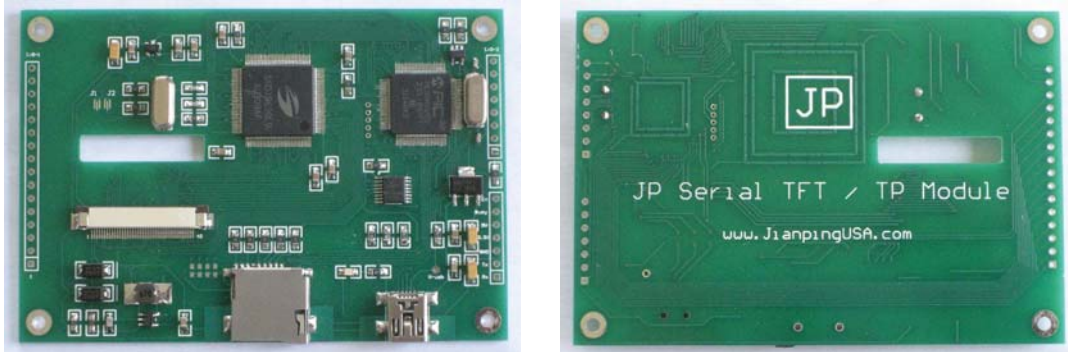


## JP Serial TFT Color LCD Display Module

(Version 2.0.0.0)



### Description:

**JP Serial TFT Color LCD Display Module** is a simple and cost effective interface controller unit. An extensive range of hardware and software peripherals have been integrated into the design, to give the user freedom to adapt the module to suit their application. Features include; micro-SD card connector, USB connector, an expansion port along with a series of GPIO and serial comm.

The JP TFT color LCD Module serves as the perfect solution to be deployed at the forefront of any product design, requiring a brilliance of color, animation or images on a 480x272 and 800x480 displays.

### Features:

Low-cost 480x272 and 800x480 LCD-TFT display graphics user interface solution.

The simplest, the easiest way build your touch screen project.

Four Wire Resistive Touch Panel.

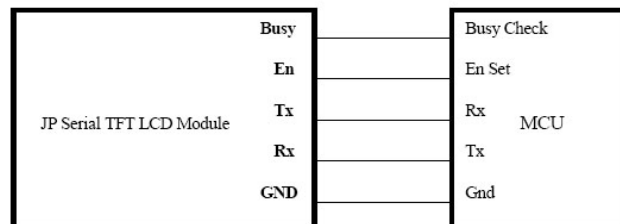
On-board micro-SD memory card adaptor and USB connector for user data logger

HC memory card support is also available for cards larger than 2GB (FAT32 format).

Load image to LCD screen from MMC card.

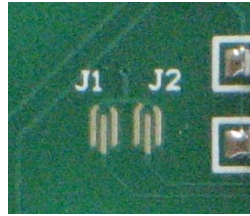
Save screen image to MMC card.

The module supports any MCU.



**JP Serial TFT LCD Module connect to MCU**

### JP TFT LCD Module Settings:



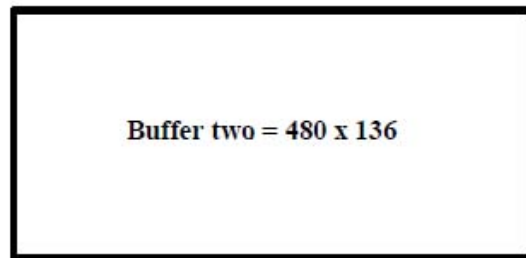
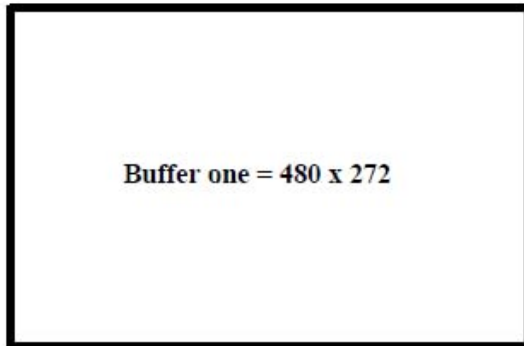
J1 is TFT LCD mode select jump, it connected for 800 x 480, unconnected for 480 x 272.  
 J2 is baud select jump, it connected for 38400, unconnected for 19200.

### JP TFT LCD connector pin order:

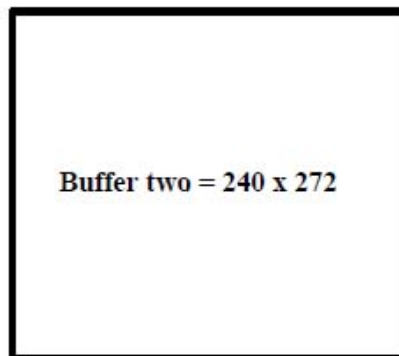
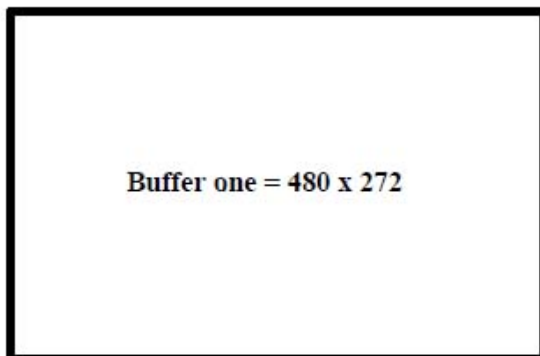
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LED-	LED+	GND	VDD	R0	R1	R2	R3	R4	R5	R6	R7	G0	G1	G2	G3	G4	G5	G6	G7
21	22	23	24	22	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
B0	B1	B2	B3	B4	B5	B6	B7	GND	PCLK	DISP	HSYNC	VSYNC	DE	NC	GND	XR	YD	XL	YU

### JP TFT LCD Module Buffer structure:

The module has two buffers for display or others. Buffer one is 65280kb, it fit 480x272 resolution display with 4bit/pix. Buffer two is 32640kb; it is 480x136 or 240x272 with 4bit/pix. (see fig)



or



## JP serial TFT LCD display module command:

1. Buffer function
2. Geometry on screen or buffer
3. Buffer shift commands
4. Data Chart Function
5. Write a character or string to buffer or LCD
6. LCD screen function
7. Meter Function
8. Touch panel function
9. MMC/SD Card function
10. Port input/output function
11. Module control function
12. Font size

### 1. Buffer function

A+,x,y

<b>Prototype</b>	A+,x,y
<b>Description</b>	This function is copy first buffer (Buffer one) to LCD Screen.
<b>Parameters</b>	<ul style="list-style-type: none"><li>• x: x coordinate of the screen location.</li><li>• y: y coordinate of the screen location.</li></ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	For 480x272, A+,0,0. For 800x480, A+,10,20
<b>Notes</b>	Buffer size is 480x272. Each Pix is 4 bits in buffer memory.

A@,Color value

<b>Prototype</b>	A@,Color value
<b>Description</b>	The function is to fill first buffer.
<b>Parameters</b>	Color value = 0 - 15
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A@,3 (3 = green)
<b>Notes</b>	Pix is 4 bits in buffer memory.

A?,Color value

<b>Prototype</b>	A?,Color value
------------------	----------------

<b>Description</b>	The function is to fill second buffer.
<b>Parameters</b>	Color value = 0 - 15
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A?,3 (3 = green)
<b>Notes</b>	Second buffer is 480x140 or 246x272. Pix is 4 bits in buffer memory.

### **A!,x\_Left,y\_bottom,x\_Right,y\_Top**

<b>Prototype</b>	A@,x_left,y_down,x_right,y_top
<b>Description</b>	Copy a partial area of the first buffer on a desired location to same place on LCD display.
<b>Parameters</b>	x_left: x coordinate of the first buffer location. y_bottom: y coordinate of the first buffer location. x_right: x coordinate of the first buffer location. Y_top: y coordinate of the first buffer location..
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A!,10,10,310,230
<b>Notes</b>	Pix is 4 bits in buffer memory.

### **A\*,x\_left,y\_bottom,x\_width,y\_height,X\_left,Y\_bottom**

<b>Prototype</b>	A*,x_left,y_bottom,x_width,y_height,X_left,Y_bottom
<b>Description</b>	Copy a partial area of the first buffer on a desired location to different place on LCD display.
<b>Parameters</b>	x_left: x coordinate of the first buffer location. y_bottom: y coordinate of the first buffer location. x_width: x coordinate of the first buffer width. Y_height: y coordinate of the first buffer height. X_left: X coordinate of LCD display location. Y_bottom: Y coordinate of LCD display location.
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A*,50,50,150,100,10,10
<b>Notes</b>	Pix is 4 bits in buffer memory.

### **A\$, data\_length**

<b>Prototype</b>	A1,data_Length
<b>Description</b>	The function is load data from MUC to First buffer
<b>Parameters</b>	

<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A\$,65280
<b>Notes</b>	Pix color is 4 bits in buffer memory.

#### A[,File\_Name

<b>Prototype</b>	A[, File_Name
<b>Description</b>	This function save a file from MMC card to first buffer
<b>Parameters</b>	
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A[,my_buffer_data
<b>Notes</b>	

#### A],File\_Name

<b>Prototype</b>	A[, File_Name
<b>Description</b>	This function load a file from MMC card to first buffer
<b>Parameters</b>	
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A[,My_MMC_File
<b>Notes</b>	

#### A^, data\_length

<b>Prototype</b>	A1,data_Length
<b>Description</b>	The function is load data from MUC to second buffer
<b>Parameters</b>	
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A^,33600
<b>Notes</b>	

#### A(,File\_Name

<b>Prototype</b>	A(, File_Name
<b>Description</b>	This function save a file from MMC card to second buffer

<b>Parameters</b>	
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A(my_buffer_data
<b>Notes</b>	

### A),File\_Name

<b>Prototype</b>	A), File_Name
<b>Description</b>	This function load a file from MMC card to second buffer
<b>Parameters</b>	
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	A),My_MMC_File
<b>Notes</b>	

## 2. Geometry on screen or buffer

### G1,n,x\_position,y\_position,color write a pix to Lcd screen or buffer

<b>Prototype</b>	G1,n,x_position,y_position,color
<b>Description</b>	Write a pix to LCD screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• x_position: x coordinate of screen or buffer</li> <li>• y_position: y coordinate of screen or buffer</li> <li>• color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G1,8,200,100,31 ( write a blue color pix to screen )
<b>Notes</b>	color volume: 4 bits for buffer, 16 bits for screen

### G2,n,x1\_position,x2\_position,y\_position,line\_size,line\_color

<b>Prototype</b>	G2,n,x1_position,x2_position,y_position,line_size,line_color
<b>Description</b>	Draw a Horizontal line to LCD screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X1_position: x coordinate of screen or buffer</li> <li>• X2_position: x coordinate of screen or buffer</li> <li>• Y_position: y coordinate of screen or buffer</li> <li>• line_size: line width.</li> </ul>

	<ul style="list-style-type: none"> <li>line_color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G2,8,10,300,100,3,31 (draw a blue line to screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### **G3,n,x1\_position,x2\_position,y\_position,line\_size,line\_color**

<b>Prototype</b>	G3,n,x1_position,x2_position,y_position,line_size,line_color
<b>Description</b>	Draw a Horizontal dash line to LCD screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>X1_position: x coordinate of screen or buffer</li> <li>X2_position: x coordinate of screen or buffer</li> <li>y_position: y coordinate of screen or buffer</li> <li>line_size: line width.</li> <li>line_color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G3,8,10,300,100,3,31 (draw a blue dash line to screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### **G4,n,x\_position,y1\_position,y2\_position,line\_size,line\_color**

<b>Prototype</b>	G4,n,x1_position,x2_position,y_position,line_size,line_color
<b>Description</b>	Draw a Vertical line to LCD screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>X_position: x coordinate of screen or buffer</li> <li>Y1_position: x coordinate of screen or buffer</li> <li>Y2_position: y coordinate of screen or buffer</li> <li>line_size: line width.</li> <li>line_color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G4,8,100,10,200,3,31 (draw a blue line to screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### **G5,n,x\_position,y1\_position,y2\_position,line\_size,line\_color**

<b>Prototype</b>	G5,n,x1_position,x2_position,y_position,line_size,line_color
<b>Description</b>	Draw a Vertical dash line to LCD screen or buffer

<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X_position: x coordinate of screen or buffer</li> <li>• Y1_position: y coordinate of screen or buffer</li> <li>• Y2_position: y coordinate of screen or buffer</li> <li>• line_size: line width.</li> <li>• line_color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G5,8,100,10,200,3,31 (draw a blue dash line to screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### G6,n,x1\_position, x2\_position,y1\_position,y2\_position,line\_color

<b>Prototype</b>	G6,n,x1_position,x2_position,y1_position,y2_position,line_color
<b>Description</b>	Draw a line to LCD screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X1_position: x coordinate of screen or buffer</li> <li>• X2_position: x coordinate of screen or buffer</li> <li>• Y1_position: y coordinate of screen or buffer</li> <li>• Y2_position: y coordinate of screen or buffer</li> <li>• line_size: line width.</li> <li>• line_color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G6,8,0,0,479,271,31 (draw a blue line to screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### G7,n,x\_center,y\_center,x\_axis,y\_axis, fill,color

<b>Prototype</b>	G7,n,x_center,y_center,x_axis,y_axis, fill,color
<b>Description</b>	Draw an ellipse to screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X_center: x coordinate of screen or buffer</li> <li>• y_center: y coordinate of screen or buffer</li> <li>• x_axis: x axis length of screen or buffer</li> <li>• Y_axis: y axis length of screen or buffer</li> <li>• Fill: fill &gt; 0 Draws a filled ellipse.</li> <li>• color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G7,8,240,136,200,130,8,31



<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen
--------------	---

### **G8,n,x1\_position,x2\_posaition,y1\_position,y2\_position,color**

<b>Prototype</b>	G8,n,x1_position,x2_posaition,y1_position,y2_position,color
<b>Description</b>	Draw an rectangle to screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X1_position: x coordinate of screen or buffer</li> <li>• X2_position: x coordinate of screen or buffer</li> <li>• Y1_position: x coordinate of screen or buffer</li> <li>• Y2_position: y coordinate of screen or buffer</li> <li>• color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G8,8,20,40,100,100,31 (draw a blue color rectangle on screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### **G9,n,x\_position, y\_position,side\_length,color**

<b>Prototype</b>	G9,n,x1_position, y1_position,side_length,color
<b>Description</b>	Draw an square to screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X_position: x coordinate of screen or buffer</li> <li>• Y_position: x coordinate of screen or buffer</li> <li>• Side_length: square side length</li> <li>• color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	G9,8,20,40,100, 31 (draw a blue color square on screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### **GA,n,x1\_position,x2\_posaition,y1\_position,y2\_position,color**

<b>Prototype</b>	GA,n,x1_position,x2_posaition,y1_position,y2_position,color
<b>Description</b>	Draw a fill box to screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X1_position: x coordinate of screen or buffer</li> <li>• X2_position: x coordinate of screen or buffer</li> <li>• Y1_position: x coordinate of screen or buffer</li> <li>• Y2_position: y coordinate of screen or buffer</li> <li>• color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>

<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	GA,8,20,40,100,100,31 (draw a blue color fill box on screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### **GB,n,Xc,Yc, Radius,Fill,color**

<b>Prototype</b>	GB,n,Xc,Yc, Radius,Fill,color
<b>Description</b>	Draw a circle to screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• Xc: x coordinate of the first buffer location</li> <li>• Yc: y coordinate of the first buffer location</li> <li>• Radius: circles radius size.</li> <li>• Fill: &gt; 0 Draws filled circle, = 0 not fill.</li> <li>• Color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	GB,8,160,120,80,8,31 (draw a filled circle on screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

### **GC,n,x\_position,y\_position, dot\_size,color**

<b>Prototype</b>	GC,n,x_position,y_position, dot_size,color
<b>Description</b>	Draw a customer size pix on screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• X_position: x coordinate of the first buffer location</li> <li>• Y_position: y coordinate of the first buffer location</li> <li>• Dot_size: a dot size.</li> <li>• Color: color volume: 4 bits for buffer, 16 bits for screen</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	GC,8,100,150,5,31 (draw a 5x5 blue dot on screen)
<b>Notes</b>	Line color is 4 bits for buffer memory,16 bits for LCD screen

## **3. Buffer shift commands**

### **JU,move\_distance,back\_color**

<b>Prototype</b>	JU,move_distance,back_color
<b>Description</b>	Move buffer position from bottom to up

<b>Parameters</b>	<ul style="list-style-type: none"> <li>• Move_distance: move distance (pix)</li> <li>• back_color: font background color = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	JU,10,0 (move buffer from bottom to up 10 pix size, back color is black)
<b>Notes</b>	color is 4 bits in buffer memory.

### JD,move\_distance,back\_color

<b>Prototype</b>	JD,move_distance,back_color
<b>Description</b>	Move buffer position from up to bottom
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• Move_distance: move distance (pix)</li> <li>• back_color: font background color = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	JD,10,0 (move buffer from up to bottom 10 pix size, back color is black)
<b>Notes</b>	color is 4 bits in buffer memory.

### JR,move\_distance,back\_color

<b>Prototype</b>	JR,move_distance,back_color
<b>Description</b>	Move buffer position from left to right
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• Move_distance: move distance (pix)</li> <li>• back_color: font background color = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	JR,10,0 (move buffer from left to right 10 pix size, back color is black)
<b>Notes</b>	color is 4 bits in buffer memory.

### JL,move\_distance,back\_color

<b>Prototype</b>	JL,move_distance,back_color
<b>Description</b>	Move buffer position from right to left
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• Move_distance: move distance (pix)</li> <li>• back_color: font background color = 0 - 15</li> </ul>
<b>Returns</b>	None

<b>Requires</b>	None.
<b>Example</b>	JL,10,0 (move buffer from right to left 10 pix size, back color is black)
<b>Notes</b>	color is 4 bits in buffer memory.

## 4. Data Chart Function

**JA,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16**

<b>Prototype</b>	JA,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16
<b>Description</b>	Reset chart lines color
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• c1 – c16: line1 to line16 color is 16 bits</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	JA,31,1024,0,63488,65504,0,0,0,0,0,0,0,0,0,0
<b>Notes</b>	Color is 16 bits.

**JC,x\_start, x\_end,l\_center,l\_step,l\_ratio,l\_size,d\_ratio,l\_number,file\_name**

<b>Prototype</b>	JC,x_start, x_end,l_center,l_step,l_ratio,l_size,d_ratio,l_number,file_name
<b>Description</b>	Draw a data chart.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x_start: data chart area x start position</li> <li>• x_end: data chart area x end position</li> <li>• l_Center: data chart area y position</li> <li>• l_step: Line step (x pix = one step)</li> <li>• l_ratio: Line ratio (x pix = data numbers)</li> <li>• l_size: Line size</li> <li>• d_ratio: data ratio ( make data to fit screen)</li> <li>• l_number: Line numbers</li> <li>• data_file_name: record data file name</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	JC,30,440,60,1,200,3,20,6,testdat6.csv
<b>Notes</b>	None

## 5. Write a character or string to buffer or LCD

**FA,n,font\_type,x\_position,y\_position,space,font\_color,back\_color,text\_string**

<b>Prototype</b>	FA,n,x_position,y_position,space,font_color,back_color,text_string
<b>Description</b>	Write a text string to LCD screen or buffer

<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• Font_type: font size and font type</li> <li>• x_position: x coordinate of screen or buffer location</li> <li>• y_position: y coordinate of screen or buffer location</li> <li>• space: adjust characters space</li> <li>• font_color: 4 bits for buffer, 16 bits for screen</li> <li>• back_color: 4 bits for buffer, 16 bits for screen</li> <li>• text_string: text string</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	FA,8,5,10,10,1,0,65535>Hello,world!
<b>Notes</b>	color is 4 bits in buffer memory.

### FB,n,font\_type,x\_position,y\_position,space,font\_color,back\_color,text\_char

<b>Prototype</b>	FA,n,x_position,y_position,space,font_color,back_color,text_string
<b>Description</b>	Write a single character to LCD screen or buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n: n = 0 write pix to buffer, n &gt; 0 write pix to LCD screen.</li> <li>• Font_type: font size and font type</li> <li>• x_position: x coordinate of screen or buffer location</li> <li>• y_position: y coordinate of screen or buffer location</li> <li>• font_color: 4 bits for buffer, 16 bits for screen</li> <li>• back_color: 4 bits for buffer, 16 bits for screen</li> <li>• text_char: text character</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	FB,8,5,100,100,0,65535,J
<b>Notes</b>	color is 4 bits for buffer memory, 16 bits for LCD screen.

## 6. LCD screen function

### L@,color

<b>Prototype</b>	L@,color
<b>Description</b>	LCD screen fill
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• color: LCD screen color, 16bits from 0 to 65535</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.

<b>Example</b>	L@,65535 (fill white color on LCD screen)
<b>Notes</b>	

### L&,x0,y0,x1,y1,color

<b>Prototype</b>	AK,x_position,y_position,font_color,back_color,char
<b>Description</b>	LCD screen partially fill
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x0_position: x coordinate of the LCD location</li> <li>• y0_position: y coordinate of the LCD location</li> <li>• x1_position: x coordinate of the LCD location</li> <li>• y1_position: y coordinate of the LCD location</li> <li>• color: LCD screen color, 16bits from 0 to 65535</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	L&,20,40,200,100,31
<b>Notes</b>	color is 4 bits in buffer memory.

### LU,x,y,Length,width, color

<b>Prototype</b>	LU,x0,y0,x1,y1, color
<b>Description</b>	Draw a up text board.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x_position: x coordinate of the LCD location</li> <li>• y_position: y coordinate of the LCD location</li> <li>• Length: length of the board</li> <li>• width: width of the board</li> <li>• color: LCD screen color, 16bits from 0 to 65535</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	LU,50,60,300,100,65535
<b>Notes</b>	

### LD,x,y,Length,width, color

<b>Prototype</b>	LD,x0,y0,x1,y1, color
<b>Description</b>	Draw a down text board.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x_position: x coordinate of the LCD location</li> <li>• y_position: y coordinate of the LCD location</li> <li>• Length: length of the board</li> <li>• width: width of the board</li> </ul>

	<ul style="list-style-type: none"> <li>color: LCD screen color, 16bits from 0 to 65535</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	LD,50,60,300,100,65535
<b>Notes</b>	

### LH,x,y,Length,width,bar\_v,bar\_color,back\_color

<b>Prototype</b>	LH,x,y,Length,width,bar_v,bar_color,back_color
<b>Description</b>	Draw a horizontal bar meter
<b>Parameters</b>	<ul style="list-style-type: none"> <li>x_position: x coordinate of the LCD location</li> <li>y_position: y coordinate of the LCD location</li> <li>Length: length of the bar</li> <li>width: width of the bar</li> <li>bar_volume bar volume</li> <li>bar color bar volume color, 16bits from 0 to 65535</li> <li>back color bar back color, 16bits from 0 to 65535</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	LH,10,10,470,50,200,31,65535
<b>Notes</b>	

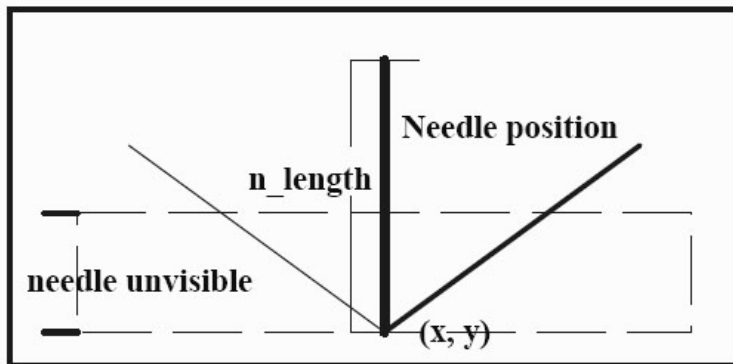
### LV,x,y,Length,width,bar\_v,bar\_color,back\_color

<b>Prototype</b>	LV,x,y,Length,width,bar_v,bar_color,back_color
<b>Description</b>	Draw a vertical bar meter
<b>Parameters</b>	<ul style="list-style-type: none"> <li>x_position: x coordinate of the LCD location</li> <li>y_position: y coordinate of the LCD location</li> <li>Length: length of the bar</li> <li>width: width of the bar</li> <li>bar_volume bar volume</li> <li>bar color bar volume color, 16bits from 0 to 65535</li> <li>back color bar back color, 16bits from 0 to 65535</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	LV,100,10,50,250,200,31,65535
<b>Notes</b>	

## 7. Meter Function

**S1,x,y,y1,n\_length,n\_width,n\_position,n\_color**

<b>Prototype</b>	S1,x,y,y1,n_length,n_width,n_position,n_color
<b>Description</b>	Draw an analog meter needle to buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x_position: x coordinate of the first buffer location</li> <li>• y_position: y coordinate of the first buffer location</li> <li>• y1: y coordinate of needle visible</li> <li>• n_length needle length</li> <li>• n_width needle size</li> <li>• needle position: meter volume(position) from 0 to 900</li> <li>• needle_color: needle color is 4 bits color = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	none
<b>Example</b>	S1,240,0,100,205,2,500,3
<b>Notes</b>	color is 4 bits in buffer memory. See fig for detail.



**S2,n\_position ,y,y1,n\_width,n\_color**

<b>Prototype</b>	S2,b_position ,y,y1,n_width,n_color
<b>Description</b>	Draw a bar meter needle to buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• n_position: meter volume(position)</li> <li>• x_position: x coordinate of the first buffer location</li> <li>• y_position: y coordinate of the first buffer location</li> <li>• n_width needle size</li> <li>• needle_color: needle color is 4 bits color = 0 - 15</li> </ul>
<b>Returns</b>	None



<b>Requires</b>	None.
<b>Example</b>	S2,240,0,100,205,2,500,3
<b>Notes</b>	color is 4 bits in buffer memory.

### S3,x0,x1,y0,y1,size,space,color

<b>Prototype</b>	S3,x0,x1,y0,y1,size,space,color
<b>Description</b>	Draw a bar meter scale to buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x0_position: x coordinate of the first buffer location</li> <li>• x1_position: x coordinate of the first buffer location</li> <li>• y0_position: y coordinate of the first buffer location</li> <li>• y1_position: y coordinate of the first buffer location</li> <li>• bar size: scale bar size (width)</li> <li>• bar soace between two bars space</li> <li>• scale bar color: scale color is 4 bits = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	s3,222,290,200,150,7,12,2
<b>Notes</b>	color is 4 bits in buffer memory.

### S4,start,end,p1,p2,color

<b>Prototype</b>	S4,start,end,p1,p2,color
<b>Description</b>	Draw an arc for meter scale to buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• start: the arc start position</li> <li>• end: the are end postion</li> <li>• p1: arc top position</li> <li>• p2: arc bottom position</li> <li>• arc color: arc color is 4 bits = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	S4,0, 900, 205, 200, 0
<b>Notes</b>	color is 4 bits in buffer memory.

### S5,start,end,p1,p2,color

<b>Prototype</b>	S4,start,end,p1,p2,color
<b>Description</b>	Draw an arc for meter scale to buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• start: the arc start position</li> <li>• end: the arc end postion</li> </ul>

	<ul style="list-style-type: none"> <li>• p1: the arc top position</li> <li>• y2: the arc bottom position</li> <li>• arc color: scale color is 4 bits = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	s5,0, 900, 190, 185, 1
<b>Notes</b>	color is 4 bits in buffer memory.

### S6,start,end,space,size,p1,p2,color

<b>Prototype</b>	S6,start,end,space,size,p1,p2,color
<b>Description</b>	Draw an arc meter scale to buffer
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• start: the arc start position</li> <li>• end: the arc end position</li> <li>• space: scale line space</li> <li>• line size: scale line size</li> <li>• p1: scale top position</li> <li>• p2: scale bottom position</li> <li>• scale color: scale color is 4 bits = 0 - 15</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	S6,0, 900, 150, 2, 185, 175, 1
<b>Notes</b>	color is 4 bits in buffer memory.

## 8. Touch panel function

### T1,num,x,y,width,high,text\_c, tx,ty,t\_type,text

<b>Prototype</b>	T1,num,x,y,width,high,text_c, tx,ty,t_type,text
<b>Description</b>	Create a user button.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• num: button ID number</li> <li>• x_position: x coordinate of button location</li> <li>• y_position: y coordinate of button location</li> <li>• width: button width</li> <li>• high: button high</li> <li>• f_color: button up text font color (16 bits color )</li> <li>• fd_color: button down text font color (16 bits color )</li> <li>• f_size: button text font size</li> <li>• fx: button text x position</li> <li>• fy: button text y position</li> <li>• text: button text string</li> </ul>

<b>Returns</b>	Button ID number
<b>Requires</b>	None.
<b>Example</b>	T1,1,120,30,100,40,0,48631,12,140,33,Erase
<b>Notes</b>	

### T2,num,x,y,tw,th,image\_up,image\_down

<b>Prototype</b>	T2,num,x,y,tw,th,image_up,image_down
<b>Description</b>	Create image button
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• num: image button ID number</li> <li>• x_position: x coordinate of button location</li> <li>• y_position: y coordinate of button location</li> <li>• tw: touch area width</li> <li>• th: touch area high</li> <li>• image_up: image button up image name</li> <li>• image_down: image button down image name</li> </ul>
<b>Returns</b>	127 + ID number
<b>Requires</b>	None.
<b>Example</b>	T2,1,336,80,46,46,keyup1,keydn1
<b>Notes</b>	<a href="#">Button image have to save in B_Image folder (direction)</a>

### T3,active

<b>Prototype</b>	T3,active
<b>Description</b>	Active user button function
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• active: volume = 0 disable, volume &gt; 0 enable</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	T3,10
<b>Notes</b>	.

### T4,active

<b>Prototype</b>	T4,active
<b>Description</b>	Active Image button function
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• active: volume = 0 disable, volume &gt; 0 enable</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	T4,10

<b>Notes</b>	.
--------------	---

### TC

<b>Prototype</b>	TC
<b>Description</b>	clean Touch panel parameter in eeprom
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	TC
<b>Notes</b>	

### TD

<b>Prototype</b>	TD
<b>Description</b>	Reload default Touch panel parameter to eeprom
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	TD
<b>Notes</b>	

### TF

<b>Prototype</b>	TF
<b>Description</b>	Touch panel point check. ( It is dead loop.)
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	TF
<b>Notes</b>	It is dead loop.

### TL

<b>Prototype</b>	TL
<b>Description</b>	Reload Touch panel default parameters.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>

<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	TL
<b>Notes</b>	

### TP,x,y,x1,y1,pen\_size,pen\_color

<b>Prototype</b>	TP,x,y,x1,y1,pen_size,pen_color
<b>Description</b>	Customer paint set.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x: paint area start of position.</li> <li>• y: paint area start of position</li> <li>• x1: paint area end of position</li> <li>• y1: paint area end of position</li> <li>• pen_size: paint pen size.</li> <li>• pen_color: paint pen color.</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	TP,50,50,280,200,5,63488 (pen_color = red)
<b>Notes</b>	

### TE,x

<b>Prototype</b>	TE,x
<b>Description</b>	Enable customer paint function
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x: x = 0 disable, x &gt; 0 enable.</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	TE,8 (function is enable.)
<b>Notes</b>	

### TT,x

<b>Prototype</b>	TT,x
<b>Description</b>	This function will send x, y parameters from Uart when TP touched.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• x: x = 0 disable, x &gt; 0 enable.</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	TT,8 (Function is enable.)

<b>Notes</b>	The function will be disabled when Button or Image button function enabled.
--------------	---

## 9. MMC/SD Card function

### M1

<b>Prototype</b>	M1
<b>Description</b>	Go to parent direction
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	M1
<b>Notes</b>	

### M2

<b>Prototype</b>	M2
<b>Description</b>	Go to root direction
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	M2
<b>Notes</b>	

### M3

<b>Prototype</b>	M3
<b>Description</b>	Show folder information
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	M3
<b>Notes</b>	This function will send MMC info from serial port.

### M4,direction\_name

<b>Prototype</b>	M4,direction_name
------------------	-------------------

<b>Description</b>	Change direction
<b>Parameters</b>	<ul style="list-style-type: none"> <li>direction_name: a direction name</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	M4,t_image
<b>Notes</b>	

#### **M5,direction\_name**

<b>Prototype</b>	M5,direction_name
<b>Description</b>	Create a new direction
<b>Parameters</b>	<ul style="list-style-type: none"> <li>direction_name: a direction name</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	M5,t_image
<b>Notes</b>	

#### **M6,direction\_name**

<b>Prototype</b>	M6,direction_name
<b>Description</b>	Delete direction
<b>Parameters</b>	<ul style="list-style-type: none"> <li>direction_name: a direction name</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	M6,t_image
<b>Notes</b>	

#### **MT,year,month,day,hour,min,sec**

<b>Prototype</b>	MT,year,month,day,hour,min,sec
<b>Description</b>	Change MMC date and time
<b>Parameters</b>	<ul style="list-style-type: none"> <li>Year, month, day</li> <li>Hour, min, second</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	MT,2014,2,15,18,38,58

<b>Notes</b>	
--------------	--

### MC,file\_name,file\_string

<b>Prototype</b>	MC,file_name,file_string
<b>Description</b>	Change MMC file and write string to this file
<b>Parameters</b>	<ul style="list-style-type: none"> <li>File name (ex. My_file.txt, my_data_csv)</li> <li>File data string (ex. Hello, world!, 12345.00,65432.23)</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	MC,your_file_name.txt,hello,world!
<b>Notes</b>	File name is 8 chars.

### MA,file\_name,file\_string

<b>Prototype</b>	MA,file_name,file_string
<b>Description</b>	append data or string to a file
<b>Parameters</b>	<ul style="list-style-type: none"> <li>File name (ex. My_file.txt, my_data_csv)</li> <li>File data string (ex. Hello, world!, 12345.00,65432.23)</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	MA,your_file_name.txt,hello,world!
<b>Notes</b>	File name is 8 chars.

### MR,file\_name

<b>Prototype</b>	MR,file_name
<b>Description</b>	Read a file
<b>Parameters</b>	<ul style="list-style-type: none"> <li>File name (ex. My_file.txt, my_data_csv)</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	MR,your_file_name.txt,
<b>Notes</b>	File name is 8 chars.

### MD,file\_name

<b>Prototype</b>	MD,file_name
<b>Description</b>	delect a file



<b>Parameters</b>	<ul style="list-style-type: none"> <li>File name (ex. My_file.txt, my_data_csv)</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	MD,your_file_name.txt,
<b>Notes</b>	

### MF,label

<b>Prototype</b>	MF,Label
<b>Description</b>	Format MMC/SD card
<b>Parameters</b>	<ul style="list-style-type: none"> <li>Label: create new label after format</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	MR,your_label
<b>Notes</b>	Warning: this function will erase ALL data on this card!

### MI,x,y,image\_name

<b>Prototype</b>	MI,x,y,image_name
<b>Description</b>	Display an image from MMC card
<b>Parameters</b>	<ul style="list-style-type: none"> <li>x_position: x coordinate of screen location</li> <li>y_position: y coordinate of screen location</li> <li>image name: image file name</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	MI,0,0,my_image
<b>Notes</b>	Image file must be 16 bits, 565 format JPG file

### MS,x,y,x1,y,image\_name

<b>Prototype</b>	MS,x,y,x1,y,image_name
<b>Description</b>	Save screen picture to MMC card
<b>Parameters</b>	<ul style="list-style-type: none"> <li>x_position: x coordinate of screen location</li> <li>y_position: y coordinate of screen location</li> <li>x1_position: x coordinate of screen location</li> <li>y1_position: y coordinate of screen location</li> <li>image name: image file name</li> </ul>
<b>Returns</b>	None

<b>Requires</b>	None.
<b>Example</b>	MS,0,0,400,250,my_image
<b>Notes</b>	Image file is 16 bits, 565 format JPG file

## 10. Port input/output function

### P1,w\_volume

<b>Prototype</b>	P1,w_volume
<b>Description</b>	Write data to port one
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• w_volume: write data to port one volume</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	P1,1 (1 = 0000000000000001)
<b>Notes</b>	Port one is 16 bits

### P2,w\_volume

<b>Prototype</b>	P2,w_volume
<b>Description</b>	Write data to port one
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• w_volume: write data to port two volume</li> </ul>
<b>Returns</b>	None
<b>Requires</b>	None.
<b>Example</b>	P1,1 (1 = 00000001)
<b>Notes</b>	Port two is 8 bits

### P3

<b>Prototype</b>	P3
<b>Description</b>	read data from port one
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	low 8bits data
<b>Requires</b>	None.
<b>Example</b>	P3
<b>Notes</b>	

### P4

<b>Prototype</b>	P4
------------------	----

<b>Description</b>	read data from port two
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	8 bits volume
<b>Requires</b>	None.
<b>Example</b>	P4
<b>Notes</b>	Port two is 8 bits

### P5,volume

<b>Prototype</b>	P5
<b>Description</b>	Set port one is input or output
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• volume: volume = 0 is output, = 1 is input</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	None.
<b>Example</b>	P4
<b>Notes</b>	Port one is 16 bits

### P6,volume

<b>Prototype</b>	P6
<b>Description</b>	Set port two is input or output
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• volume: volume = 0 is output, = 1 is input</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	None.
<b>Example</b>	P6
<b>Notes</b>	Port two is 8 bits

## 11. Module control function

### ZA,volume

<b>Prototype</b>	ZA,volume
<b>Description</b>	Adjust LCD screen back light
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• volume: volume = 0 - 15</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	none.
<b>Example</b>	ZA, 15

<b>Notes</b>	
--------------	--

### ZB,volume

<b>Prototype</b>	ZB,volume
<b>Description</b>	Adjust LCD screen back light
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• volume: volume = 0 - 65535</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	none.
<b>Example</b>	ZB, 60000
<b>Notes</b>	

### ZC

<b>Prototype</b>	ZC
<b>Description</b>	Calibrate touch panel
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	none.
<b>Example</b>	ZC
<b>Notes</b>	

### ZD, volume

<b>Prototype</b>	ZD, volume
<b>Description</b>	Set a delay for read command from MMC
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• 1 = 100ms, 2 = 200ms, 3 = 300ms, 4 = 400ms, 5 = 500ms, 6 = 1000ms</li> <li>• 7 = 1500ms, 8 = 2000ms 9 = 2500ms, 10 = 3000ms, 11 = 5000ms</li> <li>• 12 = 6000ms, 13 = 7000ms, 14 = 8000ms, 15 = 9000ms, 16 = 10000ms</li> <li>• 17 = 15000ms, 18 = 20000ms</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	none.
<b>Example</b>	ZC
<b>Notes</b>	

### ZL,loop\_number

<b>Prototype</b>	ZL,loop_number
<b>Description</b>	Set a loop for read command from MMC
<b>Parameters</b>	<ul style="list-style-type: none"> <li>Loop_number: 0 - 255</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	none.
<b>Example</b>	ZL,10 from 10 <sup>th</sup> line to do loop.
<b>Notes</b>	

### ZO,File\_Name

<b>Prototype</b>	ZO,File_Name
<b>Description</b>	Execute a command file from MMC
<b>Parameters</b>	<ul style="list-style-type: none"> <li>File_Name: an executable command file from MMC</li> </ul>
<b>Returns</b>	none
<b>Requires</b>	none.
<b>Example</b>	ZO,my_file
<b>Notes</b>	This file must be text file. (example: my_file.txt)

## 12. Font size

### Arial Narrow Font:

- 1 = Arial\_Narrow\_8x11
- 2 = Arial\_Narrow\_12x15
- 3 = Arial\_Narrow\_18x20
- 4 = Arial\_Narrow\_22x25
- 5 = Arial\_Narrow\_26x30
- 6 = Arial\_Narrow\_30x37
- 7 = Arial\_Narrow\_38x47
- 8 = Arial\_Narrow\_50x64
- 9 = Arial\_Narrow\_77x95

### Bold Arial Narrow Font:

- 11 = Bold\_Arial\_Narrow\_17x20
- 12 = Bold\_Arial\_Narrow\_23x29
- 13 = Bold\_Arial\_Narrow\_30x35
- 14 = Bold\_Arial\_Narrow\_38x46
- 15 = Bold\_Arial\_Narrow\_51x63
- 16 = Bold\_Arial\_Narrow\_76x94

### Italic Font:

- 21 = Arial\_Narrow16x19\_Italic
- 22 = Arial\_Narrow22x26\_Italic

23 = Arial\_Narrow31x36\_Italic  
24 = Arial\_Narrow40x47\_Italic  
25 = Arial\_Narrow53x63\_Italic

**Times New Roman Font:**

41 = Times\_New\_Roman20x20  
42 = Times\_New\_Roman28x25  
43 = Times\_New\_Roman37x34  
44 = Times\_New\_Roman46x44  
45 = Times\_New\_Roman63x58

**Numbers Font:**

51 = Num_15x23	normal number
52 = Dot_15x23	Dots number
53 = Seg_15x23	Seven segment number
61 = Num_36x53	normal number
62 = Dot_36x53	Dots number
63 = Seg_36x53	Seven segment number
71 = Num_50x76	normal number
72 = Dot_50x76	Dots number
73 = Seg_50x76	Seven segment number

**Customer's Font:**

77 = Second buffer font data  
88 = First buffer font data

**LIABILITY WARNING**

This device should be used only for experimental purposes. It has NOT gone through extensive testing and it could erase or corrupt some or all data on media cards that are inside the device. You assume to take your own risk when you purchase this device, and release the responsibility and liability from the manufacturer with no harm.

**REGULATORY WARNING**

This device is intended solely for experimental purposes; it is not in finished product form and is NOT FCC approved. If you wish to install these modules into non-experimental final finished products, you will be responsible to have the modules approved by the FCC at your own cost.

**Jianping Electronics**

[jp@jianpingusa.com](mailto:jp@jianpingusa.com)  
[www.JianpingUSA.com](http://www.JianpingUSA.com)