

**Введение** Этот документ описывает различные функции редактора Nextion. Редактор Nextion используется для быстрого создания GUI человеческого машинного интерфейса для устройств Nextion HMI. Таким образом, GUI может быть создан в течение нескольких часов, а не недель, и дней, а не месяцев. Поэтому, хотя мы не будем освещать основы, такие как открытие файла, мы будем нацелены на то, что может оказаться полезным, или напоминания, которые необходимы.

### Требования

- Операционная система Windows (XP или выше). Пользователи должны знать и иметь возможность использовать свою ОС Windows. Поддержка Windows выходит за рамки Nextion.
- Установлены сборки .NET 3.5. При необходимости, скачать и установить .Net 3.5 с Microsoft Website [here].
- Базовые навыки программирования являются необходимыми. Набор инструкций Nextion состоит из текстовых команд ASCII и двоичных возвращаемых данных. Событие Touch компонента "Send Component ID" может использоваться для переноса задач программирования на MCU пользователя.
- Таким образом, быстрое создание HMI GUI для Nextion не требует экстремальных навыков – но базовые навыки программирования приветствуются. Зпрограммировав Nextion логику, после этого потребители должны иметь учреждение в программировании.

Примечание: установка в других операционных системах может быть выполнена успешно, но официально не поддерживается и выходит за рамки любого руководства.

### Загрузка редактора Nextion

**Последняя версия Nextion редактор можно скачать [здесь].**

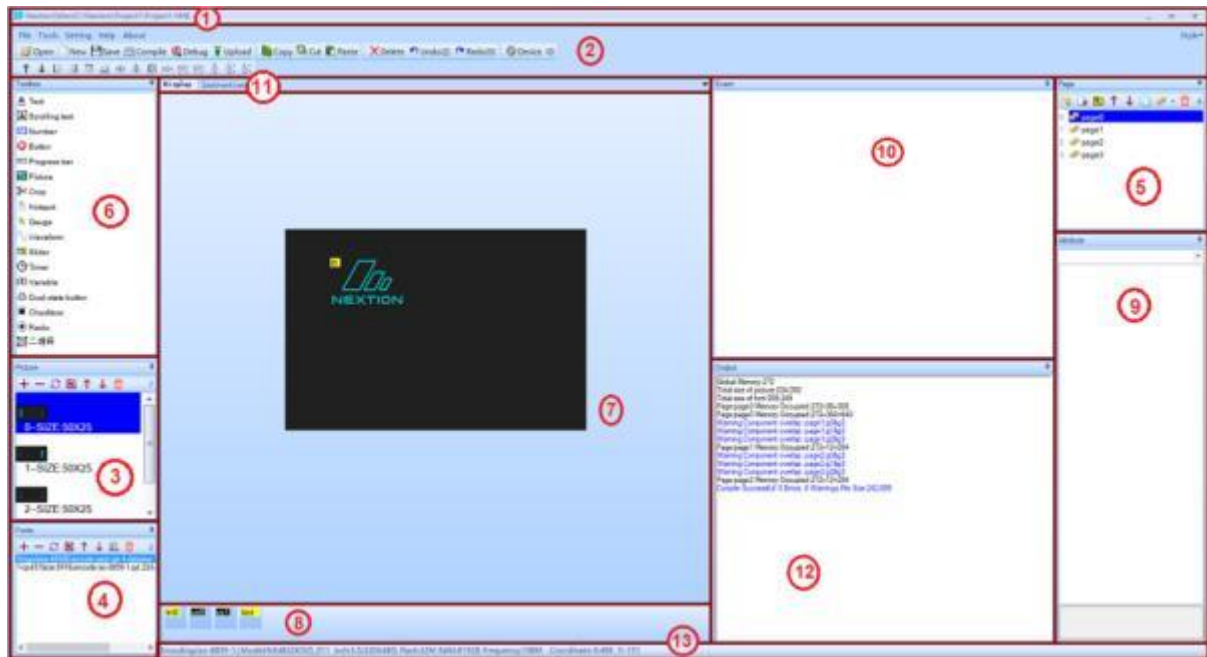
Для загрузки доступны две версии программы nextion-setup.

1. Версия EXE устанавливается через Windows MSI для более автоматической установки. Только одна версия редактора Nextion может быть зарегистрирована одновременно через версию EXE. При обновлении в редакторе Nextion, Auto Update установит версию EXE
2. Версия ZIP может быть распакована в выбранную пользователем папку и запущена непосредственно из этой папки. Для поддержания нескольких версий редактора Nextion рекомендуется использовать версию ZIP. При обновлении в редакторе Nextion, обновление вручную запустит ваш веб-браузер на странице загрузки, так что вы можете загрузить версию ZIP

### Части к этому документу

3. Основной Интерфейсом Редактора Nextion ... <перейти>
4. Отладка Симулятор Обзор ... <перейти>  
Хочу укороченный вариант: попробовать в Nextion редактора: краткий обзор

# Главный Интерфейс Nextion Editor



1. Заголовок ... <перейти>
2. а) главное меню ... <перейти> б) панели инструментов ... <перейти>
3. Панели изображения ... <перейти>
4. Панели ресурса шрифта Зи ... <перейти>
5. Панели Страницы ... <перейти>
6. Панель Компоненты ... <перейти>
7. Внешний вид Визуального чертежа Компонентов ... <перейти>
8. Внешний вид Не Визуальных Компонентов ... <перейти>
9. Панель Атрибутов ... <перейти>
10. Код Пользователя ... <перейти>
11. Дисплей/Инструкция Вкладка ... <перейти>
12. Выход ... <перейти>
13. Строке Состояния ... <перейти>

1. Title Bar ... <goto>
2. а) Main Menu ... <goto> and б) Toolbars ... <goto>
3. Picture Resource Pane ... <goto>
4. ZI Font Resource Pane ... <goto>
5. Page Pane ... <goto>
6. Components Pane ... <goto>
7. Design Canvas Visual Components ... <goto>
8. Design Non Visual Components ... <goto>
9. Attributes Pane ... <goto>
10. User Event Code ... <goto>
11. Display/Instruction Tabs ... <goto>
12. Output ... <goto>
13. Status Bar ... <goto>

Редактор Nextion может быть установлен в синий или черный тематический стиль (в правом верхнем углу). Много из форточек можно также отрегулировать как в размере, так и в их положении. При необходимости, вы можете сбросить эти настройки, выбрав **Reset layout** под **Setting** menu.

**Styles** The Nextion Editor can be set to a Blue or Black themed style (in the upper right corner). Many of the panes can also be adjusted in both size and their location. When needed, you can reset these settings by selecting the **Reset layout** under the **Setting** menu.

## 1. строка заголовка

Строка заголовка содержит путь и Имя файла проекта HMI при загрузке проекта HMI. Если проект HMI в данный момент не загружен, TFT-файл можно загрузить в Отладочный симулятор, нажав кнопку Debug.

## 2.главное меню



### Меню "Файл"

Здесь, пользователи могут создать новый проект, открыть существующий проект, сохранить текущий проект, закрыть их текущий проекта, и выйти на Nextion редактор. **Import Project** добавит существующий проект в текущий проект – как правило, с результатом последствий именованя и перенумерование вопросов. При необходимости рекомендуется импортировать отдельные страницы. **Clear Recent Projects** очистит имена файлов проекта на панели «the **Recent projects**» - он не удалит файлы с жесткого диска.

Понятно недавних проектов будет четкий проект Имена В последних проектах области – это не удалить любой из файлов с жесткого диска.

Два других полезных ярлыков папок включены здесь: открыть папку сборки и резервного копирования Dir. Все скомпилированные проекты создают \*.TFT-файл в C:\Users\Username\AppData\Roaming\Nextion Editor\bianyi папка. **Open Build Folder** откроет эту папку в Проводнике Windows. В каталоге Backup хранится копия старых проектов HMI, открытых в новой версии редактора Nextion. C:\Users\Username\AppData\Roaming\Nextion Editor\backup папка резервного копирования. **Backup Dir** будет открыть эту папку в Проводнике Windows.

## меню инструментов

В редактор Nextion встроен \*.Ширифта инструмент “Font Creator” может быть запущен при выборе **Font Generator**. Любой проект, содержащий компоненты, которые выводят .txt или .val визуальный атрибут требует, чтобы в проект HMI пользователя был включен хотя бы один шрифт. Font Creator включен как удобный способ быстро генерировать шрифт, хотя, возможно, не самое лучшее качество.

### Font Creator



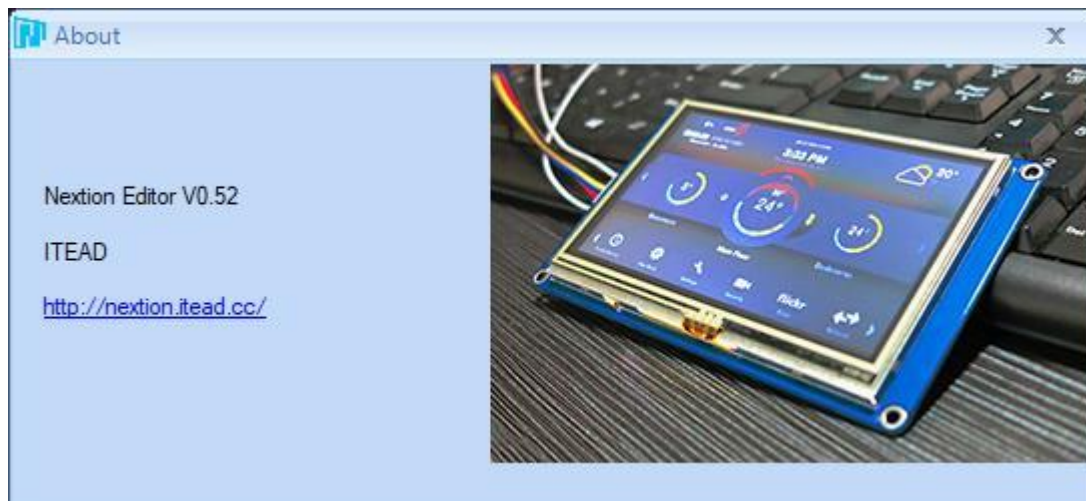
- + Высота символов в пикселях, кратно 8 от 16 до 160.
- + **Ширина символов будет 1/2 этой высоты.!!!**
- + Выберите высоту, кодирование, шрифт из установленных шрифтов, дополнительный интервал.
- + Дать свой шрифт имя и нажмите кнопку *GenerateFont..*
- + Сохранить \*.файл шрифта zi в выбранной папке шрифта
- + Вам будет предоставлена возможность добавить это в ваш проект HMI.

#### меню настроек

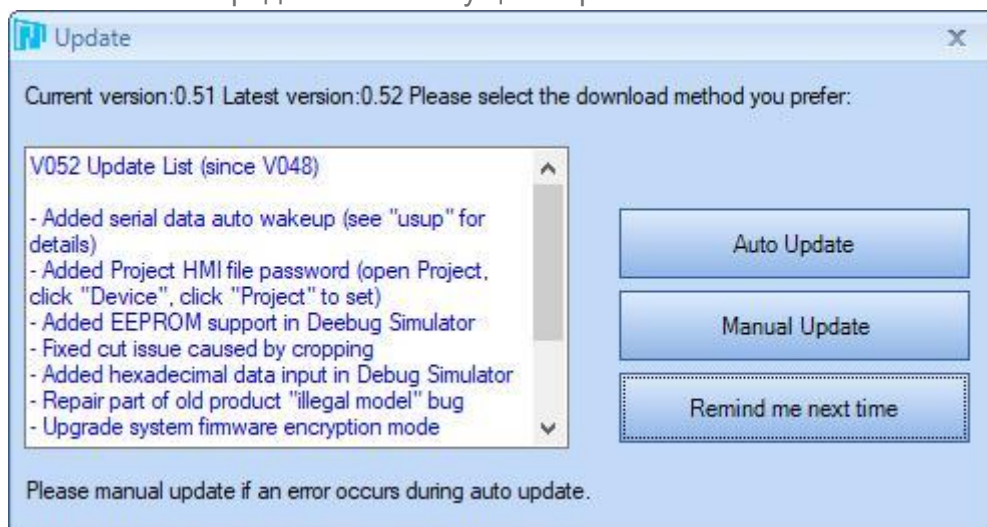
Здесь пользователь может изменить некоторые редактора параметров, выбрав Настройки, или сбросить их Редактор стилей макета выбрав сброс макета.

#### About Menu

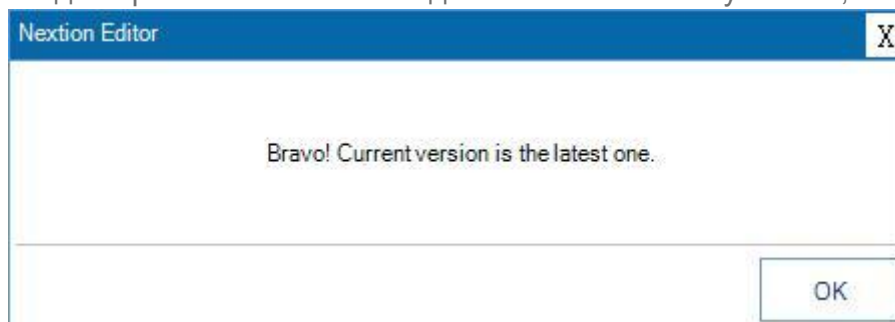
**About** покажет текущую версию Nextion редактора и Nextion сайте ссылке.



**Check for new version** будут использовать Интернет, чтобы проверить новую версию. Если доступна новая версия, появится диалоговое окно обновления. Существует две версии редактора Nextion: версия EXE и Версия ZIP. Версия EXE будет устанавливаться в папке Program Files (x86) и автоматически заботиться о ассоциациях файлов. Выбрав **Auto Update** будет установить exe-версию. ZIP-версию можно загрузить и извлечь в любую папку по выбору, а затем запустить непосредственно из этой папки. Выбор **Manual Update** запускает внешний веб-браузер пользователя на странице загрузки редактора Nextion. Здесь можно скачать версию ZIP. Выбрав **Remind me next time** закроет обновить диалоговое окно без каких-либо обновлений и продолжить с текущей версией.

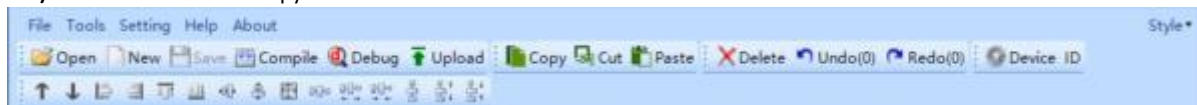


Когда версия является последней и наиболее актуальной, пользователи увидят:



**New Message** меню не функционирует в данный момент.

## 2.b) Toolbars панели инструментов



**Open, New and Save** Это позволит **открыть** существующий проект, создать **новый** проект, или **сохранить** текущий проект..

### Compile

Используйте Compile для перестроения загруженного проекта. Все предупреждения или сообщения об ошибках будут помещены в область вывода. Если ошибки нет, \*.TFT-файл будет помещен в папку bianyi, но если есть ошибка, то это \*.TFT - файл будет равен нулю байт. Не загружайте (через последовательный или через microSD) нулевой байт \*.TFT к Nextion.

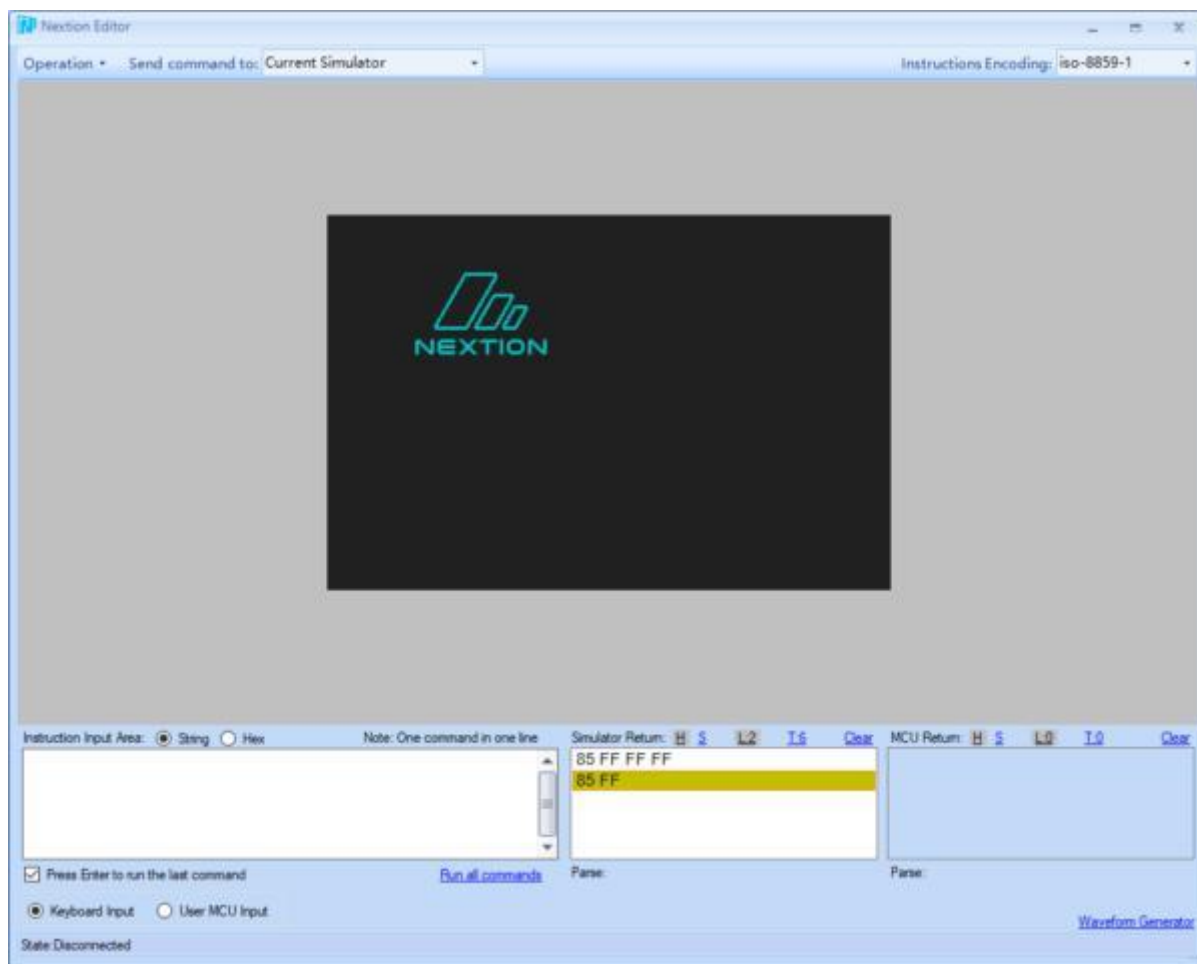
Обратите внимание на любые предупреждения, поскольку они будут означать, что ваш проект не может работать так, как вы ожидаете. Обратите внимание на любые сообщения об ошибках, поскольку они должны быть исправлены, прежде чем продолжить. Сообщения об ошибках дескрипторный, и если это ошибка кода, то пользователь может щелкнуть, чтобы перейти непосредственно к расположению ошибки кодирования.

Компиляция-это скорее процесс сборки и ассемблирования. Это только заявлено, так что пользователи не делают неправильные ожидания родного кода ARM при выполнении запросов функций и / или отчетов об ошибках. Nextion остается закрытым ресурсом.

Это указано только для того, чтобы пользователи не делали ложных ожиданий в собственном коде ARM при выполнении запросов функций и / или отчетов об ошибках.

### Debug

В Nextion редактор содержит встроенный симулятор, которые могут быть доступны через Панель инструментов **Debug**. Чтобы быть ясным, это не эмулятор точности и предназначен, чтобы быть достаточным, чтобы помочь в отладке проекта пользователей. Он ни в коем случае не предназначен для репликации устройства Nextion точно. (Любой ОС Windows достаточно, чтобы сделать такую точность недостижимой.) Этот вопрос будет рассмотрен в следующем разделе.



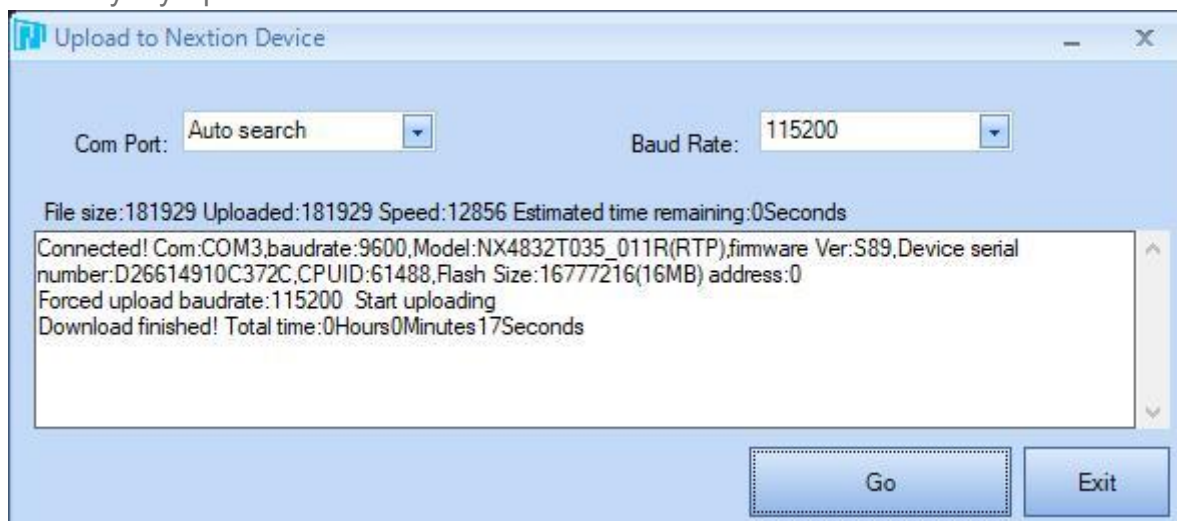
Если проект не загружен в Nextion редактором, **Debug** откроется диалоговое окно открыть скомпилированный \*.TFT-файл напрямую. Это удобно для загрузки демонстрационных версий или обмена идеями без передачи кода. Хотя Simulator может запустить \*.TFT-файл из любой серии Nextion или модели, **важно, что та же версия используется для имитации, которая была использована для создания \*.TFT-файл.**

#### Upload

Выбор Upload запускает диалоговое окно upload. Убедитесь, что Nextion подключен через последовательный (обычно через USB к TTL адаптер) перед загрузкой или порт может быть не доступен для выбора. Функция автоматического поиска будет искать ответ Вашего Nextion на команду connect, но учтите, что данные отправляются на все последовательные порты, которые ищутся (и могут помешать другим подключенным последовательным устройствам). Лучше выбрать правильный порт и скорость передачи данных. Правильная настройка последовательных адаптеров, драйверов Windows, конфликтов устройств и т. д. выходит за рамки поддержки Nextion и области ответственности пользователя, чтобы знать свою используемую операционную



систему и устройства.



Как только Nextion ответит на команду connect, процесс загрузки начнется. Не прерывайте этот процесс до завершения. Если процесс был прерван, может потребоваться сброс последовательного порта. Когда частичный \*.TFT-файл был загружен, и загрузка по последовательному больше не является опцией, то пользователь должен будет загрузить с помощью метода microSD.

**Copy, Cut, Paste and Delete** Пользователи могут выбирать компоненты или несколько компонентов, а затем **копировать**, **вырезать**, **вставить** или **удалить** по мере необходимости.

#### Undo, Redo

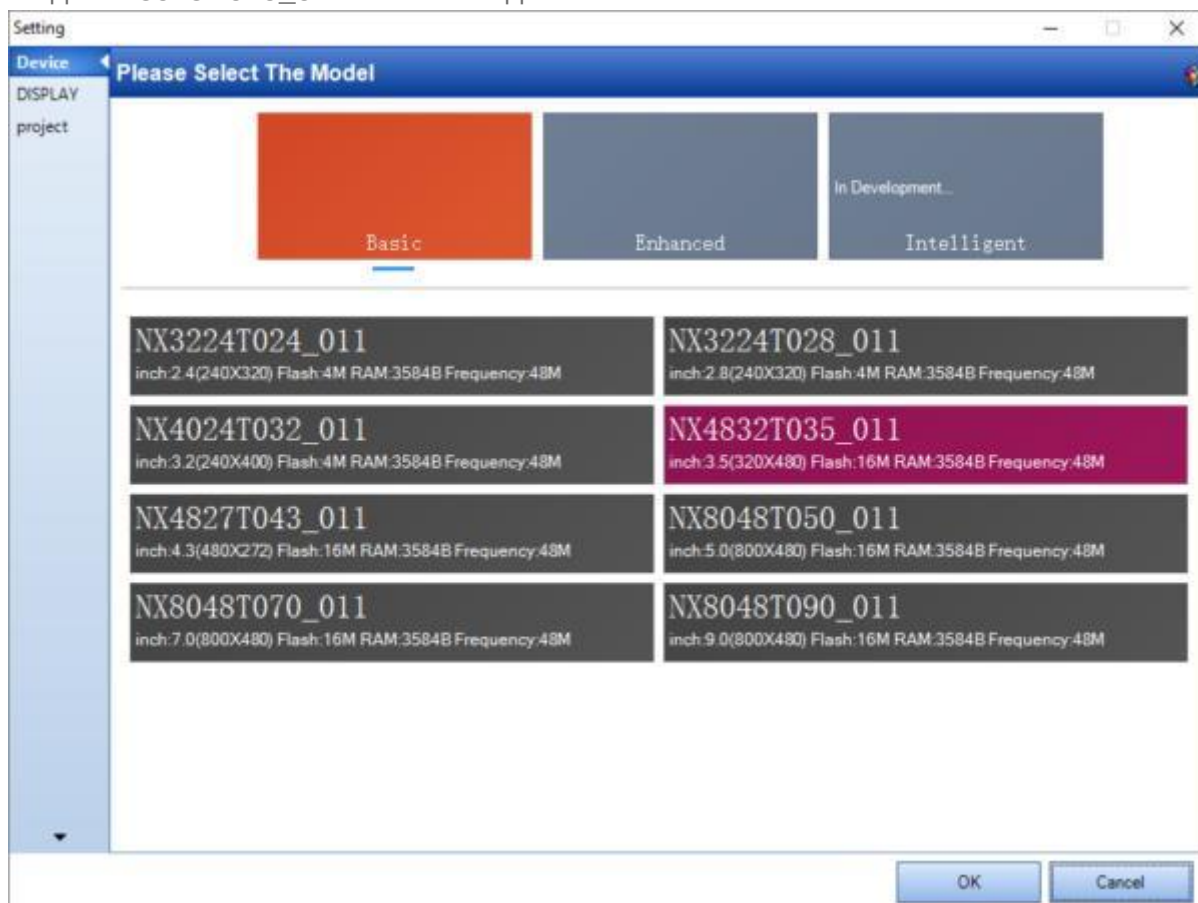
Используйте отмены и повтора, чтобы **отменить** последнее действие или **повторить** последнее отмененное действие.

#### Device

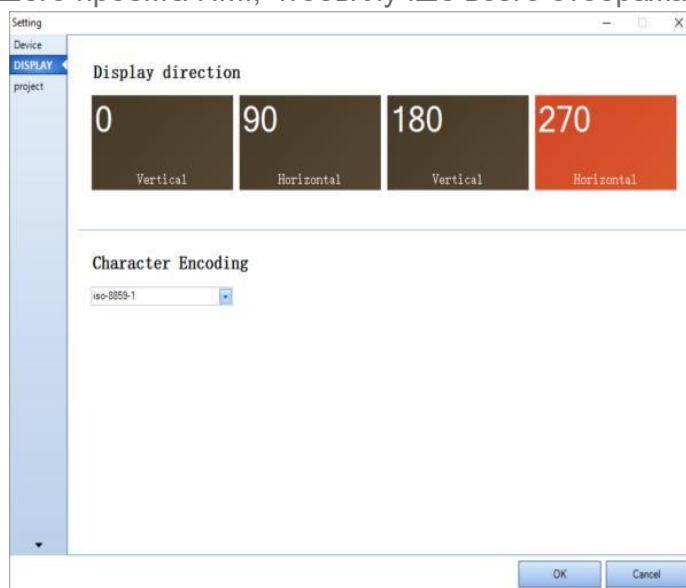
Действия по настройке проекта HMI для серии и модели Nextion обычно выполняются во время создания нового проекта. Когда вам нужно внести изменения, **устройство**будет запущено в следующем окне с **устройством** вкладке выбранное. Сначала выберите Nextion серия: **базовые** для Т-моделей и **усиленный** для К модели. Затем выберите модель Nextion. Для мульти-сенсорный емкостной NextionNX8048K070\_011C: выберите **усиленной** серии и



тогда **NX8048K070\_011** Nextion модели.



Selecting the **DISPLAY** tab пользователь может выбрать ориентацию и кодировку. 0° - собственный Угол обзора для выбранной модели. Пользователи могут выбрать альтернативные ориентации (90°, 180° или 270°), но это не будет родной Угол обзора. Кодировка символов по умолчанию iso-8859-1. Выберите из кодировок символов, которые имеют смысл для вашего проекта HMI, чтобы лучше всего отображать



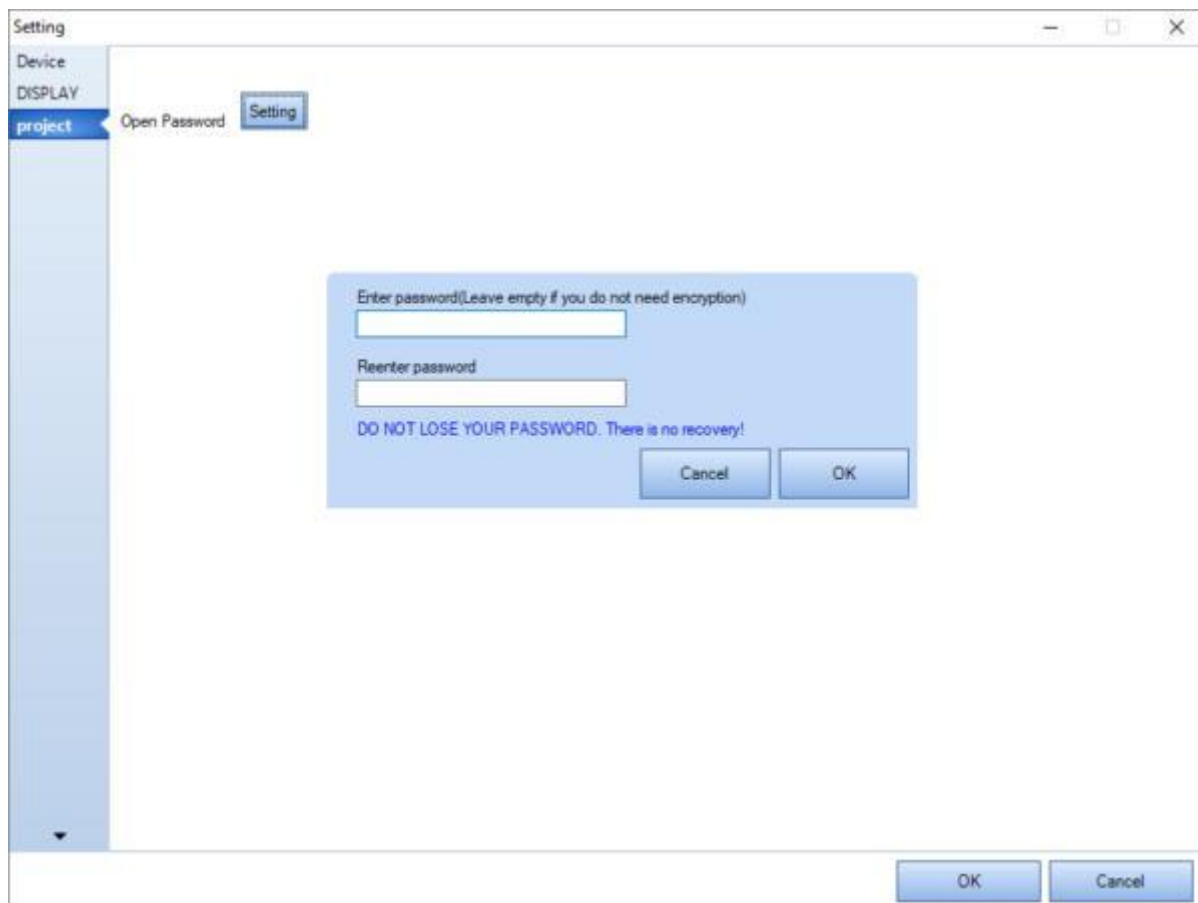
локальные наборы символов.

Существует выбор однобайтовых и двухбайтовых наборов символов. Unicode и UTF не будут в числе поддерживаемых кодировок.

В настоящее время поддерживаются кодировки включает:  
ASCII, ISO-8859 (1,2,3,4,5,6,7,8,9,11,13,15),

GB2312, BIG5, KS\_C\_5601\_1987,  
Windows 874, 1255, 1256, 1257, and 1258

Если вы желаете защитить паролем всю нм проекта, Выбор **project**, выведет пароль кнопка установки. Если существует существующий пароль, его необходимо будет ввести, прежде чем новый пароль может быть установлен. Если пароль утерян, его невозможно получить. **Предупреждение: не теряйте свой пароль. Нет никакого восстановления! Проект с утерянным паролем нужно будет перестроить-так, не потерять-или не использовать.**



## ID

Выбор **ID** будет переключаться, если компонент .objnames отображаются в верхней левой области пространства компонентов. Желтые помеченные компоненты имеют .vscode local, в то время как черные помеченные компоненты имеют .vscode of global. (Подсказка: код события никогда не является глобальным). При выборе нескольких компонентов, зеленые помеченные компоненты указывают на то, что было выбрано несколько компонентов, в то время как один **синий** помеченный компонент будет использоваться в качестве базового компонента. **Чтобы изменить базовый компонент, пока группа еще активна, просто нажмите на уже выбранный компонент, который вы хотите стать базовым компонентом.**

Атрибуты страницы.

- vscode – Видимость. Возможные значения:
  - local – видимость в пределах данной страницы
  - global – видимость на всех страницах.

### The Component Layout Toolbar



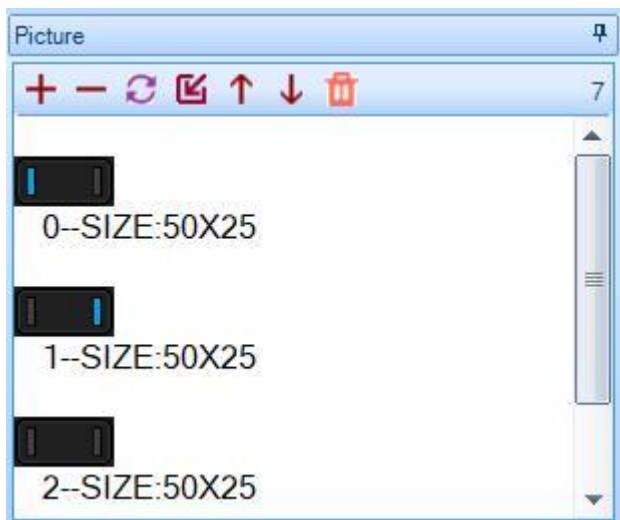
Для перенумерации компонентов: **Bring Top** (Стрелка вверх) будет принимать выбранный компонент(Ы) и изменить нумерацию на высшую .ID на странице. **Bring Bottom** (стрелка вниз) примет выбранный компонент(Ы) и изменить нумерацию на низкой (начиная с 1, страница компонента всегда равна 0) .ID на странице.

Для выравнивания компонентов: **Align Left**, **Align Right**, **Align Top** and **Align Bottom** будет взять группу выбранных компонентов (зеленая ID [маркировка](#)) и довести выравнивание в соответствие с компонентом с синей меткой ID.

Для изменения размеров компонентов: **Same Width**, **Same Height** and **Same Size** будет взять группу выбранных компонентов (зеленая ID этикетки) и установить размер (ширину, высоту или оба), которые соответствуют компонент с синим идентификатором метки.

Для интервала компоненты: **Equal Horizontal**, **Increase Horizontal spacing**, and **Decrease Horizontal spacing** берет группу выбранных компонентов (зеленая ID этикетки) и отрегулируйте горизонтальный интервал между компонентами использование компонентов с Синий ID метки в качестве базового компонента. Дополнительно: **Equal Vertical**, **Increase Vertical spacing**, and **Decrease Vertical spacing** (**равные вертикальные**, **увеличить вертикальный интервал**, и **уменьшить вертикальный интервал** ) будет взять группу выбранных компонентов (зеленая ID этикетки) и регулировать по вертикали между компонентами использование компонентов с голубой код метки в качестве базового компонента.

### 3. Picture Resource Pane *Панель Ресурсов Изображения*



Фотографии импортируются в проект HMI через панель "ресурс изображения". Добавлена фотография с **Добавить**, исключить, **удалить**, и заменить **заменить**. **Вставка** добавит импортированные фотографии до выделенную картинку. Используйте **стрелку вверх** и **стрелку вниз** , чтобы изменить нумерацию на картинке ресурсов в ресурсов области. Используя **Trash** будут удалены все снимки в рамках проекта. Обратите внимание, что Delete не удаляет ресурс изображения, если он используется с компонентом.

Допустимые типы изображений для импорта \*.jpg, \*.png, non animated \*.gif and \*.bmp. При импорте изображения изображение преобразуется в 16-битный формат цвета 565, используемый Nextion. **Nextion не является видеокартой, так как такая прозрачность и в картинной анимации не поддерживается. В собственном 16-битном цвете ресурсы изображений потребляют 16 бит на пиксель или ширину x высоту width x height x 2 bytes..**

Экспорт ресурса изображения осуществляется путем выделения ресурса изображения, щелчка правой кнопкой мыши и выбора экспорт. Доступные форматы изображений для экспорта \*.формат JPG.\* ,png и \*.формат BMP..\* jpg может сэкономить место, но является форматом с потерями. Как только данные будут потеряны, они будут потеряны.

При использовании обрезки для фона настоятельно рекомендуется использовать полноэкранное изображение. Это позволяет избежать вытягивания данных из несуществующего пространства, которое будет напоминать рандомизированные цвета (данные из других мест). Обрезка может потреблять больше циклов, чем использование изображения, использование изображения будет потреблять больше циклов, чем сплошные цвета. Когда время цикла становится высоким для рендеринга, знак и мерцание будут знаками. Nextion является устройством HMI и не предназначен для мульти-медиа и потоковой передачи. Тем не менее, удивительные эффекты все еще могут быть достигнуты с помощью целенаправленного программирования.

Назначение изображения с помощью кода во время выполнения будет выглядеть так: `p0.pic=1` over serial, or as `p0.pic=1` внутри события как код.

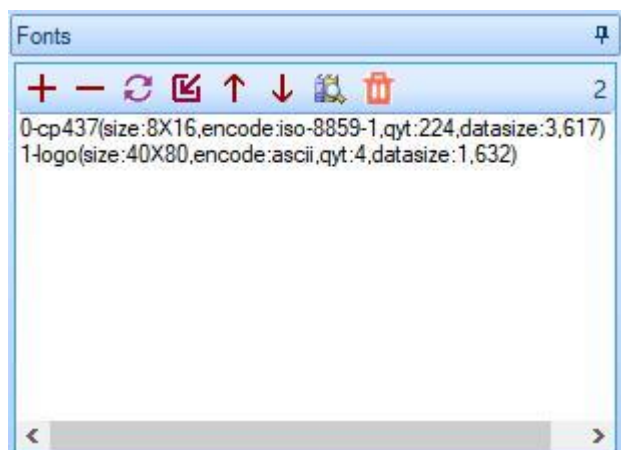
Assignment of the picture by code at runtime would look like: `p0.pic=1` over serial, or as `p0.pic=1` inside an event as code.

**4. ZI Font Resource Pane** The ZI Font Resource is very important! If any visual component used in the user's project contains a .txt or .val attribute, then at least one font must exist in the project. If a user wants to use various size of styled fonts in their HMI project, they will need to generate a ZI Font le for each style and size they will use. For most users this will mean starting with a ZI Font generated with the built-in Font Creator (covered in the Menu section above).

These \*.zi font les will need to be added into the ZI Font Resource Pane. Each ZI Font Resource will show the font resource number, it's given name user gave at creation time, it's size in pixels Width X Height, it's character encoding, number of characters in the set and the size in bytes the font contributes to the HMI project's overall lesize. The appropriate \*.zi font can then be selected by its number and assigned in a visual components .font attribute. Note that the design size of the visual component is set at design time and can not be changed at runtime.

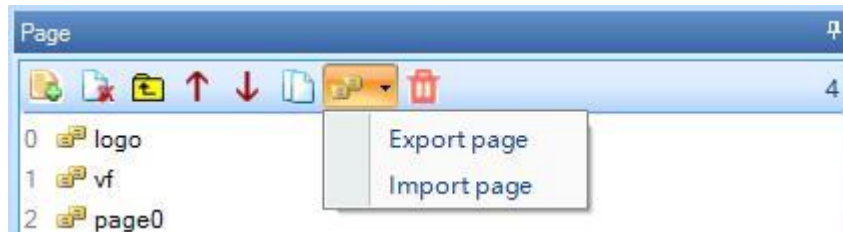
Assignment of the font by code at runtime would look like: `t0.font=1` over serial, or as `t0.font=1` inside an event as code.

Properly formed ZI font resources will have a height that is either equal to the width or a height that is double the width. Height will always be a multiple of 8 from 16 to 192. ZI fonts are monobit (pixel is either on or off) and Fixed Width. **Hint:** Generating a proportional font like Arial where W and @ are 24x24 and squeezing into a 12x24 space is like pouring 2 litres into a one litre bottle and expecting all 2 litres have been saved. Fonts that do not conform to the ZI



A ZI font resource can be viewed by selecting 0123456789 the font and pressing **Preview**. Trash will **Delete all** loaded ZI Font Resources. Multiple \*.zi fonts can be imported with **Add**. A font must not be associated to a component to remove with **Delete**. A font can be swapped out with a different font using **Replace**. **Insert** will import the font before the highlighted font. Use the **Arrow Up** and **Arrow Down** to renumber the font resource number within the ZI Font Resource Pane.

## 5. Page Pane

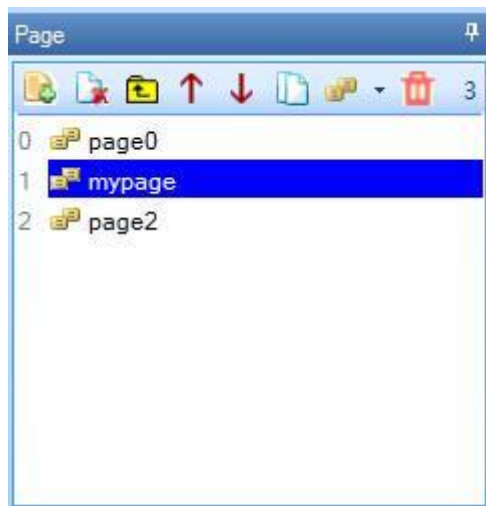


Every HMI project needs to have at least one Page. Pages are created or imported into your HMI project through the Page Pane. A Page is created with **Add**, deleted with **Delete**, and copied with **Copy**. **Insert** will create a new page before the highlighted page. Use the **Arrow Up** and **Arrow Down** to renumber the page number within the Page Pane. Using **Trash** will delete all pages within the project.

Pages can be renamed to a maximum of 14 characters and the page names are case sensitive. To rename your page highlight the page, right click, and select **Rename**. Then enter your new name (it is recommended to press Enter to ensure the change takes place). Double clicking a Highlighted page name will also trigger the page renaming function.

The page **Lock** and **Unlock** functions are only accessed by right clicking the highlighted page name and selecting Lock or Unlock. If the page has been locked with a password, the password must be entered to access the components and event code. There is no password recovery should the password becomes lost, so don't use or don't use. As an example, the keyboard pages are imported as locked, but do not use passwords (the keyboard pages are also a good coding example to review).

Pages can be exported from one project to another project with **Export page**. This is the



le, an end user can use even a locked page. When a page is imported with naming conflicts, the affected conflicting names will be renamed. It is therefore relevant to perhaps select meaningful names. To import a copy of one of the keyboard pages, one can either import the \*.page file directly or use the .key attribute of a Text or Number component.

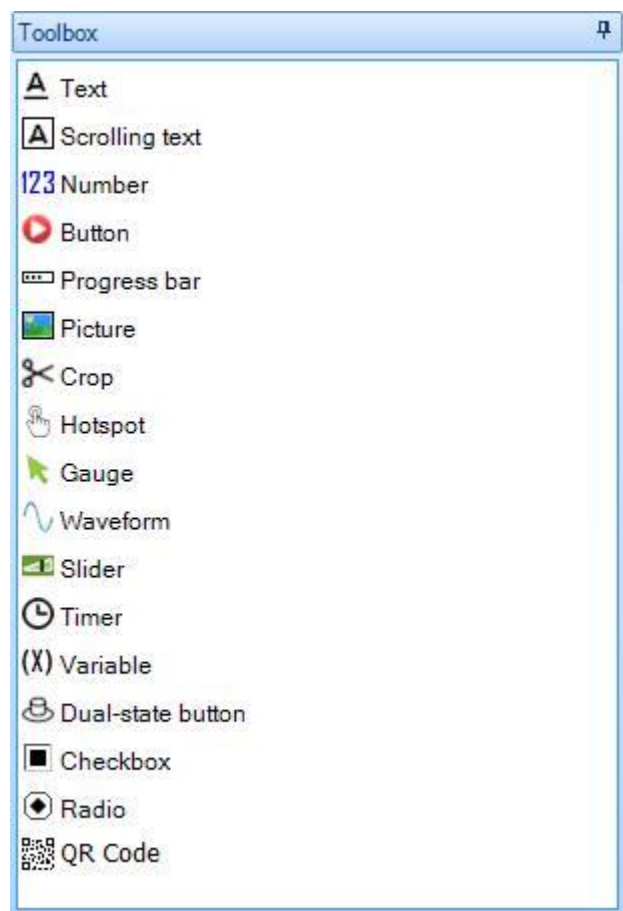
In an HMI project a page is a localized unit. When changing pages, the existing page is removed from memory and the requested page is then loaded into memory. As such components with a variable scope of local are only accessible while the page they are in is currently loaded. Components within a page that have a variable scope of global are accessible by prefixing the page name to the global component .objname.

As an Example: A global Number component n0 on page1 is accessed by page1.n0. A local Number component n0 on page1 can be accessed by page1.n0 or n0, but there is little sense to try access a local component if the page is not loaded. Only the component attributes of a global component are kept in memory. Event code is never global in nature.

A Page always contains a page component, and this page component will always have an .id of 0. Making the page component global, does not make the components within a Page global – just the very few page attributes of the page component (of which only .pic or .bco are changeable at runtime). The page component .id used with the b[id] component array is not the page index number used with the p[index] page array.

A Page component can have a background of either : Solid color, image background, or no background. An image background should use a ll screen image to avoid calling non-existent data. No background will show the current page components over top the last unloaded page, this must be used with caution. No background on a no background, on a no background soon causes unwanted side effects. There is a hard limit of 250 components allowed per page, and 254 pages per project.

**6. Components Pane** The *page* component not listed above is always created when the new Page is added to the Page pane. The page component will always have an .id of 0 and is always the bottommost layer. There is a hard limit of 250 components allowed per page.



Many components have multiple .style choices of – at (no lines will be added around edges) – border (lines will be drawn around edges) – 3D\_Down (lines will be drawn to yield impression of lowered) – 3D\_Up (lines will be drawn to yield impression of raised) – 3D\_Auto (lines will be drawn Up/Down according to state)

Many components have multiple text alignment and placement options – .xcen of Left, Center or Right – .ycen of Top, Center or Bottom

– .spay will add extra blank pixel spacing to bottom of each character – .isbr for multi-lined (set to true) or single line (set to false) – .pw for masking (Character is off, Password will mask with asterisk)

The **Text** component is a highly customizable component. The Text component has the .pw attribute for masking (Character is off, Password will mask with asterisk) and the .key attribute for integrating one of the included example keyboards (must be set to .vscope global before use).



The **Scrolling text** component combines an integrated timer component with a text component. The .pw option is not available with this component. The .key attribute allows for integrating one of the included example keyboards (must be set to .vscope global before use). There is a hard limit of 6 timer components per page within your project.

Компонент textScrolling [бегущая строка](#) объединяет интегрированный компонент таймера с текстовым компонентом. Опция .pw недоступна для этого компонента. Атрибут .key позволяет использовать клавиатуру (перед использованием должен быть установлен на .vscope global). Для вашего проекта есть таймер.

The **Number** component is used for signed 32-bit integer values. The .lenth (as spelled) sets the number of digits shown (useful for leading zeros). The new .format attribute allows for a choice of integer, currency (comma separated every three digits, not oas), or hexadecimal. Input should be in integer or hexadecimal. The .key attribute allows for integrating one of the included example keyboards (must be set to .vscope global before use).

The **Button** component is again highly customizable and integrates text in a momentary manner. Use images or event code to suit tastes.

The **Progress bar** component is for progress, thus a valid range of 0 to 100 to represent the percentage of progress. (Please no more requests to extend the range, even if many may give 110% effort). Best effects for progress are attained using images.

The **Picture** component will allow any picture resource to display in the Picture component. Example p0.pic=3. It is important that the picture resource is the user dened size in .w and .h or the picture resource will over draw the picture component boundaries, or incorrectly insert adjoining data. The Picture component is useful to represent multi-states and animation sequences.

The **Crop** component will replace its boundaries with the same location and boundaries from the picture resource pointed to with .picc. It is highly recommended that the picture resource being used is a full screen image to avoid errors (must be fullscreen image). The Crop component is useful to represent states.

**Компонент Hotspot** Невидимая кнопка. представляет собой пользовательское сенсорное пятно в область перекрытия. В области с 2 пикселями на 2 пикселя он позволяет использовать полезный держатель кода, который впоследствии вызывается командой щелчка, тем самым создавая функцию пользователя, расширенную пользователем. Как точка доступа, она превращает любую область изображения в кнопку, например, при создании [индивидуализированный](#) клавиатуры.

**Gauge Калибровочная** составляющая представляет собой полный круглый компонент со значением в градусах. Это означает jn 0 lj 360. Калибровочная составляющая всегда квадратная. Калибровочные датчики не достигаются с помощью калибровочного компонента. Компоненты калибровки не подходят для укладки (например, трехручные часы), так как перерисованный датчик перезапишет любой нижний датчик. Калибровочная составляющая всегда квадратная. Полукольцевые датчики на краю экрана не достигаются с помощью калибровочного компонента.

The **Waveform** component is used to plot y axis data points on up to 4 channels. Waveforms are never global. Up to 4 waveforms can be used on a single page. The Waveform component is limited to a y axis data range of 0 to 255 or 0 to waveform height -1. As a data point is added, it will consume one column, with the next data point using the next column. Recent changes now allow a variable to be used in the add command. Example add 1,0,h0.val. As the waveform data points are not global, changing pages away and back will revert the waveform to an emptied state. The addt command becomes useful to rell the waveforms on page load (such coding remains within the user domain).

The **Slider** component can be horizontal or vertical. The slider has the added event code for Touch Move, useful for providing updates to the sliders current position. Best results are attained with images. Slider length includes the size of the thumb as well as the range (often overlooked in calculations).

The **Timer** component is not expected to be a high precision interrupt driven component. It is however useful for queueing reoccurring event code after elapsed .tim has expired. As code is sequentially processed, it is very easy for the time to process the requested user event code to exceed the .tim intervals and therefore not interrupt driven (to avoid such stack overflows) and not high precision. There is a hard limit on the maximum number of timers running in a single page, this limit is 6. Beware that the scrolling text component integrates 1 timer. Timer attributes can have a variable scope of global, event code is never global. As such timer code can only be triggered within the current page they are designed in. As the timer is a non visual component, they are added below the Design Canvas.



The **Variable** component is a non visual component and also added below the Design Canvas. Variables are either 32 bit signed numeric or string content.

The **Dual-state button** component is an expanded Button maintaining its state between toggles.

The **Checkbox** component is another example of a lightweight dual-state component with less customization and lower memory usage. The **Radio** component is yet another example of a lightweight dual-state component with little customization and lower memory usage. Obtaining grouping is achieved via user code (remains in the user domain).

The **QR Code** component is used to generate a 2D scan able QR. It is limited to a byte maximum for the .txt attribute of 84 on Basic T models and 192 on the Enhanced K models.

To add any of the above components to the currently design page, simply click on the component and it will be added with its .id set to the number of components on the page. All components within a page are listed in .id ascending order in the Component Drop down in the Attributes Pane. Then continue with placement and adjustment of .attributes as desired.

**7. Design Canvas Visual Components** This is the main design space for the visual/touch components for the current Page. The Page's page component is always .id 0 and always the most back layer. The mouse coordinates are displayed on the Status Bar aiding in precision placement. Selected components can be moved in one pixel offsets using the keyboard arrow keys. For best precision, use the component's .x and .y attributes. Components selected with the left mouse click can be moved by dragging. As such using the right mouse click to select a component will not accidentally move the component. Resizing a component via edge dragging can only be achieved on the bottom and right edges. There is a limit of 250 components (visual and non visual) allowed per page.

**8. Design Non Visual Components** A Page's non visual components (Variables and Timers) will be listed in this area. This area is not displayed if there are no Variable or Timer components used in the page. There is a limit of 250 components (visual and non visual) allowed per page.

## 9. Attributes Pane

Attribute

n0(Number)

id	4
objname	n0
type	54
vscope	local
sta	solid color
style	flat
key	None
bco	<input type="text" value="65535"/>
pco	<input type="text" value="0"/>
font	0
xcen	Center
ycen	Center
val	0
lenth	0
format	Hex
isbr	False
spax	0
spay	0
x	19
y	11
w	100
h	30

Initial value(-2147483648 to 2147483647)

COMPONENT DROP DOWN

NO CHANGES AT RUN TIME

CAN CHANGE AT RUN TIME

NO CHANGES AT RUN TIME

DESCRIPTIVE ATTRIBUTE OPTIONS

The Attribute Pane contains the list of components included within the current design page in the Component drop down. Clicking on a component, or selecting it from the drop down will display the component's available attributes. The left side contains the attribute name, the right side contains the attributes current value. Clicking on an attribute will display the attributes meaning and valid range/options at the bottom of the Attribute Pane. Double clicking a eld with bring up resource editor for the attribute if attribute has such (ie: .pco opens color picker, .pic opens picture chooser).

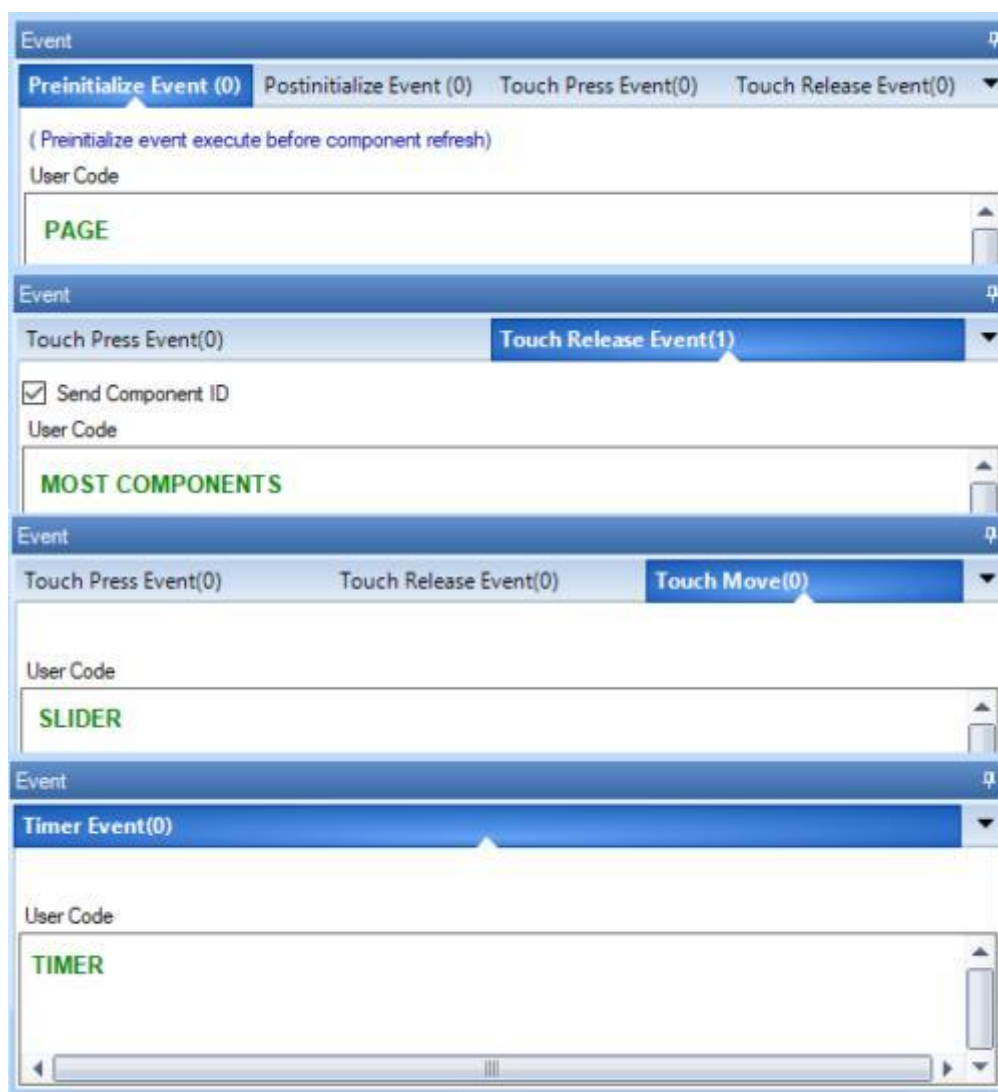
Any attribute in black is read only at runtime (with the exception of .objname and .vscope) The .objname is inaccessible, and the .vscope does not report correctly. Any attribute in green can be both read and changed by user code at runtime. Empty unassigned attributes values or invalid attribute values will need to be resolved before a successful compile can be achieved. Page name prexing is suggested to access a global component's attributes on another page. – example: page0.va0.val Page name prexing is not required to access a local component's attribute on the current page – example: va0.val

Attributes that have ranges are evaluated in full during Nextion's parsing of a complex expression and as such care is required. Nextion is stated as simplex expression, although these rules are often bent. Use care. – example: gauge z0 with a .val range of 0 to 360. z0.val=va0.val+step.val%360 Should va0.val+step.val exceed 360 the assignment fails before arriving at modulo 360. – this is the nature of simplex expressions (think assembly language), and bending the rules has side effects.

The various combinations of attribute choices provides a wide range of expected behaviours with too many combinations to cover in any manual(s). This combined with the Nextion Instruction Set creates the opportunity for very powerful HMIs.

There is a hard limit for a combined tally of attributes and user code of 65534.

#### 10. User Event Code Код События Пользователя



Код события пользователя может содержать любую допустимую инструкцию Nextion. Этот раздел не будет учить программированию, но быстро даст обзор различных типов событий, где доступна вставка пользовательского кода. Код события всегда является локальным для страницы и никогда не глобальным.

Almost every component has the **Touch Press** and **Touch Release** events.

– The Touch Press Event включает в себя флажок **Send Component ID** который при проверке отправляет данные возврата 0x65 по последовательному при физическом нажатии. Код пользователя запускается при физическом нажатии или с помощью команды click.

. Событие Touch Release включает в себя флажок **Send Component ID**, который при проверке отправляет данные возврата 0x65 по последовательному на физическом выпуске. Код пользователя запускается при физическом нажатии или с помощью команды click.

– The **Send Component ID** Команда click не может инициировать возврат данных по последовательному действию 0x65. Это зарезервировано для физического действия пользователя, полученного через сенсорный датчик. Команда click запускает только код события пользователя.

– The printh может имитировать возвращаемые данные 0x65 (фактические или поддельные)

The page component contains both Touch Press and Touch Release а ТАКЖЕ

– The **Preinitialize Event** user code is run before the loading of the HMI designed page.

– The **Postinititalize Event** user code is run after the loading of the HMI designed page.

Note глобальные значения будут сохраняться, локальные значения вернутся в их проектное состояние.

The slider component contains both Touch Press and Touch Release as well as –

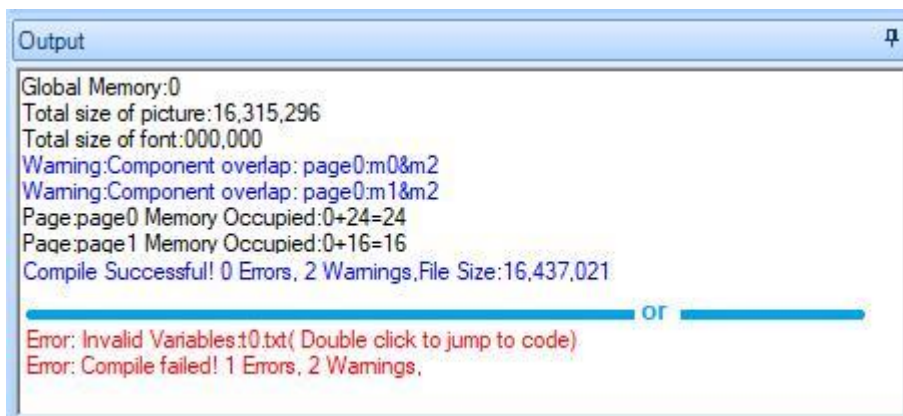
The **Touch Move Event** user code is run во время перетаскивания большого пальца, когда ползунок изменяет значений.

The timer component only contains the **Timer Event** for user code. – **Дополнительные сведения** Refer to the Timer component in the Component Pane section for more details

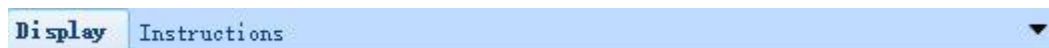
Nextion Return Data is returned at the end of command execution выполнения – иначе было бы неразумно предсказывать исход до конца.

There is a hard limit for a combined tally комбинированного подсчета of attributes and user code of 65534.

## 11. Display/Instruction Tabs



Selecting the **Instruction**Tab will load the Nextion Instruction Set in the Nextion Editor's built in



## 12. Output

The Output Pane contains details on the build process when Compile/Debug/Upload is selected. Compile needing to occur rst, the user HMI is assembled into a usable TFT le for the selected Nextion Model. The rst three lines of the output will list the total amount of global SRAM memory consumed by the HMI project, and then statistics for the total amount of Flash space the picture resources consumes, followed by the total amount of Flash space the ZI Font resources consumes. The build process then goes through the project sequentially page by page. At the end of a successful page build, the page Memory stats are listed Global+Local=Total. Should a page not build successfully, the offending page is the last listed+1.

Warnings listed in blue (such as when using the not recommended layering techniques, it will compile, but warn of potential unexpected behaviours), Errors listed in red (this will not compile, and the build process halts). Note: Do not upload a zero byte file.

File Size must be small enough to fit in your Model's Flash size. See the Status Bar or your Datasheet for your model's Flash size. All pages' Total Memory usage must be small enough to

fit your model's HMI allotted SRAM. See the Status Bar or your Datasheet for your model's HMI allotted SRAM.

Due to the nature of ash, it is possible that a compiled lesize may be under the MB size (ie: 1677216 bytes) and still be shy of the available and usable Flash on the Nextion Device. Nextion may report on upload the File is too big – this is not a hardware error but the working nature of Flash. Some allowance for unusable ash pages have to made, and even more over time.

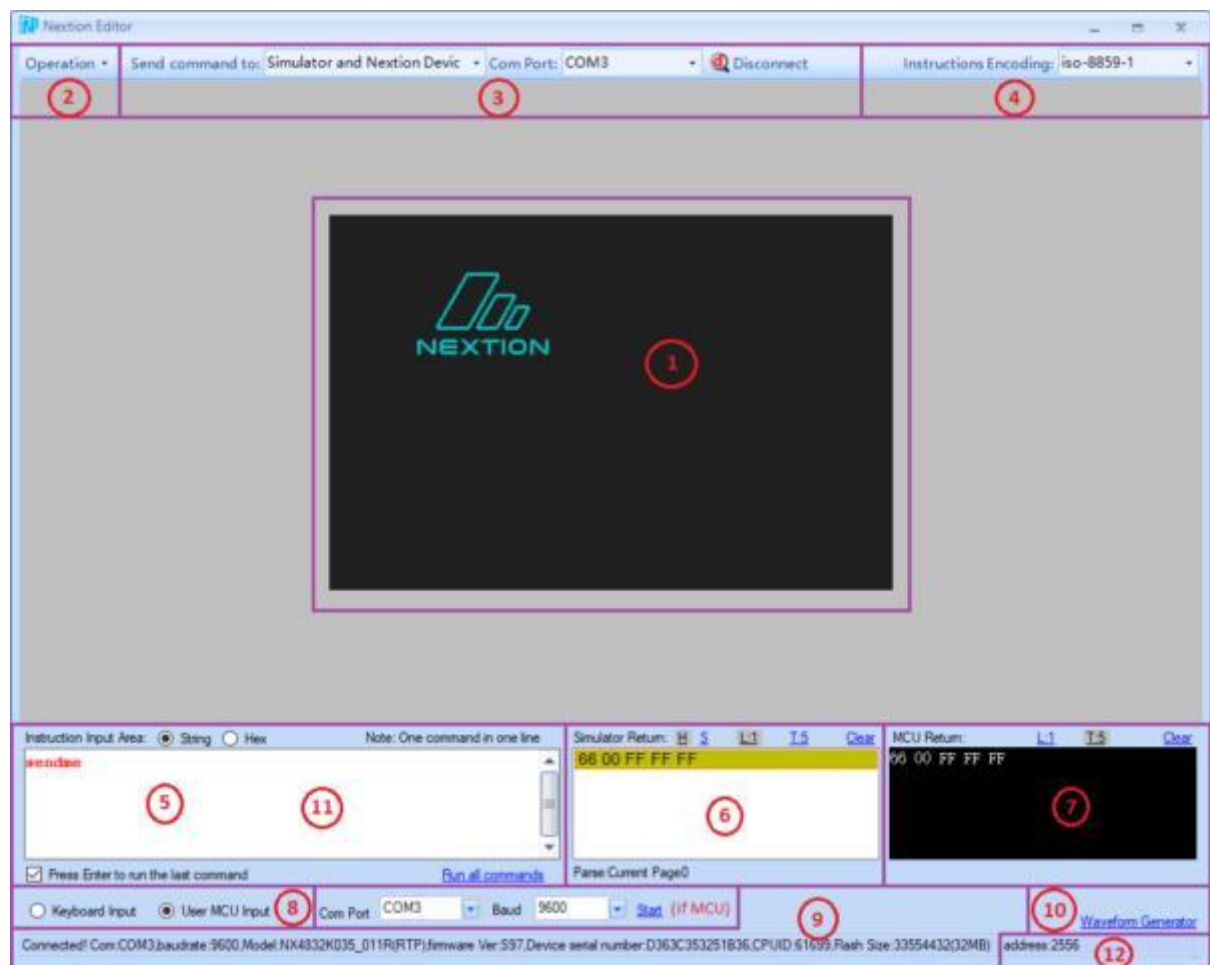
Encoding:iso-8859-1 | Model:NX4832T035\_011 | inch:3.5(320X480) Flash:16M RAM:3584B Frequency:48M Coordinate X:604 Y:599

### 13. Status Bar

The Status Bar at the bottom contains three segments. The character encoding, the quick details of the Nextion Model selected, and the Design Canvas mouse coordinates. Clicking on **Encoding** will launch the DISPLAY tab of Device Settings. The Nextion Model details provide quick design specifications – limits of Project SRAM and Flash is useful to be mindful of.

## Debug / Simulator Overview

1. Simulated Nextion ... <goto> 2. Operations Menu ... <goto> 3. Command Redirection ... <goto> 4. Input Encoding ... <goto> 5. Instruction Input ... <goto>



### 1. Simulated Nextion

This is a simulation – this is not an emulation. Purpose of the simulation is to aid in debugging commands Timers will not be precision under any Windows OS RTC is simulated, GPIO is not simulated EEPROM is simulated with le located Nextion Editor/EEPROM.bin Note: EEPROM adheres to 4 byte boundaries when writing Users may write to any address, but must care for data until boundary.

## **2. Operations Menu**

Upload, Synchronize RTC time and Reboot device.

## **3. Command Redirection**

Simulator only, Nextion device only, or both. Only one (Nextion device or User MCU) can be connected at a time When one is selected, the other is not available.

## **4. Input Encoding**

Character Encoding to use for input

## **5. Instruction Input**

text command input of any non-block Instructions (excludes if, for and while) Toggle hex mode to ensure byte precision in input. (Answers the question of how to input less frequently used characters)

## **6. Simulator Return**

Any Return Data or print commands from Simulator will display here Expected to be sufficient to aid in the debugging process Not expected to be full trace logs (Simulator – not emulator) Options added in v0.53 of the Nextion Editor View in Hex or String format View Line by Line (ÿÿÿ terminations), or by Total bytes in Hex Context Menu (right click) to: Copy HEX, Copy STR, Copy HEX+STR

## **7. User MCU Return**

Any Return Data or print commands from Nextion/MCU will display here Expected to be sufficient to aid in the debugging process Not expected to be full trace logs (Simulator – not emulator) Options added in v0.53 of the Nextion Editor View in Hex or String format View Line by Line (ÿÿÿ terminations), or by Total bytes in Hex Context Menu (right click) to: Copy HEX, Copy STR, Copy HEX+STR

## **8. Input Selection**

Keyboard input is the default simulator input method. Toggle to User MCU input to have user MCU interact with Simulator Select Com Port of user MCU, MCU operating baud rate and then Start Only one (Nextion device or User MCU) can be connected at a time



## **9. Connected Nextion Identification**

When Nextion is connected to Simulator, portions of connect string is shown here.

## **10. Waveform Generator**

Click here to show the Waveform Generator

## **11. Waveform Generator Options**

Sinewave or random waveform input at selected intervals. Expected to be sufficient to aid in the debugging process Not expected to be an oscilloscope (Simulator – not emulator)

## **12. Nextion Address**

Default is 0 – Nextion normal mode, 256 to 2815 in address mode. Address will only apply to specialized advanced applications.

Contact Us

FAQs

## RESOURCES

NEXTION Editor

Editor Guide

Instruction Set

Datasheets

## CONTACT INFO

**Address:** 8F, Building No.9,

Banri wowi, Tongsha Road No.32

, Nanshan Dist., Shenzhen, GD,

China

**Postal Code:** 518000 China

**Telephone:** +86-755-27955416

© Copyright 2011 - 2018 ITEAD STUDIO All Right Reserved 粤公网安备 44030502000000号