOCKIDEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+ SOCKIDEN {CR} or AT+ SOCKIDEN?{CR}
```

```
{CR}{LF}+ SOCKIDEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+ SOCKIDEN =status{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameters:

status:

“on”: enable show socket ID function

“off”: disable show socket ID function

#### 4.2.1.34. AT+REGEN

function : query/set enable/disable register package.

format :

query parameter introductions:

```
AT+REGEN=?{CR}
```

```
{CR}{LF}+REGEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+REGEN{CR} or AT+REGEN?{CR}
```

```
{CR}{LF}+REGEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+REGEN=status{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

status : enable or disable register package

“on” : enable

“off” : disable

#### 4.2.1.35. AT+REGTP

function : query/set register package data type.

format :

query parameter introductions:

```
AT+REGTP=?{CR}
```

```
{CR}{LF}+REGTP:<"ICCID","IMEI","REGID","REGDT">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+REGTP{CR} or AT+REGTP?{CR}
```

```
{CR}{LF}+REGTP:type{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+REGTP=type{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

type : type of register data

“ICCID” : ICCID code

“IMEI” : IMEI code

“REGID” : register ID ( for D2D function )

“REGDT” : user-defined data

#### 4.2.1.36. AT+REGID

function : query/set register ID(used for D2D).

format :

query parameter introductions:

```
AT+REGID=?{CR}
```

```
{CR}{LF}+REGID:<id>{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+REGID{CR} or AT+REGID?{CR}
```

```
{CR}{LF}+REGID:id{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+REGID=id{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

id : register ID

#### 4.2.1.37. AT+REGDT

function : query/set register user-defined data.

format :

query parameter introductions:

```
AT+REGDT=?{CR}
{CR}{LF}+REGDT:<data>{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+REGDT{CR} or AT+REGDT?{CR}
{CR}{LF}+REGDT:data{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+REGDT=data{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

data : user-defined register data, hex string, max length is 80 bytes,  
for example, "7777772E7573722E636E" means "www.usr.cn" in ASCII.

#### 4.2.1.38. AT+REGSND

function : query/set register package sending type.

format :

query parameter introductions:

```
AT+REGSND=?{CR}
{CR}{LF}+REGSND:<"link","data","link&data">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+REGSND{CR} or AT+REGSND?{CR}
{CR}{LF}+REGSND:type{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+REGSND=type{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

type : sending type

"link" : send register data when socket connects

"data" : send register data as the head of every TCP/UDP package

"link&data" : support both of above two ways

#### 4.2.1.39. AT+HEARTEN

function : query/set enable/disable heartbeat.

format :

query parameter introductions:

```
AT+HEARTEN=?{CR}
{CR}{LF}+HEARTEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HEARTEN{CR} or AT+HEARTEN?{CR}
{CR}{LF}+HEARTEN:status{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HEARTEN=status{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

status : enable/disable heartbeat function

“on” : enable

“off” : disable

#### 4.2.1.40. AT+HEARTDT

function : query/set heartbeat data.

format :

query parameter introductions:

```
AT+HEARTDT=?{CR}
{CR}{LF}+HEARTDT:<"data">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HEARTDT{CR} or AT+HEARTDT?{CR}
{CR}{LF}+HEARTDT:data{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HEARTDT=data{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

data : heartbeat data, hex string, max length is 80 bytes,

for example, “7777772E7573722E636E” means “www.usr.cn” in ASCII.

#### 4.2.1.41. AT+HEARTTP

function : query/set heartbeat package sending type.

format :

query parameter introductions:

```
AT+HEARTTP=?{CR}
{CR}{LF}+HEARTTP:<"COM","NET">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HEARTTP{CR} or AT+HEARTTP?{CR}
{CR}{LF}+HEARTTP:type{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HEARTTP=type{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

type : sending type

“COM” : send data to serial port

“NET” : send data to network



#### 4.2.1.42. AT+HEARTTM

function : query/set heartbeat time.

format :

query parameter introductions:

```
AT+HEARTTM=?{CR}
```

```
{CR}{LF}+HEARTTM:<"time">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HEARTTM{CR} or AT+HEARTTM?{CR}
```

```
{CR}{LF}+HEARTTM:time{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HEARTTM=time{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

time : strobe period, unit is second

#### 4.2.1.43. AT+HTPTP

function : query/set HTTP request type.

format :

query parameter introductions:

```
AT+HTPTP=?{CR}
```

```
{CR}{LF}+HTPTP:<"GET","POST">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HTPTP{CR} or AT+HTPTP?{CR}
```

```
{CR}{LF}+HTPTP:type{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HTPTP=type{CR}
```

```
{CR}{LF}OK{CR}{LF}
```

parameter :

type : HTTP request type

“GET” : get

“POST” : post

#### 4.2.1.44. AT+HTPURL

function : query/set HTTP URL.

format :

query parameter introductions:

```
AT+HTPURL=?{CR}
```

```
{CR}{LF}+HTPURL:<"URL">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HTPURL{CR} or AT+HTPURL?{CR}
{CR}{LF}+HTPURL:URL{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HTPURL=URL{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

URL : HTTP request URL

#### 4.2.1.45. AT+HTPSV

function : query/set HTTP server address and port.

format :

query parameter introductions:

```
AT+HTPSV=?{CR}
{CR}{LF}+HTPSV:<"address">,<port>{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HTPSV{CR} or AT+HTPSV?{CR}
{CR}{LF}+HTPSV:address,port{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HTPSV=address,port{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

address : server address, IP or domain name

port : server port, 1~65535

#### 4.2.1.46. AT+HTPHD

function : query/set HTTP request head.

format :

query parameter introductions:

```
AT+HTPHD=?{CR}
{CR}{LF}+HTPHD:<"head">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+HTPHD{CR} or AT+HTPHD?{CR}
{CR}{LF}+HTPHD:head{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+HTPHD=head{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

head : HTTP request head

#### 4.2.1.47. AT+DSTNUM

function : query/set destination phone number.

format :

query parameter introductions:

```
AT+DSTNUM=?{CR}
{CR}{LF}+DSTNUM:<"number">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+DSTNUM{CR} or AT+DSTNUM?{CR}
{CR}{LF}+DSTNUM:number{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+DSTNUM=number{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

number : destination phone number in "SMS" mode.

#### 4.2.1.48. AT+SMSSEND

function : send SMS message.

format :

query parameter introductions:

```
AT+SMSSEND=?{CR}
{CR}{LF}+SMSSEND:<"number">,<1,2,3>,<"data">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+SMSSEND=number,type,data{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

number : destination phone number

type: coding scheme

1 : ASCII coding, compressed

2 : 8 bits coding, not compressed

3 : UCS8 coding

data : content of SMS message

#### 4.2.1.49. AT+CLOUDEN

function: query/set whether enable USR CLOUD function

format:

query parameter introductions:

```
AT+ CLOUDEN =?{CR}
{CR}{LF}+ CLOUDEN:<"on","off">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+ CLOUDEN {CR} or AT+ CLOUDEN?{CR}
{CR}{LF}+ CLOUDEN: status {CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+ CLOUDEN =status{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

Status: whether enable CLOUD function, including  
"on": enable  
"off": disable

#### 4.2.1.50. AT+CLOUDID

function: query/set the 20 bit ID of device

format

query parameter introductions:

```
AT+ CLOUDID =?{CR}
{CR}{LF}+ CLOUDID:<"id">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+ CLOUDID {CR} or AT+ CLOUDID?{CR}
{CR}{LF}+ CLOUDID: id {CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+ CLOUDID =id{CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

id: 20 bit ID of device

#### 4.2.1.51. AT+CLOUDPA

function: query/set 8 bit communication password of device

format:

query parameter introductions:

```
AT+ CLOUDPA =?{CR}
{CR}{LF}+ CLOUDPA:<"pass">{CR}{LF}{CR}{LF}OK{CR}{LF}
```

query current value of parameter:

```
AT+ CLOUDPA {CR} or AT+ CLOUDPA?{CR}
{CR}{LF}+ CLOUDPA: pass {CR}{LF}{CR}{LF}OK{CR}{LF}
```

set parameter:

```
AT+ CLOUDPA = pass {CR}
{CR}{LF}OK{CR}{LF}
```

parameter :

pass: 8 bit password

## 5. Contact Us

Company: Jinan USR IOT Technology Limited  
Address: Floor 11, Building1, No.1166 Xinluo Street, Gaoxin Distric, Jinan, Shandong, 250101, China  
Tel: 86-531-55507297, 86-531-88826739  
Web: <http://www.usriot.com>  
Support : <http://h.usriot.com>  
Email: [sales@usr.cn](mailto:sales@usr.cn), [tec@usr.cn](mailto:tec@usr.cn)

## 6. Disclaimer

This document provides information about USR-C322 modules, this document does not grant any license to intellectual property rights. Except the responsibility declared in the product sale clause, USR does not assume any other responsibilities. In addition, USR does not make any warranties for the sale and use of this product, including the suitability of the product for a particular purpose, merchantability or fitness for any patent, copyright or other intellectual property infringement, etc. USR may make changes to specifications and product descriptions without notice.

## 7. Update History

2016-6-14 V1.0 created