Nextion Instruction Set

From ITEAD Wiki

Contents

Note:

1. The instruction is end with three bytes "0xff 0xff 0xff"

2. All the instrucitons and parametners are in **ASCII**

3. All the instrucitons are in **lowercase letters**

Classification I: Operation Commands of Component and System

page pageid

pageid: Page ID or Page Name

- Instance 1:

```
page 0         //Refresh the page with page ID of 0
```

- Instance 2:

```
page main    //Refresh the page with page name of main
```

Remarks:

The device automatically refresh page 0 when power on.

---

ref cmpID

cmpID: component ID or component name

- Instance

```
ref t0   //refresh component t0
```

Remarks:

The default loading mode is automatically load when you create and edit a component in Nextion Editor. If set it as manually load, you should use ref command to load the component. Or when the component is covered by the other components, you can use this command to refresh the covered component.

---

ref 0

ref 0: refresh all components of the current page

Remarks:

If you refresh one of the components,the component will be on the first top,but covers the others.Use this command "ref 0" will be work well.

---

get att

att: Name of Variable

- Instance 1

```
get t0.txt      //return t0's txt value
```

- Instance 2

```
get j0.val    //return j0's val value
```

```
get jo.val    //return to t0 txt data
```

Remarks:

1. When returned value is a string, the returned data is 0X70+ASCII code+0xff 0xff 0xff.

2. When returned value is numerical data, returned data is 0X71+4 byte binary data+0xff 0xff 0xff. The data storage mode is little-endian mode (namely, low-order in front, and high-order at back).

3. The specific returning format of data, please refer to the table: Format of Device Return Data

---

sendme

- Instance:

```
sendme  //No parameter
```

Remarks:

Device will immediately send the current page ID to serial port after receiving this instruction.

The specific sending format of data, please refer to the table: Format of Device Return Data.

---

cov

cov: variable type conversion

cov att1,att2,lenth

att1: source variable

att2: target variable

lenth: length of the string(0 is automatic length, not- 0 is fixed length)

- Instance 1:

```
cov h0.val,t0.txt,0  //convert the value variable of slider h0 val into decimal string and assign th
```

- Instance 2:

```
cov t0.txt,j0.val,0  //convert the string variable of t0 txt into value and assign the variable of s
```

Remarks:

1.lenth always represents the length of the string, when the value converts into string, it is the length of target variable; when the string converts into value, it is the length of

source variable.

2. If the target variable and source variable are of the same type, the conversion failed.

---

touch_j

- Instance 1:

```
touch_j   //Enter touch calibration function, this command does not need parameter
```

Remarks:

All the devices have been calibrated, this command is not needed under normal circumstances

---

cle_c //cancelled, go and refer to "tsw" command.

- Instance 1:

```
cle_c   //This command does not need parameter
```

Remarks:

When you use this command, all touch areas you set in the current page will neither be valid, nor automatically be identified.Until you use the "page" command, the touch areas can be reloaded.

---

tsw

tsw cmp,state

cmp: component name or component ID

state: state(0 or 1)

- Instance 1:

```
tsw b0,0  //component b0 touch invalid
```

- Instance 2:

```
tsw b0,1  //component b0 touch valid
```

- Instance 3:

```
tsw 1,0  //component of ID 1 touch invalid
```

- Instance 4:

```
tsw 1,1  //component of ID 1 touch valid
```

Remarks:

1.The first parameter 255 means all components in current page, for example: tsw 255,0 (all components in current page touch invalid) ; tsw 255,1 (all components in current page touch valid).

com_stop

- Instance 1:

```
com_stop  //This instruction does not need parameter
```

Remarks:

This command is used for pausing the execution of serial port commands, but note that the device will continue receiving the commands and store them in the buffer. Until receiving "com_star" commands, the device will execute the rest commands that store in the buffer.

When using this command to pause the execution, please make sure whether the buffer size and the maximum capacity of command queue can store all the commands you need. You will find these two parameter in Nextion Hardware manual.

---

com_star

- Instance 1:

```
com_star  //This instruction does not need parameter
```

Remarks:

After receiving this command, the device will execute all the commands that store in the buffer.

When using this command to recover the execution, please make sure whether the buffer size and the maximum capacity of command queue can store all the commands you need. You will find these two parameter in Nextion Hardware manual.

---

code_c

Instance 1:

```
code _c   //This instruction does not need parameter
```

Clear the commands that is stored in the buffer but not executed.

---

print att

print att

print: obtain variable value/constant value without format

print att

att: variable name

- Instance 1:

```
print t0.txt //returns the txt property value of component t0 in ASCII
```

- Instance 2:

```
print j0.val //returns val’s property value of component j0 in 4-byte hexadecimal data
```

- Instance 3:

```
print "123" //returns the ASCII of string "123":0x31 0x32 0x33
```

- Instance 4:

```
print 123 //returns the 4-byte hexadecimal data of value "123":0x7B 0x00 0x00 0x00
```

Remarks:

1. When the variable obtained by using print command is string type, the device directly returns string ASCII; if it is numeric type (such as progress val property) , the device will directly return the variable’s 4-byte hexadecimal data, value is stored as little-endian mode (ie low level in the front, high level at the back).

2. When use print command to obtain data, the device sends only the data content, no start identifier, nor end mark.

3.print command can match up printh command to add a piece of user-defined label in the front so as to tell the microcontroller which component this variable belongs to).

4. print command is very similar to get command, the only difference between them is get command returns data with initial identifier (0x70 or 0x71) and end mark (0xff 0xff 0xff), while print not.

---

printh hex

hex: the characters'hexadecimal string expression to be sent

- Instance

```
printh d0 a0 //let device sends these two bytes 0xd0 0xa0
```

Remarks:

1. print and printh commands are executed only in Nextion display, they will not be

shown in the software simulator.

2. When using printh command to send data, the device sends specified characters only, no Start character, Space character or End character.

3. There must have one and only Space separated between each set of characters in the parameter, both upper case and lower case are supported in the hexadecimal string expression.

---

add cmpID, ch, val

cmpID: Waveform component ID

ch: Waveform component channel number

val: value (maximum 255, minimum 0)

- Instance 1

```
add 1,0,30  //add data 30 to channel 0 of the Waveform component which ID number is 1
```

- Instance 2

```
add 1,1,50  //add data 50 to channel 1 of the Waveform component which ID number is 1
```

Remarks:

1.Each page supports up to four Waveform components, each Waveform component supports up to four channels. User can transmit data continuously, the component will automatically push and display the data. In the process of sending data, control properties can be modified at any time, such as modify the foreground or background color of each channel.

Classification II: GUI Designing Command

Note: When you can’t realize some special GUI designing in Nextion Editor, you can use some GUI commands to make it happen. Generally, the controls in Nextion editor can satisfy your GUI designing demand.

cls color

color: Decimal color value or color code

- Instance 1:

```
cls 1024  //Refresh the screen with decimal 1024 color value
```

- Instance 2:

```
cls RED  //Refresh the screen with the color of code RED (RED represents red color)
```

Both Decimal color value and Color code are supported in Nextion Editor

Both Decimal color value and Color code are supported in Nextion Editor

---

pic x,y,picid

x: x coordinate starting point;

y: y coordinate starting point;

picid: Picture ID;

- Instance 1:

```
pic 10,20,0  //Display the picture (ID is 0) in resource file at the coordinate (10,20)
```

- Instance 2:

```
pic 40,50,1  //Display the picture (ID is 1) in resource file at the coordinate (40,50)
```

---

picq x,y,w,h,picid

x: x coordinate starting point;

y: y coordinate starting point;

w: Region width;

h: Region height;

picid: Picture ID;

- Instance 1:

```
picq 20,50,30,20,0  //Crop the area from starting coordinate (20,50) , with a width 30×height 20 siz
```

Remarks:

This instruction requires that the background picture must be full-screen; otherwise, the image you crop is not the one you want. The crop image area and the display area is overlap on the screen.

---

xstr x,y,w,h,fontid,fontcolor,backcolor,xcenter,ycenter,sta,string

x: x coordinate starting point;

y: y coordinate starting point;

w: Region width;

h: Region height;

fontid: Font ID;

fontcolor: Font color;

backcolor: Background color (when set sta as Crop Image or Image, backcolor means image ID);

xcenter: Horizontal alignment (0 is left-aligned, 1 is centered, 2 is right-aligned);

ycenter: Vertical alignment (0 is upper-aligned, 1 is centered, 2 is lower-aligned);

sta: Background fill(0-crop image;1-solid color;2-Image, when set sta as Crop Image or Image, backcolor means image ID);

string: Character content;

- Instance 1:

```
xstr 0,0,100,30,1,RED,BLACK,1,1,1, China
```

- Explanation:

Use font 1, at the starting point coordinate(0,0), write"China" in an area that its width is 100, height is 30, the font color is RED, background color is BLACK(if you do not want any background color, use NULL), horizontal alignment is center, and vertical alignment is center too.

Remarks:

1.There is automatic word wrapping if characters exceeds the default set w. If there are remaining characters not written out when wrapped to h, they will be neglected.

2. For more information about color value, please refer to cls command.

---

fill x,y,w,h,color

x: x coordinate starting point;

y: y coordinate starting point;

w: Region width;

h: Region height;

color: Fill color;

- Instance 1:

```
fill 0,0,100,30,RED  //Fill color RED in the area of starting coordinate (0,0) and width 100×height
```

Remarks:

For more information about color value, please refer to cls command.

---

line x,y,x2,y2,color

x: x coordinate starting point;

y: y coordinate starting point;

x2: x coordinate ending point;

y2: y coordinate ending point;

color: Line color;

- Instance 1:

```
line 0,0,100,100,RED  //Draw a line in color RED between the coordinate (0,0) and the coordinate (10
```

Remarks:

For more information about color value, please refer to cls command.

---

draw x,y,x2,y2,color

x: x coordinate of starting point;

y: y coordinate of starting point;

x2: x coordinate of ending point;

y2: y coordinate of ending point;

color: Line color;

- Instance 1:

```
draw 0,0,100,100,RED    //Draw a rectangle, the top left coordinate is (0,0) and bottom right corner
```

Remarks:

1. What is drawn by draw is hollow rectangle. Please directly use area fill instruction of fill if the filled rectangle needs filling. 2. For more information about color value, please refer to cls command.

---

cir x,y,r,color

x: Coordinate x of the center of a circle

y: Coordinate y of the center of a circle

r: Radius

color: Line color;

- Instance 1:

```
cir 100,100,30,RED    //Draw a hollow circle of radius of 30 with the coordinate (100,100) as the ce
```

Remarks: For more information about color value, please refer to cls command.

Nextion Editor supports decimal color value and color code using in all GUI designing commands, for more information, please refer to Color Code List.

Nextion HMI: System Variables List

Serial Number	Name	Meaning	Instance/Remarks
1	dim	Current value of backlight b1ightness	1.dim=50 2.dim=dim+l0 3.dim=dim-l0 When you set dim=80, it means you set the brightness to 80,but doesn't save the data. Next time you power the the Nextion TFT, the brightness of backlight will keep its default setting.
2	dims	default backlight brightness when Nextion power on	1.dims=50 2.dims=dims+10 3.dims=dims-l0 When you set dims=80 to Nextion TFT, it means you set the brightness to 80, and save the data as default. Next time you power the Nextion TFT, the brightness of backlight will be 80 by default.
3	baud	Current value of baud	1.baud=2400 2.baud=4800 3.baud=9600 4.baud=19200 5.baud=38400 6.baud=57600 7.baud=115200
4	bauds	Default value of baud when Nextion power on	1.bauds=9600 is the default value of baud from factory. 2.when you set bauds=115200 to Nextion, it means you set the baud to115200, and save the data as default. Next time you power the Nextion, the value of baud will be 115200 by detault.
5	spax	Hori spacing of font display (Powered default is 0)	spax=2
6	spay	Vertical spacing of font display (	spay=2

		(Powered default is 0)	
7	thc	Brush color at touch drawing	l.thc=RED 2.thc=l024
8	thdra	Switch of touch drawing function	thdra=0(C1ose) thdra=1(Open)
9	ussp	If no serial data, it automatically activates sleep time (unit: second, minimum 3, maximum 65,535, power-on default 0)	ussp=30(No serial data within 30 seconds, it automatically enters into sleep mode) ussp=0(invalid)
10	thsp	If no touch operation, it automatically enters into sleep time (unit: second, minimum 3, maximum 65,535, power-on default 0)	thsp=30(No touch operation within 30 seconds, it automatically enters into sleep mode) thsp=0(invalid)
11	thup	Touch in sleep mode will auto-awake switch (power-on default 0)	thup=0(Touch will not automatically awake switch during sleep mode) thup=1(Touch will automatically awake switch during sleep mode) Remarks: No matter thup is 0 or 1, if any touch operation in sleep mode, the device will send touch coordinates to the serial port.
12	sendxy	Switch of real-time sending function of touch coordinate	sendxy=0(C1ose) sendxy=1(Open) Remarks: 1. When sending function is open, the device will send touch coordinate through serial port when you touch the screen. 2. Please refer to the table: Format of Device Return Data to know more about the format of sending the coordinate.
13	delay		Delay=100(Pause the device for l00ms) Remarks: When delay command is executed, the CPU of the device will not execute any command, but will continue receiving serial port command and store them into buffer.
14	sleep		sleep=0 (Exit Sleep) sleep=l (Enter Sleep) Remarks: When the device is enabled from sleep mode

14	sleep		When the device is enabled from sleep mode, the device will automatically re-fresh the current page, and the backlight brightness will recover to the default brightness value. Two commands available for changing the brightness of backlight, dim and dims.
15	bkcmd	Set serial command successful execution or fail execution data return(Default is 2)	bkcmd=0 (No return) bkcmd=l (Only return the successful result) bkcmd=2(Only return the failed result) bkcmd=3（Always return） Remarks: This setting only affects the serial command's successful execution or fail execution's data return. In Nextion Editor's editing interface, when there is a command execution error, it will return error data; when the command executes successfully, it will not return execution result data.

Conditional Statements In Nextion: if

if Conditional Statements **Note:this statement is only run in Nextion itself, add your code in the "user code" of Nextion editor.**

- Instance:

```
if(t0.txt==”123”)
{     //Note that the 2 braces must be on a single line, it does not support if internal embedded if
t1.txt=”correct password”
}
if(t0.txt!=”123”)
{
t1.txt=”incorrect password”
}
```

Remarks:

1. value type variable support:

1.1 greater than the judgment (>)

1.2 less than the judgment (<)

1.3 equal to the judgment (==)

1.4 not equal to the judgment (!=)

1.5 Greater than or equal to the judgment (>=)

1.6 less than or equal to the judgment.(<=)

2. Character string type variable only supports 1.equal to the judgment(==) 2. not equal to the judgment (!=)

3. There must be strict open parentheses and close parentheses, it does not support multiple parentheses. It only supports two variables directly judge, it does not support judge after calculation, such as: if (j0.val + 1> 0) this is not supported.

4.The brace must be on a single line, it does not support if internal embedded if, it neither supports else, nor supports if else.

5.Download "if" HMI demo here: Demo (http://support.iteadstudio.com/helpdesk/attachments/1029054663)

Nextion HMI: Color Code List

Code	Decimal System	Indicator Color
RED	63488	Red
BLUE	31	Blue
GRAY	33840	Gray
BLACK	0	Black
WHITE	65535	White
GREEN	2016	Green
BROWN	48192	Brown
YELLOW	65504	Yellow

Format of Device Return Data

Table 1: serial l instruction execution success or failure notification format

1.Only when the system variable bkcmd is not zero will return instruction execution succeed or fail data, bkcmd defaults to 0 after each power on, which means it does not return the result of instruction execution.

2.The code of the source file is not affected by bkcmd when software is under editing, the error data will be returned when there is an execution error, and the error data will not be returned when execute success.

3.The returned data is end with three bytes of “0XFF 0XFF 0XFF”.

The first byte of returned data	Meaning	Format
0X00	Invalid instruction	0X00+End
		Return this data when receiving the invalid instruction sent by the user
0X01	Successful execution of instruction	0X01+End
		Return this data when the instruction sent by the user is successfully executed

0X03	Page ID invalid	0X03+End
		Return this data when the instruction sent by the user contains invalid page ID or invalid page name
0X04	Picture ID invalid	0X04+End
		Return this data when the instruction sent by the user contains invalid picture ID
0X05	Font ID invalid	0X05+End
		Return this data when the instruction sent by the user contains invalid font ID
0X11	Baud rate setting invalid	0X11+End
		The baud rate setting instruction sent by the user contains invalid baud rate parameter
		Baud rate supported by the device including:2400 4800 9600 19200 38400 57600 115200
0X12	Curve control ID number or channel number is invalid	0X12+End
		When users use the add commands to add data to curve, this data will be returned as curve control ID number or channel number is invalid
0X1A	Variable name invalid	0X1A+End
		Return this data when the serial port receives invalid variable name
		Note: control attribute is also called variable. For example, when you set the attribute of a control, it will return this data if the unavailable attribute name is input.
0X1B	Variable operation invalid	0X1B+End
		For example, when txt attribute of text component t0 is assigned, it should be written as t0.txt="abc" It is wrong if you write t0.txt=abc.
		For another example, the val attribute of progress bar j0 should be numerical, so it should be written as j0.val=50; it will be wrong if you write j0.val="50" or j0.val=abc.

Table 2: other data return format

1.The returned data is end with three bytes of "0XFF 0XFF 0XFF".

2.The following data's return does not be affected by bkcmd.

The first byte of returned data	Meaning	Format
0X65	Touch event return data	0X65+Page ID+Component ID+TouchEvent+End
		Return this data when the touch event created by the user is pressed.
		Definition of TouchEvent: Press Event 0x01, Release Event 0X00)
		Instance: 0X65 0X00 0X02 0X01 0XFF 0XFF 0XFF'

		Instance: 0X65 0X00 0X02 0X01 0XFF 0XFF 0XFF
		Meaning: Page 0, Button 2, Press
0X66	Current page ID number returns	0X66+Page ID+End
		The device returns this data after receiving “sendme” instruction)
		Instance: 0X66 0X02 0XFF 0XFF 0XFF
		Meaning: Current page ID is 2
0X67	Touch coordinate data returns	0X67++ Coordinate X High-order+Coordinate X Low-order+Coordinate Y High-order+Coordinate Y Low-order+TouchEvent State+End
		When the system variable “sendxy” is 1, return this data at TouchEvent occurring
		Definition of TouchEvent: Press Event 0x01, Release Event 0X00
		Instance: 0X67 0X00 0X7A 0X00 0X1E 0X01 0XFF 0XFF 0XFF
		Meaning: Coordinate (122,30), Touch Event: Press
0X68	Touch Event in sleep mode	0X68++Coordinate X High-order+Coordinate X Low-order+Coordinate Y High-order+Coordinate Y Low-order+TouchEvent State+End
		When the device enters sleep mode, return this data at TouchEvent occurring
		Definition of TouchEvent: Press Event 0x01, Release Event 0X00
		Instance: 0X68 0X00 0X7A 0X00 0X1E 0X01 0XFF 0XFF 0XFF
		Meaning: Coordinate (122,30), Touch Event: Press
0X70	String variable data returns	0X70+Variable Content in ASCII code+End
		When the variable obtained through get command is string type, return this data
		Instance: 0X70 0X61 0X62 0X63 0XFF 0XFF 0XFF
		Meaning: Return the string data: “abc”
0X71	Numeric variable data returns	0X71+variable binary data(4 bytes little endian mode, low in front)+End
		When the variable obtained by get command is value, this data returns.
		Instance:0X71 0X66 0X00 0X00 0X00 0XFF 0XFF 0XFF
		Meaning:return value data:102
0X86	Device automatically enters into sleep mode	0X86+End Only when the device automatically enters into sleep mode will return this data. If execute serial command “sleep = 1” to enter into sleep mode, it will not return this data.
0X87	Device automatically	0X87+End Only when the device automatically wake up will return this data. If execute serial command “sleep=0” to wake up, it

	wake up	data. If execute serial command "sleep=0" to wake up, it will not return this data.
0X88	System successful start up	This data is sent after a successful power-on initialization on the device
0X89	Start SD card upgrade	This data is sent after the device power on and detect SD card, and then enter upgrade interface

Useful link

Discuss Nextion here: http://support.iteadstudio.com/discussions/1000058038
