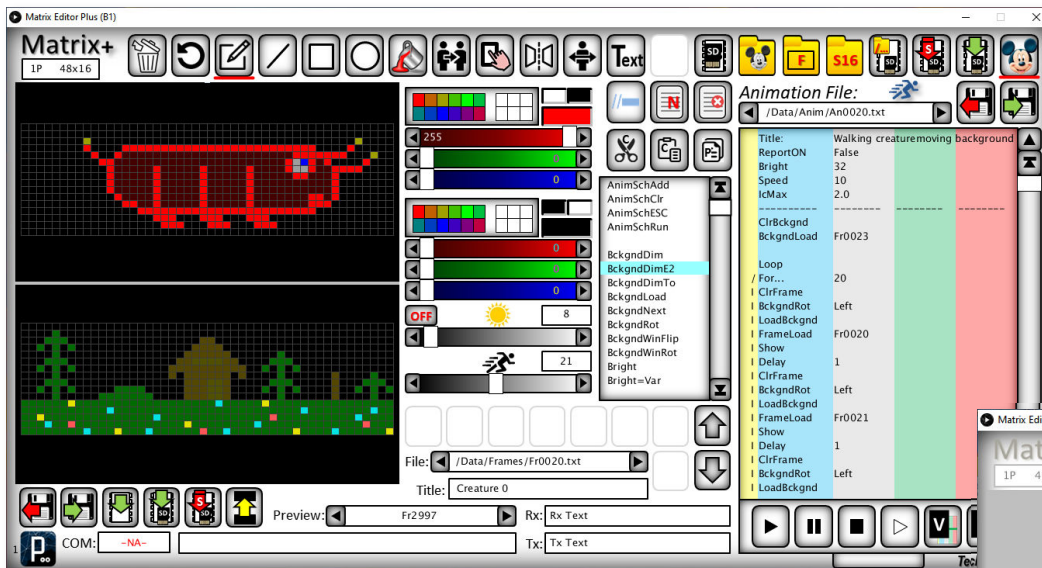
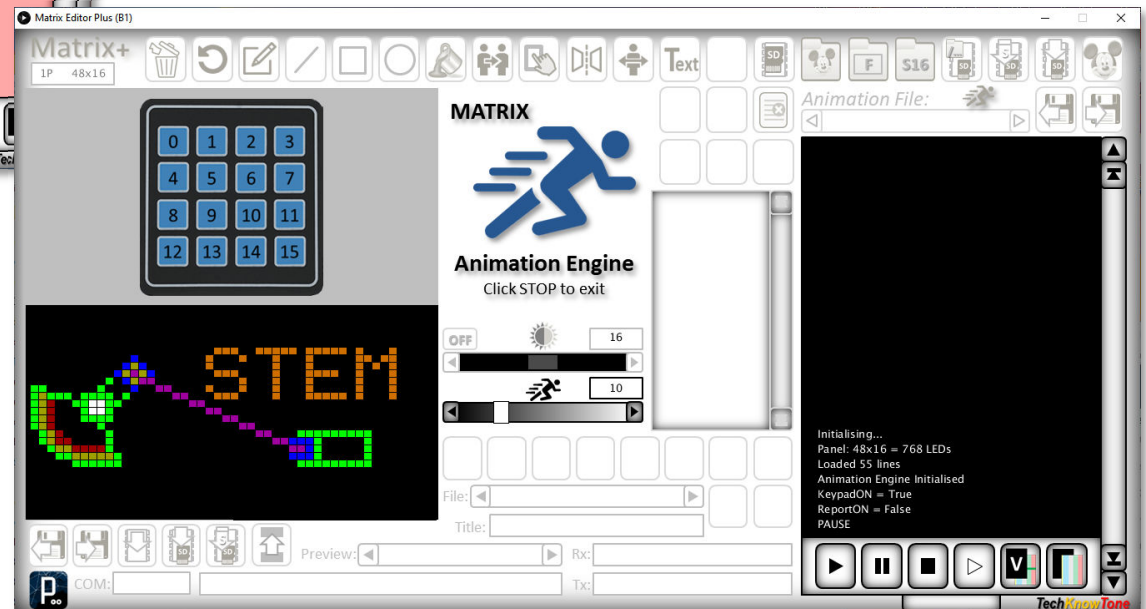


MATRIX+ Animation Editor Quick Guide (R1)



MATRIX+ includes a built-in **Animation Engine**!

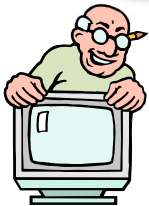
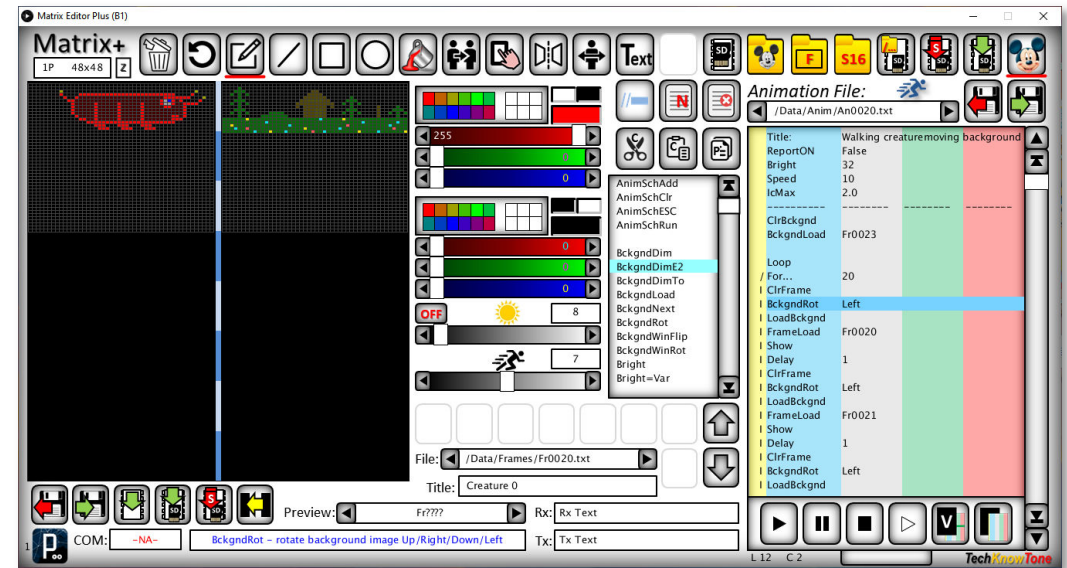
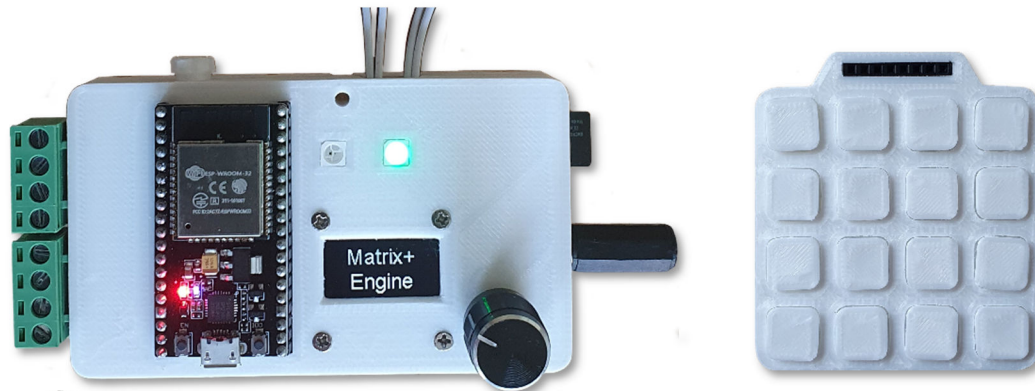
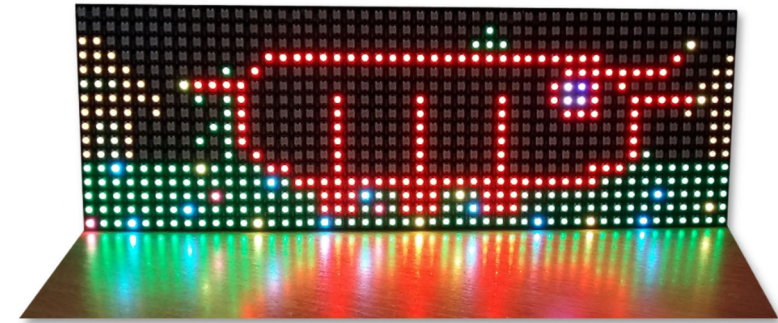
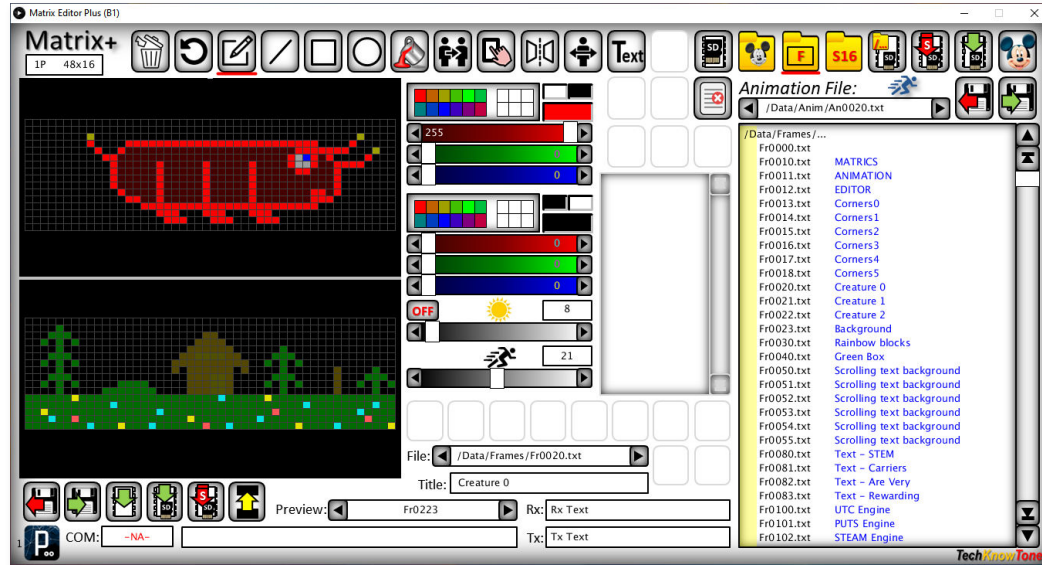
You can now test your animation scripts, without connecting an LED panel.



The controller has also been upgraded for frame speed, and the connection of external switches.

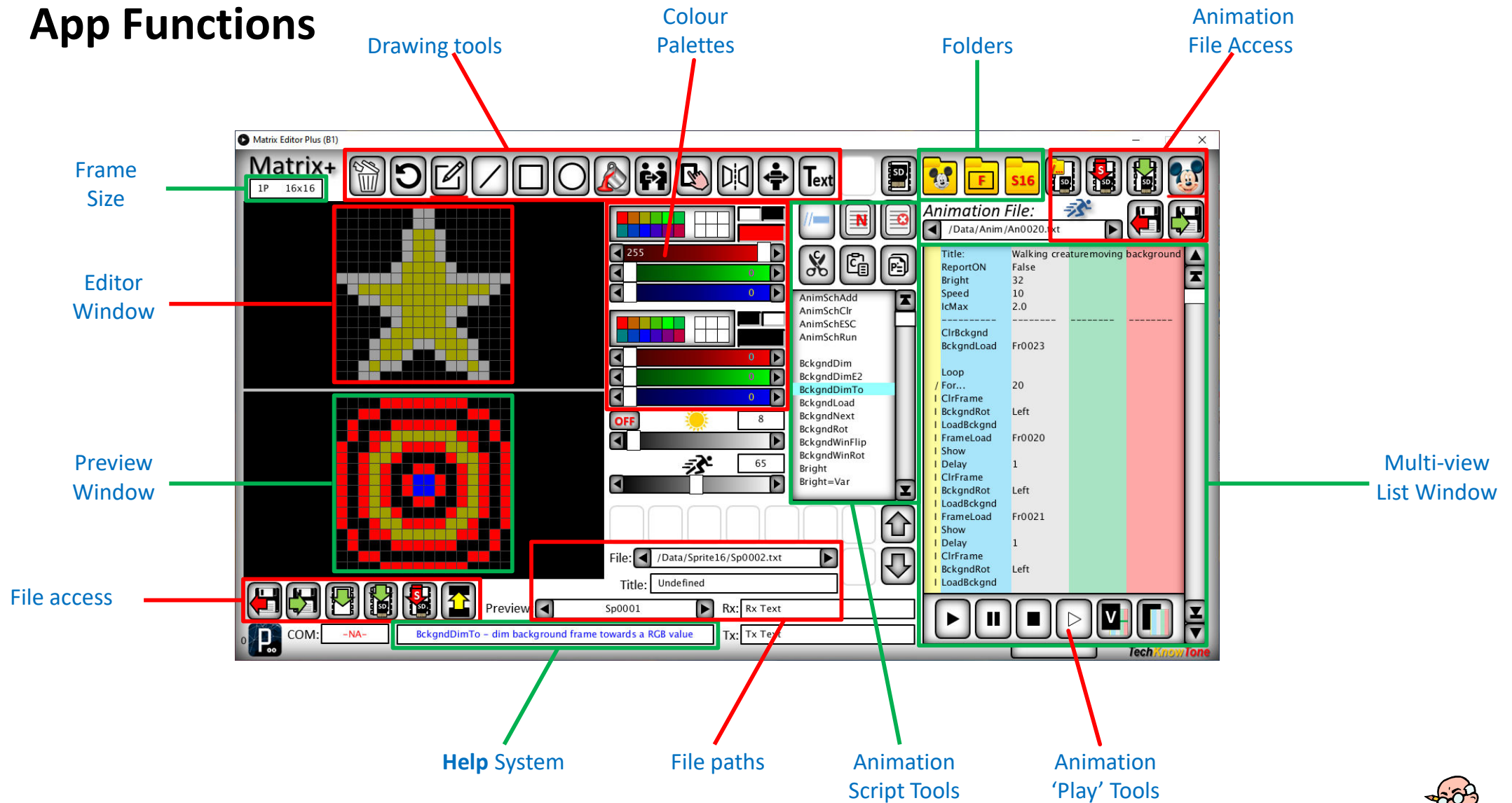
So you can now interact with your animations in real time.

Introduction



Matrix+ is a Windows app, designed for use with WS2812B RGB LED panels, to perform two functions, when used in conjunction with a suitable microcontroller. It enables you to design colourful images, and to bring them to life by creating animation scripts, which are all stored in files on a microSD card. A number of LED panel configurations are supported, ranging from a single 16x16 LED panel, up to a 3x8 panel 48x128 LED array. This simple guide shows you the key features of the Matrix+ app, which should be sufficient to get you started. Matrix+ was written in C++ using the Processing development platform, compiled into Java 32-bit and 64-bit, to run on Windows based PC's, with Java pre-installed, and a minimum screen resolution of 1366 x 768 pixels.

App Functions

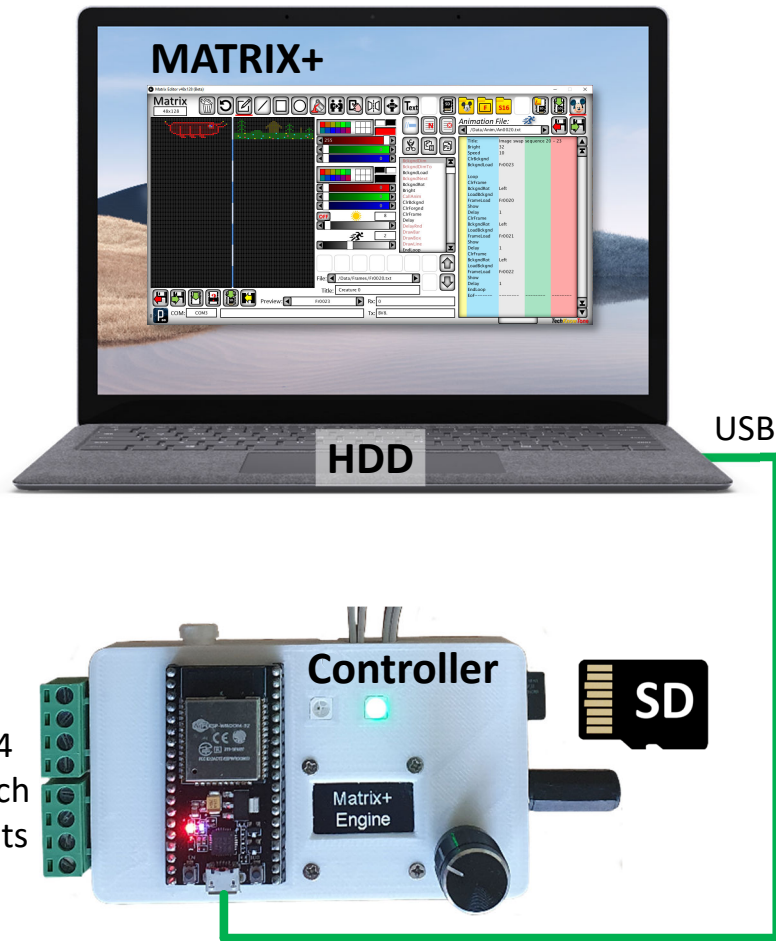


The left hand side of the Matrix+ app is used for creating graphics, whilst the right hand side is used for creating animation scripts and viewing file lists. The 'Help' system field at the bottom of the app displays key information about items under the mouse pointer. Try moving the mouse pointer around to view the help messages; you can learn a great deal from reading these prompts.



Note that some items only appear under certain conditions; like the additional controls for the animation script editor. The Preview window also acts as a source for colour picking, and copying graphics and cloning from existing image files.

File Storage



App root/Data

/Anim



Text scripts

/Frames



Graphics

/Sprite16



Graphics 16x16

SD root/Data

/Anim



Numeric scripts

/Frames

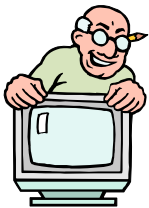


Graphics

/Sprite16



Graphics 16x16



Matrix+ creates files for graphics and animation scripts, which are stored on your PC, and also transferred to your micro as data, over the serial USB link, and stored on its microSD card. The microSD file system is based on FAT32, hence filenames are limited to 8 characters, plus a 3 character extension; like 'FR0000.txt'. For consistency this naming convention is also used for PC filenames.

The data storage format used in all cases is readable ASCII text, using hexadecimal numbers for improved efficiency. However files are not directly exchangeable between microSD and PC HDD systems, you must use the MATRIX app to read/write, and manage both systems. It is strongly recommended that you save backup copies of your PC files to a backup folder, in case you accidentally delete or corrupt your files. Bulk transfer of selected files to the controllers microSD card is a time saving option.

Drawing Graphics

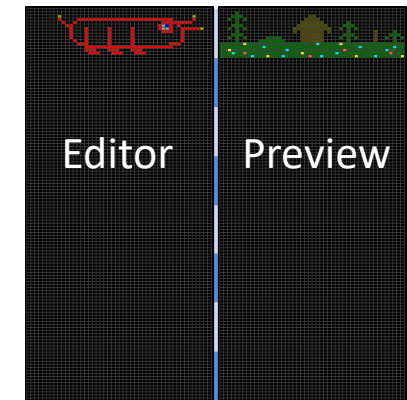
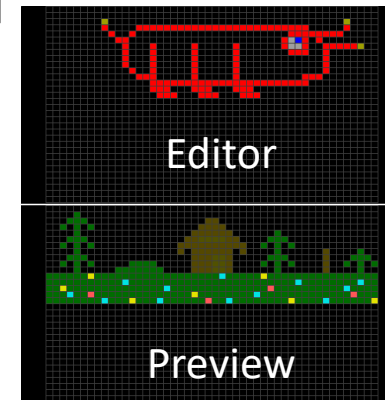
Start by choosing the correct frame size for your project. Click on the field beneath the **Matrix+** logo to do this. As LED panels are 16x16, a project using 3x6 panels has 48x96 LEDs. You can send this and the controller mode to the microcontroller, to be stored in EEPROM flash memory.



Frame sizes: 16x16 to 16x128,
32x16 to 32x128,
48x16 to 48x128



Matrix+ displays two windows, one for drawing in (Editor) and one used to preview files on your system. At low resolutions, the Editor is above the Preview pane; at higher vertical values, above 32 pixels, they are side by side, as seen here:

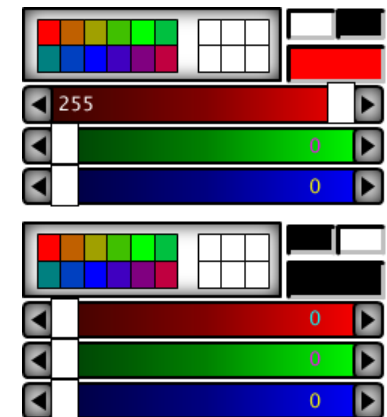


By default the freestyle pen tool is selected. Clicking on the other tool buttons changes the editors mode. The active mode is underlined in red. To draw simply click and drag with the mouse buttons. The left-hand button is the Pen colour, and the right-hand button is the background, or secondary colour. As the outcome of all of the tools is dependent on the mouse buttons, it is recommended that a mouse is used, rather than a mouse pad.



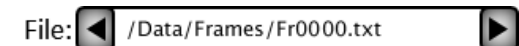
A 'Zoom' feature option appears in the higher frame size settings. This causes the editor to display the image twice normal size, through a view port. Hold the SPACE key down and move the mouse to navigate around. When in this mode however, you temporarily lose the file preview window, as it can only displays the edited image.

The ink colours can be changed by either clicking on the pre-set values or moving their respective sliders. Each colour component, Red, Green and Blue has a range of 0 – 255. When working on an image with many colours, you can easily switch to an existing colour, picking it by holding down the CTRL key whilst clicking on the pixel.



Start cloning operations by holding the CTRL key down whilst clicking on a reference point. Now drawing in another area, will effectively copy the source pixels into that new area. Pressing the ESC key ends cloning, and is also used to abort all of the other drawing modes.

Select a file path using the file selector and save your image by clicking on the SAVE button. As there is only one level of UNDO, it is recommended that you save your work frequently. Giving your image a recognisable title will help you find it in the file list later, as all FAT32 filenames are limited to 8 character numeric.



Files & Folders

As explained in a previous slide, files are store either on your PC, or in the microSD card connected to your microcontroller, and finally on both. Images must be stored on the microSD for animations to run, but scripts can be sent dynamically for testing; then transferred to microSD later.



File path: /Data/Anim/An0000.txt



File: /Data/Frames/Fr0000.txt

Title: Enter your title here

Preview: Fr0001

The yellow 'folder' buttons enable you to view lists of files stored on your PC, as either Animation, Frame graphics or Sprite16 graphics files. Along with the filenames, Matrix+ reads the 'Title' information entered for each file and displays this too, in blue. You can load files by double clicking on the list item. For graphics files you need to have the correct frame size setting. Sprite16 files will only load into a frame size of a 16x16 array, but can be copied from Preview.

If you click and drag on filenames in either list of graphics files, they will appear in the preview window. This is also true if you click and drag in either the file path or preview path fields. Think of them as sliders.

Whilst there are separate buttons for storing graphics files to either the PC HDD or microSD card, via USB; you can hold down the SHIFT key when saving graphics to your PC and it will also be transferred to the micro SD card.



Whilst editing graphics, MATRIX+ will send data to the micro, such that it can mimic what you are drawing on an LED display. But this is only a temporary process, and not the same as saving the file to the microSD. Save files regularly to avoid data loss.

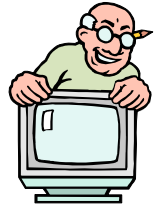
Double clicking on a file, listed in the Animation list, will invoke the script editor mode and load that file from your PC HDD. See animation scripts described on the next page. You can also select one or more files for deletion or for sending to the microSD card using the synch buttons.

<p>/Data/Anim/...</p> <ul style="list-style-type: none"> An0000.txt <i>Boot Animation</i> An0001.txt <i>Image swap sequence</i> An0002.txt <i>Red, Green, Blue Frames</i> An0003.txt <i>Creature walking, 2 frames</i> An0004.txt <i>Image swap test</i> An0005.txt <i>---- Null file ----</i> An0006.txt <i>Gosub test</i> An0020.txt <i>Walking creaturemoving background</i> An0030.txt <i>Scrolling ForRnd tests</i> An0040.txt <i>Hey You!</i> An0050.txt <i>Scrolling Text</i> <p>Animation files</p>	<p>/Data/Frames/...</p> <ul style="list-style-type: none"> Fr0000.txt <i>3 random pattern</i> Fr0001.txt <i>1 random pattern</i> Fr0002.txt <i>2 random pattern</i> Fr0003.txt <i>3 random pattern</i> Fr0004.txt <i>4 random pattern</i> Fr0005.txt <i>5 random pattern</i> Fr0006.txt <i>Space shuttle</i> Fr0007.txt <i>Creature 0</i> Fr0008.txt <i>Creature 1</i> Fr0009.txt <i>Hovercraft</i> Fr0010.txt <i>Horizontal lines - full frame</i> Fr0011.txt <i>Checker board - full frame</i> Fr0012.txt <i>Vertical lines - full frame</i> Fr0013.txt <i>Rainbow blocks</i> Fr0015.txt <i>Boxed-in</i> Fr0020.txt <i>Creature 0</i> Fr0021.txt <i>Creature 1</i> Fr0022.txt <i>Creature 2</i> Fr0023.txt <i>Background</i> Fr0030.txt <i>Rainbow blocks</i> Fr0040.txt <i>Green Box</i> Fr0050.txt <i>Scrolling text background</i> Fr0051.txt <i>Scrolling text background</i> Fr0052.txt <i>Scrolling text background</i> Fr0053.txt <i>Scrolling text background</i> Fr0054.txt <i>Scrolling text background</i> Fr0055.txt <i>Scrolling text background</i> <p>Frame graphics files</p>	<p>/Data/Sprite16/...</p> <ul style="list-style-type: none"> Sp0000.txt <i>Square</i> Sp0001.txt <i>Circle</i> Sp0002.txt <i>Star</i> Sp0003.txt <i>Cross</i> <p>Sprite graphics files</p>
--	--	--

PC file lists



Animation Scripts



The animation script editor is accessed by clicking on the button, top right, or by double-clicking on a file in the file list. This reveals a multi-coloured list. New lines are added to the list by clicking on the New LINE button, or SPACE bar. When lines are added they are set as being selected, and are therefore highlighted.

To cancel line selections press the ESC key at any time. The left mouse button adds lines before the last or selected line, where as the right mouse button adds a line after the first selected line. This helps when adding lines to existing scripts. To select lines, click in the left-most yellow column.

The animation editor mode also reveals a new set of hidden tools, such as line DELETE, CUT, COPY and PASTE buttons. DISABLE is used in debugging.

An important tool is the command reference list, which holds all of the commands that the associated micros animation engine will respond to. Placeholder commands, not yet coded are in red. Commands are added to selected lines by double clicking on items the reference list. This will add commands to new lines or overwrite existing commands, whichever is selected; and the process moves down the list of selected lines. Point with the mouse and press a letter for fast searching.

The 'Help' system describes each command under the mouse pointer, and the line is also highlighted, in either list. Pointing at RGB values in the script list, will change the line highlight to reflect the set colour. To change values associated with commands, simply click on them, mouse left or right. RGB values are changed using the pen colour selector. Other values are incremented or decremented using the left and right mouse buttons. Text is modified using a simple line editor; press <RET> or <ESC> keys to either accept or reject changes.

Enter a title for your script, to make the file more recognisable later, as seen in the animation file list, which displays the title entered next to the filename.

The animation scripts are saved to the PC HDD using the file selector and SAVE button. They can also be transferred to the microSD card using a button at the top of the MATRIX+ app. That is required for standalone operation, when a PC is not connected to the ESP32 microcontroller, and the animation is run directly from data stored on the microSD card.

Click in this column to select a line.

Click to switch to Animation editor



Animation Planes

Traditional animation frames are made from transparent sheets, layered one on top of another, to give the combined scene. This saves on having to redraw everything in the scene for each frame, and makes it easier to move or change each layer of the scene independently.

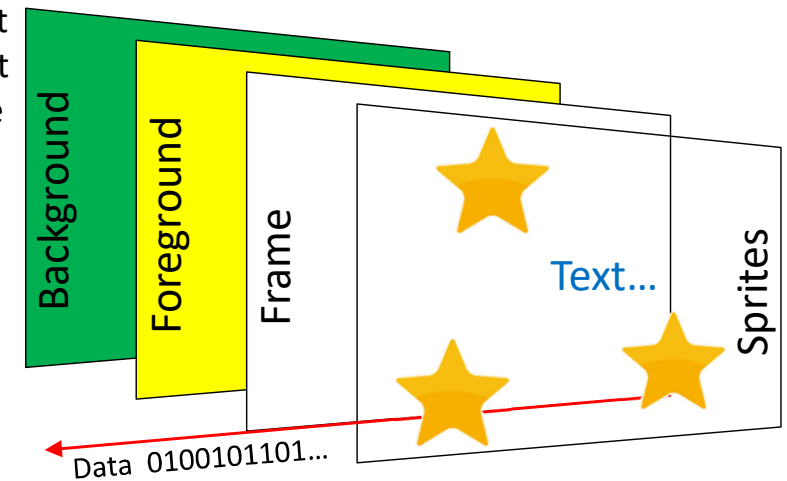


In a similar way the micro used to drive the LED matrix display, can be instructed to store RGB pixel information in separate arrays, like layers, and then combine them into one array before sending the data to the LED matrix, when given a 'show' command.

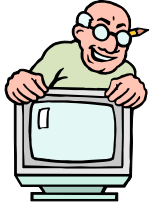
The array holding the final image is known as the 'Frame' and the two supporting arrays are the 'Foreground' and 'Background'. All three arrays can have images loaded into them, from instructions in the animation script. Once loaded the images can have functions applied, like Dim and Rotate, or be cleared before loading another image. That is often necessary, as images are often transparent (ie. pixels not set) and we don't want to leave remanence behind. The background and foreground need to be loaded into the frame, to be included in a scene, and can be loaded in either order, one effectively overlaying on top of the other.

The Frame array has additional functions like the placement of characters or text, and even scrolling text. Up to eight small arrays of 16x16 LEDs, defined here as sprites, can also be positioned in the scene and switched ON or OFF. Sprites that are set to visible, will be automatically drawn in the Frame, prior to Show events; to overlay other graphics. Simple animations might only use the Frame array, for example to cycle through a series of images; where as complex scenes will make use of all three panes, and incorporate sprites, with scrolling and dimming effects.

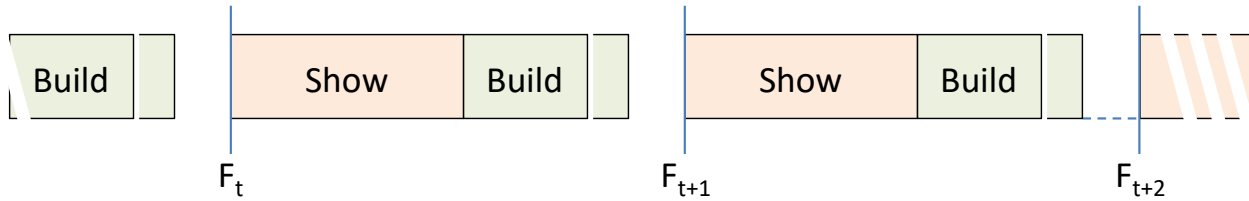
It should be known that the time needed to load an image into an array from the microSD card is proportional to its contents, as only set pixels contain data. The data from the Frame array is sent to the WS2812B LEDs at 800 kb/s, so the overall number of 16x16 LED panels used in your project will have an inverse effect on frame rate. For example a 3 panel 48x16 display can run at > 40 fps, where as a 18 panel 48x96 display will only run at 7 fps. Acceptable frame rates for most animations vary between 5 – 15 fps. Use the MATRIX+ controllers dual output mode (2H) to increase the frame rate for large panels, above 48x32 pixels.



Animation Engine



The animation scripts are sent to the microcontroller, which runs code on the ESP32 known as the animation engine, AE for short. It's role is to interpret each command in the script in sequence, in order to build the output frame, and then send it to the LED panel display at regular intervals.



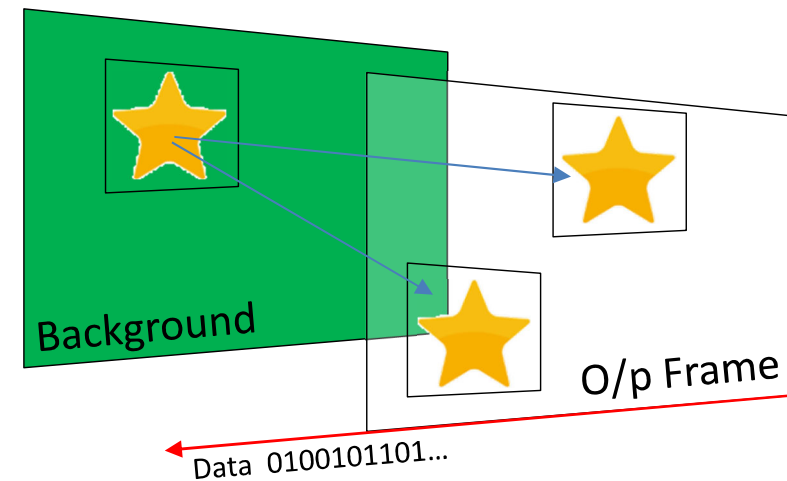
In the timing diagram above the AE is sending LED colour data to the display panel at regular intervals using the FastLED 'Show' command. The frame rate speed is set such that there is time after each burst of 'Show' data, to build the contents of the next frame. The time needed to build the frame is dependent on the animation script, and it will vary.

A healthy state is when there is always a time gap between building a frame, and the pending show event. In this condition the desired frame rate is being achieved. The 'Show' data output period is constant for a given panel size, and dependent on the number of LEDs in your display, with RGB data being sent at a precise 800 kbps. Therefore the show period is proportional to the size of your LED panel, and the maximum frame rate is inversely proportional to the number of LEDs. So large panels limit the maximum frame rate achievable (fps).

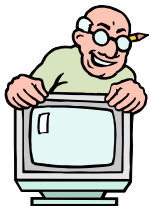
When creating scripts, which will build your animations, try to consider the amount of data you are moving around in the process. For example we could load a background frame which contains only a small image. Whilst there are commands to copy the whole of the background into the output Frame, it is far more time efficient to only copy in the area of the small image from the background, you specifically want to display.

You could in fact build the output Frame by copying in a series of small images taken from the background. And these in turn could be placed in different locations, using variables to control the copy process; there by developing movement from a static background source.

The same applies to use of the Foreground frame, for Frame construction.



Playing Animations



The animation script and graphics, created with the MATRIX+ editor, are transferred to the micro over USB, which in turn sends serial data to the LED matrix. At the base of the script window is a broad thin button which, when clicked, reveals controls for playing the animations.

For animations to run, the graphics need to have been stored on the microSD card, but the script can be sent dynamically; allowing you to make changes and run them on the fly. Once you are happy with the animation you can transfer it to the microSD and run it directly by clicking on the running man icon above the list.



In this way the PC could then be disconnected and the micro would run the script independently, until reset or power is removed. The MATRIX+ controls enable you to PLAY, PAUSE, STOP and single STEP through the script.

As the micro code runs the script, it reports to the PC via a black console window, what it is doing.



This form of verbose messaging can be reduced to simple line numbers, by clicking on the VERBOSE toggle button. Then when you return to the script view, the line highlight will move in line with the micro's messaging, to show which part of the script the animation is running. Loaded images will also be previewed, as they are being loaded in the animation.

Auto-Boot

In order to initiate animations being run, from power-up or reset, without the need for a PC to be connected, the code in the micro checks to see if the file [An0000.txt](#) is present in the [Root/Anim/..](#) folder on the microSD card. If it is, then the micro will load it into the animation engine and run the script stored in it.

The Matrix app can create a script of scheduled animation files, which the user wants to run consecutively as a whole series of short animations, one after the other. The short animations, referenced in this list, once run simply use an AnimSchESC command, to instruct the background scheduler to load the next animation script in the sequence.



```

Title: Scrolling Text
Bright 8
Speed 12

ClrFrame
FrameLoad Fr0050
TextRGB [ RGB ]

Loop
TextScrollY 4
Text UTC students support Lumiere 2021
Text+ !
Text+ See our display in Durham on 18-
Text+ 21 November.

//
// This code scrolls the text whilst changing the
// background image using FrameNext
Label: Scroll Loop
TextScroll
Show
FrameNext Fr0050 Fr0055
GolfScroll Scroll Loop
EndLoop
EoF-----
    
```

Play toggle

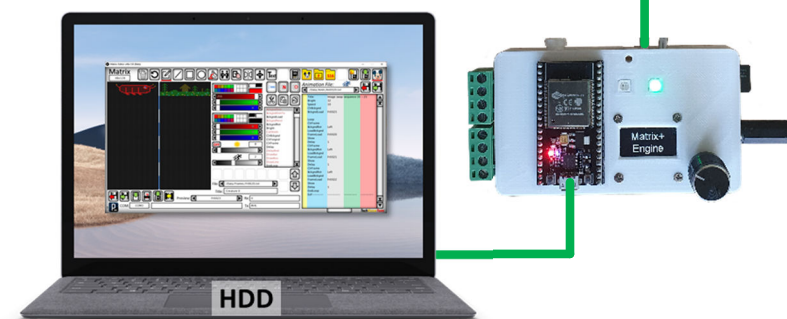
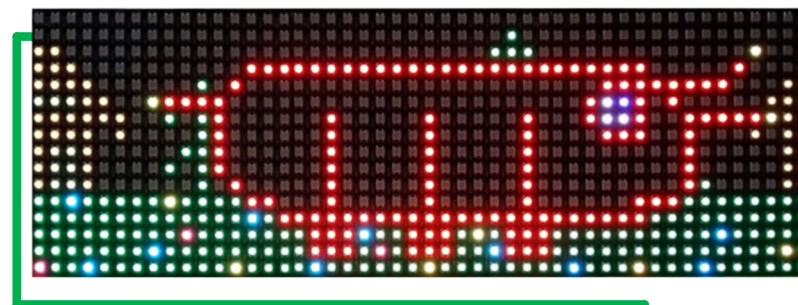
```

Title: Scrolling Text
Bright 8
Speed 12

ClrFrame
FrameLoad Fr0050
TextRGB [ RGB ]

Loop
TextScrollY 4
Text UTC students support Lumiere 2021
Text+ !
Text+ See our display in Durham on 18-
Text+ 21 November.

//
// This code scrolls the text whilst changing the
// background image using FrameNext
Label: Scroll Loop
TextScroll
Show
FrameNext Fr0050 Fr0055
GolfScroll Scroll Loop
EndLoop
EoF-----
    
```

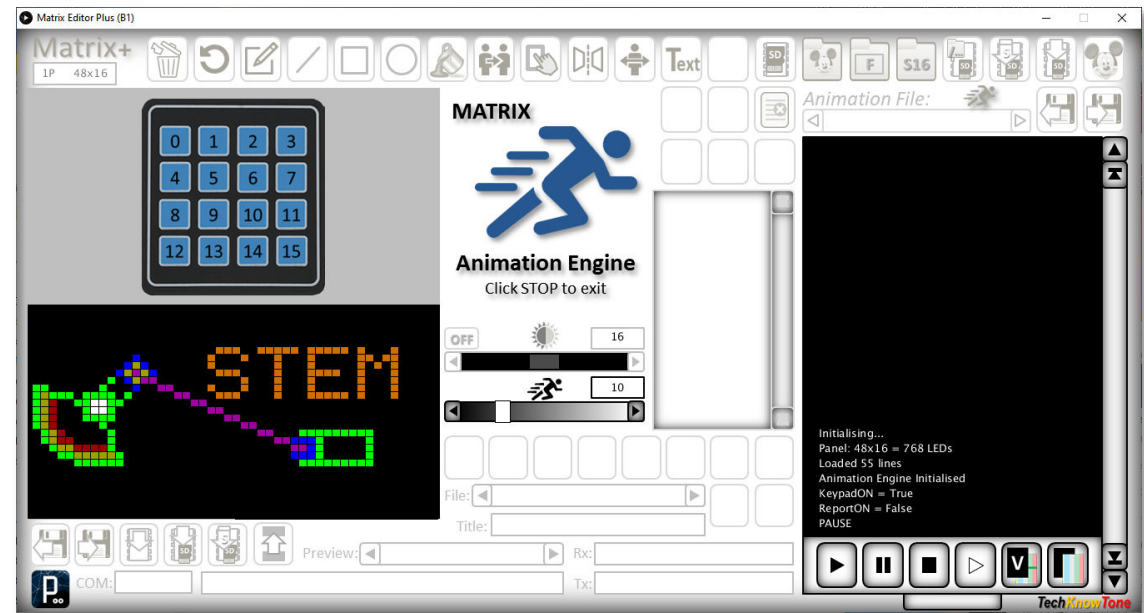


MATRIX+ Animation Engine



MATRIX+ contains an Animation Engine, which is capable of displaying your creations directly within the app. It runs automatically when you click 'Play', if there is no controller connected to the USB serial port.

This feature provides a great way of testing your projects, without the need to transfer images and scripts to the controllers microSD card first. All of the editors features are naturally disabled when this mode is invoked, but you can return to the editor simply by clicking on the 'STOP' button at any time. The 4x4 keypad enables mouse clicks to simulate the operation of external push switches and touchpads.



Whilst the animation engine interprets script commands in the same way as the micro controller, there are some differences, due to it running within the Windows environment, so frame speed time may be slightly off for example. Displayed colours are also limited.

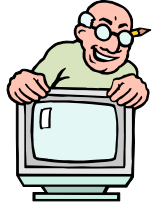
Colour – RGB LEDs are very bright, and can be controlled over a wide range, 0 – 255. But screen colours do not have the same level of brightness, making them difficult to see at low levels. For this reason the dynamic range of the colours is constrained within the app, and the brightness level is shown numerically next to the brightness control slider, with a bar that varies in width, representing brightness.

Frame speed – the Windows screen refresh can be up to 100 Hz, but is not as easy to control as it is within the micro controller, using the `FastLED.show()` events. So the app animation will be close to, but may not quite match that of the micro controllers frame rate.

Verbose text – the micro controller is free to send messages to the app, whenever it wishes during an animation run, and the MATRIX+ app displays then on receipt in the black console window, as soon as they arrive over the serial port. But when the Animation Engine is running, the presentation of verbose messages needs to coincide with screen refresh cycles, so their appearance in time may be quite different.

Keypads & Touchpads – the new micro controller now has 8 GPIO ports connected to screw terminals, for the purpose of reading up to 16 switches, wired as a 4 x 4 matrix, or 8 touchpads. In order to test animation scripts interactively using the key state commands, the MATRIX+ animation engine displays a keypad, whenever one or the other is turned on in the script.

ESP32 microcontroller



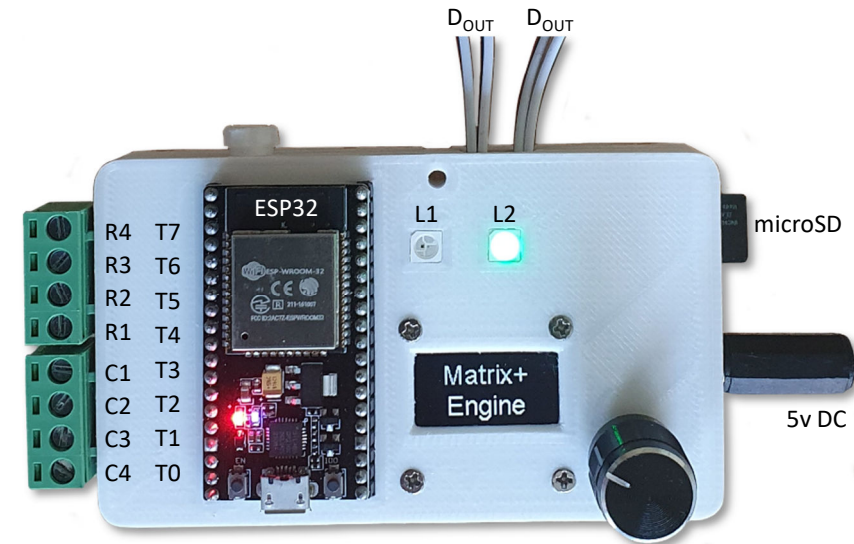
The ESP32 microcontroller is an essential part of this system, as it contains the C++ code which forms the animation engine, which pulls data from files held on the microSD card, to generate your light shows. This great microcontroller is similar in many ways to an Arduino, but it is much more powerful, running two 32-bit cores at 80 MHz, with a wealth of memory.

In this project the ESP32 has been programmed using the Arduino IDE, which works well, provided that you install the ESP32 board libraries and associated USB drivers. You can see the libraries used in this project in the image. The new controller now has two data outputs, for driving larger panels more quickly.

By default the code is set for a display, consisting of 3 horizontal 16x16 panels, giving a matrix of 48x16 pixels (WxH). But the Matrix app and micro can handle much larger display, up to 48x128 pixels, that's 6144 LEDs! This limit was chosen due to the fact that the technology effective slows down as the number of LEDs increase, as the data rate is fixed at 800 kbps. The MATRIX+ app enables you to define the panel arrangement you are using, and then instruct the micro to store these values in EEPROM memory, as the default power-up settings.

The controller has a 64x128 OLED display and a rotary encoder included in the design, but you can if you wish build it without these components. If you don't wish to have a controller with a menu system, the only essential components are the ESP32, the microSD card reader and the two WS2812B LEDs. As explained in the wiring diagrams, the LEDs act as electronic level-shifters, between the 3.3v micro and the 5v used by the LED panel chips.

The microcontroller, if instructed by an ICMAX command, can determine the predicted current to be used by the LED display on the fly, and therefore limit the brightness used. This is a great feature, as it allows you to design your graphics, without worrying whether or not your 2A limited supply can cope with the peak demand.



```
Panel_Plus | Arduino IDE 2.2.1
File Edit Sketch Tools Help
ESP32 Dev Module
Panel_Plus.ino AnimEngine.ino Display.ino Functions.ino GFX48x128.ino GFX_Lib.ino Tasks.ino microSD.ino ...
32 */
33 // Declare Libraries
34 #include <Arduino.h> // needed when using an ESP32 micro
35 #include <HardwareSerial.h> // serial library
36 #include <Wire.h> // I2C library
37 #include <FastLED.h> // Neopixel library
38 #include "SSD1306Wire.h" // OLED display library
39 #include "FS.h" // file system library
40 #include "SD.h" // SD card library
41 #include "SPI.h" // SPI library
42 #include <stdlib.h> // C utility library
43 #include <EEPROM.h> // EEPROM flash access library
44
45 // Configuration
46 String Release = "Panel Plus (B1+)";
47 String Date = "28/09/2023";
48 #define BUILD true // == true if display and rotary encoder are connected
49
50 // Display Settings
51 const int I2C_DISPLAY_ADDRESS = 0x3c;
52 const int SDA_PIN = 21; // I2C SDA GPIO pin
53 const int SCL_PIN = 22; // I2C SCL GPIO pin
54
55 // Initialize the OLED display for address 0x3c
56 SSD1306Wire display(I2C_DISPLAY_ADDRESS, SDA_PIN, SCL_PIN);
57
58 // Define a task for core 0
59 TaskHandle_t Core0;
60
61 // Define constants and GPIO pins
62 #define ACK 6 // link acknowledge character
63 #define BatPin 36 // ADC1 Ch0 potential divider pin, aka battery voltage sample
64 #define C1Pin 12 // matrix switch pin C1
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
246
```

Command Reference List



There is a wide range of commands available in the MATRIX+ Editor, which are described in much more detail in the Command Reference document, than in the simple dynamic Help system provided by the MATRIX+ App.

As well as loading images for frames and sprites, you can also set variables and use conditional 'If' statements. Variables can also be used to position images and adjust brightness and speed.

There are also commands for creating schedules, which control the order in which animation files are loaded and played.

It is well worth spending time reading through this list, to familiarise yourself with what is available. Each command is listed alphabetically for easy reference. Missing features?... let me know.

The parameters associated with each command are deliberately constrained within the list editor. All you need to do is click on them with either the left or right mouse key to change their values, or to enter into special editing modes for text and RGB colours.

In order to reduce the length of a script, some of the commands use single bit assignments to apply the same parameter to all of those with a bit set in the mask. An example of this is the SpriteOn{M} command, where the 8-bit field {xxxxxxxx} relates to all 8 sprites, so you can control all eight of them with one command.

An animation script can be up to 1,000 lines in length, so there is plenty of scope for making quite complex animations. If you wanted more, then you could easily combine them as a schedule.

Matrix Command Reference (R3)

Command mnemonics are listed alphabetically:

AnimSchAdd AnXXXX S	Adds an animation file AnXXXX to the animation schedulers list. Files will be loaded in the order they are listed, when the animation engine encounters a AnimSchESC command in the animation file it is running. The S text reference option can be edited to describe the animation file, making the list easier to understand
AnimSchClr	Clear the animation schedulers list, and reset the scheduler. In effect turning off this function, or preparing it for a new list of animation files to be added using AnimSchAdd. As a schedule list is stored in an animation file, one list can call up another, which can clear the current list and add new schedule. Making the process unlimited.
AnimSchESC	Animation exit point, used only if the scheduler is running, having been pre-loaded with animation file references. This command is placed at a point in your animation script where an exit to another animation is permitted; normally after a number of counted cycles, or prior to a cycle repeating itself. On encountering this command, the animation engine will check to see if a schedule has been loaded. If so, it will load the designated animation file, otherwise it will ignore this command and continue with the current animation indefinitely.
AnimSchRun N	Invoke the animation scheduler, if one or more file references have been loaded using the AnimSchAdd command. This command is normally placed at the end of a list of scheduled files. The number N is used to reference the first animation file to run; which does not have to be the first one in the list. Note that setting N=0 refers to the first file reference, and N=1 the second, and so on.
BckgndDim N	Dim the whole background frame by an amount N, towards a pixel RGB level of zero. Depending on the value of the pixel, this command many need to be called several times to get to a value of zero. The same value of N is subtracted from each RGB component. Hence the contribution of each colour and the combined colour effect, will change, as the values get smaller.
BckgndDimE2	Dim all background frame buffer pixels by a half of their current value. If called repeatedly, this will dim the background buffer pixels to RGB 0,0,0 in 9 cycles (ie. 255, 127, 63, 31, 15, 7, 3, 1, 0) from max to off.
BckgndDimTo [RGB] D	Dim background frame towards a target RGB value, by an incrementing value of D. Depending on the initial colour of the background, and to what degree it differs from the target RGB colour, this command may need to be used several times in order for the target RGB colour to be reached.
BckgndLoad FrXXXX	Load image file FrXXXX, from the microSD card, into background frame buffer. The time needed for this operation to complete is proportional to the number of pixels set in the source image file FrXXXX, and could have an adverse effect on achieving high frame rate settings.
BckgndNext FrXXXX FrYYYY	Load next background image file in a range, starting with FrXXXX and ending with FrYYYY. Once the last image file has been loaded, the next time this command is encountered the animation engine will cycle back to the first file, and repeat the whole cycle continuously.
BckgndRot Up/Right/Down/Left	Rotate the whole of the background image, in the buffer in one of 4 directions: Up/Right/Down/Left. This effect is like having the image wrapped round a barrel, which is rotating one pixel at each command.
BckgndWinFlip X0,Y0 X1,Y1 Left-Right/Up-Down	Flip part of the image in the background buffer, in a region defined by X0Y0,X1Y1 in one of 2 directions; left-Right about the Y-axis, or Up-Down about the X-axis.
BckgndWinRot X0,Y0 X1,Y1 Up/Right/Down/Left	Rotate part of the image in the background buffer, in a region defined by X0Y0,X1Y1, in one of 4 directions; Up, Right, Down, Left. This effect, is like having that part of the image wrapped round a barrel, which is rotating one pixel at each command. This is useful for scrolling objects or text in a scene.
Bright B	Set the overall brightness value 0 – 255 for the whole panel. The FastLED library used in this project has a function which effectively controls the brightness of each and every pixel value as they are sent to the LED strip display.

Release: R3 19/11/2021

TechKnowTone

me to adjust the this command.

variable 'A' - 'Z'. This mically. The FastLED and every pixel value as pp, works in the same

This is often used to t and only contain data

y setting the value of t part of the

This is often used to t and only contain data

setting the value of t part of the

is often used to remove ly contain data on pixel

ing the value of each dow into which text or

e the number of led by the rightmost pple, to clear Sprite 1, n be set in this way, to

= A - 1

which 'Show' commands process. The delay For example, a Delay of each cycle.

ist the 'Speed' value want to temporarily event is skipped, before the animation, by

will be generated. This

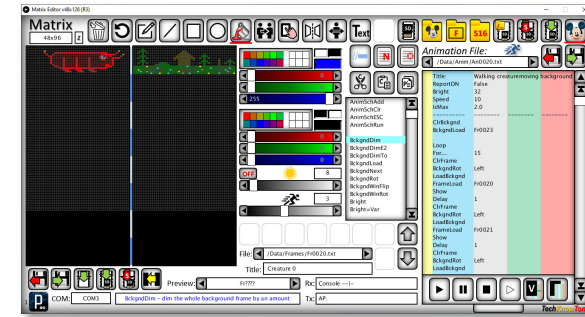
is used to set a 'Show' delay, based on the value of the assigned variable, in speed period units. Whilst the 'Speed' value sets the over rate at which 'Show' commands are executed, there are time when we may want to

Release: R3 19/11/2021

TechKnowTone

Hints & Tips

Despite the aid of the built in Help system, the MATRIX+ app has a number of features which aren't immediately obvious, but could save you a lot of time once you are aware of them.



Esc - the keyboard Esc key helps you to exit from drawing operations, which you may have started but don't wish to complete. It also clears the line selection in lists, and cancels dialogues.

CTRL – the keyboard Ctrl key, used in conjunction with the mouse buttons, sets the start point for cloning. In the list editor you can quickly copy and paste single lines, using Ctrl+Right button to copy and Ctrl+Left to paste the chosen command line. It also produces a narrow font when selecting and using the Text tool.

Shift – the keyboard Shift key, used in conjunction with the mouse buttons, doubles up on file saving actions; saving an image to the PC HDD as well as the microSD in one click. When changing values in the editor it also increase the +/- increments to +/-10, making it much easier to get to large values. The Delete line(s) button in the editor will delete the whole list, if Shift is held down whilst clicking on it.

COM – the USB COM field will make or remake a connection when clicked with the left mouse button, and disconnect the COM port when clicked with the right mouse button, freeing up the port for the IDE Serial Monitor or other Windows apps. If the MATRIX+ app connects to the wrong port, simply click the field again to make it select the next available port. When disconnected the text is in red ---- or -NA-.

Projects – animations are often produced as short sequences, pooled together with the use of the animation scheduler list. Simple file organisation can make things much easier to manage. For example spacing the animation file names at intervals of say 10 to 50, and naming the associated image files within those ranges. As in: An0010.txt, with Fr0010.txt, Fr0011.txt and Fr0012.txt; then An0020.txt, etc

Backup – as a project grows in size, it is relatively easy to accidentally overwrite a valuable image or script file. So, to avoid losing valuable work, make a backup folder of the Data folder, and regularly copy the contents of your working folder into the backup folder.

Frame size – should be set to suite the physical size of your project, as in 48x16 pixels. The value of frame width is critical in determining the X,Y co-ordinates of a pixel, for drawing and file loading operations to work correctly. Set that at the outset.

Feedback on any part of this project would be most welcomed.

Enjoy!

