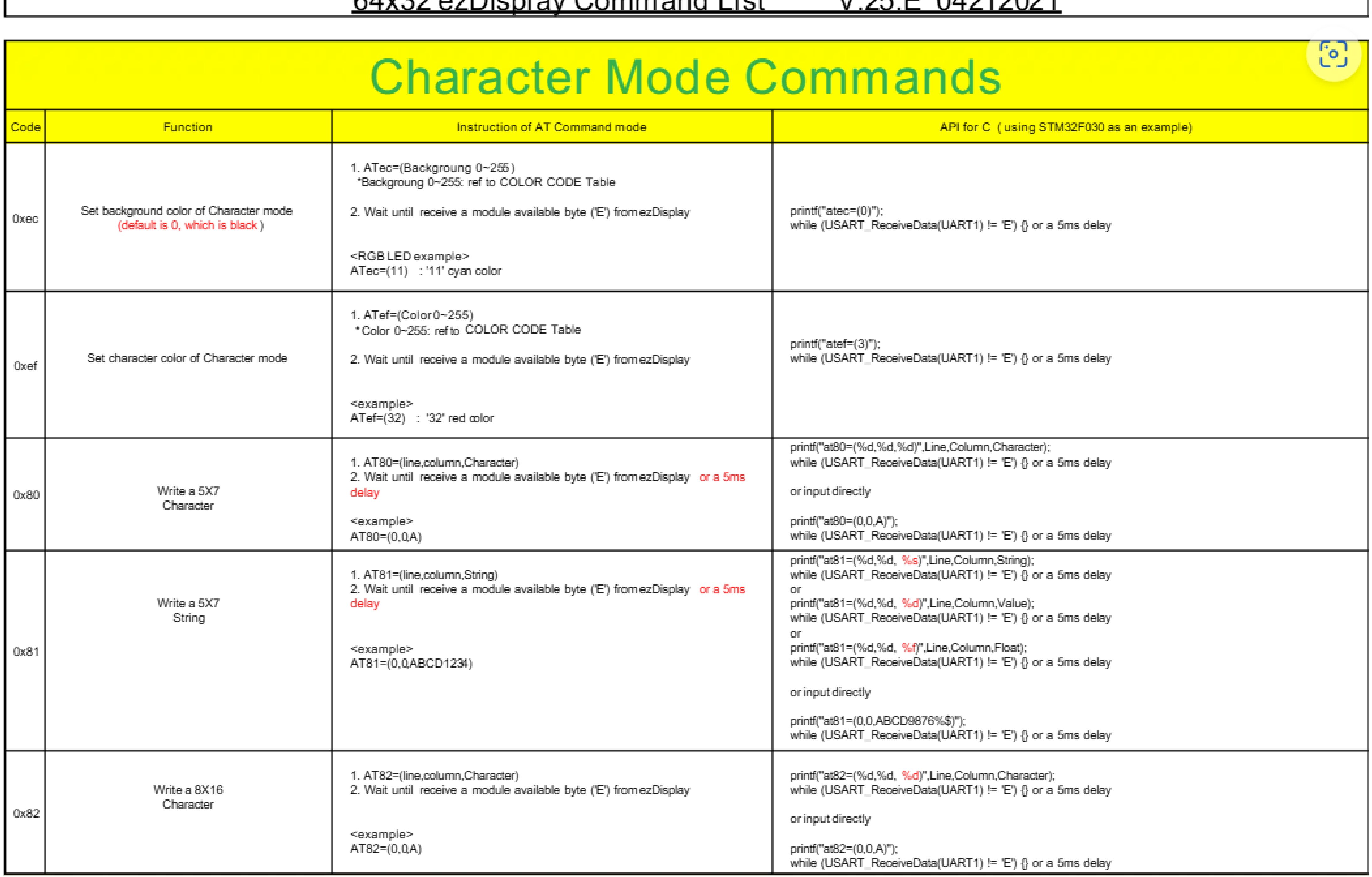


Character Mode Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)

0xec	Set background color of Character mode
	(default is 0, which is black)
		1. ATec=(Backgroung 0~255)
		*Backgroung 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
			printf("atec=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

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
			printf("atef=(3)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

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)
			printf("at80=(%d,%d,%d)",Line,Column,Character);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at80=(0,0,A)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

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)
			printf("at81=(%d,%d, %s)",Line,Column,String);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("at81=(%d,%d, %d)",Line,Column,Value);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("at81=(%d,%d, %f)",Line,Column,Float);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at81=(0,0,ABCD9876%$)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x82	Write a 8X16
	Character
		1. AT82=(line,column,Character)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT82=(0,0,A)
			printf("at82=(%d,%d, %d)",Line,Column,Character);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at82=(0,0,A)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x83	Write a 8X16
	String
		1. AT83=(line,column,String)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT83=(0,0,ABCD1234)
			printf("at81=(%d,%d, %s)",Line,Column,String);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("at81=(%d,%d, %d)",Line,Column,Value);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("at81=(%d,%d, %f)",Line,Column,Value);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at81=(0,0,ABCD9876%$)");
			while (USART_ReceiveData(UART1) != 'E') {}

Graphic Mode Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)

N/A	Send Image
	(64x32 332/RGB bitmap) to LED Display
	(An array consist of 2048 bytes bitmap)
		1. A "for" loop to send 2048 bytes user define display information
		2. Wait until receive a module available byte ('E') from ezDisplay or a 100ms delay
			for (i = 0 ; i < 2048; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {} or a 100ms delay

0x90	Draw a line	1. AT90=(X0 position,Y0 position,X1 position,Y1 position, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT90=(1,4,54,4,4) : '4' green color
			printf("at90=(%d,%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Xcoordinate2,Ycoordinate2,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at90=(1,4,54,4,4)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x91	Draw a Rectangle	1. AT91=(X0 position,Y0 position,X1 position,Y1 position, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT91=(0,0,55,7,2) : '2' blue color
			printf("at91=(%d,%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,Xcoordinate2,Ycoordinate2,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at91=(0,0,55,7,2)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x92	Draw a filled Rectangle	1. AT92=(X0 position,Y0 position,X1 position,Y1 position, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT92=(1,1,54,6,32) : '32' red color
			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

0x93	Draw a Square	1. AT93=(X position,Y position,Width, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT93=(48,1,4,2) : '2' blue color
			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

0x94	Draw a Circle	1. AT94=(X position,Y position,Radius, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT94=(48,4,3,2) : '2' blue color
			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

0x95	Draw a filled Circle	1. AT95=(X position,Y position,Radius, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT95=(48,4,3,4) : '4' green color
			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

0x96	Draw a tip upward Triangle	1. AT96=(X position,Y position,Height, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<RGB LED example>
		AT96=(48,1,3,4) : '4' green color
			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

0x97	Draw a filled tip upward Triangle	1. AT97=(X position,Y position,Height, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT97=(48,1,3,32) : '32' red color
			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

0x98	Draw a tip downward Triangle	1. AT98=(X position,Y position,Height, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT98=(48,6,3,4) : '4' green color
			printf("at98=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at98=(48,1,3,4");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x99	Draw a filled tip downward Triangle	1. AT99=(X position,Y position,Height, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT99=(48,6,3,4) : '4' greencolor
			printf("at99=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at99=(48,1,3,4");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x9a	Draw a tip leftward Triangle	1. AT9a=(X position,Y position,Width, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT9a=(48,4,2,4) : '4' green color
			printf("at9a=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at9a=(48,1,3,4");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x9b	Draw a filled tip l eftward Triangle	1. AT9b=(X position,Y position,Width, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms
		<example>
		AT9b=(48,4,2,4) : '4' green color
			printf("at9b=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at9b=(48,1,3,4");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x9c	Draw a tip rightward Triangle	1. AT9c=(Xposition,Y position,Width, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT9c=(48,4,2,4) : '4' green color
			printf("at9c=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at9c=(48,1,3,4");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x9d	Draw a filled tip rightward Triangle	1. AT9d=(X position,Y position,Width, 0~255)
		*0~255: ref to COLOR CODE Table
		2. Wait until receive a module available byte ('E') from ezDisplay
		<RGB LED example>
		AT9d=(48,4,2,4) : '4' greencolor
			printf("at9d=(%d,%d,%d,%d)",Xcoordinate1,Ycoordinate1,height,Color_code);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("at9d=(48,1,3,4");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

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

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

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	1. ATa0=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa0=(20)
			printf("ata0=(%d)",speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("ata0=(20)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xa1	Display image row by row Down Ward	1. ATa1=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa1=(20)
			printf("ata1=(%d)",speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("ata1=(20)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xa2	Display image column by column Left Ward	1. ATa2=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa2=(20)
			printf("ata2=(%d)",speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("ata2=(20)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xa3	Display image column by column Right Ward	1. ATa3=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa3=(20)
			printf("ata3=(%d)",speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("ata3=(20)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xa4	Erase image row by row Up Ward	1. ATa4=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa4=(20)
			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

0xa5	Erase image row by row Down Ward	1. ATa5=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa5=(20)
			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

0xa6	Erase image column by column Left Ward	1. ATa6=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa6=(20)
			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

0xa7	Erase image column by column Right Ward	1. ATa7=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa7=(20)
			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

0xa8	Display image Inside Out	1. ATa8=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa8=(20)
			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

0xa9	Display image Outside In	1. ATa9=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATa9=(20)
			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

0xaa	Erase image Inside Out	1. ATaa=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATaa=(20)
			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

0xab	Erase image Outside In	1. ATab=(Speed in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATab=(20)
			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

0xac	Shift one row up	1. ATac=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATac=()
			printf("atac=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xad	Shift one row down	1. ATad=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATad=()
			printf("atad=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xae	Shift one column left	1. ATae=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATae=()
			printf("atae=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xaf	Shift one column right	1. ATaf=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATaf=()
			printf("ataf=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xb0	Fly in up-ward	1. ATb0=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb0=(1,3)
			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

0xb1	Fly in down-ward	1. ATb1=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb1=(3,2)
			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

0xb2	Fly in left-ward	1. ATb2=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb2=(1,3)
			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

0xb3	Fly in right-ward	1. ATb3=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb3=(1,3)
			printf("atb3=(%d,%d)",Page_address,Speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("atb3=(0,3)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xb4	Fly in up-left-ward	1. ATb4=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb4=(1,3)
			printf("atb4=(%d,%d)",Page_address,Speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay}
			or input directly
			printf("atb4=(0,3)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xb5	Fly in up-right-ward	1. ATb5=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb5=(1,3)
			printf("atb5=(%d,%d)",Page_address,Speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("atb5=(0,1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xb6	Fly in down-left-ward	1. ATb6=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb6=(1,3)
			printf("atb6=(%d,%d)",Page_address,Speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("atb6=(0,1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xb7	Fly in down-right-ward	1. ATb7=(Page address,Speed)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATb7=(1,3)
			printf("atb7=(%d,%d)",Page_address,Speed);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or input directly
			printf("atb7=(0,1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xbc	Set scroll flag for pattern edit	1. ATbc=(0 for without empty page , 1 for with empty page)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATbc=(0)
			printf("atbc=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("atbc=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xbe	Set Page interval of multi-page animation	1. ATbe=(1~10)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATbe=(5)
			printf("atbe=(5)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("atbe=(10)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xbf	Set animation speed	1. ATbf=(1~10)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATbf=(5)
			printf("atbf=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("atbf=(10)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xd2	Scroll the whole display upward	1. ATd2=(shif time in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATd2=(20)
			printf("atd2=(30)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

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

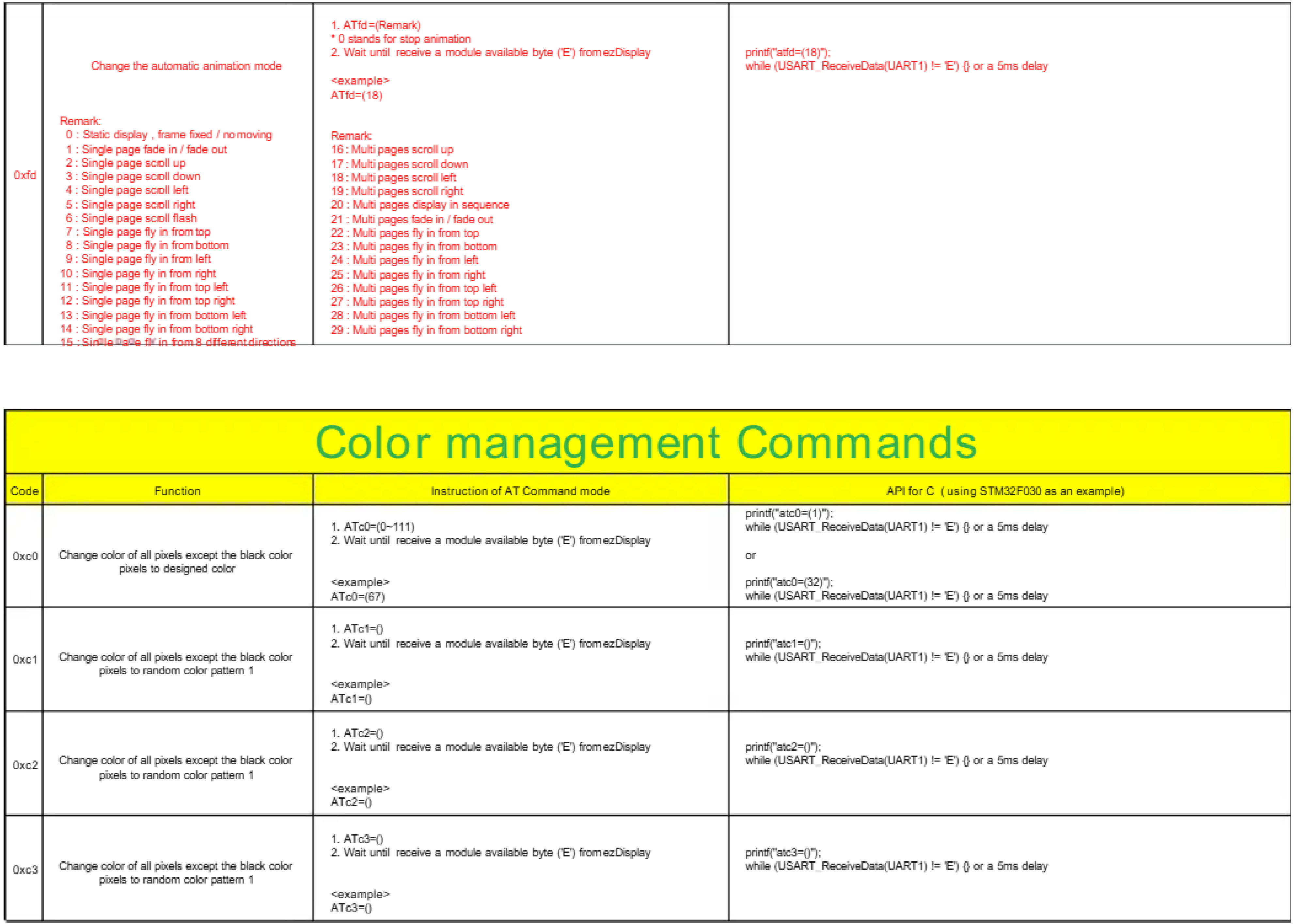
0xd6	Scroll section display upward	1. ATd6=(upper left corner X, upper left corner Y, lower right corner X, lower right corner Y,shift time in ms)
		2. Wait until receive a module available byte ('E') from ezDisplay
		< example>
		ATd6=(18,3,55,28,10)
			printf("atd6=(18,3,55,28,10)");
			while (USART_ReceiveData(UART1) != '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	Change the automatic animation mode
	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
		1. ATfd=(Remark)
		* 0 stands for stop animation
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATfd=(18)
		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
			printf("atfd=(18)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

Color management Commands

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)

0xc0	Change color of all pixels except the black color pixels to designed color	1. ATc0=(0~111)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc0=(67)
			printf("atc0=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("atc0=(32)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc1	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc1=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc1=()
			printf("atc1=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc2	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc2=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc2=()
			printf("atc2=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc3	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc3=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc3=()
			printf("atc3=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc4	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc4=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc4=()
			printf("atc4=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc5	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc5=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc5=()
			printf("atc5=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc6	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc6=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc6=()
			printf("atc6=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc7	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc7=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc7=()
			printf("atc7=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc8	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc8=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc8=()
			printf("atc8=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xc9	Change color of all pixels except the black color pixels to random color pattern 1	1. ATc9=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATc9=()
			printf("atc9=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xca	Change color of all pixels except the black color pixels to random color pattern 1	1. ATca=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATca=()
			printf("atca=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xcc	Swap displayped color to designated color for whole display	1. ATcc=(Color on screen want to swapped, Designated color)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATcc=(32,3)
			printf("atcc=(32,3)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xcd	Color change effect flag for animation	1. ATcd=(Color change effect flag 0~10) * 0 stands for no color effect
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		Atcd=(1)
			printf("atcd=(2)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xce	Change effect flag for fade in / fade out animation	1. ATce=(Color change effect flag 1~6)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		Atce=(1)
			printf("atce=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xcf	Swap one color to another color in designated area	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
		<example>
		ATcf=(32,10,8,8,32,3)
			printf("atcf=(32,10,8,8,32,3)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xd0	Clear display	1. ATd0=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATd0=()
			printf("atd0=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xd1	Show the data in the display memory	1. ATd1=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATd1=()
			printf("atd1=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

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

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
	Set to disable the page 0 eeprom write flag can increase the speed of image transfer
		1. ATbd=(0 for disable, 1 enable)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATbd=(0)
			printf("atbd=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("atbd=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xf0	Turn display Off	1. ATf0=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATf0=()
			printf("atf0=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xf1	Turn display On	1. ATf1=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATf1=()
			printf("atf1=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xf2	Set the brightness of RGB LED Display	1. ATf2=(level of brightness 0~11)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATf2=(3)
			printf("atf2=(8)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xf7	Change the configuration mode
	(only available for 64x64, 256x32 and 128x64 resolution)
		1. ATf7=(0 or 1)
		* 0 stands for 256x32, 1 stands for 128x64 , 2 stands for 64x64
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATf7=(1) : Change to 128x64 configuration.
			printf("atf7=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xfc	Change the display page	1. ATfc=(Page address 0~7)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATfc=(3)
			printf("atfc=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0xfe	Write dipslay contents to current displayed EEPROM page address	1. ATfe=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		ATfe=()
			printf("atfe=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

Advanced User Define Pattern Commands

It is recommanded to use the "pattern tool" of Lumex's ezDisplay console software to design the user define pattern.
When project moves into mass production phase please console Lumex for the technique of how to store user pattern in EEPROM

Code	Function	Instruction of AT Command mode	API for C (using STM32F030 as an example)

0x84	Dsiplay a 8X8 pattern	1. AT84=(X position,Y position, pattern ID)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT84=(16,32,1)
			printf("at84=(16,32,1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x85	Dsiplay a 8X16 pattern	1. AT85=(X position,Y position,pattern ID)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT85=(16,32,1)
			printf("at85=(16,32,1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x86	Dsiplay a 16X16 pattern	1. AT86=(X position,Y position,pattern ID)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT86=(16,32,1)
			printf("at86=(16,32,1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x20	Display firmware Revision	1. AT20=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT20=()
			printf("at20=()");
			while (USART_ReceiveData(UART1)!= 'E') {} or a 5ms delay

0x21	Define the user pattern's upper left X coordinate that is going to place	1. AT21=(pattern's upper left X coordinate)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT21=(16)
			printf("at21=(%d)",Xcoordinate);
			while (USART_ReceiveData(UART1)!= 'E') {} or a 5ms delay

0x22	Define the user pattern's upper left Y coordinate that is going to place	1. AT22=(pattern's upper left Y coordinate)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT22=(16)
			printf("at22=(%d)",Ycoordinate);
			while (USART_ReceiveData(UART1)!= 'E') {} or a 5ms delay

0x23	Define the size of user pattern	1. AT23=(Patterm's Column size, Patterm's Row size)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT23=(12,12)
			printf("at23=(%d,%d)",Column_Size,Row_Size);
			while (USART_ReceiveData(UART1)!= 'E') {} or a 5ms delay

0x24	Define the color depth of pattern	1. AT24=(Pattern's color bits depth 1,2,3,8,or24)
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT24=(8)
			printf("at24=(%d)",Color_depth);
			while (USART_ReceiveData(UART1)!= 'E') {} or a 5ms delay

0x25	Enable or Disable the "store pattern into to eeprom flag"	1. AT25=(1 for Enable / 0 for Disable)
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT25=(1)
			printf("at25=(1)");
			while (USART_ReceiveData(UART1)!= 'E') {} or a 5ms delay

0x26	Define the pattern ID of the pattern that is going to be stored	1. AT26=(pattern ID strat from 0 to 49)
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT26=(0)
			printf("at26=(10)");
			while (USART_ReceiveData(UART1)!= 'E') {} or a 5ms delay

0x27	Define reading the pattern from RAM or eeprom	1. AT27=(1 from RAM / 0 for eeprom)
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT27=(1)
			printf("at27=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			
			or
			
			printf("at27=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x29	Dsiplay one user deifined pattern from eeprom or RAM to designated coordinate
	
	(available for 5x5, 8x8, 12x12, and 16x16 pattern size)
		1. AT29=(X position,Y position,pattern's Column size, Pattern's Row size, pattern's ID)
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT29=(16,32,8,8,1)
			printf("at29=(%d,%d,%d,%d,%d)",Xcoordinate,Ycoordinate,Column_Size,Row_Size,Pattern_ID);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x2a	Read back a user defined pattern from eeprom
	
	(available for 5x5, 8x8, 12x12, and 16x16 pattern size)
		1. AT2a=(pattern's Column size, Pattern's Row size, Pattern ID)
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT8a=(8,8,0)
			printf("at2a=(%d,%d,%d)",Column_Size,Row_Size,Pattern_ID);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x2b	Define the pattern is overwrite or overlap to the placed area	1. AT2b=(0 for overwrite / 1 for overlapping)
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT2b=(1)
			printf("at2b=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			
			or
			
			printf("at2b=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x2c	Store the current display contents to temporary memory	1. AT2c=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT2c=()
			printf("at2c=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x2d	Retrieve the temporary memory as displayed contents	1. AT2d=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		
		<example>
		AT2c=()
			printf("at2d=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x2e	Dsiplay muiti patterns to designated coordinate in background from eeprom or RAM (available for 5x5, 8x8, 12x12, and 16x16 pattern size)	1. AT2e=(X position,Y position,pattern's Column size, Pattern's Row size,pattern ID) 2. Wait until receive a module available byte ('E') fromezDisplay 3. AT2e=(X position,Y position,pattern's Column size, Pattern's Row size, pattern ID)) 4. Wait until receive a module available byte ('E') fromezDisplay <example> AT8e=(16,32,5,5,0) AT8e=(32,32,5,5,1)	printf("at2e=(16,32,5,5,0)") ; while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay printf("at2e=(32,32,5,5,1)") ; while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x2f	Display the multi patterns input result in background	1. AT2f=()
		2. Wait until receive a module available byte ('E') fromezDisplay
		<example>
		AT2f=()
			printf("at2f=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x32	Move defined pattern up one row	1. AT32=(pattern's Column size, Pattern's Row size, pattern's ID)
		2. Wait until receive a module available byte ('E') fromezDisplay
		<example>
		AT32=(8,8,1)
			printf("at32=(%d,%d,%d)",Column_Size,Row_Size,Pattern_ID);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x33	Move defined pattern down one row	1. AT33=(pattern's Column size, Pattern's Row size, pattern's ID)
		2. Wait until receive a module available byte ('E') fromezDisplay
		<example>
		AT33=(8,8,1)
			printf("at33=(%d,%d,%d)",Column_Size,Row_Size,Pattern_ID);
			while (USART_ReceiveData(UART2) != 'E') {} or a 5ms delay

0x34	Move defined pattern left one column	1. AT34=(pattern's Column size, Pattern's Row size, pattern's ID)
		2. Wait until receive a module available byte ('E') fromezDisplay
		<example>
		AT34=(8,8,1)
			printf("at34=(%d,%d,%d)",Column_Size,Row_Size,Pattern_ID);
			while (USART_ReceiveData(UART3) != 'E') {} or a 5ms delay

0x35	Move defined pattern leftright one column	1. AT35=(pattern's Column size, Pattern's Row size, pattern's ID)
		2. Wait until receive a module available byte ('E') fromezDisplay
		<example>
		AT35=(8,8,1)
			printf("at35=(%d,%d,%d)",Column_Size,Row_Size,Pattern_ID);
			while (USART_ReceiveData(UART4) != 'E') {} or a 5ms delay

0x8a	Read back a
	pattern's color at designated coordinate of display memory
		1. AT8a=(X position,Y position,pattern's Column size, Pattern's Row size)
		2. Wait until receive a module available byte ('E') fromezDisplay
		<example>
		AT8a=(16,32,8,8)
			printf("at8a=(%d,%d,%d)",X_coordinate,Y_coordinate,Column_Size,Row_Size);
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

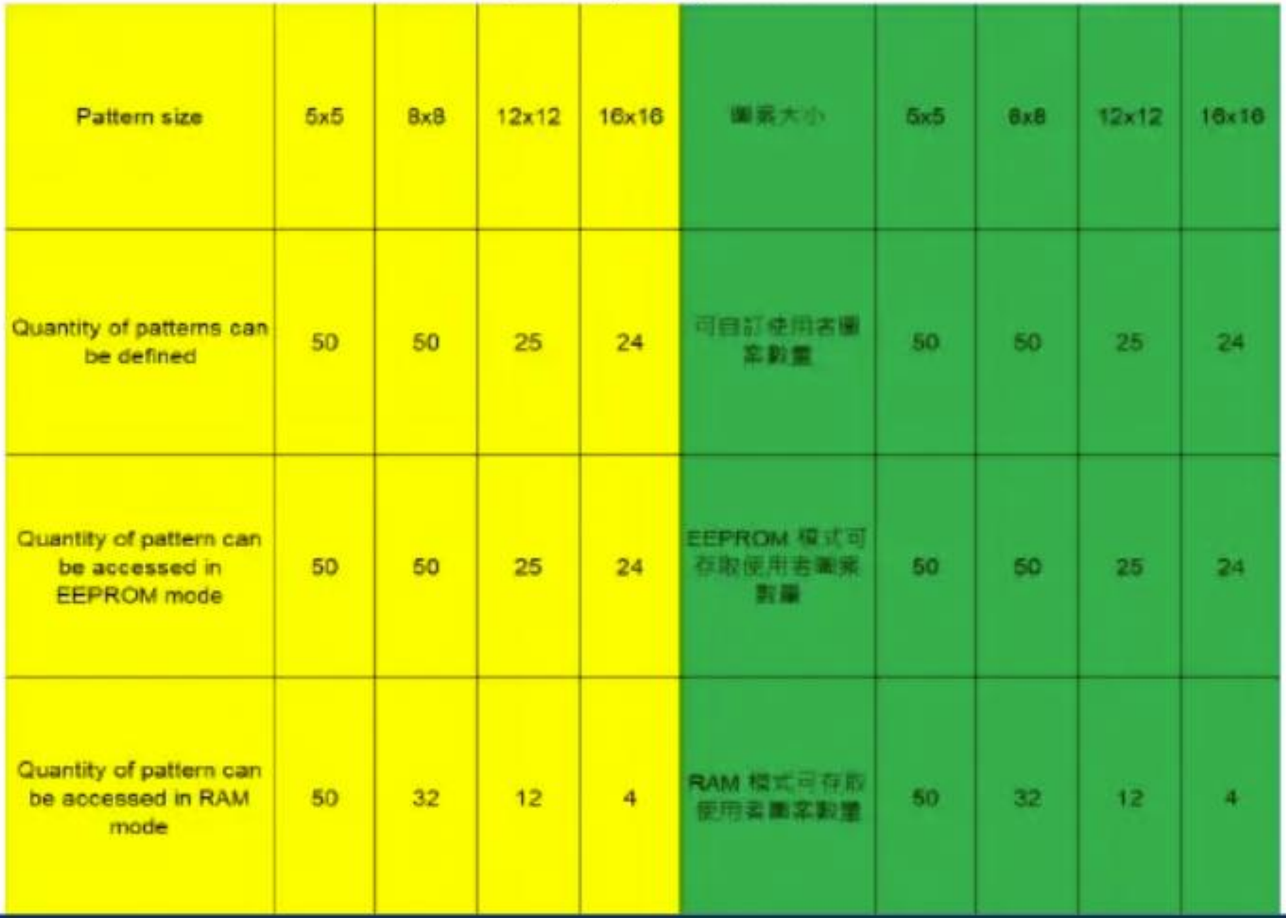
0x8b	Define the pattern is overwrite or overlap to the placed area	1. AT8b=(0 for overwrite / 1 for overlapping)
		2. Wait until receive a module available byte ('E') fromezDisplay
		<example>
		AT8b=(0)
			printf("at8b=(1)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay
			or
			printf("at8b=(0)");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x8c	Store the current display contents as background to temporary memory	1. AT8c=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT8c=()
			printf("at8c=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

0x8d	Retrieve the temporary memory as displayed contents	1. AT8d=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT8d=()
			printf("at8d=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

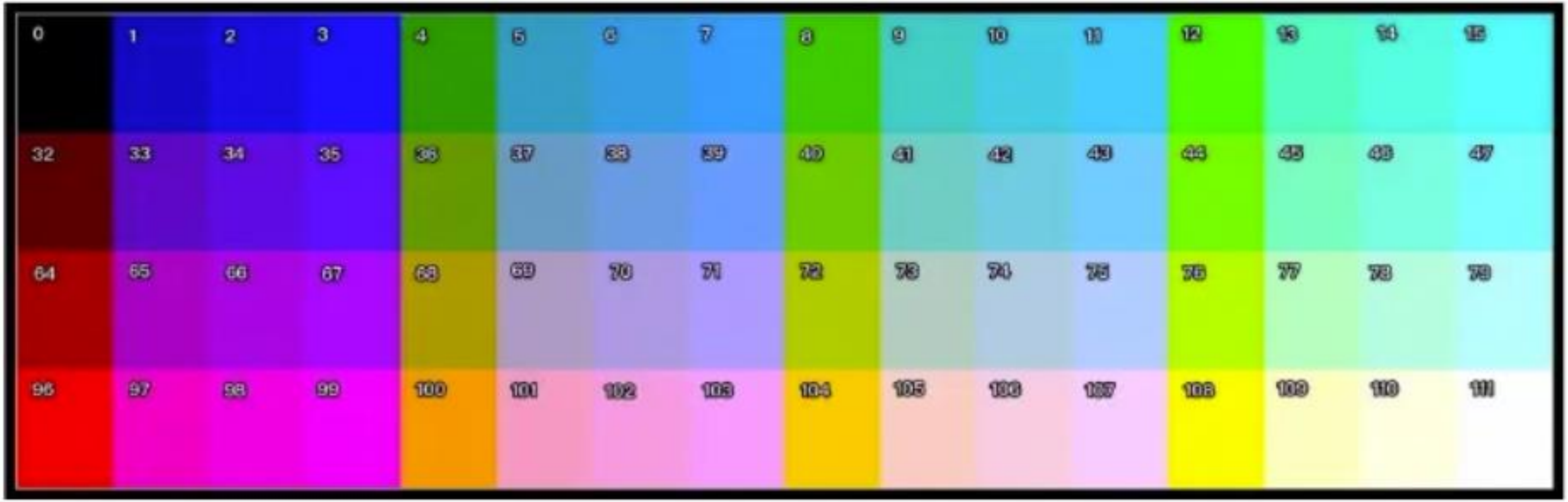
0x8f	Display the multi patterns input result in background	1. AT8f=()
		2. Wait until receive a module available byte ('E') from ezDisplay
		<example>
		AT8f=()
			printf("at8f=()");
			while (USART_ReceiveData(UART1) != 'E') {} or a 5ms delay

User define pattern quantity allocation



Pattern size	5x5	8x8	12x12	16x16	圖素大小	5x5	8x8	12x12	16x16
Quantity of patterns can be defined	50	50	25	24	可自訂使用者圖案數量	50	50	25	24
Quantity of pattern can be accessed in EEPROM mode	50	50	25	24	EEPROM 模式可存取使用者圖案數量	50	50	25	24
Quantity of pattern can be accessed in RAM mode	50	32	12	4	RAM 模式可存取使用者圖案數量	50	32	12	4

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