

Сведения о версии документа

Версия 2.

По сравнению с версией 1 изменения следующие:

- исключено упоминание о порядке воспроизведения скриптов и кодировки символов имени файла, т.к. это определяется конкретной реализацией плеера;
- уточнено описание «терминатора» строки скрипта;
- дополнен раздел о команде CLR;
- дополнено описание команды INF;
- исправлена ошибка в описании команды вычитания в разделе «Команды для работы с переменными»;
- добавлены команды GI, PI, RGC, RPC, BM, FM, RRC, RLC, RRB, RLB, SRC, SLC, SRB, SLB;
- добавлена функция DP;
- изменено название команд PC, PF, PB;
- изменено описание команд REV, WT, GC, GB, GF;
- добавлен раздел «Сводная таблица команд»;
- добавлен раздел «Индексы цвета»;
- дополнен раздел «Порядок составления скрипта».

Скрипт световых эффектов

Файл скрипта должен иметь имя в формате MS-DOS, т.е. не более 8 символов имени и обязательное расширение .SC.

На SD-карту скрипты помещаются в корневую директорию.

Скрипт представляет собой текстовый файл в кодировке «один символ – один байт», т.е. любая кодировка, допускающая кодирование 1 символа несколькими байтами, недопустима.

Требования к строкам скрипта

Обязательные требования

Каждая строка скрипта должна завершаться «терминатором» - символами `\n\r` (перевод строки и возврат каретки, допустимо использование только одного из этих символов).

Каждая строка скрипта должна содержать не более 255 символов, включая терминатор.

Команды скрипта

Все команды, функции и числа в скрипте должны вводиться заглавными символами английского алфавита.

Все числа в скрипте ограничены 1 байтом.

Для представления чисел в командах используется шестнадцатеричный формат без префиксов и суффиксов, каждое число записывается всегда двумя знаками. То есть вместо `0xE2` следует писать просто `E2`, вместо `0x3` следует писать `03`.

Перечень всех команд с примерами дан далее.

В одной строке скрипта может быть одна или несколько команд.

Для разделения команд в строке используются символы-разделители: пробел и табуляция. Терминатор строки так же считается символом-разделителем.

Внутри команды не должно быть символов-разделителей.

Часть строки между двумя символами-разделителями, не распознанная, как корректная команда, считается комментарием и игнорируется.

Пустая строка считается комментарием и игнорируется.

Файл скрипта должен заканчиваться пустой строкой.

Переменные и функции

Для хранения числовых значений в скрипте можно использовать 16 переменных V0, V1 ... VF. Эти переменные могут быть использованы во всех командах, требующих числового значения.

Для получения случайного числа в диапазоне 0...255 можно использовать функцию **RD**.

Для получения случайного номера пиксела можно использовать функцию **RP**. Значение этой функции всегда будет соответствовать номеру одного из физически имеющихся пикселов.

Для гарантированного получения номера пиксела, который не светится, можно использовать функцию **DP**. Функция анализирует текущее состояние пикселов и, если светятся все, команда, которая использует эту функцию, будет пропущена – это следует учитывать при разработке скрипта.

Функция **TP** всегда равна максимально доступному номеру пиксела. То есть при физически имеющихся 16 пикселах эта функция будет возвращать 15, т.к. нумерация пикселов идет с нуля.

Команды для работы с переменными

Значение переменной **Vn** (где n – номер переменной от 0 до F) может быть задано командой **Vn=XX**, где (здесь и далее) **XX** – это *число, переменная* или *функция*.

Значение переменной **Vn** может быть модифицировано при помощи арифметических команд:

Vn+XX – увеличение переменной на **XX**;

Vn-XX – уменьшение переменной на **XX**;

Vn*XX – результат произведения **Vn** на **XX**;

Vn/XX – частное от деления **Vn** на **XX**;

Vn%XX – остаток от деления **Vn** на **XX**.

Арифметические действия сложения и умножения выполняется «с насыщением», т.е. результат никогда не выходит за пределы 0...255. Операция уменьшения вызывает переполнение, т.е. при вычитании из нуля результат будет FF.

В командах деления и получения остатка от деления значение делителя, равное 0, игнорируется, деление не производится, переменная не изменяется.

Команды скрипта

END – конец скрипта

Команда может использоваться для обозначения последней исполняемой команды скрипта. В файле может быть далее сколько угодно текста – он никогда не будет считан.

Формат команды: **END**

Примечание: при наличии команды **END** требование завершать файл скрипта пустой строкой не актуально, строка с командой **END** фактически и будет последней строкой. Ниже удобно помещать описание скрипта, комментарии автора и т.п. информацию для пользователя.

CLR – очистка

CLEAR

Установка всех параметров в начальное состояние: все пиксели гасятся, все переменные обнуляются, BRIGHT=FF, FADE=00, COLOR=000000.

Формат команды: **CLR**

Примечание: команда помимо прочего очищает и стек вложенных циклов, поэтому она не должна использоваться внутри цикла.

PB – яркость рисования

PAINT BRIGHT

Команда устанавливает значение параметра BRIGHT, которое затем используется при формировании «рисунка».

Формат команды: **PB=XX**

GB – глобальная яркость

GLOBAL BRIGHT

Команда устанавливает одновременно для всех пикселей одинаковое значение яркости. Текущее значение яркости рисования не изменяется.

Формат команды: **GB=XX**

PF – затухание рисования

PAINT FADE

Команда устанавливает значение параметра FADE, используемое при формировании «рисунка».

Формат команды: **PF=XX**

Примечание: отрицательные значения должны быть приведены в нужный формат, например, вместо -1 надо указывать FF, вместо -2 соответственно FE и т.д. Число 80 обозначает предельно допустимое значение -128. Предельно допустимое положительное значение 7F соответствует десятичному 127.

GF – глобальное затухание

GLOBAL FADE

Команда устанавливает одновременно для всех пикселей одинаковое значение FADE. Текущее значение затухания рисования не изменяется.

Формат команды: **GF=XX**

PC – цвет рисования

PAINT COLOR

Команда устанавливает текущий цвет пикселей, используемый для формирования «рисунка».

Формат команды: **PC=XXXXXX**

GC – глобальный цвет

GLOBAL COLOR

Команда устанавливает в указанный цвет одновременно все пиксели. Текущий цвет «рисования» не изменяется.

Формат команды: **GC=XXXXXX**

GI – глобальный цвет по индексу

GLOBAL INDEX

Команда устанавливает одновременно все пиксели в цвет, определяемый индексом. Индекс является аналогом значения Н в цветовой системе HSV. Текущий цвет «рисования» не изменяется.

Формат команды: **GI=XX**

PI – цвет пиксела по индексу

PAINT INDEX

Команда устанавливает текущий цвет, используемый для формирования «рисунка», по индексу цвета. Индекс является аналогом значения Н в цветовой системе HSV.

Формат команды: **PI=XX**

RGC – случайный глобальный цвет

RANDOM GLOBAL COLOR

Команда устанавливает все пиксели в случайный цвет. Текущий цвет «рисования» не изменяется.

Формат команды: **RGC**

RPC – случайный цвет рисования

RANDOM PAINT COLOR

Команда устанавливает случайный цвет рисования.

Формат команды: **RPC**

PM – пиксельная карта

PAINT MAP

Команда формирования «рисунка»: в ней перечисляются номера пикселей, получающих ранее установленные значения параметров.

Формат команды:

PM=XX или **PM=XXXXXX...XX**

Первый вариант задает единственный пиксел, второй – произвольную группу пикселей.

Примечание: числа **XX** могут содержать номер пикселя больше, чем физически существует – такой некорректный номер пикселя игнорируется.

BM – карта яркости

BRIGHT MAP

Команда, в которой аналогично **PM**, перечисляются номера пикселей, для которых устанавливается ранее заданное значение параметра **BRIGHT**, остальные свойства пикселя не меняются.

Формат команды:

BM=XX или **BM=XXXXXX...XX**

Первый вариант задает единственный пиксел, второй – произвольную группу пикселей.

Примечание: числа **XX** могут содержать номер пикселя больше, чем физически существует – такой некорректный номер пикселя игнорируется.

FM – карта «затухания»

FADE MAP

Команда, в которой аналогично **PM** и **BM**, перечисляются номера пикселей, для которых устанавливается ранее заданное значение параметра **FADE**, остальные свойства пикселя не меняются.

Формат команды:

FM=XX или **FM=XXXXXX...XX**

Первый вариант задает единственный пиксел, второй – произвольную группу пикселей.

Примечание: числа **XX** могут содержать номер пикселя больше, чем физически существует – такой некорректный номер пикселя игнорируется.

REV – обратная нумерация

REVERSE

Команда меняет направление нумерации пикселей на обратное для команд **PM**, **BM** и **FM** (далее - **xM**).

После этой команды команда **xM=00** фактически будет «закрашивать» самый последний пиксел, а не самый первый, **xM=01** закрасит предпоследний и т.д.

Формат команды: **REV**

Примечание: команда меняет направление нумерации всякий раз, когда выполняется, т.е. 2 подряд идущие команды **REV** оставят нумерацию неизменной.

WT - задержка

WAIT

Задержка на заданное количество десятимиллисекундных интервалов. Эта команда обязательно должна присутствовать в описании эффекта, так как именно во время задержки происходит визуальное обновление информации, реализованное предыдущими командами.

Формат команды: **WT=XX**

Примечание: задержка осуществляется на **XX*10** миллисекунд.

PNT – отрисовка

PAINT

Вызывает немедленное отображение «нарисованной» предыдущими командами картинки. Эквивалентна команде **WT=01**, т.е. приостанавливает выполнение скрипта на 10 мс, в течение которых происходит отрисовка.

Формат команды: **PNT**

RST – рестарт

RESTART

Команда вызывает немедленное выполнение скрипта с самой первой его строки.

Формат команды: **RST**

Примечание: строки скрипта после команды **RST** никогда не будут считаны и исполнены, поэтому могут содержать любые данные.

Все встроенные переменные сохраняют свое значение.

NEG – инверсия

NEGATIVE

Инвертирует биты яркости у всех пикселей, т.е. яркость **FF** превращается в **00**, яркость **55** превращается в **AA** и т.д.

Формат команды: **NEG**

RPT – начало цикла

REPEAT

Начало циклически повторяемой области.

Формат команды: **RPT**

Каждой команде **RPT** обязательно должна соответствовать нижерасположенная команда конца цикла **LV**. Наличие в скрипте команд **RPT** без парных им **LV**, является грубой ошибкой. Так как возможности выдать какое-либо предупреждение пользователю не имеется, поведение скрипта в этом случае будет (непредсказуемо) не корректным.

LV – конец цикла

LOOP if VAR

Конец циклически повторяемой области команд.

Формат команды: **LVn** или **LVn#XX**, где n – номер переменной, а # - знак отношения (>, <, = или !).

Второй вариант формата команды позволяет выполнять повтор тела цикла только если значение переменной больше, меньше, равно или не равно числу.

Команда по первому варианту формата работает следующим образом. Значение указанной переменной уменьшается на 1, и если в результате переменная станет равна нулю, цикл завершается, а если переменная не равна нулю, то происходит переход на соответствующую строку с командой **REP**. Таким образом, тело цикла всегда выполняется хотя бы 1 раз.

Циклы могут быть вложенными, глубина вложенности лимитирована пятью уровнями. Если встречается команда **LV** без соответствующей ей команды **RPT**, она приводит к перезапуску скрипта с самого начала.

Если вложенность циклов превышает лимит, это грубая ошибка в скрипте. Так как возможности выдать какое-либо предупреждение пользователю не имеется, поведение скрипта в этом случае будет (непредсказуемо) не корректным.

INF – бесконечное повторение

INFINITY

Эта команда, в отличие от **LV**, вызывает бесконечное повторение области цикла, прервать которое можно только органами управления устройства.

Формат команды: **INF**

Примечание: строки скрипта после команды **INF** никогда не будут считаны и исполнены, поэтому могут содержать любые данные.

Если команде **INF** не предшествовала команда **RPT**, это ошибочная ситуация, при которой произойдет выполнение скрипта сначала, но при этом уменьшится на 1 допустимая глубина вложенности циклов. Наличие **INF** без соответствующего ей **RPT** следует избегать.

RRC и RLC – циклический сдвиг цвета

ROTATE RIGHT COLOR

ROTATE LEFT COLOR

Команды циклического сдвига вправо и влево соответственно. Работают так: при сдвиге вправо цвет последнего пиксела переносится в первый пиксел, цвет предпоследнего – в последний, пред-предпоследнего в предпоследний и т.д. При сдвиге влево цвет первого пиксела переносится в последний, цвет второго в первый, третьего во второй и т.д.

Исходное состояние пикселей	Команда	Состояние пикселей после выполнения команды
	RRC	
	RLC	

Другие параметры пикселей и параметры рисования не изменяются.

Формат команд:

RRC

RLC

SRC и SLC – сдвиг цвета пикселей

SHIFT RIGHT COLOR

SHIFT LEFT COLOR

Команды сдвига цвета пикселей вправо и влево соответственно. В отличие от команд циклического сдвига цвета **RRC** и **RLC**, при выполнении сдвига вправо цвет первого символа становится черным, а при сдвиге влево – черным становится последний символ.

Исходное состояние пикселей	Команда	Состояние пикселей после выполнения команды
	SRC	
	SLC	

Другие параметры пикселей и параметры рисования не изменяются.

Формат команд:

RRC

RLC

RRB и RLB – циклический сдвиг яркости пикселей

ROTATE RIGHT BRIGHT

ROTATE LEFT BRIGHT

Команды циклического сдвига яркости пикселей вправо и влево соответственно. Действуют аналогично **RRC** и **RLC**, только перемещается не цвет пикселей, а яркость.

Другие параметры пикселей и параметры рисования не изменяются.

Формат команд:

RRB

RLB

SRB и SLC – сдвиг яркости пикселей

SHIFT RIGHT BRIGHT

SHIFT LEFT BRIGHT

Команды сдвига яркости пикселей вправо и влево соответственно. Действуют аналогично командам SRC и SLC, но перемещается яркость пикселей, а не цвет. Соответственно, при выполнении SRB обнуляется яркость первого пикселя, а при выполнении SLB - последнего.

Другие параметры пикселей и параметры рисования не изменяются.

Формат команд:

SRB

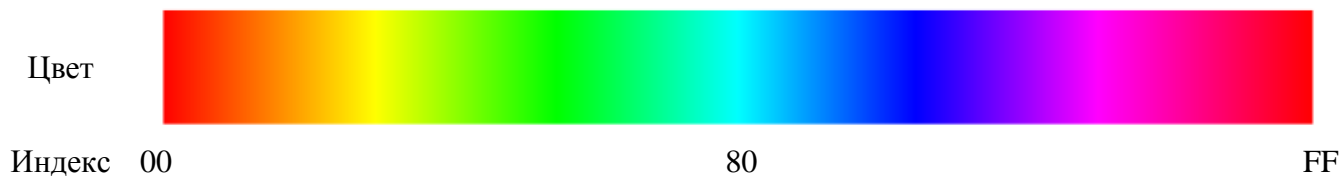
SLB

Сводная таблица команд

Команда	Описание
CLR	Очистка свойств пикселей в начальное значение
END	Принудительное завершение скрипта
RST	Рестарт скрипта с начала
INF	Бесконечный цикл
PNT	Немедленное отображение пикселей
RGC	Случайный цвет для всех пикселей
REV	Реверс нумерации пикселей для команд «рисования»
NEG	Инверсия яркости всех пикселей
RPC	Случайный цвет для «рисования»
RPT	Начало цикла
RRC	Циклический сдвиг цвета пикселей вправо
RLC	Циклический сдвиг цвета пикселей влево
SRC	Сдвиг цвета пикселей вправо
SLC	Сдвиг цвета пикселей влево
RRB	Циклический сдвиг яркости пикселей вправо
RLB	Циклический сдвиг яркости пикселей влево
SRB	Сдвиг яркости пикселей вправо
SLB	Сдвиг яркости пикселей влево
LV	Конец цикла
PB	Установить яркость для «рисования»
GB	Установить яркость всех пикселей
PC	Установить RGB-цвет для «рисования»
GC	Установить RGB-цвет для всех пикселей
PF	Установить для «рисования» параметр FADE
GF	Установить параметр FADE для всех пикселей
PM	Рисование по карте пикселей
BM	Установка яркости по карте пикселей
FM	Установка FADE по карте пикселей
GI	Установить HSV-цвет по индексу для всех пикселей
PI	Установить HSV-цвет по индексу для «рисования»
WT	Отображение пикселей и задержка скрипта

Индексы цвета

Для команд индексного задания цвета можно ориентироваться на следующие соответствия индекса цвету:



Так же для характерных цветов можно использовать следующую таблицу:

Цвет	Индекс
Красный	00
Оранжевый	18
Желтый	2B
Зеленый	55
Голубой	93
Синий	AA
Фиолетовый	D5
Лайм (светло-зелёный)	3D
Небесно-голубой	80
Сиреневый	C3
Малиновый	E8

Белый и черный цвет задать индексом невозможно, следует использовать команды RGB-задания цвета.

Порядок составления скрипта

Разбор каждой команды и её выполнение требует определенных временных затрат, поэтому рекомендуется придерживаться ряда простых правил, с целью получения максимальной производительности при воспроизведении скрипта.

1. Не смотря на то, что комментарии в тексте скрипта игнорируются, сам процесс их выявления и «игнорирования» требует времени. Чем больше «лишних» символов в скрипте, тем ниже скорость его исполнения. Настоятельно рекомендуется после отладки скрипта очистить его от комментариев полностью. Аналогичные рекомендации действуют и в отношении пустых строк.
2. Следует стремиться к достижению нужного результата предельно коротким скриптом. Чтение файла скрипта осуществляется блоками по 256 байт – это самая длительная операция, и чем меньше таких считываний будет, тем лучше. Идеально, если размер файла не превышает 256 байт.
3. Следует стремиться к минимуму циклов в скрипте, т.к. цикл – самая ресурсоемкая команда. Размер самого цикла тоже следует по возможности уменьшать, т.к. если считывание очередного блока с карты будет осуществляться в цикле, это наверняка станет причиной заметных глазу «замираний» эффекта.
4. Следует стремиться использовать команды РМ с максимально возможным количеством перечисляемых в ней номеров пикселей, т.к. 5 однопиксельных команд РМ будут выполняться дольше, чем одна, но сразу для тех же 5 пикселей.
5. Всегда, когда это возможно, следует применять «глобальные» команды установки цвета, затухания и яркости, т.к. выполнение тех же действий в циклах потребует значительно больше времени.

Свойства пикселей

Каждый пиксел, из числа аппаратно доступных, имеет четыре основных свойства: яркость, затухание и цвет.

FADE – затухание

Затухание – это скорость уменьшения яркости, число от -127 до 127¹. Это число показывает, на какую величину автоматически уменьшится яркость пикселя через 10 мс (см. команду WT).

Отрицательное значение означает, что яркость будет увеличиваться, но, когда яркость достигнет предельного значения 255, знак параметра яркости автоматически изменится на противоположный. Т.е. пиксел, для которого было задано отрицательное значение FADE будет увеличивать свою яркость до максимума, а затем с тем же темпом будет погасать.

BRIGHT - яркость

Яркость это число от 0 до 255, определяющее условную яркость пикселя, 0 – нет свечения, 255 – яркость максимальна.

¹ Десятичное отрицательное число должно быть преобразовано в шестнадцатеричный байт. Проще всего это сделать при помощи калькулятора Windows в режиме «программист».

За счет свойства FADE яркость пиксела может автоматически увеличиваться или уменьшаться, но при этом никогда не выходит за пределы допустимого диапазона 00...FF.

COLOR – цвет

Цвет пиксела задается тремя байтами, определяющими уровень RGB-цветовых составляющих.

Принцип формирования эффектов

Общий подход к формированию эффектов заключается в создании последовательностей команд скрипта, формирующих эффект «кадр за кадром». Под кадром тут подразумевается статичное состояние светящихся пикселей. Так как обновление состояния пикселей происходит только при исполнении команды задержки WT, можно считать кадром все, что сформировано предшествующими задержке командами. Такой кадр виден в течение указанного интервала времени задержки, при этом яркость пикселей может автоматически меняться (см. FADE) каждые 10 мс, а затем начинается формирование очередного кадра.

То есть командами задержки в скрипте разделены отдельные кадры эффекта. А последовательная смена кадров, как и в мультипликации, формирует динамическое изменение картинки, т.е. реализует эффект во времени.

Рисунок создается включением тех или иных пикселей в нужный цвет. Делается это, в общем случае, такой последовательностью команд.

Сначала задается исходное состояние всех пикселей. Как правило, это полное отсутствие свечения. Это может быть сделано командой CLR. Так же можно использовать «глобальные» команды GC или GB, ведь установка нулевой яркости приведет к погасанию всех пикселей независимо от их заданного цвета, а установка «черного» цвета так же погасит пиксели независимо от их яркости.

Какую команду использовать лучше, зависит от того, как очередной кадр будет отличаться от предыдущего. Если новый кадр должен лишь поменять яркость отдельных пикселей, следует сохранить их цвет. Если требуется поменять цвет выбранных пикселей, не меняя их яркости, то следует оставить неизменной яркость. И так далее.

После установки исходного состояния пикселей следует сформировать новый рисунок. Для этого командами PB, PC и PF устанавливаются требуемые свойства вновь включаемых (обновляемых) пикселей, а затем командой PM указываются номера пикселей, принимающих эти свойства. Если требуется, следует повторить эту последовательность нужное количество раз.

Завершает рисование кадра, как уже было сказано, команда задержки. В этот момент сформированный рисунок отображается на светодиодных пикселях. Если задержка более, чем на 10 мс, то пока она длится, каждые 10 мс происходит автоматическое изменение яркости пикселей в соответствии с заданными для них значениями FADE. К концу задержки яркость некоторых (или всех) пикселей может быть равной 0, при этом цвет и значение FADE не изменится (кроме отрицательных значений FADE – см. описание команды).