

vLcds I2C Instruction set

Document update date	Update content
20171101	Original version
20180401	Add touch edit

Catalog

VlcdS Instruction set.....	3
1. Configuration instruction list.....	3
2. Basic instruction list.....	12

VLCDS INSTRUCTION SET

The instruction set of vLcds LCM is divided into the configuration instruction set and the basic instruction set. The configuration instruction is oriented to the control operation, the parameters of the control are pre configured in the host computer software VisualLcdStudio, and downloaded to the memory of the screen together with the image. The basic instructions are the basic display operations, including the XY coordinates, color, font size operation, etc.

1. Configuration instruction list

Function	Command	Parameter	Illustrate
Display page	0x70	Page_Id	Page_Id: page Id(1 bytes)
Read page Id	0x71	null	<p>Read the current page Id number.</p> <p>Read data format:</p> <p>0x71 0xff Page_Id Statu 0xff 0xff</p> <p>Page_Id: current page Id(1 bytes)</p> <p>Status: Read state(1 bytes)</p> <p>0x6f:read successfully</p> <p>0x65: read failure</p> <p>Note: the return value of Page_Id is 0xff, indicating that there is no page currently displayed</p>
Update text control	0x74	Control_Id +Page_Id	<p>Writes text data to a text controls.</p> <p>Control_Id: control Id(1 bytes)</p>

		+String	<p>Page_Id: page Id(1 bytes)</p> <p>String: Text data (the longest 58 bytes).</p>
Update number controls	0x6E	<p>Control_Id</p> <p>+Page_Id</p> <p>+Number</p>	<p>Write numeric data to a number controls.</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>Number: number data (2 bytes), high in the front.</p>
Upload button controls data	null	null	<p>Press the button control, the LCM will automatically upload button information.</p> <p>Upload data format:</p> <p>0x62 Control_Id Page_Id Status keytype keyvalue</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>Status: Button status (1 bytes).</p> <p>0x44:Press button</p> <p>0x55: Button release</p> <p>keytype:Button function(1 bytes)</p> <p>0x30: Character key for ASCII code</p> <p>0x31: Clear all text character functions</p> <p>0x32: Delete 1 text character functions</p> <p>0x33: Add a space character</p> <p>0x34: Confirm the function key</p> <p>0x35: Switch page function</p>

			<p>0x36: Upload button Id function</p> <p>Keyvalue: The button value (1 bytes)</p>
Update Edit controls data	0x65	<p>Control_Id</p> <p>+Page_Id</p> <p>+String</p>	<p>Writes text data to edit controls</p> <p>Control_Id: Control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>String: Text data (the longest 58 bytes)</p>
Read Edit controls data	0x75	<p>Control_Id</p> <p>+Page_Id</p>	<p>Read Edit controls text data.</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>Read data format:</p> <p>0x75 Control_Id Page_Id Status StringLength String</p> <p>parameter:</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>Status: Read state(1 bytes)</p> <p>0x6f: read successfully</p> <p>0x6e: Invalid text data for Edit control</p> <p>0x65: read failure</p> <p>StringLength: String Length (the longest 58 bytes)</p> <p>String:Text string data</p> <p>Note: Edit control and button control in conjunction with, press the button to confirm the function, in order to read the text data.</p>

Upload touch Edit data	null	null	<p>The edit box control keyboard presses Enter to confirm the key.</p> <p>After the bounce, the display will pull the INT pin down, and notify the external controller to read the key information.</p> <p>Upload data format:</p> <p>0x7c Control_Id Page_Id Status keytype keyvalue</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>Status: Button status (1 bytes).</p> <p>0x55: Button release</p> <p>keytype:Button function(1 bytes)</p> <p>0x34: Confirm the function key</p> <p>Keyvalue: The button value (1 bytes)</p> <p>0xff</p>
Read touch Edit controls data	0x7C	Control_Id +Page_Id	<p>Read the edit box control keyboard for text data.</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>Read data format:</p> <p>0x7c Control_Id Page_Id Status StringLength String</p>
Update progress bar control data	0x6F	Control_Id +Page_Id +ProgressV	<p>Writes numeric values to the progress bar control.</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p>

		alue	ProgressValue: Progress bar value (1 bytes). Valid range (0~100)c
Update the check box control status	0x63	Control_Id +Page_Id +CheckBoxS tatu	Writes data to the check box controls. Control_Id: control Id(1 bytes) Page_Id: page Id(1 bytes) CheckBoxStatu: Check box state settings (1 bytes) 0x55: No selection state 0x53: Select state
Check box control data upload	null	null	Click the check box control, the screen will automatically upload the check box control state. Upload format: 0x60 Control_Id Page_Id Status keytype keyvalue Control_Id: control Id(1 bytes) Page_Id: page Id(1 bytes) Status: Fixed value 0x6f (1 bytes) key type: Fixed value 0xff (1 bytes) keyvalue: Check box status (1 bytes) 0x55: No selection state 0x53: Select state
Read the status of the checkbox	0x64	Control_Id +Page_Id	Read the status of the marquee controls. Control_Id: control Id(1 bytes) Page_Id: page Id(1 bytes)

control			<p>Read data format:</p> <p>0x64 Control_Id Page_Id Status keytype keyvalue</p> <p>parameter:</p> <p>Status: Read state(1 bytes)</p> <p>0x6f: read successfully</p> <p>0x65: read failure</p> <p>key type: Fixed value 0xff (1 bytes)</p> <p>keyvalue: Check box status (1 bytes)</p> <p>0x55: No selection state</p> <p>0x53: Select state</p>
Update CircleGauge control data	0x7A	Control_Id +Page_Id + CircleGauge Value	<p>Writes data to CircleGauge controls.</p> <p>Control_Id: control Id(1 bytes)</p> <p>Page_Id: page Id(1 bytes)</p> <p>CircleGaugeValue: CircleGauge pointer radian data (2 bytes).</p>
Update BarGauge control data	0x61	Control_Id +Page_Id + BarGaugeVa lue	<p>Writes data to BarGauge controls.</p> <p>Control_Id: control Id(1 bytes).</p> <p>Page_Id: page Id(1 bytes)</p> <p>BarGaugeValue: BarGauge level data (2 bytes). Valid range (0~100).</p>
Update WaterGauge	0x77	Control_Id +Page_Id	<p>Writes data to WaterGauge controls.</p> <p>Control_Id: control Id(1 bytes).</p>

control data		+ WaterGaug eValue	Page_Id: page Id(1 bytes) WaterGaugeValue: WaterGauge level data (2 bytes). Valid range (0~100).
Update thermometer control data	0x6D	Control_Id +Page_Id + Temperatur eValue	Writes data to thermometer controls. Control_Id: control Id(1 bytes). Page_Id: page Id(1 bytes). TemperatureValue: Thermometer data (2 bytes). Valid range (0~120).
Update battery control data	0x79	Control_Id +Page_Id + BatteryValu e	Writes data to battery controls. Control_Id: control Id(1 bytes). Page_Id: page Id(1 bytes). BatteryValue: Battery capacity data (2 bytes). Valid range (0~100).
Update curve waveform control data	0x66	Control_Id +Page_Id + Channel_No +WaveForm Value	Write a data point to the curve waveform control, and refresh the display. Control_Id: control Id(1 bytes). Page_Id: page Id(1 bytes). Channel_No: Curve channel number (1 bytes). 0x01: Update data with curve 1 0x02: Update data with curve 2 WaveFormValue: Curve waveform data (1 bytes). Valid range (0~250).

Clear curve waveform control data	0x76	Control_Id +Page_Id	Clear the curve waveform control data and clear display. Control_Id: control Id(1 bytes). Page_Id: page Id(1 bytes).
Input curve waveform control data point	0x6A	Control_Id +Page_Id + Channel_No +WaveForm Value	Input waveform control a data point, but not refresh the curve. Control_Id: control Id(1 bytes). Page_Id: page Id(1 bytes). Channel_No: Curve channel number (1 bytes). 0x01: Curve 1 input data 0x02: Curve 2 input data WaveFormValue: Curve waveform data (1 bytes). Valid range (0~250).
Refreshing curve waveform control data	0x6B	Control_Id +Page_Id + Channel_No	Refresh the curve from the new display. Control_Id: control Id(1 bytes). Page_Id: page Id(1 bytes). Channel_No: Curve channel number (1 bytes). 0x01: Refresh curve 1 0x02: Refresh curve 2 0x03: Simultaneous update of curves 1 and 2
Update slider control data	0x5F	Control_Id +Page_Id + SliderValue	Write data to slider controls. Control_Id: control Id(1 bytes). Page_Id: page Id(1 bytes). SliderValue: Slide bar data (1 bytes). Valid range (0~100).

Read slider control data	0x68	Control_Id +Page_Id	<p>Reads the slider controls current slider data.</p> <p>Control_Id: control Id(1 bytes).</p> <p>Page_Id: page Id(1 bytes).</p> <p>Read data format:</p> <p>0x68 Control_Id Page_Id Status keytype keyvalue</p> <p>parameter:</p> <p>Status: Read state(1 bytes).</p> <p>0x6f: read successfully</p> <p>0x65: read failure</p> <p>keytype: Fixed value 0xff (1 bytes).</p> <p>keyvalue: Current sliding bar data (1 bytes). Valid range (0~100).</p>
Data upload of slider control	null	null	<p>After dragging the slider control, the slider is released and the display will automatically upload the position data of the slider control.</p> <p>Upload format:</p> <p>0x67 Control_Id Page_Id Status keytype keyvalue</p> <p>Control_Id: control Id(1 bytes).</p> <p>Page_Id: page Id(1 bytes).</p> <p>Status: Fixed value 0x55 (1 bytes).</p> <p>key type: Fixed value 0xff (1 bytes).</p> <p>keyvalue: Position data (1 bytes), range 0~100.</p>

2. BASIC INSTRUCTION LIST

Function	Command	Parameter	Illustrate
Connect LCM	0x48	null	The LCM receives the connection command and returns the data to the controller, indicating the online state.
LCM reset	0x4A	0x52+0x65+0x73 +0x65+0x74	LCM reset, clear internal data and display.
Clear screen	0x43	BackColor	Specify color clear screen. BackColor: RGB565 color value (2 bytes), high position in front.
Display image	0x49	X +Y + Image_No	Display image with specified coordinates. X: X address coordinates (2 bytes), high in the front Y: Y address coordinates (2 bytes), high in the front Image_No: Image number (1 bytes).
Image cut display	0x4F	Image_X + Image_Y +Image_W +Image_H + Image_No	Cutting the full area of the full screen image. Image_X: Cut the image starting X address (2 bytes), high in the front. Image_Y: Cut the image starting Y address (2 bytes), high in the front.

			<p>Image_W: The width of the cut image (2 bytes) and the high position in front.</p> <p>Image_H: The height of the image is cut (2 bytes) and the high position is in front.</p> <p>Image_No: image number (1 bytes)</p>
Text display	0x53	<p>X</p> <p>+Y</p> <p>+DisplayType</p> <p>+FontSize</p> <p>+ Image_No</p> <p>+BackColor</p> <p>+ForeColor</p> <p>+String</p>	<p>Specify coordinate display text.</p> <p>X: X address (2 bytes), high in front..</p> <p>Y: Y address (2 bytes), high in front.</p> <p>DisplayType: Display type (1 bytes).</p> <p>0x00: Transparent(transparent display).</p> <p>0x01: Color(with color background display).</p> <p>0x02: CutImage(cut image background display).</p> <p>FontSize: Text size number (1 bytes).</p> <p>0x10: Height 16 font.</p> <p>0x18: Height 24 font.</p> <p>0x20: Height 32 font.</p> <p>0x30: Height 48 font.</p> <p>0x40: Height 64 font.</p> <p>Image_No: image number (1 bytes).</p> <p>BackColor: Background color, RGB565 color</p>

			<p>value (2 bytes), high position in front..</p> <p>ForeColor: Text color, RGB565 color value (2 bytes), high position in front.</p> <p>String: String (the longest 48 bytes), high in the front.</p>
Message prompt display	0x4D	<p>FontSize</p> <p>+DelayTime</p> <p>+String</p>	<p>Message display with form.</p> <p>FontSize: Message literal size number (1 bytes)</p> <p>0x18: Height 24 font.</p> <p>0x20: Height 32 font.</p> <p>DelayTime: The message displays delay time, in seconds (1 bytes).</p> <p>String: Message string (the longest 58 bytes), high in the front..</p>
Draw rectangle box	0x52	<p>X +Y</p> <p>+Width</p> <p>+Height</p> <p>+ ForeColor</p>	<p>Draw rectangle box at the specified location</p> <p>X: X address (2 bytes), high in the front.</p> <p>Y:Y address (2 bytes), high in the front.</p> <p>Width: Width of rectangular box.</p> <p>Height: Height of rectangular box.</p> <p>ForeColor: Rectangle box color, RGB565 color value (2 bytes), high position in front.</p>
Draw fill	0x46	X +Y	Fill color in specified area

		+Width +Height + ForeColor	X: X address (2 bytes), high in the front. Y: Y address (2 bytes), high in the front. Width: Rectangular width. Height: Rectangle height. ForeColor: Fill in color, RGB565 color value (2 bytes), high position in front.
Draw line	0x4C	X1+Y1 +X2+Y2 + ForeColor	Draw a straight line at the specified position. X1: The line start X coordinates (2 bytes), and the high position is in front.. Y1: The line start Y coordinates (2 bytes), and the high position is in front.. X2: The line ends the X coordinate (2 bytes) and the high position is in front.. Y2: The line ends the Y coordinate (2 bytes) and the high position is in front. ForeColor: Linear color, RGB565 color value (2 bytes), high position in front.
Backlight brightness adjustment	0x42	LightValue	Adjust backlight brightness. LightValue: Range 0~100 (1 bytes). 0: Turn off backlight 100: Backlight most bright. Note: with touch screen, when the backlight

			is turned off and the touch function is opened, the backlight will automatically open after touch.
Resistance touch calibration	0x50	null	Resistance touch calibration switch. Note: capacitive touch does not require calibration.
Touch test	0x54	Enable	Touch test switch. Enable: Test switch (1 bytes). 0x00:disable test 0x01:enable test
Touch settings	0x51	Enable	Touch enable switch. Enable: Enable switch (1 bytes). 0x00: Close touch function 0x01: Open touch function Note: power on default opens the touch function.

