

User Manual and Test Guide

HIS Script

Rev1.0

1. INTRODUCTION.....	4
2. GRAMMER.....	5
2.1. Data Structure.....	5
2.2. Operator.....	5
2.3. Operation.....	5
2.4. IF Sentence.....	6
2.5. FOR Sentence.....	6
2.6. WHILE Sentence.....	7
2.7. Reserved Word.....	7
2.8. External Interface.....	7
2.9. Visiting Industrial Device.....	7
2.10. Industrial Device Input.....	8
2.11. Function.....	9
2.12. Variable Scope.....	9
3. HisUI INTRODUCTION.....	11
4. HFPackTools INTRODUCTION.....	12
5. SCRIPT UPDATE.....	13
6. GPIOE10 GPIO CONTROL.....	15
6.1. Function Demand.....	15
6.2. Function Achieve.....	15
6.3. HisUI Test.....	16
6.4. Script Transferred and Uploaded.....	17

6.5. E10 Test.....	17
APPENDIX A. HEARTBEAT CASE.....	19
A.1. Demands.....	19
A.2. Achieve.....	19
A.3. Script Transferred and Uploaded.....	20
A.4. Product Test.....	20
APPENDIX A. REGISTER PACKAGE.....	21
B.1. Demands.....	21
B.2. Achieve.....	21
B.3. Other Description.....	22

Version List :

2017-06-19 First Draft

1. INTRODUCTION

Hi-flying I.O.T script(HF IOT Script, short fot HIS). **Note that version must be 1.09i or above.** It is used for data transfer of IOT industrial device. When download the script in to the industrial device, the following functions can be achieved.

- Automatically send data to UART or Socket at regular time, which can be recognized as Modbus primary station.
- After receive data from UART or Socket, data will transferred based on script.

The download address is able to acquire all the tools and example scripts:

<http://pan.baidu.com/s/1qY0QPcS>

Or :

<https://www.dropbox.com/s/phnpvbqve43n86y/HIS%E8%84%9A%E6%9C%AC%E5%B7%A5%E5%85%B7.rar?dl=0>

All supported product is shown as following figures, the next chapters will take Eport-E10 or HF2211 as example.

Super-Ethernet :

产品	Eport-E10	Eport-E30	Eport Pro-EP10	Eport Pro-EP20
				

Serial-to-Ethernet :

产品	HF5111A	HF5111B	HF5142
			

Serial-to-WiFi:

(W10 and W20 are the bulid-in modules of HF2211 and HF2221respectively)

产品	Wport-W10	Wport-W20	HF2211	HF2221
				

2. GRAMMER

2.1. Data Structure

All the data structure HIS script supported: Int, Float, String

Variable is automatically defined and assigned, such as:

```
a=10
```

Define an integer and assign 10 to it.

```
f=1.04
```

Define a float and assign 1.04 to it.

```
s=" 1234"    or  s=[0x31,0x32,0x33,0x34]
```

Define a string and assign it.

Note: HIS grammer is sensitive to capital and small letter.

2.2. Operator

Supported operator and its priority:

High to Low	Operator
0	(,) , [,] , .
1	* , /
2	+ , -
3	>> , << , & ,
4	< , <= , > , >= , == , !=
5	&& ,
6	=

Note: Cannot support ++, --, +-

2.3. Operation

Integer and float can support "+, -, *, /" operation, and interger can also support bit operation like <<, >>, &, |

In comparson operation, source operation and destination operation must be the same type(integer and float are reconized as same type, but string can only be compared to string)

When doing assign operation, it will involve in data structure transfer. For example:

a is integer, f is float and s is string

a=f f is assigned to a, the type of a is float at the same time(if f=a, it does not involve structure transfer)

a=s a will turn to string and assigned as the content of s

s=f s is float and assigned

Plus operation of string:

Like : s1=" 123" , s2=" 456"

s3=s1+s2 Result: s3=" 123456"

String comparison operation: string can only support "=" and "!="

String function, for example:

Like : s1=" 12345678" , a1=0x010203

a=s1.leng() a is integer 8

b=s1.charAt(2) b is integer 2, parameter means which character

c=s1.subString(1,5) c is string "2345"

d=s1.replaceAt(1, '5') d is string "15345678"

Note: subString() function, the former bit is start position while the last is for end. Position is calculated from 0. String contain the data from start position but the end position.

d=a1.toString(0,3) d is string : [0x03,0x02,0x01,0x00]

e=a1.toString(2,0) e is string : [0x01,0x02,0x03]

f=a1.prtString() f is string : " 66051" , (0x010203,66051 decimalism)

Note: toString() source operator is integer. Two parameters is the four bytes in integer(so the value is $\geq 0, \leq 3$). In example one, d is from 0byte to 4byte. In example two, e is two bytes from 2byte to 0bytes.

2.4. IF Sentence

For example :

IF (a>1)

.....

ELSE

.....

END

No ELSE is ok,like:

IF (a>1)

.....

END

2.5. FOR Sentence

For example :

FOR i,0,5,1 // i is variable and the following parameter is initial value, end value and step value

k=i

END

Note: initial, end and step value must be integer. When variable is equal to end value, the loop is finished. And the step value can be negative and positive

2.6. WHILE Sentence

For example :

```
i=0
WHILE (i<10)
    i=i+1          Not for++,--,+=,-=,etc. operation
END
```

2.7. Reserved Word

The reserved words in HIS are always in capital character, for example:

"TIMER","END","MAC","SOCKBEAT","SYSTIME","RECV","CONN","DISCONN","SOCK","UART","SEND","WAIT","IF","ELSE","FOR","WHILE","INT","FLOAT","INPUT","FUNCTION","RETURN","TRUE","FALSE","STRING"

2.8. External Interface

External interface is the interactive interface between script and industrial device. It has two parts, visiting industrial device and industrial device input respectively.

2.9. Visiting Industrial Device

Reserved word MAC is the mac address of industrial device, such as:

The mac address of industrial device is "ACCF23001234"

mac= MAC Define variable mac and assigned with "ACCF23001234"

SYSTIME is seconds for reading system

t= SYSTIME

SOCKBEAT is heartbeat string of a socket, like:

bt= SOCKBEAT(netp) Read the heartbeat string named netp of Socket

SEND is to send data to some interface of industrial device, like:

SEND(UART,uart0," 1234")

SEND(SOCK,netp," 3456")

Note: the first parameter is the interface type of industrial device. The second one is the name of it while the third one is data content.

Read/Write device GPIO, like:

P10= GETHW(GPIO, 10) // Read the value of GPIO Pin10

SETHW(GPIO, OUT, 10, 1) // Set GPIO Pin10 as output high

SETHW(GPIO, IN, 10) // Set GPIO Pin10 as input

Close Socket , like :

```
DISCONN(SOCK, netp)           // netp disable connection
```

2.10. Industrial Device Input

Industrial device input interface is represented as program part. For example:

- Timer interface

```
TIMER HeartBeat 1000
    SEND(UART,uart0," 1234" )
END
```

Timer interface named as "HeartBeat" ,timing for 1000ms. END is the end mark of program part.

- Receive interface

```
RECV SOCK netp
    len=INPUT.length()
    IF (len>10)
        tmp=INPUT.subString(3,10)
    ELSE
        tmp=" ERR"
    END
    SEND(UART,uart0,tmp)
    RETURN(FALSE)
END
```

Note: Socket netp is automatically called for receiving data. INPUT is input parameter, RETURN(FALSE) indicates it is no need to transfer the original definition of ROUT at the end of this function(Typically, this function call for SEND to send data). RETURN(TRUE), it needs transfer according to the definition of ROUT and the content is the string of OUTPUT.

- RECV interface is UART recv:

```
RECV UART uart0
.....
END
```

- Socket connection interface:

```
CONN SOCK netp
.....
END
```

When Socket netp generate connection, automatically invoke.

- Disconnection interface:

```
DISCONN SOCK netp
.....
```


END

When Socket netp disconnect,automatically invoke.

2.11. Function

For example :

```
FUNCTION func1 (arg1, arg2)
    tmp1= arg1
    tmp2= arg2
    RETURN(tmp1)
END
```

Invoke :

```
TIMER xxx 1000
.....
tmp= func1( "1234" , "abcd" )
.....
END
```

2.12. Variable Scope

所有变量分为全局变量和局部变量，全局变量为程序块外定义的变量，全局变量在工控设备运行期间一直有效，可以被其他程序块访问，也可以被工控设备外部读取，如网页等。

局部变量以程序的外部接口运行为单位，一个外部接口运行期间，产生的中间变量都为局部变量，同名的变量视为同一个变量。如：

All variable can be divided into global or local. Global variable is defined out of the program block and it is always effective when the module is executing. It can be visited by other program block or be externally read by industrial device like webpage etc.

Local variable is in unit of external interface of program. During the period of executing one external interface. The intermediate variable is local variable. The variables with same name are recognized as the same variable. For example:

```
a=""
b=" 1234"
TIMER HB 1000
j=10
ret= func(a);#The visit without error, j in TIMER can be visited by func
END

CONN SOCK netp
ret=func(b)          #The visit is error because j in func is not defined
```

END

FUNCTION func (arg)

 b=arg

 c=a

 IF (j==2)

 b=" abcd"

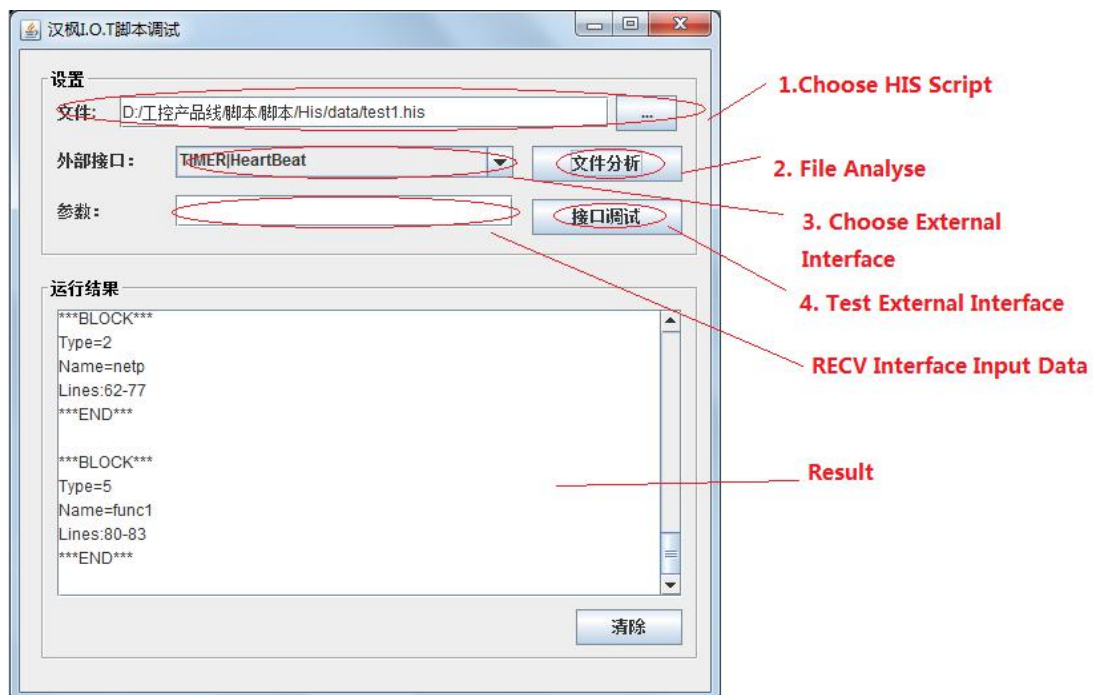
 END

END

In above code, a and b are gloable variable while j is local variable. Func is called by TIMER, local variable in func is the same as TIMER. So j can be invoked.(It is not recommended because other external interface will be error when visit func). When a external interface end, it will release all local and global variable. Two ret variable in above code is the diffierent variable.

3. HISUI INTRODUCTION

HisUI is visual debugging tool which is convenient for script debugging. Operate "HisUI.vbs" and upload file for testing.



Note: This and later tool must excute under JRE environment. Following command is to comfirm JRE environment.

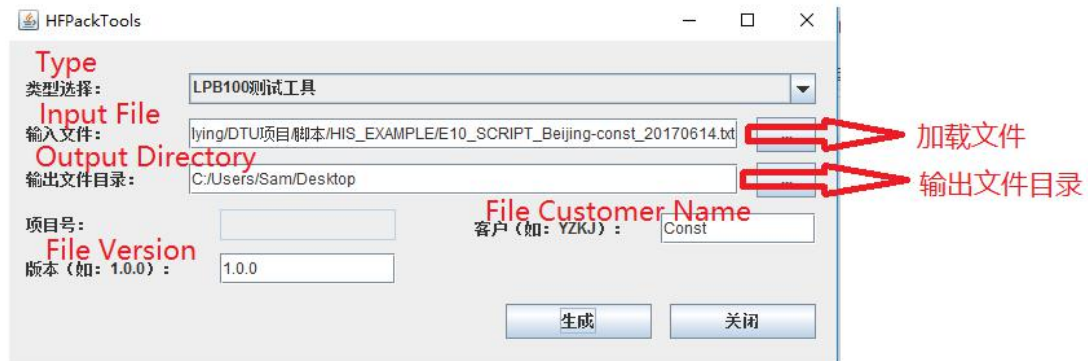
```
C:\Users\Sam>java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b15)
Java HotSpot(TM) Client VM (build 25.91-b15, mixed mode, sharing)
```

If not installed, download it in the following address.

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

4. HFPACKTOOLS INTRODUCTION

HFPackTools is script transfer tool. If script is already tested by HisUI tool, it needs transfer before download into the device. The aim of transfer is to add checkup information and confirm its completeness. Only check passed script can be downloaded successfully. Operate "HFPackTools.vbs" to open tool. After transferred, file named "Eport_Const_1.0.0_20170618.his" is derived for device to upload script.



5. SCRIPT UPDATE

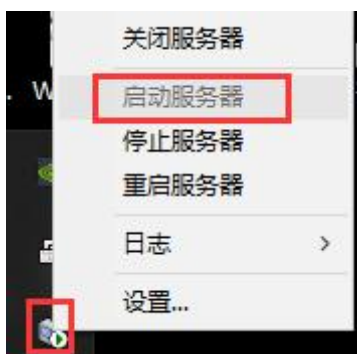
Step1 : Open local server software "EasyWebSvr.exe" and right click setting option.



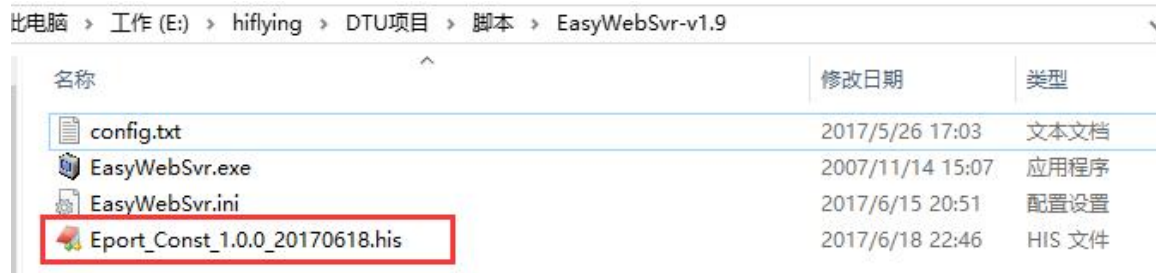
Step2 : Choose the main catalogue and start server.



Step3 : Choose start server.



Step4 : Put the derived script into this catalogue.



Step5 : Execute SecureCRT software(others as well) to enter ninto CLI mode.
Command Script is shown under SYS catalogue.

```

EPORT/SYS>
Version          Auth          Network      Telnet       web
NTP             MAC          JCMD         NAT          Ping
ProductID      CustomerID   UserID       CfgProtect   FactoryCfg
Script
EPORT/SYS>
  
```

Supported command is below:

Script Query : Script

Script Delete : Script Del

Script Download : Script http://192.168.199.100/Eport_Const_1.0.0_20170618.his

“http://192.168.199.100/Eport_Const_1.0.0_20170618.his” is download
address for script(example above)

Step6 : Execute download command and it will hint “Download Script
OK” after download. Please restart after download successfully(valid after restart)

```

EPORT/SYS>script http://192.168.199.100/Eport_Const_1.0.0_20170618.his
Download script OK!
EPORT/SYS>
  
```

Note: it can search the content of the script.

```

EPORT/SYS>script
Script file:
SETHW(GPIO, OUT, 2, 1)

low1=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x30,0x0D,0x0A]
low2=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x30,0x00]
high1=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x31,0x0D,0x0A]
high2=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x31,0x00]

RECV SOCK netp

    #设置GPIO1电平为高
    IF (INPUT==high1)
        SETHW(GPIO, OUT, 2, 1)
        RETURN(FALSE)
    END
    IF (INPUT==high2)
        SETHW(GPIO, OUT, 2, 1)
        RETURN(FALSE)
    END
    #设置GPIO1电平为低
    IF (INPUT==low1)
        SETHW(GPIO, OUT, 2, 0)
        RETURN(FALSE)
    END
    IF (INPUT==low2)
        SETHW(GPIO, OUT, 2, 0)
        RETURN(FALSE)
    END
    OUTPUT=INPUT
    RETURN(TRUE)
END
  
```

6. GPIOE10 GPIO CONTROL

6.1. Function Demand

When netp Socket channel receive fixed network data, it will control GPIO2 output high or low level. Other data is used for transparent transmission as default.

When receive "255:W:OSLED:1\0" or "255:W:OSLED:1\r\n ", GPIO2 output high level, the related hexadecimal data is following:

```
32 35 35 3A 57 3A 4F 53 4C 45 44 3A 31 00
32 35 35 3A 57 3A 4F 53 4C 45 44 3A 31 0D 0A
```

When receive "255:W:OSLED:1\0" or "255:W:OSLED:1\r\n ", GPIO2 output low level, the related hexadecimal data is following:

```
32 35 35 3A 57 3A 4F 53 4C 45 44 3A 30 00
32 35 35 3A 57 3A 4F 53 4C 45 44 3A 30 0D 0A
```

6.2. Function Achieve

Test script "E10_SCRIPT_Beijing-const_20170614.txt" is under catalogue of HIS_EXAMPLE. The content is following:

```
SETHW(GPIO, OUT, 2, 1) //初始化 GPIO2 输出高电平
```

```
low1=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x30,0x0D,0x0A]
low2=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x30,0x00]
high1=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x31,0x0D,0x0A]
high2=[0x32,0x35,0x35,0x3A,0x57,0x3A,0x4F,0x53,0x4C,0x45,0x44,0x3A,0x31,0x00]
```

```
RECV SOCK netp //接收到 netp socket 的数据
#设置 GPIO1 电平为高
IF (INPUT==high1) //比较接收到的数据和 high1
    SETHW(GPIO, OUT, 2, 1)
    RETURN(FALSE)
END

IF (INPUT==high2)
    SETHW(GPIO, OUT, 2, 1)
    RETURN(FALSE)
END

#设置 GPIO1 电平为低
IF (INPUT==low1)
    SETHW(GPIO, OUT, 2, 0)
```

```
        RETURN(FALSE)
    END

    IF (INPUT==low2)
        SETHW(GPIO, OUT, 2, 0)
        RETURN(FALSE)
    END

    OUTPUT=INPUT
    RETURN(TRUE)

```

END

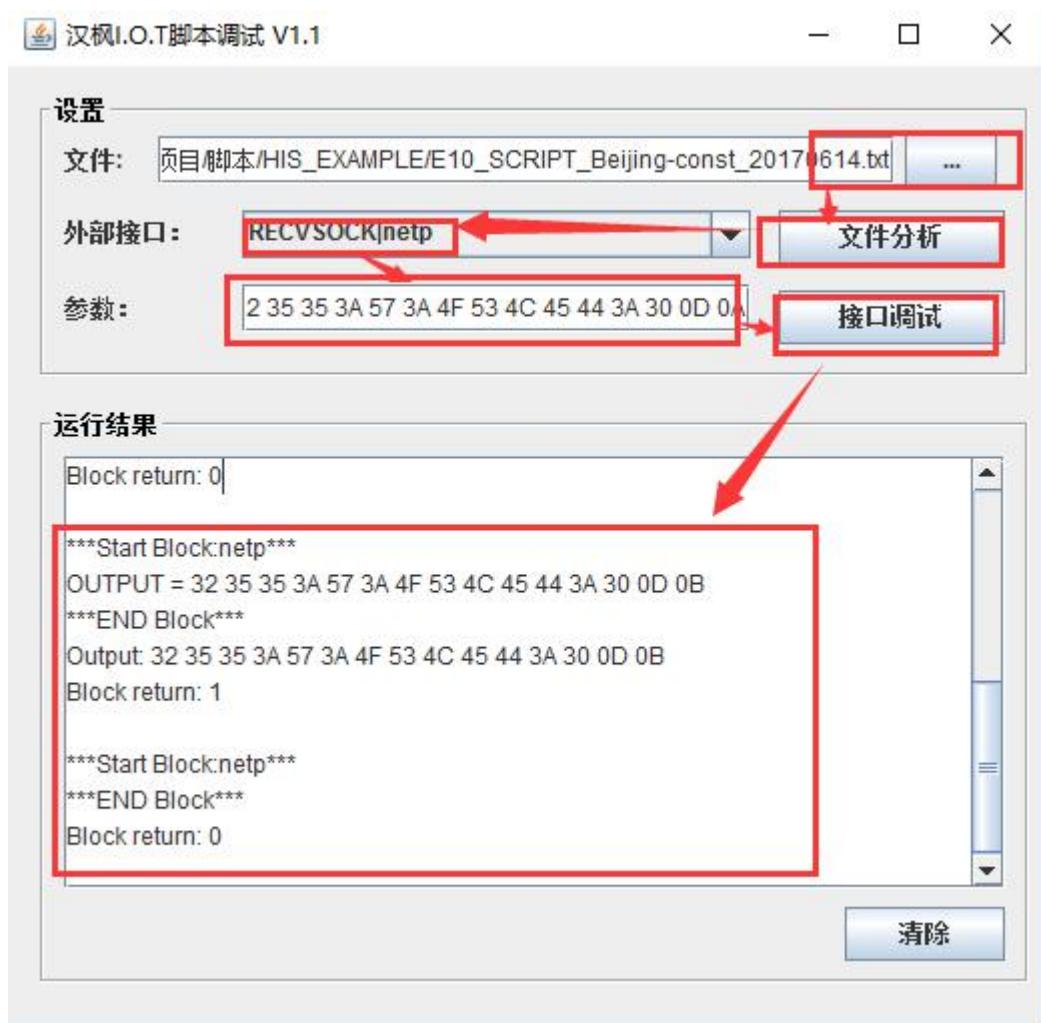
6.3. HisUI Test

Step1 : Open HisUI and upload script.

Step2 : Click [File Analyse] and choose test interface.

Step3 : Input test parameter and click [Interface Debugging] to test. If the parameters are separated by blankspace, it is recognized as hexadecimal format(as follow). Otherwise, recognized as ASCII format.

Step4 : According to the different input parameter, it can test if the function is achieved. In the following example, if Block return 1, the data can transmit normally. Otherwise, it will not transmit.

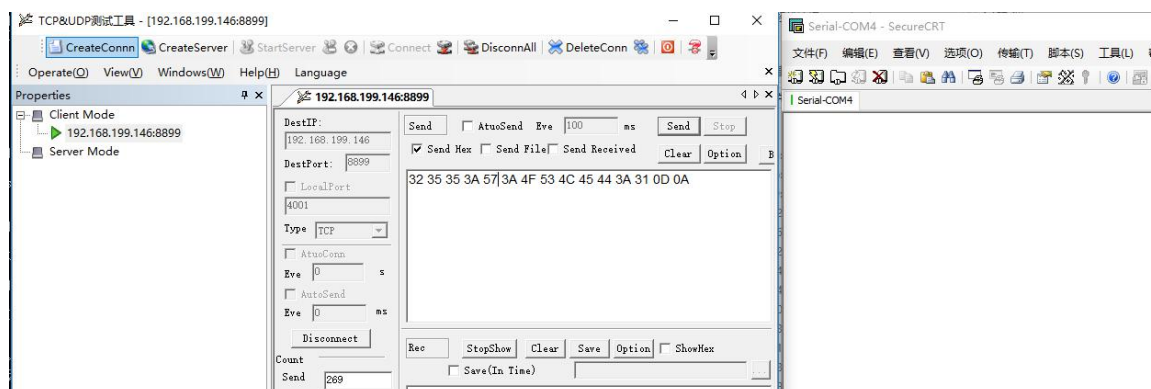


6.4. Script Transferred and Uploaded

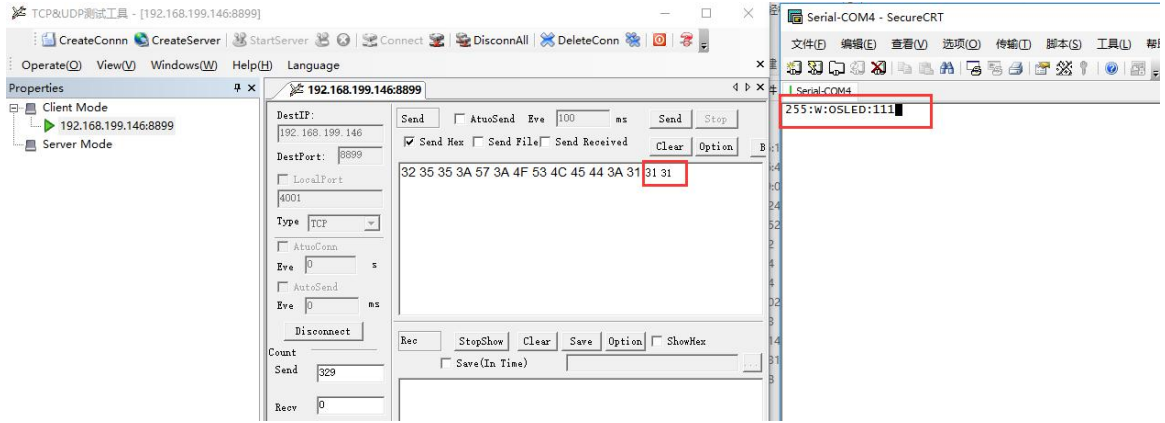
Details in chapter 4 and 5.

6.5. E10 Test

When TCP connection is built with E10, TCP debug software send following data can let GPIO2 output high level and the data will not output from serial port.



Serial port will output relative data if modified data.



APPENDIX A. HEARTBEAT CASE

A.1. Demands

When netp socket channel has conneted, serial port will send "netp connected" ." netp disconnected" when the network is shut down.

When netp Socket channel is normal, it send "hearbeat data" in every 30 seconds.

A.2. Achieve

Test script "E10_E30_HF5111B_SCRIPT_example_heartbeat.txt" is under catalogue of HIS_EXAMPLE. The content is following:

```
# "netp" //Connection Flag, 0-disconnected, 1-connected
connFlag = 0
# "netp" //Time when receive data
recvTime = 0

# "netp" //Heartbeat data
space = [0x20]
beatData = "heartbeat" + space + "data"

connStr = "netp connected\r\n"

# "netp" //Channel connected
CONN SOCK netp
connFlag=1
recvTime=SYSTIME
SEND(UART, uart0, connStr)
END

# "socka" //Channel disconnected
DISCONN SOCK netp
connFlag=0
SEND(UART, uart0, "netp disconnected\r\n")
END

RECV SOCK netp
recvTime=SYSTIME
OUTPUT=INPUT
RETURN(TRUE)
END
```

```

TIMER HeartBeat 1000
t=SYSTIME
t=t-recvTime
IF (connFlag == 1)
    IF (t > 30000)
        SEND(SOCK, netp, beatData)
        recvTime=SYSTIME
    END
END
END
END

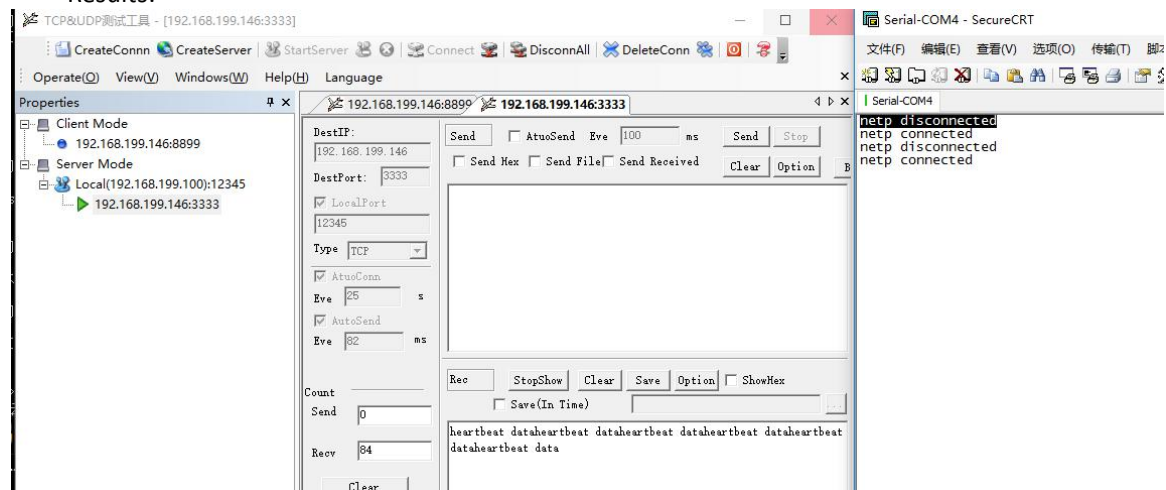
```

A.3.Script Transferred and Uploaded

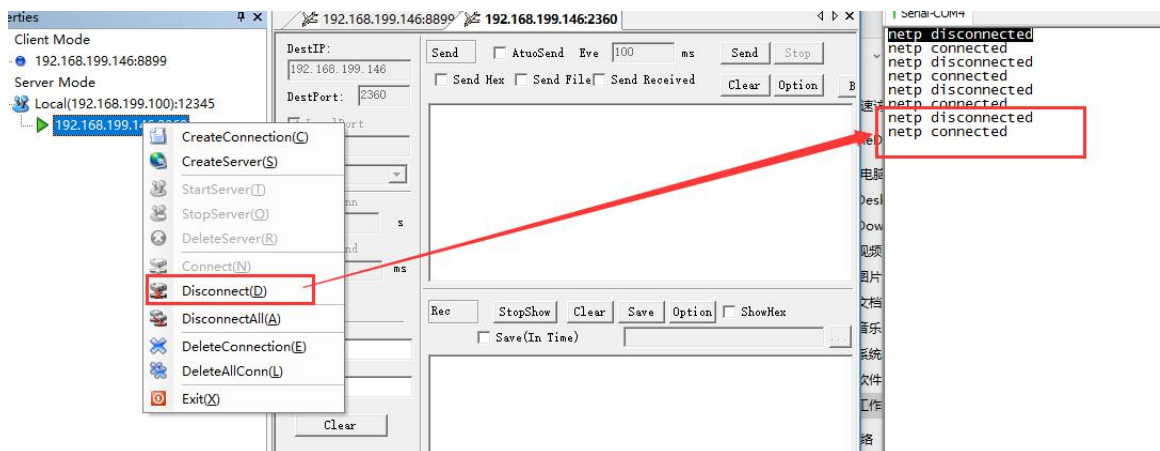
“ E10_E30_HF5111B_SCRIPT_example_heartbeat.his” is already transferrd. Download it into device to update.

A.4.Product Test

Results:



Relative serial data when click Disconnect.



APPENDIX A. REGISTER PACKAGE

B.1. Demands

Everytime when netp channel is built or sending data. Extra id information or MAC address will be put into the header for server to distinguish the device.

B.2. Achieve

“E10_E30_HF5111B_SCRIPT_example_login.txt” is under the HIS_EXAMPLE catalogue. The content of script below, default script dose not start using register package function. It can modify REGEN and REGTCP two parameter to finish this test.

```
#----Configure Parameter-----
# Register Type , 0-disable , 1-id mode , 2-MAC mode
REGEN = 0
# Register Method , 1-first connected , 2-every sending data
REGTCP = 1
# Register ID , 0-65535 , hexadecimal array format
REGID = [0x04, 0x57, 0xFB, 0xA8]
#-----

# Acquire MAC address
mac = ""
macStr = MAC
mac = macStr.stringHex()

# "netp" callback when connected
CONN SOCK netp
IF (REGTCP == 1)
    IF (REGEN == 1)
        SEND(SOCK, netp, REGID)
    END
    IF (REGEN == 2)
        SEND(SOCK, netp, mac)
    END
END
END

# callback when it receive serial data
RECV UART uart0
flag = 0
```

```
IF (REGTCP == 2)
  IF (REGEN == 1)
    OUTPUT = REGID + INPUT
    flag = 1
  END

  IF (REGEN == 2)
    OUTPUT = mac + INPUT
    flag = 1
  END
END

IF (flag == 0)
  OUTPUT = INPUT
END
RETURN(TRUE)
END
```

B.3. Other Description

The initial paramter can be modified by webpage. As a result, it is convenient to modify the function of open or close in the webpage.