

ezDisplay RGB LED Display Command List V.17.E 03152019

Code	Function	Instruction of AT Command mode	API for Arduino (AT Command mode)
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	for (i = 0 ; i < 2048; i++) { Serial.write(User_define_array[i]); } while (Serial.read() != 'E') {}
0x20	Display firmware Revision	1. AT20=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> AT20=()	Write_AT_Command("AT20=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
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)	Write_AT_Command("AT21=(16)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
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)	Write_AT_Command("AT22=(16)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0x23	Define the size of user pattern	1. AT23=(Pattern's Column size, Pattern's Row size) 2. Wait until receive a module available byte ('E') from ezDisplay <example> AT23=(12,12)	Write_AT_Command("AT23=(12,12)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
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)	Write_AT_Command("AT24=(8)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
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)	Write_AT_Command("AT25=(1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }

0x26	Define the pattern ID of the pattern that is going to be stored	<p>1. AT26=(pattern ID strat from 0 to 49) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT26=(0)</p>	<pre>Write_AT_Command("AT26=(0)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x27	Define reading the pattern from RAM or eeprom	<p>1. AT27=(1 from RAM / 0 for eeprom) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT27=(1)</p>	<pre>Write_AT_Command("AT27=(1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x29	<p>Dsplay one user deifined pattern from eeprom or RAM to designated coordinate</p> <p>(only available for 5x5, 8x8, 12x12, and 16x16 pattern size)</p> <p>(not available for 96x8 RGB LED)</p>	<p>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</p> <p><example> AT29=(16,32,8,8,1)</p>	<pre>Write_AT_Command("AT29=(16,32,8,8,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x2a	<p>Read back a user defined pattern from eeprom</p> <p>(only available for 5x5, 8x8, 12x12, and 16x16 pattern size)</p> <p>(not available for 96x8 RGB LED)</p>	<p>1. AT2a=(pattern's Column size, Pattern's Row size, Pattern ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT2a=(8,8,0)</p>	<pre>Write_AT_Command("AT2a=(8,8,0)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x2b	Define the pattern is overwrite or overlap to the placed area	<p>1. AT2b=(0 for overwrite / 1 for overlapping) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT2b=(1)</p>	<pre>Write_AT_Command("AT2b=(0)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x2c	Store the current display contents to temporary memory	<p>1. AT2c=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT2c=()</p>	<pre>Write_AT_Command("AT2c=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x2d	Retrieve the temporary memory as displayed contents	<p>1. AT2d=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT2d=()</p>	<pre>Write_AT_Command("AT2d=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0x2e	<p>Dsplay multi patterns to designated coordinate in background from eeprom or RAM</p> <p>(only available for 5x5, 8x8, 12x12, and 16x16 pattern size)</p> <p>(not available for 96x8 RGB LED)</p>	<p>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') from ezDisplay 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') from ezDisplay</p> <p><example> AT2e=(16,32,5,5,0) AT2e=(32,32,5,5,1)</p>	<pre>Write_AT_Command("AT2e=(16,32,5,5,0)") while (Serial.read() !='E') {} Write_AT_Command("AT8e=(32,32,5,5,1)") while (Serial.read() !='E') {} void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x2f	<p>Display the multi patterns input result in background</p>	<p>1. AT2f=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT2f=()</p>	<pre>Write_AT_Command("AT2f=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x32	<p>Move defined pattern up one row</p> <p>(Pattern ID don't care for pattern size other than 5x5, 8x8, 12x12, and 16x16)</p> <p>(not available for 96x8 RGB LED)</p>	<p>1. AT32=(pattern's Column size, Pattern's Row size, pattern's ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT32=(8,8,1)</p>	<pre>Write_AT_Command("AT32=(8,8,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x33	<p>Move defined pattern down one row</p> <p>(Pattern ID don't care for pattern size other than 5x5, 8x8, 12x12, and 16x16)</p> <p>(not available for 96x8 RGB LED)</p>	<p>1. AT33=(pattern's Column size, Pattern's Row size, pattern's ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT33=(8,8,1)</p>	<pre>Write_AT_Command("AT33=(8,8,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x34	<p>Move defined pattern left one column</p> <p>(Pattern ID don't care for pattern size other than 5x5, 8x8, 12x12, and 16x16)</p> <p>(not available for 96x8 RGB LED)</p>	<p>1. AT34=(pattern's Column size, Pattern's Row size, pattern's ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT34=(8,8,1)</p>	<pre>Write_AT_Command("AT34=(8,8,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x35	<p>Move defined pattern leftright one column</p> <p>(Pattern ID don't care for pattern size other than 5x5, 8x8, 12x12, and 16x16)</p> <p>(not available for 96x8 RGB LED)</p>	<p>1. AT35=(pattern's Column size, Pattern's Row size, pattern's ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT35=(8,8,1)</p>	<pre>Write_AT_Command("AT35=(8,8,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0x89	<p style="text-align: center;">Display a pattern to designed coordinate</p> <p style="text-align: center;">(not available for 96x8 RGB LED)</p>	<ol style="list-style-type: none"> 1. AT89=(X position,Y position,pattern's Column size, Pattern's Row size, pattern's color bits 1,2,3,8 or24) 2. Wait until receive a module available byte ('E') from ezDisplay 3. Sent pattern's color array 4. Wait until receive a module available byte ('E') from ezDisplay 5. Delay 5ms <p><example> AT89=(16,32,5,5,8) send an array</p>	<pre>Write_AT_Command("AT89=(16,32,5,5,8)") for (i = 0 ; i < (5x5); i++) { Serial.write(User_define_array[i]); } while (Serial.read() != 'E') {} delay(5); void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x8a	<p style="text-align: center;">Read back a pattern's color at designated coordinate of display memory</p> <p style="text-align: center;">(not available for 96x8 RGB LED)</p>	<ol style="list-style-type: none"> 1. AT8a=(X position,Y position,pattern's Column size, Pattern's Row size) 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT8a=(16,32,8,8)</p>	<pre>Write_AT_Command("AT8a=(16,32,8,8)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x8b	<p style="text-align: center;">Define the pattern is overwrite or overlap to the placed area</p>	<ol style="list-style-type: none"> 1. AT8b=(0 for overwrite / 1 for overlapping) 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT8b=(0)</p>	<pre>Write_AT_Command("AT8b=(0)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x8c	<p style="text-align: center;">Store the current display contents as background to temporary memory</p>	<ol style="list-style-type: none"> 1. AT8c=() 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT8c=()</p>	<pre>Write_AT_Command("AT8c=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x8d	<p style="text-align: center;">Retrieve the temporary memory as displayed contents</p>	<ol style="list-style-type: none"> 1. AT8d=() 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT8d=()</p>	<pre>Write_AT_Command("AT8d=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0x8e	<p>Dsisplay muiti patterns to designed coordinate in background</p> <p>(not available for 96x8 RGB LED)</p>	<ol style="list-style-type: none"> 1. AT8e=(X position,Y position,pattern's Column size, Pattern's Row size, pattern's color bits 1,2,8 or24) 2. Wait until receive a module available byte ('E') from ezDisplay 3. Sent pattern's color array 4. Wait until receive a module available byte ('E') from ezDisplay 5. Delay 5ms 6. AT8e=(X position,Y position,pattern's Column size, Pattern's Row size, pattern's color bits 1,2,8 or24) 7. Wait until receive a module available byte ('E') from ezDisplay 8. Sent pattern's color array 9. Wait until receive a module available byte ('E') from ezDisplay 10. Delay 5ms <p><example> AT8e=(16,32,5,5,8) send an array AT8e=(32,32,5,5,8) send an array</p>	<pre>Write_AT_Command("AT8e=(16,32,5,5,8)") for (i = 0 ; i < (5x5); i++) { Serial.write(User_define_array[i]); } while (Serial.read() !='E') {} delay_ms(5); Write_AT_Command("AT8e=(32,32,5,5,8)") for (i = 0 ; i < (5x5); i++) { Serial.write(User_define_array[i]); } while (Serial.read() !='E') {} delay_ms(5) void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x8f	<p>Display the multi patterns input result in background</p>	<ol style="list-style-type: none"> 1. AT8f=() 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT8f=()</p>	<pre>Write_AT_Command("AT8f=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x80	<p>Write a 5X7 Character</p>	<ol style="list-style-type: none"> 1. AT80=(line,column,Character) 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT80=(0,0,A)</p>	<pre>Write_AT_Command("AT80=(0,0,A)"); void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x81	<p>Write a 5X7 String</p>	<ol style="list-style-type: none"> 1. AT81=(line,column,String) 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT81=(0,0,ABCD1234)</p>	<pre>Write_AT_Command("AT81=(0,0,ABCD1234)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x82	<p>Write a 8X16 Character</p> <p>(not available for 96x8 RGB LED)</p>	<ol style="list-style-type: none"> 1. AT82=(line,column,Character) 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT82=(0,0,A)</p>	<pre>Write_AT_Command("AT82=(0,0,A)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x83	<p>Write a 8X16 String</p> <p>(not available for 96x8 RGB LED)</p>	<ol style="list-style-type: none"> 1. AT83=(line,column,String) 2. Wait until receive a module available byte ('E') from ezDisplay <p><example> AT83=(0,0,ABCD1234)</p>	<pre>Write_AT_Command("AT83=(0,0,ABCD1234)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0x84	<p>Dsisplay a 8X8 pattern</p>	<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>Write_AT_Command("AT84=(16,32,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x85	<p>Dsisplay a 8X16 pattern</p> <p>(not available for 96x8 RGB LED)</p>	<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>Write_AT_Command("AT85=(16,32,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x86	<p>Dsisplay a 16X16 pattern</p> <p>(not available for 96x8 and 64x32 RGB LED)</p>	<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>Write_AT_Command("AT86=(16,32,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x87	<p>Dsisplay a 32X32 pattern</p> <p>(not available for 96x8 and 64x32 RGB LED)</p>	<p>1. AT87=(X position,Y position,pattern ID) 2. Wait until receive a module available byte ('E') from ezDisplay 3. Wait 2ms</p> <p><example> AT87=(16,32,1)</p>	<pre>Write_AT_Command("AT87=(16,32,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x88	<p>Dsisplay a 16X32 pattern</p> <p>(not available for 96x8 and 64x32 RGB LED)</p>	<p>1. AT88=(X position,Y position,pattern ID) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT88=(16,32,1)</p>	<pre>Write_AT_Command("AT88=(16,32,1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x90	<p>Draw a line</p>	<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><RGB LED example> AT90=(1,4,54,4,4) : '4' green color</p>	<pre>Write_AT_Command("AT90=(1,4,54,4,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x91	<p>Draw a Rectangle</p>	<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><RGB LED example> AT91=(0,0,55,7,2) : '2' blue color</p>	<pre>Write_AT_Command("AT91=(0,0,55,7,2)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0x92	Draw a filled Rectangle	<p>1. AT92=(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><RGB LED example> AT92=(1,1,54,6,32) : '32' red color</p>	<pre>Write_AT_Command("AT92=(1,1,54,6,32)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x93	Draw a Square	<p>1. AT93=(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> AT93=(48,1,4,2) : '2' blue color</p>	<pre>Write_AT_Command("AT93=(48,1,4,2)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x94	Draw a Circle	<p>1. AT94=(X position,Y position,Radius,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> AT94=(48,4,3,2) : '2' blue color</p>	<pre>Write_AT_Command("AT94=(48,4,3,2)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x95	Draw a filled Circle	<p>1. AT95=(X position,Y position,Radius,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> AT95=(48,4,3,4) : '4' green color</p>	<pre>Write_AT_Command("AT95=(48,4,3,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x96	Draw a tip upward Triangle	<p>1. AT96=(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><RGB LED example> AT96=(48,1,3,4) : '4' green color</p>	<pre>Write_AT_Command("AT96=(48,1,3,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x97	Draw a filled tip upward Triangle	<p>1. AT97=(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><RGB LED example> AT97=(48,1,3,32) : '32' red color</p>	<pre>Write_AT_Command("AT97=(48,1,3,32)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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><RGB LED example> AT98=(48,6,3,4) : '4' green color</p>	<pre>Write_AT_Command("AT98=(48,6,3,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }</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><RGB LED example> AT99=(48,6,3,4) : '4' green color</p>	<pre>Write_AT_Command("AT99=(48,6,3,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }</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><RGB LED example> AT9a=(48,4,2,4) : '4' green color</p>	<pre>Write_AT_Command("AT9a=(48,4,2,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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 3. Wait 2ms</p> <p><RGB LED example> AT9b=(48,4,2,4) : '4' green color</p>	<pre>Write_AT_Command("AT9b=(48,4,2,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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><RGB LED example> AT9c=(48,4,2,4) : '4' green color</p>	<pre>Write_AT_Command("AT9c=(48,4,2,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("AT9d=(48,4,2,4)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0x9e	Set a Pixel as default Color	<p>1. AT9e=(X position,Y position)</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> AT9e=(120,32)</p>	<pre>Write_AT_Command("AT9e=(120,32)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("AT9f=(32,32)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
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>Write_AT_Command("ATa0=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa1=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa2=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa3=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa4=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa5=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa6=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa7=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa8=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATa9=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATaa=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATab=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATac=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATad=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATae=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATaf=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb0=(1,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb1=(3,2)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb2=(1,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb3=(1,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb4=(1,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb5=(1,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb6=(1,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATb7=(1,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xbc	Set scroll flag for pattern edit	<p>1. ATbc=(0 for without empty page , 1 for with empty page) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATbc=(0)</p>	<pre>Write_AT_Command("ATbc=(0)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xbd	Set Page 's EEPROM write enable flag	<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>Write_AT_Command("ATbd=(0)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATbe=(5)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATbf=(5)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xc0	Change color of all pixels except the black color pixels to designed color	<p>1. ATc0=(0~111) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATc0=(67)</p>	<pre>Write_AT_Command("ATc0=(67)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xc1	Change color of all pixels except the black color pixels to random color pattern 1	<p>1. ATc1=(2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATc1=(</p>	<pre>Write_AT_Command("ATc1=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xc2	Change color of all pixels except the black color pixels to random color pattern 2	<p>1. ATc2=(2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATc2=(</p>	<pre>Write_AT_Command("ATc2=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0xc3	Change color of all pixels except the black color pixels to random color pattern 3	1. ATc3=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATc3=()	Write_AT_Command("ATc3=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0xc4	Change color of all pixels except the black color pixels to random color pattern 4	1. ATc4=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATc4=()	Write_AT_Command("ATc4=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0xc5	Change color of all pixels except the black color pixels to random color pattern 5	1. ATc5=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATc5=()	Write_AT_Command("ATc5=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0xc6	Change color of all pixels except the black color pixels to random color pattern 6	1. ATc6=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATc6=()	Write_AT_Command("ATc6=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0xc7	Change color of all pixels except the black color pixels to random color pattern 7	1. ATc7=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATc7=()	Write_AT_Command("ATc7=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0xc8	Change color of all pixels except the black color pixels to random color pattern 8	1. ATc8=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATc8=()	Write_AT_Command("ATc8=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0xc9	Change color of all pixels except the black color pixels to random color pattern 9	1. ATc9=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATc9=()	Write_AT_Command("ATc9=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }
0xca	Change color of all pixels except the black color pixels to random color pattern 10 Not available for 96x8 and 64x32 modules	1. ATca=() 2. Wait until receive a module available byte ('E') from ezDisplay <example> ATca=()	Write_AT_Command("ATca=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }

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>Write_AT_Command("ATcc=(32,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xcd	<p>Color change effect flag for animation</p> <p>Not available for 96x8 and 64x32 modules</p>	<p>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</p> <p><example> Atcd=(1)</p>	<pre>Write_AT_Command("ATcd=(1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xce	<p>Change effect flag for fade in / fade out animation</p> <p>Not available for 96x8 module</p>	<p>1. ATce=(Color change effect flag 1~6) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> Atce=(1)</p>	<pre>Write_AT_Command("ATcd=(1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATc=(32,10,8,8,32,3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xd0	Clear display	<p>1. ATd0=() 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATd0=()</p>	<pre>Write_AT_Command("ATd0=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATd1=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATd2=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xd3	Scroll the whole display downward	<p>1. ATd3=(shif time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATd3=(20)</p>	<pre>Write_AT_Command("ATd3=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0xd4	Scroll the whole display leftward	<p>1. ATd4=(shif time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATd4=(20)</p>	<pre>Write_AT_Command("ATd4=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xd5	Scroll the whole display rightward	<p>1. ATd5=(shif time in ms) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATd5=(20)</p>	<pre>Write_AT_Command("ATd5=(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xd6	Scroll section display upward	<p>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</p> <p>< example> ATd6=(18,3,55,28,10)</p>	<pre>Write_AT_Command("ATd6==(18,3,55,28,10)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xd7	Scroll section display downward	<p>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</p> <p>< example> ATd7=(18,3,55,28,10)</p>	<pre>Write_AT_Command("ATd7==(18,3,55,28,10)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xd8	Scroll section display leftward	<p>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</p> <p>< example> ATd8=(18,3,55,28,10)</p>	<pre>Write_AT_Command("ATd8==(18,3,55,28,10)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xd9	Scroll section display rightward	<p>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</p> <p>< example> ATd9=(18,3,55,28,10)</p>	<pre>Write_AT_Command("ATd9==(18,3,55,28,10)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xdf	Desinate the number of pages for animamtion	<p>1. ATdf=(page number 2~7) 2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATdf=(5)</p>	<pre>Write_AT_Command("ATdf=(5)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xec	Set default Background	<p>1. ATec=(Background 0~255) *Background 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> ATec=(11) : '11' cyan color</p>	<pre>Write_AT_Command("ATec=(11)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATee=(48,6,96)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xef	Set pixel's default Color	<p>1. ATef=(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><example> ATef=(32) : '32' red color</p>	<pre>Write_AT_Command("ATef=(32)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }</pre>
0xf0	Turn display Off	<p>1. ATf0=()</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf0=()</p>	<pre>Write_AT_Command("ATf0=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }</pre>
0xf1	Turn display On	<p>1. ATf1=()</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf1=()</p>	<pre>Write_AT_Command("ATf1=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }</pre>
0xf2	Set the brightness of RGB LED Display	<p>1. ATf2=(level of brightness 0~11)</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf2=(3)</p>	<pre>Write_AT_Command("ATf2=(3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xf6	<p>Change connected hardware</p> <p>1 for dot matrix display , 2 for digital LEDs Ring or stripe</p> <p>*Only available for LDM-6432-4LT</p>	<p>1. ATf6=(1 or 3)</p> <p>2. Wait until receive a module available byte ('E') from ezDisplay</p> <p><example> ATf6=(3)</p>	<pre>Write_AT_Command("ATf6=(1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xf7	<p>Change the configuration mode</p> <p>**not available for 96x8 and 64x32 module</p>	<p>1. ATf7=(0 or 1) * 0 stands for 256x32, 1 stands for 128x64</p> <p>2. Change user's MCU UART baudrate to desinated baudrate</p> <p><example> ATf7=(1) : Change to 128x64 configuration.</p>	<pre>Write_AT_Command("ATf7=(1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>

0xfb	Change Baudrate	<p>1. ATfb=(0 ~3) * 0 stands for 9600, 1 stands for 115200, 2 stands for 576000, 3 stands for 921600,</p> <p>2. Change user's MCU UART baudrate to desinated baudrate</p> <p><example> ATfb=(1) : Change to 115200 bps.</p>	<pre>Write_AT_Command("ATfb=(1)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</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>Write_AT_Command("ATfc=(3)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xfe	Write dipslay 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>Write_AT_Command("ATfe=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
0xfd	Change the animation	<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>	<pre>Write_AT_Command("ATfd=(18)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} }</pre>
	Remark:	<p>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>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>