

Please power on the controller and module at the same time

Character Mode Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)
N/A	Direct ASCII string input to display (8x16 characters) (No AT command needed)	1. "012345678ABCDEFGH" 2. Wait until receive a module available byte ('E') from ezDisplay or 5ms delay	<pre>printf("012345678ABCDEFGH"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xec	Set background color of Character mode (default is 0, which is black)	1. ATec=(Background 0~255) *Background 0~255: ref to COLOR CODE Table 2. Wait until receive a module available byte ('E') from ezDisplay <RGB LED example> ATec=(11) : '11' cyan color	<pre>printf("atec=(0)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xef	Set character color of Character mode	1. ATef=(Color 0~255) *Color 0~255: ref to COLOR CODE Table 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATef=(32) : '32' red color	<pre>printf("atef=(3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x80	Write a 5X7 Character	1. AT80=(line,column,Character) 2. Wait until receive a module available byte ('E') from ezDisplay or a 5ms delay <example> AT80=(0,0,A)	<pre>printf("at80=(%d,%d,%d)",Line,Column,Character); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at80=(0,0,A)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x81	Write a 5X7 String	1. AT81=(line,column,String) 2. Wait until receive a module available byte ('E') from ezDisplay or a 5ms delay <example> AT81=(0,0,ABCD1234)	<pre>printf("at81=(%d,%d,%s)",Line,Column,String); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or</p> <pre>printf("at81=(%d,%d,%d)",Line,Column,Value); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or</p> <pre>printf("at81=(%d,%d,%f)",Line,Column,Float); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at81=(0,0,ABCD9876%\$)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0x82	Write a 8X16 Character	<p>1. AT82=(line,column,Character) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT82=(0,0,A)</p>	<pre>printf("at82=(%d,%d,%d)",Line,Column,Character); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at82=(0,0,A)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x83	Write a 8X16 String	<p>1. AT83=(line,column,String) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT83=(0,0,ABCD1234)</p>	<pre>printf("at81=(%d,%d,%s)",Line,Column,String); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or</p> <pre>printf("at81=(%d,%d,%d)",Line,Column,Value); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or</p> <pre>printf("at81=(%d,%d,%f)",Line,Column,Value); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at81=(0,0,ABCD9876%\$)"); while (USART_ReceiveData(UART1) != 'E') {}</pre>
0x84	Dsiplay a 8X8 pattern	<p>1. AT84=(X position,Y position, pattern ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT84=(16,32,1)</p>	<pre>printf("at84=(16,32,1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x85	Dsiplay a 8X16 pattern	<p>1. AT85=(X position,Y position,pattern ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT85=(16,32,1)</p>	<pre>printf("at85=(16,32,1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x86	Dsiplay a 16X16 pattern	<p>1. AT86=(X position,Y position,pattern ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT86=(16,32,1)</p>	<pre>printf("at86=(16,32,1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x20	Display firmware Revision	<p>1. AT20=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT20=()</p>	<pre>printf("at20=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

Graphic Mode Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)
0xf7	Change the configuration (resolution) mode	<p>1. ATf7=(0~4) * 0 or 1 stands for 32x16, 2 stands for 64x16 * 3 stands for 96x16, 4 stands for 128x16</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf7=(2) : Change to 64x16 configuration.</p>	<pre>printf("atf7=(2)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
N/A	<p>Under 32x16 configuration Send 32x16 Image (32x16 332/RGB bitmap) to LED Display</p> <p>(An array consist of 512 bytes bitmap)</p>	<p>1. A "for" loop to send 512 bytes user define display information 2. Wait until receive a module available byte ('E') from ezDisplay or 100ms delay</p>	<pre>for (i = 0 ; i < 512; i++) { UART_SendData(User_define_array[i]); } while (USART_ReceiveData(UART1) != 'E') {}</pre>
N/A	<p>Under 64x16 configuration Send 64x16 Image (64x16 332/RGB bitmap) to LED Display</p> <p>(An array consist of 1024 bytes bitmap)</p>	<p>1. A "for" loop to send 1024 bytes user define display information 2. Wait until receive a module available byte ('E') from ezDisplay or 100ms delay</p>	<pre>for (i = 0 ; i < 1024; i++) { UART_SendData(User_define_array[i]); } while (USART_ReceiveData(UART1) != 'E') {}</pre>
N/A	<p>Under 96x16 configuration Send 96x16 Image (96x16 332/RGB bitmap) to LED Display</p> <p>(An array consist of 1536 bytes bitmap)</p>	<p>1. A "for" loop to send 1536 bytes user define display information 2. Wait until receive a module available byte ('E') from ezDisplay or 100ms delay</p>	<pre>for (i = 0 ; i < 1536; i++) { UART_SendData(User_define_array[i]); } while (USART_ReceiveData(UART1) != 'E') {}</pre>
N/A	<p>Under 128x16 configuration Send 128x16 Image (128x16 332/RGB bitmap) to LED Display</p> <p>(An array consist of 2048 bytes bitmap)</p>	<p>1. A "for" loop to send 2048 bytes user define display information 2. Wait until receive a module available byte ('E') from ezDisplay or 100ms delay</p>	<pre>for (i = 0 ; i < 2048; i++) { UART_SendData(User_define_array[i]); } while (USART_ReceiveData(UART1) != 'E') {}</pre>
0x90	Draw a line	<p>1. AT90=(X0 position,Y0 position,X1 position,Y1 position,0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT90=(1,4,54,4,4) : '4' green color</p>	<pre>printf("at90=(%d,%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Xcoordinate2,Ycoordinate2,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at90=(1,4,54,4,4)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x91	Draw a Rectangle	<p>1. AT91=(X0 position,Y0 position,X1 position,Y1 position,0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT91=(0,0,55,7,2) : '2' blue color</p>	<pre>printf("at91=(%d,%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Xcoordinate2,Ycoordinate2,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at91=(0,0,55,7,2)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0x92	Draw a filled Rectangle	<p>1. AT92=(X0 position,Y0 position,X1 position,Y1 position,0~255) *0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT92=(1,1,54,6,32) : '32' red color</p>	<pre>printf("at92=(%d,%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Xcoordinate2,Ycoordinate2,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("at92=(0,0,55,7,2)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x93	Draw a Square	<p>1. AT93=(X position,Y position,Width,0~255) *0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT93=(48,1,4,2) : '2' blue color</p>	<pre>printf("at93=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Width,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("at93=(48,1,4,2)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x94	Draw a Circle	<p>1. AT94=(X position,Y position,Radius,0~255) *0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT94=(48,4,3,2) : '2' blue color</p>	<pre>printf("at94=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Radius,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("at94=(48,4,3,2)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x95	Draw a filled Circle	<p>1. AT95=(X position,Y position,Radius,0~255) *0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT95=(48,4,3,4) : '4' green color</p>	<pre>printf("at95=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Radius,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("at95=(48,4,3,4)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x96	Draw a tip upward Triangle	<p>1. AT96=(X position,Y position,Height,0~255) *0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><RGB LED example> AT96=(48,1,3,4) : '4' green color</p>	<pre>printf("at96=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("at96=(48,1,3,4)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x97	Draw a filled tip upward Triangle	<p>1. AT97=(X position,Y position,Height,0~255) *0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT97=(48,1,3,32) : '32' red color</p>	<pre>printf("at97=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("at97=(48,1,3,32)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0x98	Draw a tip downward Triangle	<p>1. AT98=(X position,Y position,Height,0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT98=(48,6,3,4) : '4' green color</p>	<pre>printf("at98=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at98=(48,1,3,4"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x99	Draw a filled tip downward Triangle	<p>1. AT99=(X position,Y position,Height,0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT99=(48,6,3,4) : '4' green color</p>	<pre>printf("at99=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at99=(48,1,3,4"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x9a	Draw a tip leftward Triangle	<p>1. AT9a=(X position,Y position,Width,0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT9a=(48,4,2,4) : '4' green color</p>	<pre>printf("at9a=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at9a=(48,1,3,4"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x9b	Draw a filled tip leftward Triangle	<p>1. AT9b=(X position,Y position,Width,0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p>3. Wait 2ms</p> <p><example> AT9b=(48,4,2,4) : '4' green color</p>	<pre>printf("at9b=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at9b=(48,1,3,4"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x9c	Draw a tip rightward Triangle	<p>1. AT9c=(X position,Y position,Width, 0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT9c=(48,4,2,4) : '4' green color</p>	<pre>printf("at9c=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at9c=(48,1,3,4"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x9d	Draw a filled tip rightward Triangle	<p>1. AT9d=(X position,Y position,Width,0~255)</p> <p>*0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><RGB LED example> AT9d=(48,4,2,4) : '4' green color</p>	<pre>printf("at9d=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at9d=(48,1,3,4"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0x9e	Set a Pixel as default Color	<p>1. AT9e=(X position,Y position) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT9e=(120,32)</p>	<pre>printf("at9e=(%d,%d)",Xcoordinate1,Ycoordinate1); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at9e=(12,8)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0x9f	Clear a Pixel	<p>1. AT9f=(X position,Y position) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT9f=(32,32)</p>	<pre>printf("at9f=(%d,%d)",Xcoordinate1,Ycoordinate1); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("at9f=(12,8)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

Animation Mode Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)
0xa0	Display image row by row Up Ward	<p>1. ATa0=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa0=(20)</p>	<pre>printf("ata0=(%d)",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("ata0=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa1	Display image row by row Down Ward	<p>1. ATa1=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa1=(20)</p>	<pre>printf("ata1=(%d)",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("ata1=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa2	Display image column by column Left Ward	<p>1. ATa2=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa2=(20)</p>	<pre>printf("ata2=(%d)",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly</pre> <pre>printf("ata2=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa3	Display image column by column Right Ward	<p>1. ATa3=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa3=(20)</p>	<pre>printf("ata3=(%d)",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("ata3=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0xa4	Erase image row by row Up Ward	<p>1. ATa4=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa4=(20)</p>	<pre>printf("ata4=%d",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("ata4=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa5	Erase image row by row Down Ward	<p>1. ATa5=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa5=(20)</p>	<pre>printf("ata5=%d",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("ata5=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa6	Erase image column by column Left Ward	<p>1. ATa6=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa6=(20)</p>	<pre>printf("ata6=%d",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("ata6=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa7	Erase image column by column Right Ward	<p>1. ATa7=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa7=(20)</p>	<pre>printf("ata7=%d",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("ata7=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa8	Display image Inside Out	<p>1. ATa8=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa8=(20)</p>	<pre>printf("ata8=%d",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("ata8=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xa9	Display image Outside In	<p>1. ATa9=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATa9=(20)</p>	<pre>printf("ata9=%d",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("ata9=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xaa	Erase image Inside Out	<p>1. ATaa=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATaa=(20)</p>	<pre>printf("ataa=%d",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("ataa=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0xab	Erase image Outside In	<p>1. ATab=(Speed in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATab=(20)</p>	<pre>printf("atab=(%d)",speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("atab=(20)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xac	Shift one row up	<p>1. ATac=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATac=()</p>	<pre>printf("atac=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xad	Shift one row down	<p>1. ATad=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATad=()</p>	<pre>printf("atad=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xae	Shift one column left	<p>1. ATae=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATae=()</p>	<pre>printf("atae=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xaf	Shift one column right	<p>1. ATaf=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATaf=()</p>	<pre>printf("ataf=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xb0	Fly in up-ward	<p>1. ATb0=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb0=(1,3)</p>	<pre>printf("atb0=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("atb0=(0,3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xb1	Fly in down-ward	<p>1. ATb1=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb1=(3,2)</p>	<pre>printf("atb1=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("atb1=(0,3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xb2	Fly in left-ward	<p>1. ATb2=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb2=(1,3)</p>	<pre>printf("atb2=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay or input directly printf("atb2=(3,3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0xb3	Fly in right-ward	<p>1. ATb3=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb3=(1,3)</p>	<pre>printf("atb3=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("atb3=(0,3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xb4	Fly in up-left-ward	<p>1. ATb4=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb4=(1,3)</p>	<pre>printf("atb4=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay)</pre> <p>or input directly</p> <pre>printf("atb4=(0,3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xb5	Fly in up-right-ward	<p>1. ATb5=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb5=(1,3)</p>	<pre>printf("atb5=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("atb5=(0,1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xb6	Fly in down-left-ward	<p>1. ATb6=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb6=(1,3)</p>	<pre>printf("atb6=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("atb6=(0,1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xb7	Fly in down-right-ward	<p>1. ATb7=(Page address,Speed) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATb7=(1,3)</p>	<pre>printf("atb7=(%d,%d)",Page_address,Speed); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or input directly</p> <pre>printf("atb7=(0,1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xbe	Set Page interval of multi-page animation	<p>1. ATbe=(1~10) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATbe=(5)</p>	<pre>printf("atbe=(5)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or</p> <pre>printf("atbe=(10)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xbf	Set animation speed	<p>1. ATbf=(1~10) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATbf=(5)</p>	<pre>printf("atbf=(1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or</p> <pre>printf("atbf=(10)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xd2	Scroll the whole display upward	<p>1. ATd2=(shif time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATd2=(20)</p>	<pre>printf("atd2=(30)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0xd3	Scroll the whole display downward	1. ATd3=(shif time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATd3=(20)	printf("atd3=(30)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
0xd4	Scroll the whole display leftward	1. ATd4=(shif time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATd4=(20)	printf("atd4=(30)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
0xd5	Scroll the whole display rightward	1. ATd5=(shif time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATd5=(20)	printf("atd5=(30)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
0xd6	Scroll section display upward	1. ATd6=(upper left corner X, upper left corner Y, lower right corner X, lower right corner Y,shift time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay < example> ATd6=(18,3,55,28,10)	printf("atd6=(18,3,55,28,10)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
0xd7	Scroll section display downward	1. ATd7=(upper left corner X, upper left corner Y, lower right corner X, lower right corner Y,shift time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay < example> ATd7=(18,3,55,28,10)	printf("atd7=(18,3,55,28,10)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
0xd8	Scroll section display leftward	1. ATd8=(upper left corner X, upper left corner Y, lower right corner X, lower right corner Y,shift time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay < example> ATd8=(18,3,55,28,10)	printf("atd8=(18,3,55,28,10)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
0xd9	Scroll section display rightward	1. ATd9=(upper left corner X, upper left corner Y, lower right corner X, lower right corner Y,shift time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay < example> ATd9=(18,3,55,28,10)	printf("atd9=(18,3,55,28,10)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
0xdf	Desinate the number of pages for animamtion	1. ATdf=(page number 2-7) 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATdf=(5)	printf("atdf=(2)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xfd	<p>Change the automatic animation mode</p> <p>Remark: 0 : Static display , frame fixed / no moving 1 : Single page fade in / fade out 2 : Single page scroll up 3 : Single page scroll down 4 : Single page scroll left 5 : Single page scroll right 6 : Single page scroll flash 7 : Single page fly in from top 8 : Single page fly in from bottom 9 : Single page fly in from left 10 : Single page fly in from right 11 : Single page fly in from top left 12 : Single page fly in from top right 13 : Single page fly in from bottom left 14 : Single page fly in from bottom right 15 : Single page fly in from 8 different directions</p>	<p>1. ATfd=(Remark) * 0 stands for stop animation 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATfd=(18)</p> <p>Remark: 16 : Multi pages scroll up 17 : Multi pages scroll down 18 : Multi pages scroll left 19 : Multi pages scroll right 20 : Multi pages display in sequence 21 : Multi pages fade in / fade out 22 : Multi pages fly in from top 23 : Multi pages fly in from bottom 24 : Multi pages fly in from left 25 : Multi pages fly in from right 26 : Multi pages fly in from top left 27 : Multi pages fly in from top right 28 : Multi pages fly in from bottom left 29 : Multi pages fly in from bottom right</p>	<pre>printf("atfd=(18)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
------	--	---	---

Color management Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)
0xcc	Swap displayed color to designated color for whole display	<p>1. ATcc=(Color on screen want to swapped, Designated color) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATcc=(32,3)</p>	<pre>printf("atcc=(32,3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xcf	Swap one color to another color in designated area	<p>1. ATcf=(X position,Y position,pattern's Column size, Pattern's Row size,Color code to be swapped, Color code) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATcf=(32,10,8,8,32,3)</p>	<pre>printf("atcf=(32,10,8,8,32,3)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xd0	Clear display	<p>1. ATd0=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATd0=()</p>	<pre>printf("atd0=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xd1	Show the data in the display memory	<p>1. ATd1=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATd1=()</p>	<pre>printf("atd1=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

0xee	Set a Color Pixel	<p>1. ATee=(X position,Y position, Color 0~255) *Color 0~255: ref to COLOR CODE Table</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><RGB LED example> ATee=(48,6,96) : '96' Red color</p>	<pre>printf("atee=(48,6,96)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
------	-------------------	---	--

System Control Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)
0xbd	Set Page0 EEPROM write enable flag <i>Set to disable the page 0 eeprom write flag can increase the speed of image transfer</i>	<p>1. ATbd=(0 for disable, 1 enable) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATbd=(0)</p>	<pre>printf("atbd=(1)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre> <p>or</p> <pre>printf("atbd=(0)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xf0	Turn display Off	<p>1. ATf0=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf0=()</p>	<pre>printf("atf0=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xf1	Turn display On	<p>1. ATf1=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf1=()</p>	<pre>printf("atf1=()"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xf2	Set the brightness of RGB LED Display	<p>1. ATf2=(level of brightness 0~11) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf2=(3)</p>	<pre>printf("atf2=(8)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xfc	Change the display page	<p>1. ATfc=(Page address 0~7) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATfc=(3)</p>	<pre>printf("atfc=(0)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>
0xfe	Write display contents to current displayed EEPROM page address	<p>1. ATfe=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATfe=()</p>	<pre>printf("atfe=(0)"); while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay</pre>

ezDisplay Color Code

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111