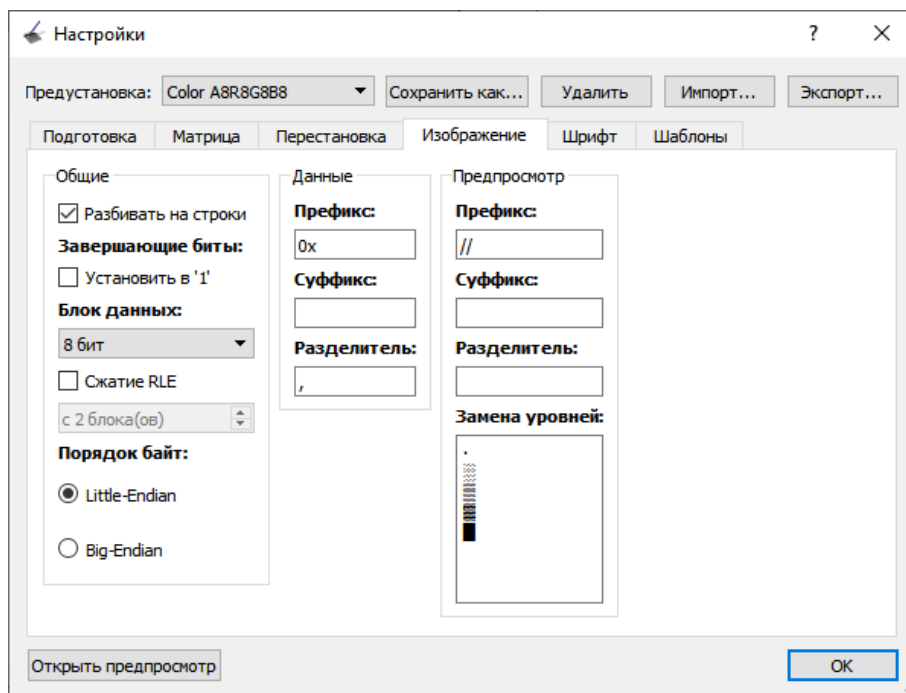


Lcd-image-converter. Подготовка данных

Байты в четверках нужно менять местами. Однако lcd-image-converter именно и меняет байты в словах. Что вовсе не означает, что он меняет порядок вывода байтов при выводе в этих самых байтах.

То есть:

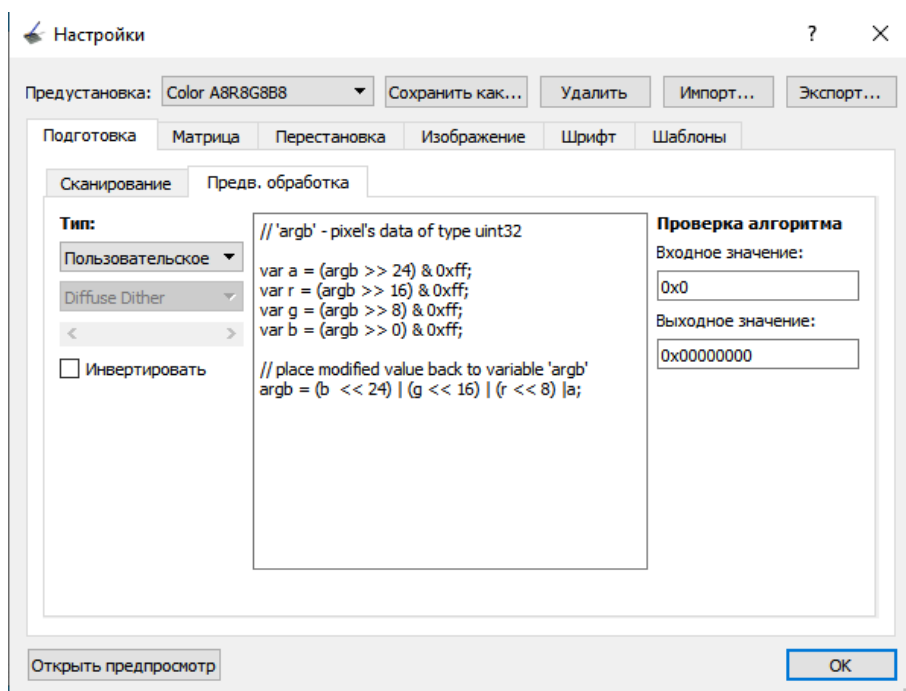
Настройки -> Преобразование -> Изображение -> Блок данных -> 8 бит



Little-Endian (Big-Endian) нам не помогут, поскольку работают в режимах 16 бит и 32 бита.

Для подготовки байтовых данных нужно:

Настройки -> Преобразование -> Подготовка -> Предв. обработка -> Тип: -> Пользовательское

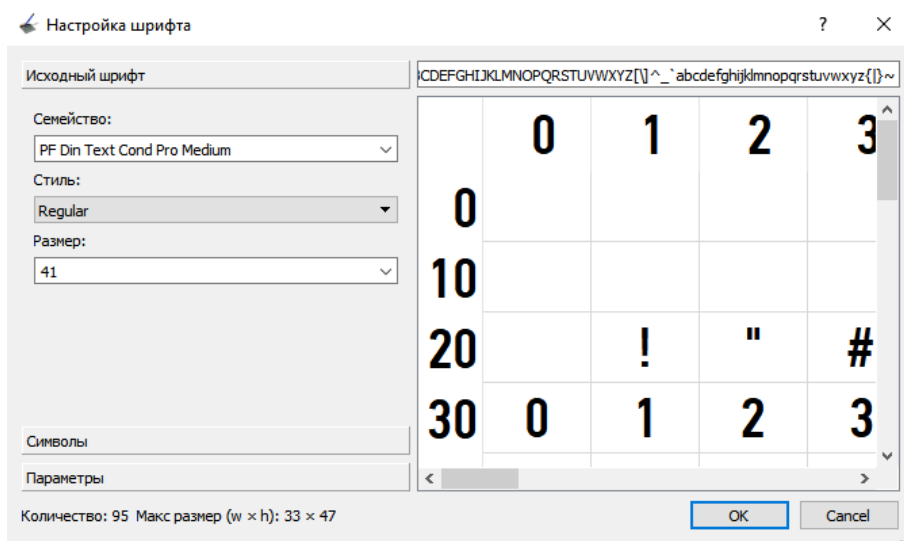


То есть просто поменять порядок ручками. Тогда меняется порядок байт в четверках. И на монитор выводится чудненько!

Пропорциональные шрифты

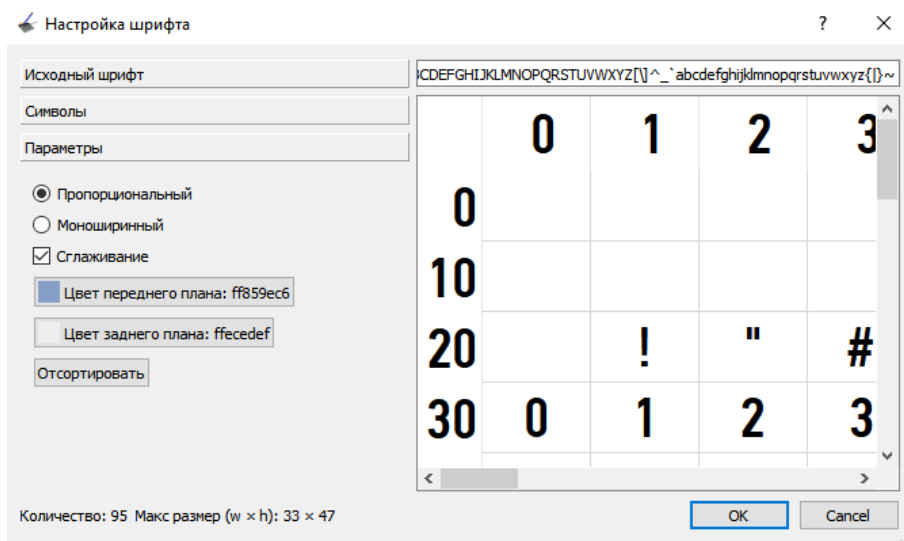
Шрифт создать Pf_Dn_33

Шрифт изменить



PF Din Text Cond Pro Medium. Regular. 41.

Basic Latin



Пропорциональный. Сглаживание.

Цвет переднего плана #859ec6. Цвет заднего плана #ecedef.

Изменение размеров

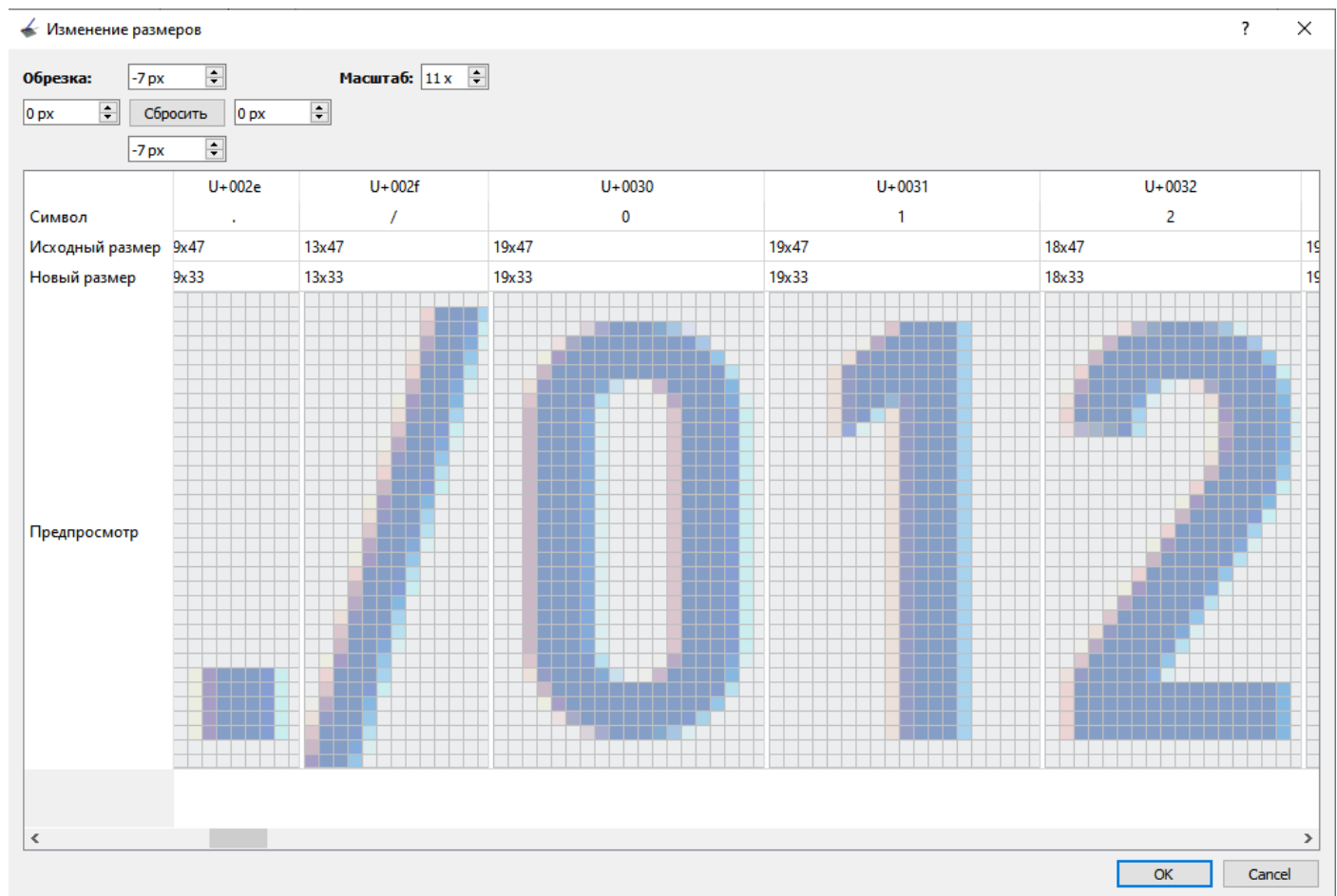
Обрезка 0 -7 0 -7

Собственно нам нужно, чтобы ноль был с учетом всех сглаживаний влезал в строку индикатора

На картинке обрезки видно, что остается по 2 пикселя сверху и снизу, чего нам и нужно. Из этих соображений выбирается кегль (размер) шрифта 41.

Кегль — параметр шрифта, означающий высоту его литер. Кегль включает в себя высоту строчной буквы с самым длинным выносным элементом и пробельное расстояние снизу неё. Величина кегля определяется числом пунктов.

Поскольку выносных элементов у нас не предвидится ни нижних, ни верхних, то и режем по нулю.



Понятно, что при таком раскладе мы не сможем отображать ни «Й», ни «j». Кроме того, при попытке вывести несколько строк, мы получим не здоровый интрильяж. Но у нас одна строка.

В отличие от проекта calculator_429I_DIS поворачивать символы через меню изображение -> вращение необходимости у нас нет.

Преобразовываем в соответствии с пользовательскими настройками, описанными выше.

Затем чистим ручками лишние символы. Как удалить не нужные сразу я пока не научился.

