Serial command set

Verr:V1.0

list

One、Instruction Set..... 2

1.1 Serial command overview 2

1.2 Basic instruction set..... 4

1.3 Configuration control instruction set..... 9

1.4 Serial HMI statement（Can only be written to the control's event when the project is edited） 18

1.5 Serial HMI system variable list..... 19

One、Instruction Set

1.1 Serial command overview

◆ Instruction format specification:

1. All instruction names and parameters are all in ASCII string format, non-binary data, easy to read and debug.
2. All instruction names use lowercase letters (only the instruction name is lowercase, regardless of the parameter size status).

◆ Color format:

The LCD module uses 16-bit true color display, the picture is exquisite, the color is rich, and the data format adopts the standard 565 color format:

65536 color setting method:

	D15	D14	D13	D12	D11		D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
65536 color	R4	R3	R2	R1	R0		G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

For example: pure red = F800 (hexadecimal) / 63488 (decimal), pure blue = 001F (hexadecimal) / 31 (decimal)

Note: The user can debug and obtain the desired color value through software. The color value can only be expressed in decimal value;

◆ Operation instructions:

The instruction set is divided into two parts: the configuration control instruction set and the basic instruction set.

The main difference between the two is that the configuration instruction set is directly object-oriented operation, and the relevant parameters of these objects are all configured in advance in the host computer software, and downloaded together with the picture to the screen memory; the basic instruction set can be understood as the most The underlying instruction set, most operations must contain coordinate, color, font and other parameter information. When the host computer interface editing software cannot fulfill

some of your special display requirements, the user uses the basic instructions to achieve the desired display effect, such as clearing the screen. In most cases, these drawing instructions are not needed. Most of the applications can be implemented through the control operations of the interface editing software.

◆ Description:

The instruction end character is two bytes of “0x0D 0x0A”. When the command is sent by the FY HMI system, the terminator is automatically added. When using the universal serial port assistant, ensure that the serial port assistant can automatically append “0x0D 0x0A”, otherwise you need to manually append “0x0D 0x0A”

1.2 Basic instruction set

Instruction	Function	Parameter	Instance	Description
cls	Clear screen	Quantity: 1 parameter 1: Decimal color value	cls 1024 (screen with a decimal 1024 color value)	1. To get the decimal data of a certain color, you can enter the software menu bar "Tools" - "Color Picker". 2. The color parameters in all the instructions in this instruction list all use the decimal color value.
pic	Brush map	Quantity: 3 parameter 1: Starting point x coordinate parameter 2: Starting point y coordinate parameter 3: Image ID	1.pic 50,50,0 (Display the image with the picture ID 0 in the resource file at coordinates (50, 50))	
xpic	Advanced cut	Quantity: 7 parameter 1: Screen start point x coordinate parameter 2: Screen start point y coordinate parameter 3: Area width	1. xpic 50,50,30,20,30,30,0 (Picture 0 start coordinate (30, 30) width 30 height 20 This area is cut to the screen display, the display start coordinate on the screen is (50, 50))	

		parameter 4: Area height parameter 5: Image starting point x coordinate parameter 6: Image starting point y coordinate parameter 7: Image ID		
picq	crop	Quantity: 5 parameter 1: Screen start point x coordinate parameter 2: Screen start point y coordinate parameter 3: Area width parameter 4: Area height parameter 5: Image ID	1.picq 20,50,30,20,0 (Picture 0 starting coordinate (0,0) width 30 height 20 This area is cut to the screen display, the display starting coordinate on the screen is (20, 50))	This command requires the image to be a full-screen image, otherwise the cut image is not what you want. The cut area on the picture overlaps with the display area on the screen.
wristr	Writing instruction	Quantity: 11 parameter 1: Starting point x coordinate parameter 2: Starting point y coordinate parameter 3: Area width	1. wristr 0,0,200,40,0,63488,1024,1,1,1,"FY HMI" (Use font 1 to write "FY HMI" in the area of starting coordinates (0,0), width 100, height 30, the font color is 63488, and the background color is 31 (if you do not want to write the background color (ie no background) you can set The sta parameter is 3), the horizontal alignment is centered,	1. When the character is written beyond the set w, it will automatically wrap. If the remaining characters are not written after the line is changed to h, it will be ignored. 2. For a description of the color values, see the notes for the cls command.

		parameter 4: Area height parameter 5: Image ID parameter 6: font color parameter 7: Background color (backcolor indicates the image ID when sta is set to cut or picture); parameter 8: Horizontal alignment (0 is left-aligned, 1 is centered, and 2 is right-aligned); parameter 9: Vertical alignment (0 is top aligned, 1 is centered, 2 is bottom aligned); parameter 10: Background fill mode (0 is cut, 1 is monochrome, 2 is picture, 3 is no background, when sta is set to cut or picture, backcolor means picture ID) parameter 11: Character content	and the vertical alignment is also centered.)	
fill	Area fill	Quantity: 5 parameter 1: Starting point x coordinate parameter 2: Starting point y coordinate	1.fill 50,50,100,30,1024 (In the starting coordinate (0,0) width 100, height 30, this area is filled with a color value of 1024)	For a description of the color values, see the notes for the cls command.

		parameter 3: Area width parameter 4: Area height parameter 5: Fill color		
line	Draw line	Quantity: 5 parameter 1: Starting point x coordinate parameter 2: Starting point y coordinate parameter 3: End point x coordinate parameter 4: End point y coordinate parameter 5: Line color value	1. line 0,0,100,100,1024 (Draw a line with a color value of 1024 colors between coordinates (0,0) and coordinates (100,100))	For a description of the color values, see the notes for the cls command.
draw	Draw rectangle	Quantity: 5 parameter 1: Starting point x coordinate parameter 2: Starting point y coordinate parameter 3: End point x coordinate parameter 4: End point y coordinate parameter 5: Line color value	1. draw 0,0,100,100,1024 (Draw a rectangle with (0,0) in the upper left corner, (100,100) in the lower right corner, and RED in color)	1. Draw draws a hollow rectangle. If you need to fill a solid rectangle, use the fill area to fill the instruction directly. 2. For a description of the color values, see the notes for the cls command.

cir	rawing a hollow circle	Quantity: 4 parameter 1: Center x coordinate parameter 2: Center y coordinate parameter 3: Circle radius parameter 4: Line color value	1. cir 100,100,30,1024 (Draw a hollow circle with a radius of 30 with the coordinates (100, 100) as the center, the color value is 1024)	For a description of the color values, see the notes for the cls command.
solcir	Draw a solid circle	Quantity: 4 parameter 1: Center x coordinate parameter 2: Center y coordinate parameter 3: Circle radius parameter 4: Line color value	1. solcir 100,100,30,1024 (Draw a hollow circle with a radius of 30 with the coordinates (100, 100) as the center, the color value is 1024)	For a description of the color values, see the notes for the cls command.

1.3 Configuration control instruction set

Instruction	Function	Parameter	Instance	Description
page	refresh page	Quantity: 1 parameter: Page ID or page name instance	1. page 0(Refresh the page with ID 0) 2. page main(Refresh the page with the name main)	When you need to jump to other interfaces, you can use the page directive to achieve
ref	Redraw control	Quantity: 1 parameter: Item ID or control name	1. ref 1(Redraw the control with ID 1) 2. ref textBox0(Redraw the control whose name is textBox0)	If a control is occluded by the content drawn by the GUI instruction or needs to be displayed after being occluded by another control, you can use ref to redraw.
click	Press/bounce event	Quantity: 2 parameter 1: Item ID or control name parameter 2: Event type, 0 is bounce and 1 is pressed	1. click btn0,0 (activates the pop-up event of the control named btn0) 2. click btn1,1 (activates the press event of the control named btn1)	The control's push/bounce event is automatically activated when touched on the screen. If you want to activate manually without touching it, use the click command.
get	Get value	Quantity: 1 parameter: Variable name	1. get textBox0.txt (returns the txt attribute value of the control textBox0) 2. get number0.val(returns the val attribute value of control number0)	
read	Get variable value / constant value	Quantity: 1 parameter 1: Variable name	1. read textBox0.txt (returns the txt attribute value of the control textBox0)	1. When the variable obtained by the read instruction is a string type, the device directly returns the string inner code. If it is a numeric type (such as the val attribute of the progress

			2. read number0.val get number0.val(returns the val attribute value of control number0) 3. read “123” (Return the constant string "123" ie: 0x31 0x32 0x33) 4. read 123(Return constant value: 123 ie: 0x7b 0x00 0x00 0x00)	bar), the device directly returns the variable 4-byte hexadecimal data (int type)), the value storage mode is little endian mode (ie, the low bit is first and the high bit is after). 2. When using the read command to get data, the device only sends the data content, there is no start identifier, and there is no terminator. 3. The read command can be combined with the readh instruction to add a custom flag to tell the microcontroller which variable belongs to which control.
readh	Get a custom hexadecimal byte	Quantity: 1 parameter: The hexadecimal string expression of the character to be sent	1.readh d0 a0(Let the device send 0xd0 0xa0 two bytes)	1. When using the readh command to send data, the device only sends the specified characters, does not send the start character, does not send a space, does not issue a terminator. 2. Each character in the parameter must have one space and only one space, and the hexadecimal string expression is supported.
vis	Hide/show control	Quantity: 2 parameter 1: Control name or control ID parameter 2: Status (0 is hidden, 1 is displayed)	1.vis button0,0(hidden button0 control) 2.vis button0,1(Show button0 control)	The first parameter is 255 for all controls on the current page, for example: vis 255, 0 (hiding all controls on the current page) vis 255, 1 (displays all controls on the current page)
tsw	Control touch enable	Quantity: 2 parameter 1: Control name or control ID parameter 2: Status (0 is hidden, 1 is displayed)	1.tsw button0,0 (Let the control whose name is button0 touch invalid) 2.tsw button0,1 (Let the control with the name button0 be valid)	The first parameter is 255 for all controls on the current page, for example: tsw 255, 0 (all controls on the current page are invalid) tsw 255, 1 (all controls on the current page are valid).

randset	Random number range setting	Quantity: 2 parameter 1: Minimum parameter 2: Maximum	1. randset 0,100 (Set the current random number generation range to a minimum of 0 and a maximum of 100)	1. Before using the random number, you need to use the randset command to set the random number generation range. If not, the default is minimum 0 and maximum 2147483647. After setting the range, each time the system variable rand is read, it will get a random number. 2. Each time the range is set using the randset command, it will remain in effect until the power is restored or the device is reset. 3. The data type of the random number setting range is int type (ie: minimum -2147483648, maximum 2147483647).
add	Add data to the curve control	Quantity: 3 parameter 1: Curve control ID number parameter 2: Curve control channel number parameter 3: (maximum 255, minimum 0)	1. add 1,0,60 (Add data to channel 0 of the curve control with ID 1)	1. Curve data only supports 8-bit data, with a minimum of 0 and a maximum of 255. 2. Each page page supports up to 4 curve controls, and each curve control supports up to 4 channels. Data can be sent continuously, and the control will automatically push the data. In the process of sending data, you can also modify the control properties at any time, such as modifying the foreground or background color of each channel at any time.
addw	Add specified waveform data to the curve control	Quantity: 5 parameter 1: Curve control ID number parameter 2: Curve control channel number parameter 3: Waveform number (0-2)	1. addw 1,0,0,100,50 (Add a sine wave data with an amplitude of 100 and a bias of 50 to the 0 channel of the curve control with ID 1)	1. The amplitude and offset values only support 8-bit data, with a minimum of 0 and a maximum of 255. 2. There are three waveform numbers: 0-sine wave; 1-square wave; 2: triangular wave;

		parameter 4: Curve waveform amplitude parameter 5: Curve waveform offset		
cle	Clear curve control data	Quantity: 2 parameter 1: Curve control ID number parameter 2: Curve control channel number	1.cle 1,0(Clear the 0 channel data of the curve control with ID 1) 2.cle 1,255(Clear all channel data for the curve control with ID 1)	1. A channel number of 255 means that all channel data in this curve control is cleared.
addt	Curve data transparent transmission instruction	Quantity: 3 parameter 1: Curve control ID number parameter 2: Curve control channel number parameter 3: The number of points for this passthrough data	1.addt 1,0,100(The curve control with ID 1 enters the data transparent transmission mode, and the number of transparent transmission points is 100 points)	1. Curve data only supports 8-bit data, with a minimum of 0 and a maximum of 255. Single passthrough data volume up to 1024 bytes 2. After the transparent transmission command is issued, the user needs to wait for the response of the device to start transparent transmission of data. After receiving the transparent transmission command, the device needs to transparently transmit the initialization data, which takes about 5ms (if there is still a lot of serial buffer before the transparent transmission instruction is executed) The instruction will be longer.) After the device transparent transmission is ready, it will send a transparent data to the user (0XFE+terminator), indicating that the device is ready, and can start sending transparent data. The transparent data is pure hexadecimal data. The string is no longer used, and the terminator is no longer needed. After the device receives the specified amount of data, the command receiving state is restored. Otherwise,

				the data is transparently transmitted. After the transparent transmission data is completed, the device will send an end tag to the user (0XFD+terminator). 3. The curve will not be refreshed until the specified number of transparent transmissions is completed, and will be automatically refreshed immediately after the transmission is completed.
doevents	Transfer system control to screen refresh	no	1.doevents	1. In the execution of a process with more instructions, or in a long-term loop statement, all control of the system is occupied by this process. Before the process ends, although the corresponding memory data can be read and written normally, the screen is not Will refresh the display, add doevents can transfer control to the screen refresh, after executing the results, the screen will refresh all the changed controls, after the refresh, the control is returned to the current process to continue. Prevent the screen from showing a display state of suspended animation. 2. In many cases, doevents are used with the while or for statements. For the use method, please refer to the instance of the while or for statement.
repageid	Get the current page ID number to the serial port	no	1.repageid	When the device receives this command, it will immediately send the ID number of the current page to the serial port. If you want to automatically send the page ID every time you refresh the page, please write the repageid statement in the initialization event of the page.

covx	Variable type conversion	Quantity: 4 parameter 1: Source variable parameter 2: Target variable parameter 3: The length of the string (0 is the automatic length, non-zero is the fixed length) parameter 4: Declared value type (0-number; 1-currency; 2-Hex)	1. covx slider0.val,textBox0.txt,0,0 (Convert the val variable of slider slider0 into a decimal substring and assign it to the txt variable of text textBox0, the length is automatic) 2. covx textBox0.txt, slider0.val,0,0 (Converts the txt decimal digit string variable of text textBox0 to a numeric value and assigns it to the val numeric variable of the slider slider0, the length is automatic)	1. lenth always indicates the length of the string. When the value is converted to a string, it is the length of the target variable. When the string is converted to a value, it is the length of the source variable. 2. If the target variable and the source variable type are the same, the conversion fails.
strlen	String variable character length test	Quantity: 2 parameter 1: String variable to test parameter 2: Assign test results to this variable	1. strlen textBox0.txt,n0.val (Assign the actual character length of the string variable textBox0.txt to n0.val)	1. strlen tests the length in characters, and btlen tests the length in bytes. For example, the length of a Chinese character tested with btlen is 2 bytes, and the length tested with strlen is 1 character. 2. The variable being tested must be a string type, and the variable to be written must be a numeric type, otherwise an error will be reported.
btlen	String variable byte length test	Quantity: 2 parameter 1: String variable to test parameter 2: Assign test results to this variable	1. btlen textBox0.txt,number0.val (Assign the actual byte length of the string variable textBox0.txt to n0.val)	1. btlen tests the length in bytes, and strlen tests the length in characters. For example, the length of a Chinese character tested with btlen is 2 bytes, and the length tested with strlen is 1 character. 2. The variable being tested must be a string type, and the variable to be written must be a numeric type, otherwise an error will be reported.

substr	String interception	Quantity: 4 parameter 1: Source variable parameter 2: Target variable parameter 3: The starting position of the character in the source variable parameter 4: Intercept string length instance	1. substr textBox0.txt,t1.txt,0,2 (Through the 0 position in textBox0.txt, 2 characters are assigned to t1.txt)	
touch_j	Touch calibration	no	touch_j (Enter touch calibration function)	All devices have been calibrated at the factory and are generally not required.
ref_stop	Pause screen refresh	no	ref_stop	1. After the screen refresh is paused, all statements will continue to be parsed and executed, and the corresponding attribute assignment operation will also run normally, but the controls on the screen will not be refreshed. Modifying any of the properties of any control will not automatically refresh the display (but the attribute is already Was modified normally). The modified control will refresh the display immediately after the device receives the recovery refresh command (ref_star). 2. After the refresh is paused, even if the ref instruction is used, it will not be refreshed immediately, until the ref_star instruction is executed, the system will be uniformly refreshed, but all gui drawing instructions (such as drawing points, lines, circles, etc.) are not affected. , will be displayed immediately.

ref_star	Restore screen refresh	no	ref_star	This directive is used in conjunction with ref_stop.
com_stop	Suspend serial command execution	no	com_stop	1. After the serial port instruction is suspended, the device will continue to accept the instruction, but it will not be executed. All of them will be placed in the instruction buffer area. After receiving the “com_star” instruction, the device will start from the time of the paused instruction to all the instructions up to the present. carried out. 2. When using the command suspend and resume function, evaluate whether the serial buffer size of your device and the maximum number of instruction cache queues are sufficient to support the number of instructions you need to cache. These two parameters can be found in the parameter list in the equipment specification you purchased.
com_star	Restore serial command execution	no	com_star	
code_c	Clear all instructions that have not been executed in the serial command buffer	no	code_c	

rest	Reset	no	rest	

1.4 Serial HMI statement（Can only be written to the control's event when the project is edited）

Statement	Instance	Description

if	1. If the system variable sys0 is equal to "100" then page 0 is refreshed
	`if(sys0 ==100)`
	`{`
	`page 0`
	`}`
		1. Numerical type variable support: 1. Greater than judgment (>) 2. Less than judgment (<) 3. Equal judgment (==) 4. Not equal to judgment (!=) 5. Greater than or equal to judgment (>=). 6. Less than or equal to judgment (<=).
		2. String type only supports 1. Equal to judgment (==) 2. Not equal to judgment (!=).

while	1. n0.val has been added to 100 until the screen is not refreshed during the self-adding process until the end of all statements in the whole process
	`while(number0.val<100)`
	`{`
	`number0.val++`
	`}`
	2. n0.val has been automatically added to 100, the screen will continue to refresh the display of the n0 control during the self-addition process.
	`while(number0.val<100)`
	`{`
	`number0.val++`
	`doevents`
		1. In the execution of a process with more instructions, or in a long-term loop statement, all control rights of the system are occupied by this process. Before the end of the process, although the corresponding memory data can be arbitrarily read and written normally, The screen will not refresh the display. After adding the doevents, you can transfer control to the screen refresh. After performing the results, the screen will refresh all the changed controls. After the refresh, the control will be returned to the current process to continue. Prevent the screen from showing a display state of suspended animation.
		2. During the loop of the while statement, the device will not respond to the touch event. The serial port command will receive the buffer, but it will not be executed until all the statements in the current process are executed. Please use it carefully to prevent an infinite loop.

	}	

for	1. number0.val is incremented by one each time, and the loop is not repeated until the end of all the statements in the loop process.
	for(number0.val=0; number0.val<100; number0.val++)
	{
	prog0.val= number0.val
	}
	2. var0.val Add one each time, loop 100 times, the screen will continuously refresh the display of prog0 control during the loop
	for(var0.val=0;var0.val<100;var0.val++)
	{
	doevents
	prog0.val=var0.val
	}
		1. In the execution of a process with more instructions, or in a long-term loop statement, all control rights of the system are occupied by this process. Before the end of the process, although the corresponding memory data can be arbitrarily read and written normally, The screen will not refresh the display. After adding the doevents, you can transfer control to the screen refresh. After performing the results, the screen will refresh all the changed controls. After the refresh, the control will be returned to the current process to continue. Prevent the screen from showing a display state of suspended animation.
		2. During the loop of the for statement, the device will not respond to the touch event. The serial port will receive the buffer, but it will not be executed until all the statements in the current process are executed. Please use it carefully to prevent an infinite loop.

1.5 Serial HMI system variable list

All variable names use lowercase characters

instruction	Function	Instance	Remarks
dp	Current page ID	1. dp=1 (Set the current page to 1, which is equivalent to page 1)	

		2. read dp(send the face page ID to the serial port) 3. number0.val=dp (The current page ID is assigned to number0.val)	
presdim	Current backlight brightness value (0-100)	1. presdim=50 2. presdim=presdim +10	
defdim	Power-on default backlight brightness value (0-100)	1.defdim=50 2.defdim=defdim +10 3.defdim=defdim -10	
presbaud	Current baud rate value	presbaud=9600	The baud rate supported by the device is :9600 19200 38400 57600 115200 230400 460800
defbaud	Power-on default baud rate value	defbaud=9600	
spax	Character display horizontal spacing (power-on default is 0)	spax=2	Only valid for characters written by the wristr instruction. The character display spacing of the control is determined by the properties inside the control.
spay	Character display vertical spacing (power-on default is 0)	spay=2	Only valid for characters written by the wristr instruction. The character display spacing of the control is determined by the internal properties of the control.
thc	Brush color when touching a drawing	1.thc=1024	
thdra	Touch drawing function switch	1.thdra=0 (close) 2.thdra=1(open)	

ussp	No serial data automatic sleep time (unit: second, minimum 3, maximum 65535, power-on default 0)	ussp=30(30 seconds without serial data automatically enters sleep mode)	
thsp	Automatic sleep time without touch operation (unit: second, minimum 3, maximum 65535, power-on default 0)	thsp=30(30 seconds without touch operation automatically enters sleep mode)	
thup	Touch the automatic wake-up switch in sleep mode (power-on default 0)	1. thup=0(Touch does not wake up automatically after sleep) 2. thup=1(Touch wake up automatically after sleep)	Regardless of whether thup is 0 or 1, the device will send touch coordinates to the serial port when there is a touch operation in sleep mode.
usup	Serial data automatic wake-up switch in sleep mode (power-on default 0)	1. usup=0(The serial port does not wake up automatically after sleep) 2. usup=1(Automatic wake-up of the serial port after sleep)	Power-on default is 0, does not customize wake-up, you need to send sleep=0 to wake up the screen, if set to 1, the serial port will automatically wake up immediately after receiving any data.
wup	Refresh page settings after wake-up from sleep	1. wup=255(Power-on default, refresh sleep before wake-up page) 2. wup=2 (Refresh the page specified page after wake-up: 2)	The device is already in sleep state, and can also perform the wup=X assignment from the serial port.
sleep	Sleep	1. sleep=0 (Exit sleep) 2. sleep=1 (Going to sleep)	In the sleep state, you can execute the following commands: get, read, readh. It is also possible to execute the assignment statement of sleep=1, wup=X, and support the upper software to be online, other instructions will not be executed. In the case of the enhanced and above series and the extended IO is configured to bind control events, no interrupt events are generated in sleep mode.

bkcmd	Set the data return of the serial command execution success or failure (power-on default is 2)	1. bkcmd=0(Do not return results) 2. bkcmd=1(Only return successful results) 3. bkcmd=2(Only return failed results) 4. bkcmd=3(Return results if successful or failed)	This setting only affects the success or failure of the serial command execution. When the command written by the upper software editing interface is executed incorrectly, the error result will be returned. If it succeeds, the execution result will not be returned. This setting also does not affect the return of data when acquiring device control data.
rexy	Get touch coordinate function switch in real time	1. rexy=0(close) 2. rexy=1(open)	After the send function is turned on, the device will acquire the touch coordinates through the serial port when the touch is pressed.
delay	Delay	delay=100 (Let the device pause 100ms)	After executing the delay instruction, the device CPU will not execute any instructions, but will continue to accept the serial port instructions to save to the serial command buffer.
rand	random number	1. presdim=rand (Assign a random number to the backlight brightness) 2. number0.val=rand (Assign a random number to the number0.val variable)	1. Before using the random number, you need to use the randset command to set the random number generation range. If not set, the default is minimum 0 and maximum 2147483647. After setting the range, each time the system variable rand is read, a random number will be obtained. 2. Each time the range is set using the randset command, it will remain in effect until the power is restored or the device is reset.
sys0 sys1 sys2	Built-in numerical variables	1. sys0=10 2. sys1=40 3. sys2=60 4. number0.val=sys2	Sys0, sys1, sys2 The three numeric variables are global types, no definition, no creation, and any page can be used at any time. The power-on default is 0, it can be read, it can be assigned, and the data type is int type (ie: minimum -2147483648, maximum 2147483647). Recommended when passing

			values between pages. Using built-in numeric variables to do operations is faster than using control property variables.
tch0-tch3	Touch coordinates	1. get tch0: Current touch coordinate X 2. get tch1: Current touch coordinate Y 3. get tch2: Coordinate X when last pressed 4. get tch3: The coordinates of the last press y	The touch coordinates can only be read, and cannot be assigned. When not pressed, the real-time coordinate data is 0.