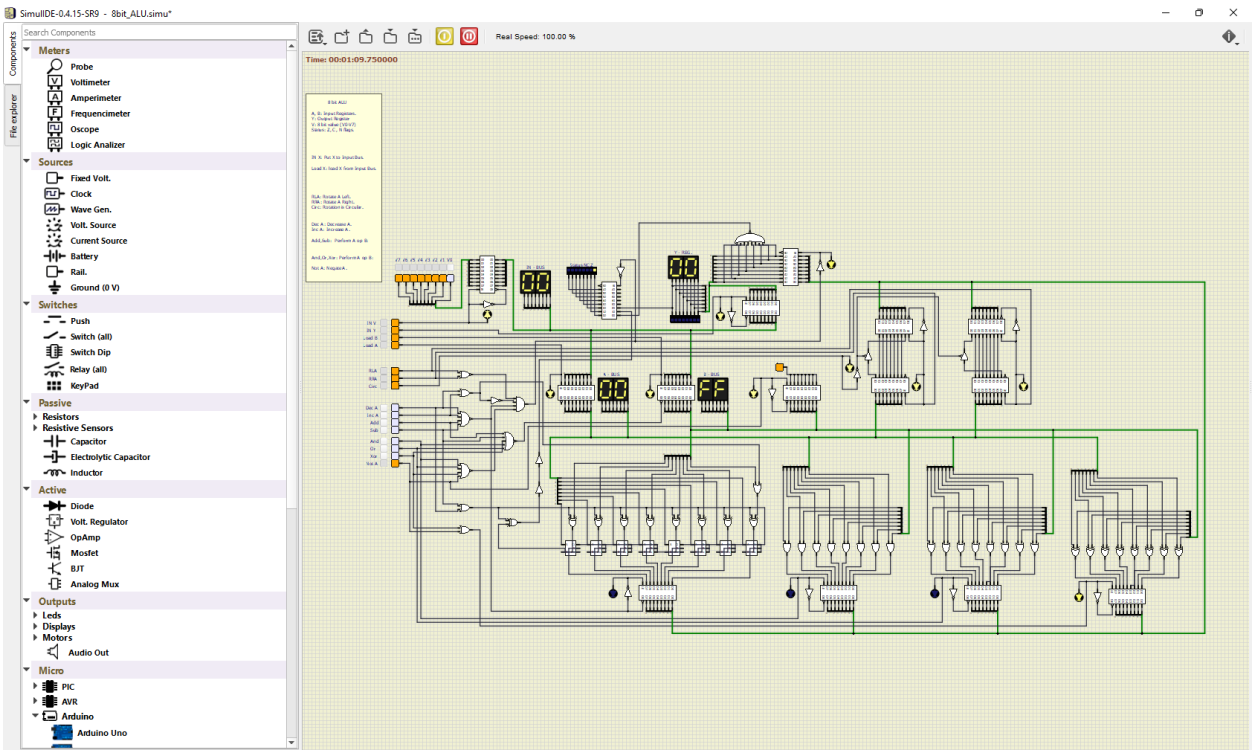
SIMULIDE

Руководство пользователя по мотивам сайта с версией 0.4.15-SR9

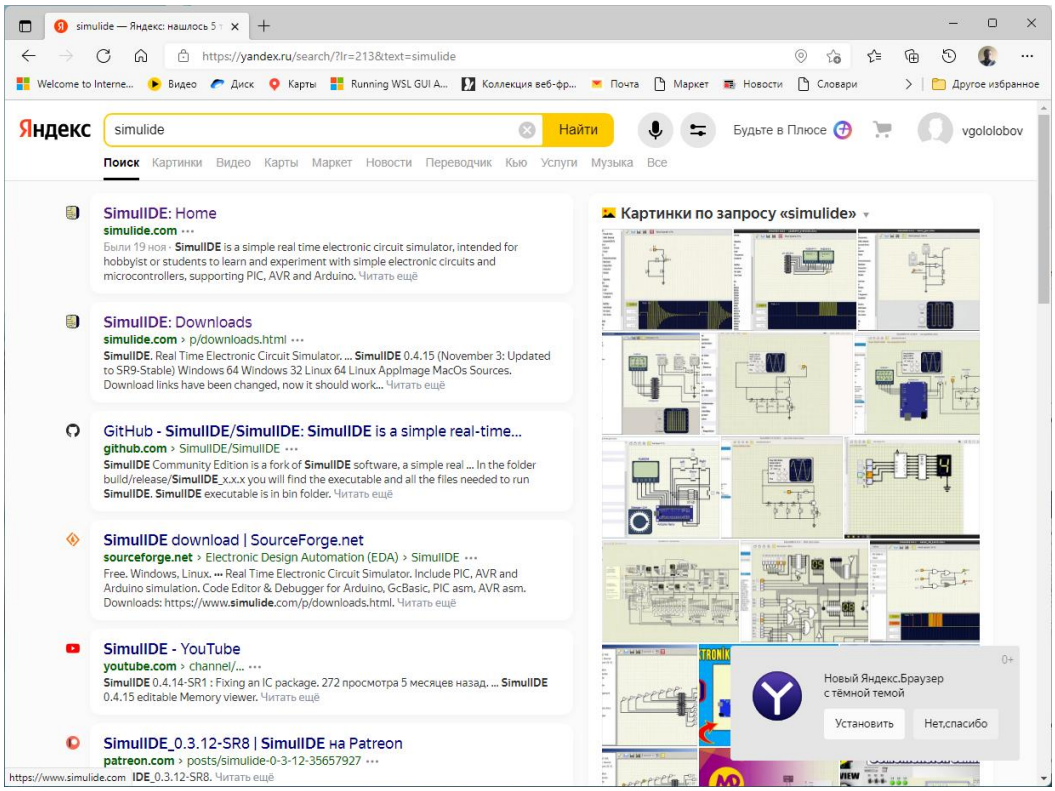


SimulIDE - это простой многоплатформенный симулятор электронных схем в реальном времени, предназначенный для любителей или студентов, при изучении и экспериментах с простыми электронными схемами и микроконтроллерами: поддерживаются PIC, AVR и Arduino.

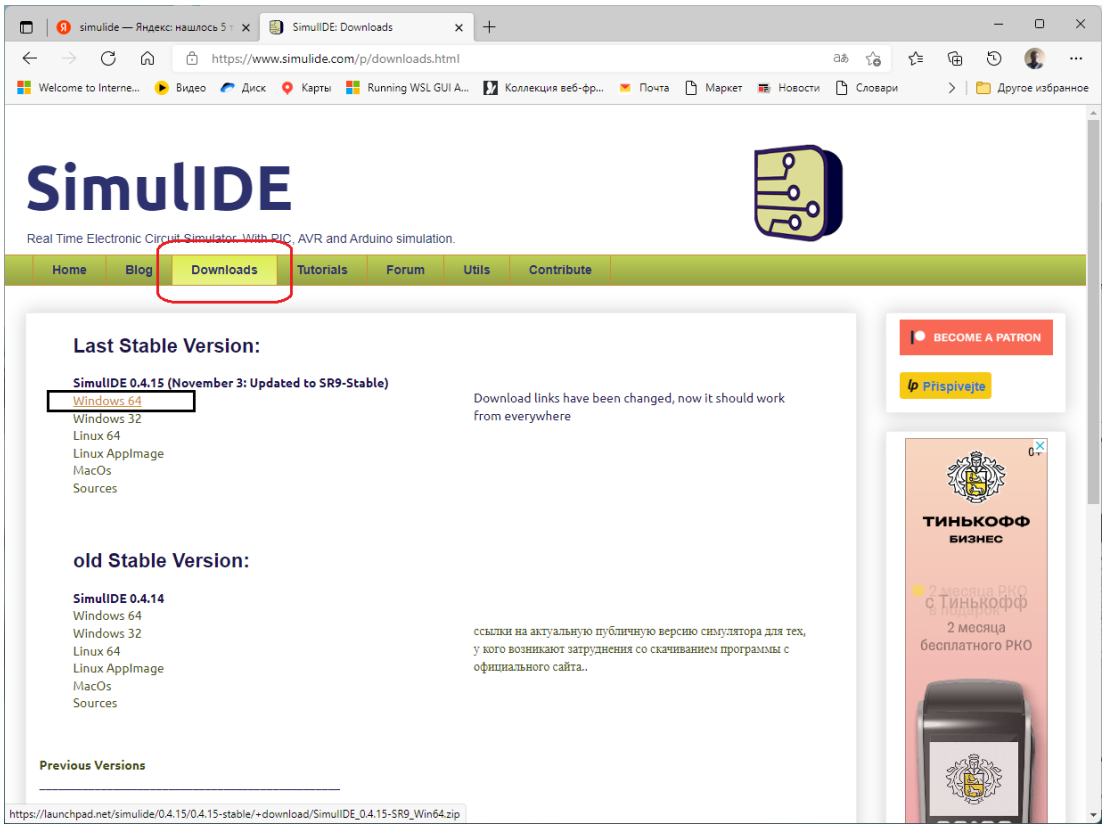
Оглавление

Загружаем программу

Чтобы загрузить программу, достаточно ввести в поисковую строку её название.



Как видно на рисунке, есть домашняя страница: SimulIDE: Home. На ней и следует искать программу.



У меня на компьютере сейчас операционная система Windows 11 64-битовой версии. Если у вас другая ОС, то можно выбрать из доступных вариантов загрузки, представленных на странице.

Программа загружается без особенностей, полученный архивированный файл в формате .zip легко разархивируется.

После распаковки можно папку с программой перенести в корневой раздел диска C, что не обязательно, но достаточно удобно. При первом запуске программы моя операционная система выразила неудовольствие тем, что не знает издателя. Но есть кнопка (ссылка) «Подробнее», где появляется возможность запустить программу вопреки устоявшейся традиции требовать от всего и всех сертификаты.

Базовые сведения

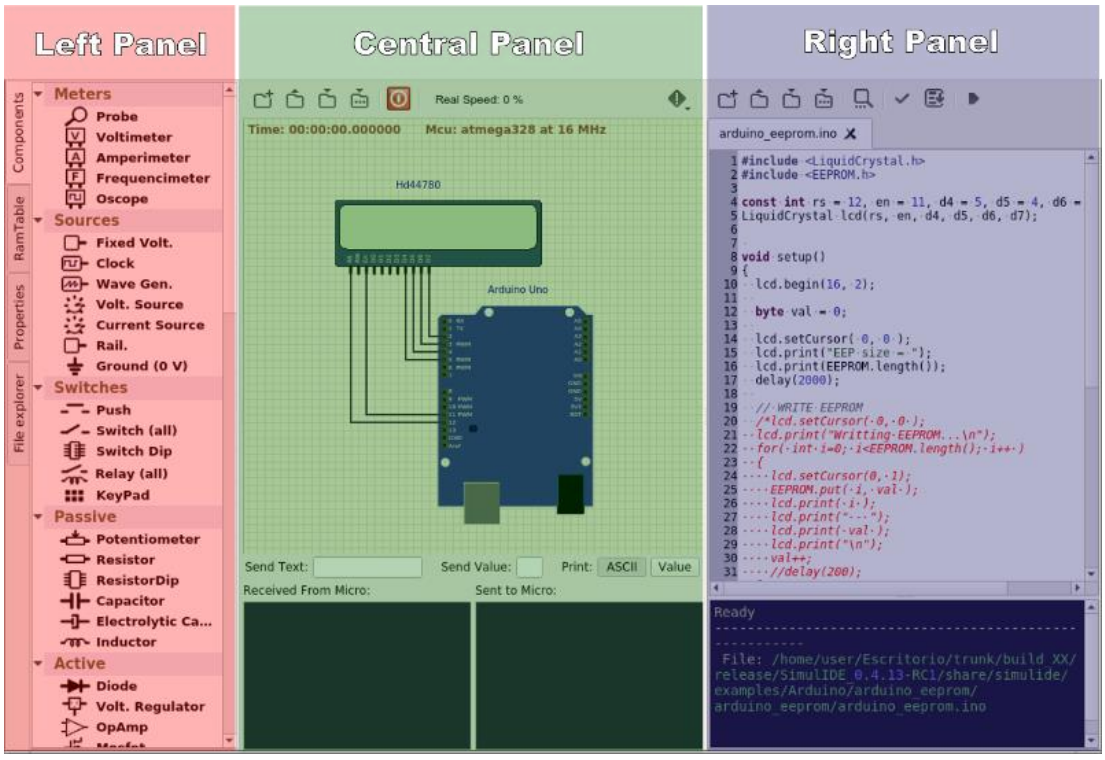
Для запуска приложения после загрузки версии, которая совместима с вашей операционной системой: распакуйте файл, который вы загрузили (.zip или .tar.gz). Папка «SimulIDE_x.x.x» содержит всё необходимое для запуска программы.

Вы можете скопировать эту папку туда, где вам удобно держать всё как есть. **Не удаляйте файлы**, пока вам не ясно, что именно вы делаете.

При использовании Windows или Linux AppImage (загрузка из готовых пакетов дистрибутива) вам больше ничего не надо устанавливать. Для использования скачанной программы в Linux от вас потребуется установить ряд пакетов (по зависимостям). (Эти пакеты вы можете посмотреть на оригинальной странице руководства к программе).

Исполняемый файл находится в папке «bin», достаточно двойного щелчка по нему для запуска. Но можно запустить программу из терминала, если вы хотите видеть некоторые сообщения о выполнении SimulIDE. Это полезно, если приложение не запускается или не работает как следует, запуск из терминала покажет ряд полезных сообщений.

Графический пользовательский интерфейс разделён на три основные части:



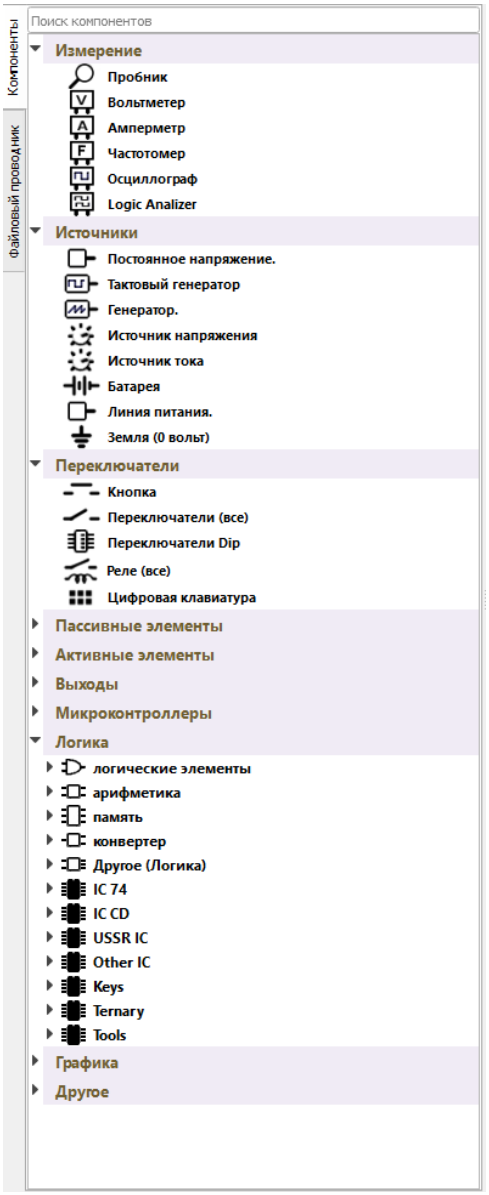
- **Left Panel** имеет закладки: Компоненты, Таблица памяти, Свойства и файловый проводник (в последней версии есть не все закладки). Ниже мы рассмотрим описание всех панелей.
- **Center Panel** для схемы и инструментов (частотомер, осциллограф, пробник и т.д.).

- **Right Panel** с редактором/компилятором/отладчиком. Этот текстовый редактор с базовой кодировочной и отладочной функциональностью.

Вы можете изменить размер каждой панели или убрать её для удобства в работе.

Левая панель

Начиная с версии 0.4.14, левая панель не имеет закладок свойств и таблицы памяти. Эти виджеты доступны из контекстного меню через щелчок правой клавишей мышки по каждому из компонентов.



Закладка «Компоненты»:

Это список доступных компонентов для добавления в схему.

Они организованы по категориям и подкатегориям, которые открываются и закрываются с помощью маленьких треугольных стрелок левее названия категории.

Открывание/закрывание списка сохраняется и возобновляется при следующей сессии, так что вид панели будет тем же в следующий раз.

Можно скрыть отдельные категории или компоненты в списке. Так вы можете создать простой список компонентов, если используете только некоторые из них. Например, вы можете создать список только для логических схем.

Чтобы сделать это щёлкните правой клавишей мышки по любой части списка и выберите «Управление компонентами».

Откроется виджет, где вы можете выбрать, какие категории/компоненты будут видимы.

Эти изменения сохраняются автоматически, чтобы в следующий раз вы видели знакомую картину.

Примечание: в оригинале на сайте вид панели сопровождается видео. Посмотрите!

Центральная панель

Жёлтое пространство центральной панели – это холст схемы. То есть, область, где создаётся и симулируется ваша схема.

Создание схем…

Компоненты:

- Перетащите компоненты с левой панели (закладка «Компоненты») и оставьте их на «холсте» схемы.
- Щёлкните левой клавишей мышки по компонентам, чтобы переместить их.
- Щелкните правой клавишей мышки по компонентам, чтобы вызвать контекстное меню.

Соединения:

- Подведите курсор к выводу компонента или другому проводнику для начала и завершения соединения. Когда курсор превратится в перекрестие, щёлкните левой клавишей мышки.
- Проведите соединение к другому выводу. Когда на выводе появится перекрестие, щёлкните левой клавишей мышки, завершая соединение.

Схема:

- Когда курсор находится в центре схемы, вращение колёсика мышки увеличивает или уменьшает вид.
- Когда курсор находится слева от схемы, вращение колёсика мышки одновременно с изменением вида сдвигает схему вправо.
- Когда курсор находится справа от схемы, вращение колёсика мышки одновременно с изменение вида сдвигает схему влево.
- Щелчок правой клавиши мышки вызывает контекстное меню.
- Кнопка включения запускает/останавливает моделирование.

Контекстное меню:

В контекстном меню схемы вы найдёте ряд полезных функций.

- **Вставить**: вставляет ранее скопированный объект.
- **Отмена действия**: отменяет последнее изменение.
- **Выполнить повторно**: возвращает последнюю отмену.
- **Импортировать схему:** открывает диалоговое окно для выбора .simu файла (файл схемы). Эта схема будет импортирована в текущую схему.
- **Сохранить схему как изображение**: открывает диалоговое окно для сохранения текущего файла в виде изображения (jpg, png, bmp или svg формата).
- **Спецификация материалов**: создаёт текстовый файл, содержащий спецификацию текущей схемы.

Свойства схемы:

Вы можете редактировать некоторые свойства, относящиеся к появлению и поведению схемы. Щёлкните правой клавишей мышки на свободном месте схемы; появится меню, в котором можно выбрать раздел «Свойства». Щелчок левой клавиши мышки по нему вызывает появление панели свойств схемы.

- **Speed per** (скорость в…), **Speed sps** (шагов в секунду): задают скорость работы основного цикла. Так скорость в шагах в секунду допускает Default Value: 1.000.000, Max Value:

1.000.000, Min Value: 1. При изменении этого значения симуляция будет проходить быстрее или медленнее. Например, при значении по умолчанию вы будете видеть изменения каждую секунду; при значении 500.000 это произойдёт каждые 2 секунды. Полезная настройка при медленных процессах.

- **Simu Step nS**: шаг симуляции в наносекундах.
- **Шаг отклика**: количество шагов основного цикла для запуска реактивного подцикла. Default Value: 50, Max Value: 100, Min Value: 1. Уменьшение этого значения приводит к большей точности при симуляции реактивных компонентов и большей загрузке процессора. Значение лучше изменить для малых значений ёмкости и индуктивности. Например, для 50 кГц осциллятора, использующего конденсатор, вы должны задать это значение, по крайней мере, ниже 10; запуск каждые 10 шагов, что равно 100 кГц.
- **Накопление NoLin**: точность нелинейных компонентов. Базовое значение допускает наименьшие ошибки. NoLinAcc = 5 означает ошибку в +- 1*10e-5. Увеличение этого значения приводит к большей точности, но замедляет моделирование. Default Value: 5, Max Value: 14, Min Value: 3.
- **Показать сетку**: показывает или скрывает сетку. С сеткой скорость моделирования может несколько улучшаться.
- **Показать полосы прокрутки**: показывает или скрывает полосы прокрутки.
- **Анимация**: расцвечивает соединения, отображая цифровое состояние (для цифровых схем). Красный цвет – высокое состояние, синий – низкое состояние. Может замедлять моделирование.
- **Масштаб шрифта**: масштабирует размер шрифта интерфейса, например 1,5 = 150%.

Правая панель

Редактор-Компилятор

Начиная с версии 0.1.5 существует встроенный текстовый редактор с некоторыми возможностями кодирования. Он включает в себя выделение кода, компиляцию, загрузку в микроконтроллер, присутствующий в схеме, и базовую отладку для некоторых компиляторов.

- Arduino:
Файлы должны иметь расширение .ino. Среда Arduino должна быть установлена в вашей системе. Вы должны указать путь к компилятору Arduino. Щелкните правой клавишей мышки на закладке открытого документа (файл .ino). По умолчанию Arduino будет компилироваться для платы «Uno». Вы можете изменить плату в свойствах: щелкните правой клавишей мышки внутри документа .ino и откройте виджет свойств. Выберите доступную плату (Плата->правое окошко->стрелка вниз) или выберите «Custom» и задайте свойство «Заказная плата».

- **Компилятор GcBasic:**
Файлы должны иметь расширение .gcb. Компилятор GcBasic должен быть установлен в вашей системе. Необходимо задать путь к компилятору GcBasic щелчком правой клавишей мышки на закладке документа.

- Компилятор Avra Avr asm:
Файлы должны иметь расширение .asm. Для систем Linux в вашей системе должен быть установлен компилятор avra. Он доступен в большинстве репозиториев дистрибутивов Linux. Для систем Windows компилятор avra поставляется с SimulIDE.

- **Компилятор Gpasm Pic asm:**
Файлы должны иметь расширение .asm. Для систем Linux в вашей системе должен быть установлен компилятор gpasm. Он доступен в большинстве репозиториев дистрибутивов Linux. Для систем Windows компилятор gpasm поставляется с SimulIDE.

SimulIDE может автоматически определять, является ли файл asm для Pic или Avr микроконтроллеров. Некоторые свойства можно редактировать в виджете свойств (щелкните правой клавишей мышки по разделу «Свойства»).

- **Размер шрифта:** устанавливает размер шрифта (также работает ctrl+ и ctrl-), по умолчанию 9.
- **Шаг табуляции:** устанавливает размер табуляции, по умолчанию 4.
- **Табуляция пробела:** если true, табуляции будут пробелами, если false, то они будут табуляциями (по умолчанию false).
- **Показать пробелы:** если true, вы будете различать пустое от фактических пробелов или табуляций. Вы увидите крошечные точки, представляющие пробелы,
и крошечные стрелки, представляющие табуляцию (по умолчанию false).

Вот что вы видите с «Показать пробелы» = false:

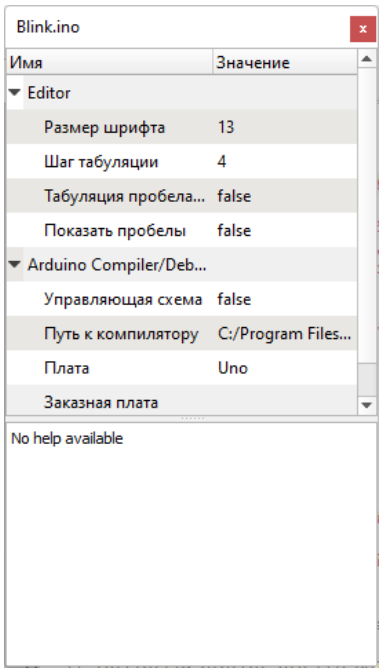
```
26
27
28  main:
29
30    count = ReadAD(AN0)
31    'Limit CW travel
32    if count < 75 then
33        count = 75
34    end if
35    'Limit CCW travel
36    if count > 225 then
37       count = 225
38    end if
39    pulseout PORTB.0 ,count 10us
40    wait 20 ms
41
42  goto main
43
```

А это с «Показать пробелы» = true: точки — это пробелы, а стрелки — табуляция.

```
26
27
28 ··main:
29
30 ····count·=·ReadAD(AN0)
31 ····'Limit·CW·travel
32 ····if·count·<·75·then
33 ········count·=·75→  →   →   →   →
34 ····end·if
35 ····'Limit·CCW·travel
36 ····if·count·>·225·then
37 ·····count·=·225→  →   →   →   →
38 ····end·if
39 ····pulseout·PORTB.0·,count·10us
40 ····wait·20·ms
41
42 ··goto·main
43
```

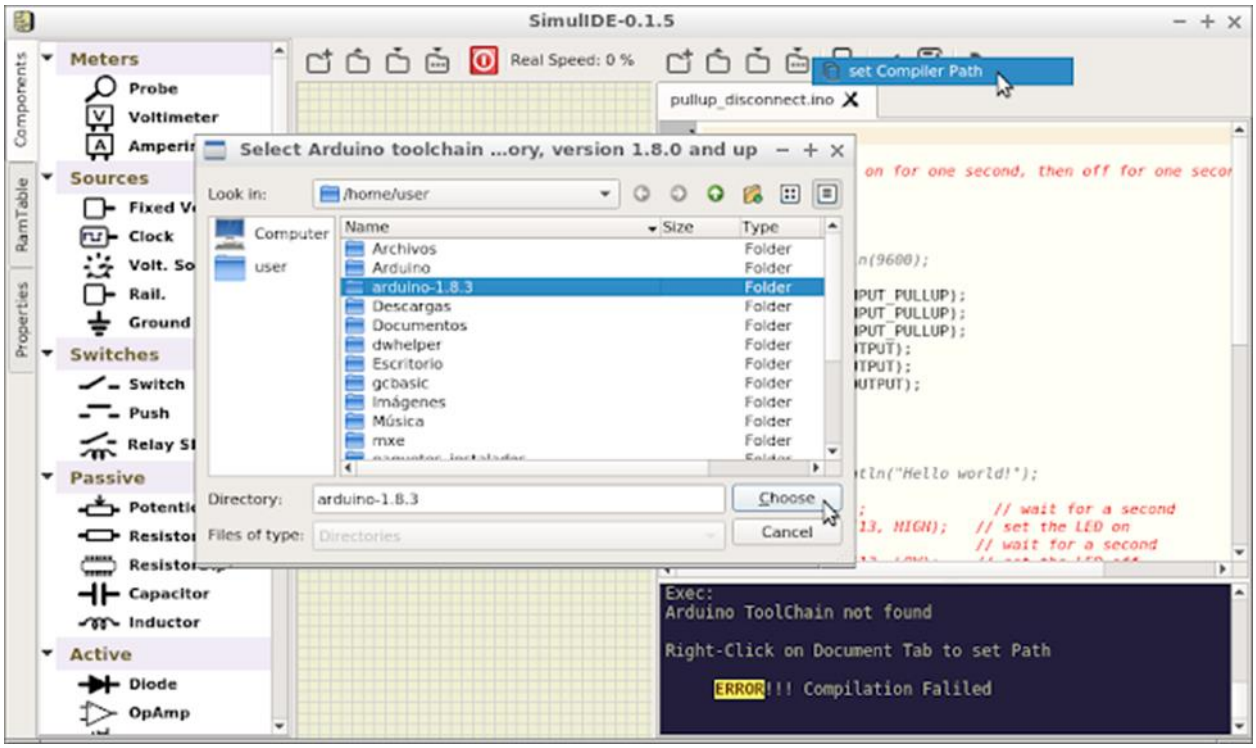
Обратите внимание, что есть некоторые табуляции, о которых вы, вероятно, не знали, что они там были.

Если для текущего типа файлов доступен компилятор, вы увидите дополнительное свойство, чтобы увидеть или задать путь к компилятору вручную.



Вы также можете задать путь к компилятору, выбрав папку в диалоговом окне файла. Чтобы установить Arduino или GcBasic Path, сначала откройте файл .ino или .gcb, затем щелкните правой клавишей мышки на закладке документа и выберите «Путь к компилятору».

Появится диалоговое окно файла, необходимо выбрать корневую папку arduino или установку GcBasic. Не переходите в папку, просто нажмите один раз на нее, чтобы выбрать, затем нажмите «Выбрать, Choose»; теперь вы можете работать с компиляторами.



Элементы управления слева направо:

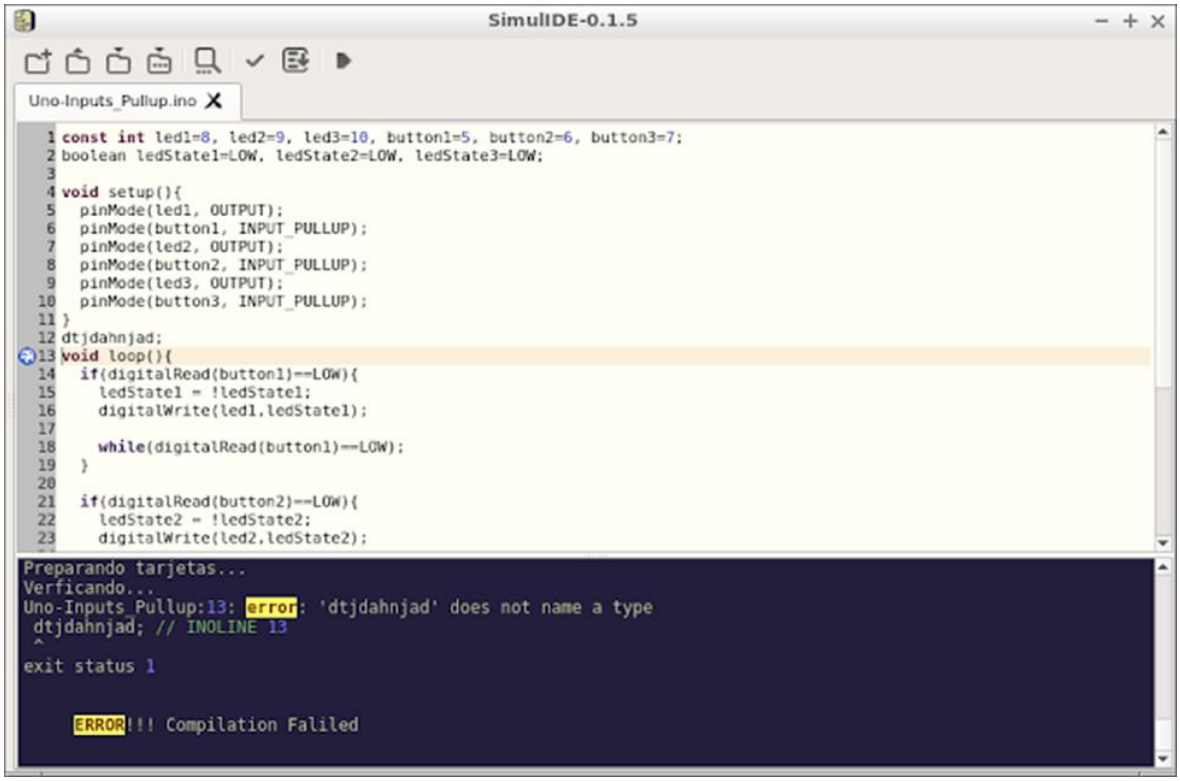
- **Последние файлы**: открывается список последних файлов.
- **Новый файл:** создает новый пустой файл.
- **Открыть файл:** появляется диалоговое окно проводника, чтобы открыть файл.
- **Сохранить файл:** сохранить текущий файл.
- **Сохранить файл как:** появляется диалоговое окно проводника для сохранения текущего файла.
- **Найти и заменить:** появляется диалоговое окно для поиска или замены строк в текущем файле.
- **Компилировать**: если тип файла поддерживается, он будет скомпилирован. Вы увидите выходные данные компилятора в нижней панели.
- **Загрузить:** загружает прошивку, соответствующую текущему файлу, в MCU, присутствующий на схеме.
- **Отладка:** запуск сеанса отладки.

Таким образом, вы можете открыть файл или создать новый файл и написать свою программу.

Компилятор читает файл с вашего диска, поэтому при нажатии кнопки компиляции текущий файл будет сохранен на диск, а затем скомпилирован.

Обратите внимание, что сохраняется только основной файл, если у вас есть код в других включенных файлах, вы должны сохранить их перед компиляцией.
При компиляции все выходные данные процесса компиляции будут показаны в нижней панели.

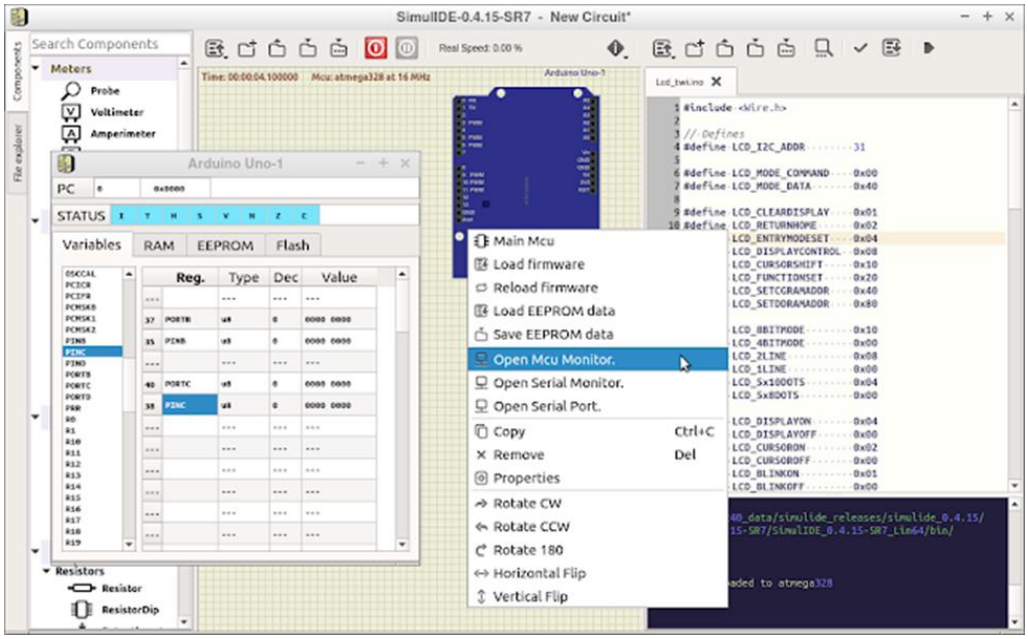
Если компиляция не удалась, появится сообщение ERROR и знак стрелки укажет на строку, в которой была обнаружена ошибка:



Как только программа скомпилируется, вы можете загрузить ее в MCU, который есть в схеме, затем запитать схему (кнопка включения) и моделировать ее обычным образом.

Вы можете наблюдать за регистрами Mcu и переменными программы в виджете «Монитор Mcu» (щелкните правой клавишей мышки по микроконтроллеру, выберите раздел «Open Mcu Monitor»).

Если регистр или переменная найдена, вы увидите, что десятичный адрес есть в самом левом столбце, чуть левее имени. Если имя не найдено, адрес будет читаться как 0, и вы не увидите никакого значения.

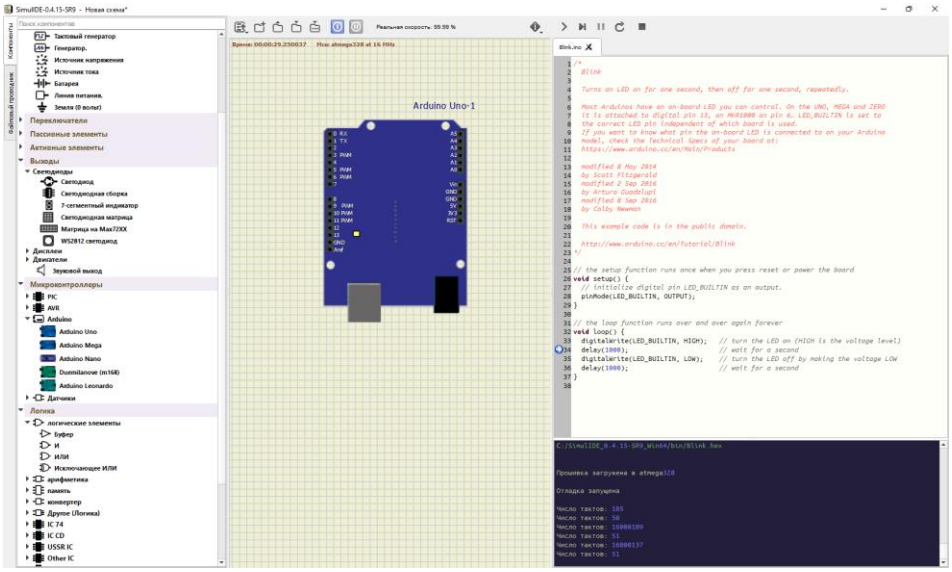


Редактор-Отладчик

Для запуска режима отладки нужно предварительно убедиться, что исходный файл уже скомпилирован и проверен, что он не содержит ошибок. В схеме должен быть микроконтроллер, соответствующий целевому в исходном тексте программы.

Затем вы можете нажать кнопку «Отладка» и, если все будет правильно, будет запущен режим отладки.

Сообщение в нижней панели подтвердит, правильно ли запущен режим отладки, а «значок стрелки» укажет на первую строку кода: кнопка питания изменится на синий, указывая, что отладчик взял под контроль моделирование схемы.



По умолчанию моделирование схемы будет выполняться с нормальной скоростью, в то время как выполнение MCU будет контролироваться шагами отладчика.
Начиная с 0.2.9 существует возможность «синхронизации» моделирования схем с выполнением отладчика.

Новое в последующих версиях

Начиная с версии 0.2.9 существует возможность «синхронизации» моделирования схем с выполнением отладчика. Щелкните документ редактора, который вы хотите отладить, и перейдите на вкладку свойств.

В разделе компилятора вы увидите свойство «Управляющая схема», которое вы можете установить в значение true или false (выбор из списка после двойного щелчка левой клавишей мышки!).

По умолчанию свойство имеет значение false, поэтому моделирование схемы будет выполняться с нормальной скоростью, в то время как выполнение MCU будет контролироваться шагами отладчика.

Это эквивалентно запуску MCU в реальной схеме, но с выполнением прошивки, которое контролируется интерфейсом JTAG или аналогичным.

Но этого способа недостаточно, чтобы наблюдать некоторые характеристики или особенности вашей прошивки.

Например, осциллограф, прикрепленный к mcu ouput, который генерирует волну, не будет «видеть» эту волну или не сможет считывать ее частоту.

Или, например, серводвигатель не будет вести себя правильно при изменении частот. Когда отладчиком является «Управляющая схема» (Управляющая схема=true), моделирование схемы будет выполняться синхронно с выполнением отладчика, то есть: схема будет выполнять только точное количество шагов, которые она будет выполнять при обычном моделировании для каждого шага отладчика, затем она будет останавливаться до тех пор, пока не будет выполнен другой шаг отладчика.

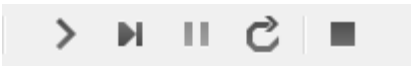
Таким образом, любой элемент в цепи будет «видеть» изменения выводов MCU в нужное время и вести себя как при обычном моделировании схемы.

Но у этого метода есть свои недостатки: так в случае мигающего светодиода, когда выход принимает высокое состояние, светодиод не включается немедленно, что выглядит неправильно, но в действительности правильно, потому что моделирование схемы «зависнет»; просто после того, как выход станет высоким, в это мгновение через светодиод не проходит ток.

Если вы поместите зонд на этот выход, вы увидите, что напряжение уже изменилось на высокое, но светодиод все еще медлит, потому что моделирование схемы «застыло» только на этом этапе.

Редактор-Отладчик (продолжение)

Обратите внимание, что вид инструментальной панели изменился: вместо иконок открыть файл, сохранить и т.д. теперь появился новый набор доступных элементов управления. На рисунке слева направо:



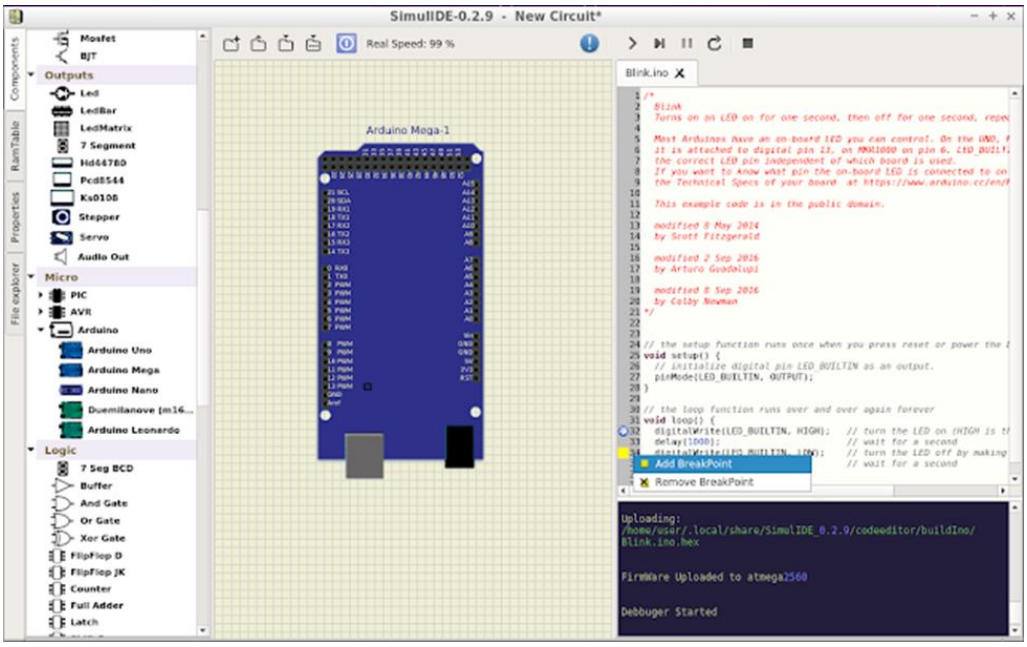
- Шаг: запустить одну строку исходного кода.
- **Перейти к точке останова:** запуск до следующей точки останова.
- Пауза: останавливает запущенный процесс.
- **Сброс:** перезагрузка для запуска программы.
- **Остановить отладчик:** останавливает отладчик и возвращается в режим редактирования.

Обратите внимание, что «Пауза» останавливает запущенный процесс, но остаётся режим отладки, а «Остановить отладчик» фактически останавливает отладчик и возвращает всё в режим редактирования.

В режиме отладки вы можете запустить свой код шаг за шагом, или вы можете добавить точки останова, для этого:

- Щелкните правой клавишей мышки по номеру строки, на которую вы хотите добавить точку останова.
- Выберите «Добавить точку останова» или «Удалить точку останова».

На выбранной строке появится желтый квадрат. А вы можете «Перейти к точке останова», и стрелка перейдет к точке останова (если она доступна) или продолжит работать до тех пор, пока точка останова не будет достигнута.

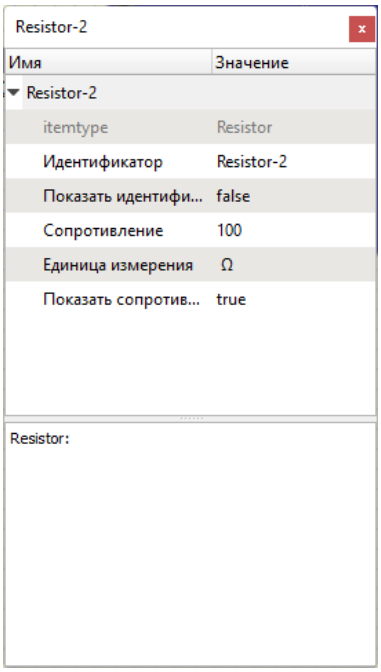


Отладчик установит для микроконтроллера MHZ значение 0, а предыдущее значение восстановится после закрытия отладчика.

Поэтому обязательно остановите отладчик после завершения сеанса отладки.
Вы по-прежнему можете наблюдать за регистрами и переменными MCU в таблице закладки Ram (Mcu Monitor).

Будут обнаружены только глобальные переменные. После завершения сеанса отладки необходимо остановить его и выйти из режима отладки.

Компоненты



Многие компоненты просты в использовании – перетащите их в схему и при необходимости измените значение.

Значения и конфигурация доступна в разделе «Свойства» компонента.

Чтобы перейти к этому разделу, щёлкните правой клавишей мышки по компоненту и выберите раздел «Свойства» в выпадающем меню.

Есть два свойства, присущие всем компонентам:

id – это имя (идентификатор), показанное на схеме, оно может быть любым по вашему усмотрению.

Show id – показать или скрыть id на схеме…

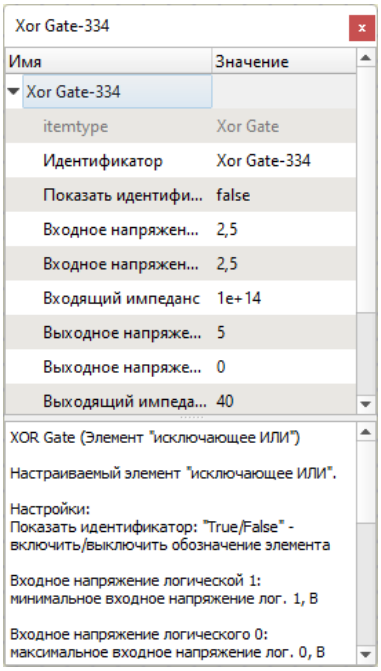
Каждый тип компонента имеет собственный набор свойств, чтобы редактировать значения достаточно двойного щелчка по значению для выбора из списка или ввода значения.

Некоторые значения зависят от единиц измерения; например, у резистора значение сопротивления будет меняться на kΩ, если значение больше (равно) 1000, или на mΩ, если оно меньше (равно) 0,001. Это может немного беспокоить первое время.

Логические компоненты

Большинство логических компонентов имеет общую группу свойств для настройки входных и выходных характеристик. Вы можете увидеть/отредактировать эти свойства – щёлкните правой клавишей мышки на компоненте и выберите «Свойства» в выпадающем меню.

Помимо типичных свойств «Идентификатор» и «Показать идентификатор», общих для всех компонентов, есть несколько свойств, касающихся входов и выходов логических компонентов общего назначения:



- **Входное напряжение логической единицы:** минимальное входное напряжение, которое считается высоким состоянием (высокий порог).
- **Входное напряжение логического нуля:** максимальное входное напряжение, которое считается низким состоянием (низкий порог).
- **Входной импеданс:** входное сопротивление.
- **Выходное напряжение логической единицы:** выходное напряжение для высокого состояния.
- **Выходное напряжение логического нуля:** выходное напряжение для низкого состояния.
- **Выходящий импеданс:** выходное сопротивление.

Редактируя эти свойства, вы можете «имитировать» различные логические семейства/технологии.

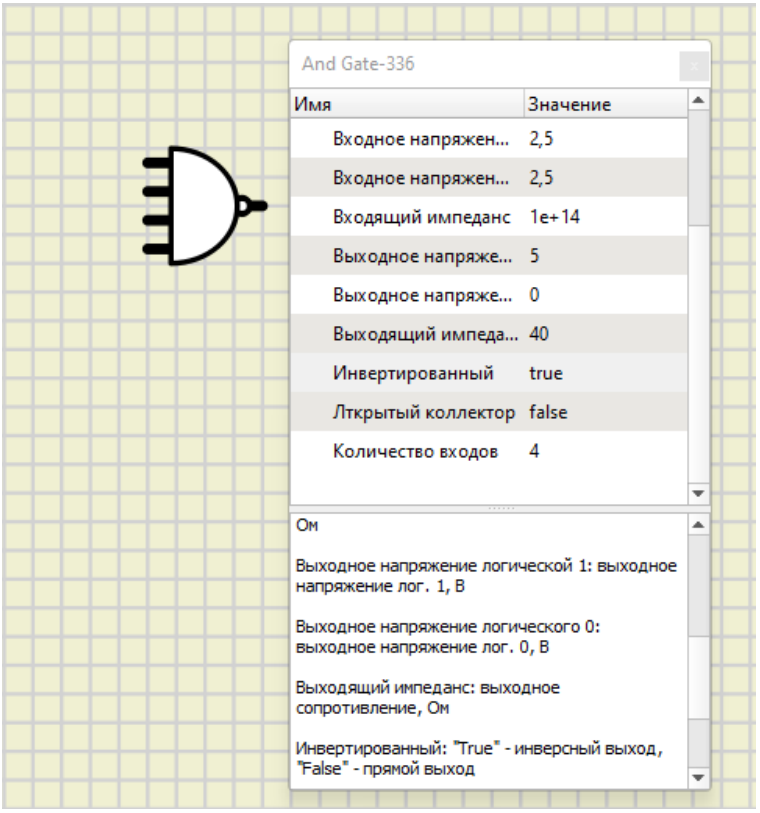
Так, например, вам для входов триггера Шмитта может потребоваться установить желаемые входные напряжения: 3 В для высокого порога и 2 В для низкого порога срабатывания. Или вы можете настроить входное и выходное сопротивление, чтобы получить различные характеристики.

Для каждого типа компонентов есть и другие дополнительные свойства. Давайте посмотрим на характеристики каждого логического компонента.

- Логический вентиль:

- **Инвертированный:** является ли выход инвертирован или нет.
- **Открытый коллектор:** выход высокого состояния – высокое сопротивление.
- **Количество входов:** количество входов от затвора (если применимо).

На первый взгляд кажется, что SimulIDE не имеет инверторных, NAND, NOR, NXOR вентилей. Но они на самом деле есть; все, что вам нужно сделать, это установить «Инвертированный» на «true» (список открывается двойным щелчком левой клавиши мышки в поле определения!).



Буфер имеет дополнительную опцию:

- **Третье состояние:** подключает или отключает выход.

При активации появится дополнительный вывод включения. Этот контакт контролирует, является ли выход высоким импедансом или нет, когда этот контакт в высоком состоянии, выход - высокое сопротивление, когда в низком, он действует как обычный буфер.

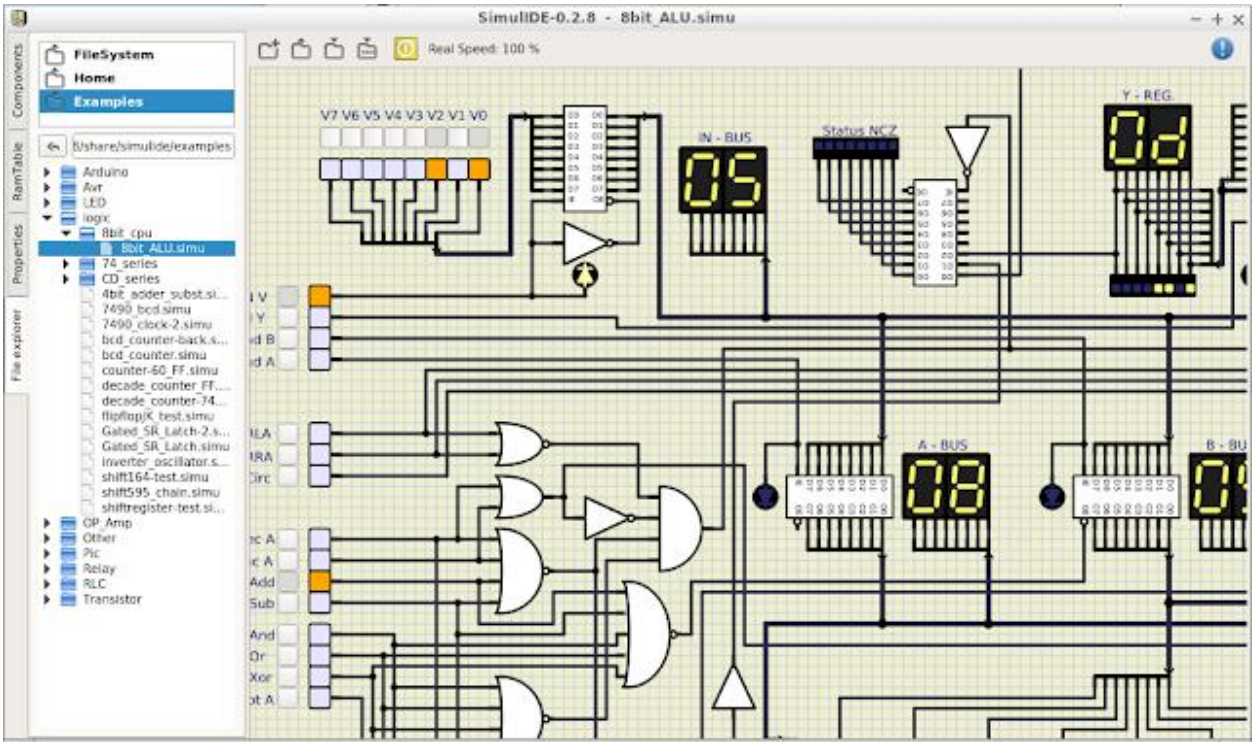
- Шины:

При использовании нескольких параллельных линий иногда полезно группировать их в шинах. SimulIDE предоставляет компонент для этого.

Свойства:

- **Разрядность бит**: задайте количество битов шины.
- **Стартовый бит**: установите начальный бит, остальные биты будут пронумерованы от этого.

Линии шин показаны толще, чем обычные линии, и вы не можете подключить обычные линии к шине.



Микроконтроллеры

SimulIDE поддерживает моделирование микроконтроллеров Pic, Avr и модуля Arduino.

Сам микроконтроллер моделируется внешними симуляторами:

- Моделирование AVR предоставлено simavr: https://github.com/buserror/simavr

- Моделирование PIC предоставлено GpSim: http://gpsim.sourceforge.net/

Базовое использование:

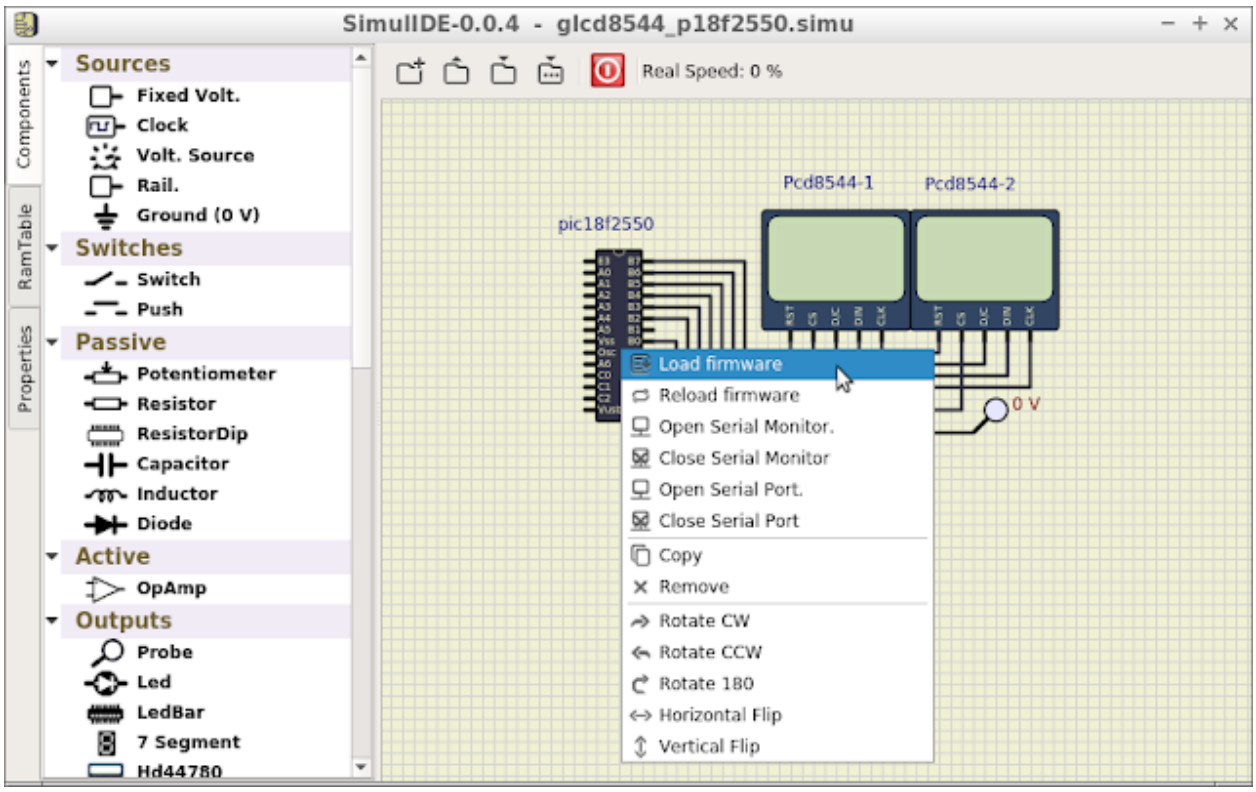
Чтобы выполнить моделирование микроконтроллера, просто выберите микроконтроллер, который вам нужен, и добавьте на рабочее поле, а затем постройте схему, необходимую для тестирования вашего устройства. Когда схема будет готова, щелкните правой клавишей мышки по микроконтроллеру и выберите «Загрузить прошивку». Выберите HEX-файл для загрузки в микроконтроллер. Если сообщение об ошибке не отображается, это означает, что прошивка была правильно загружена.

Чтобы установить тактовую частоту микроконтроллера:

- Выберите его.
- Щёлкните правой клавишей мышки, а из выпадающего меню выберите «Свойства».
- Установите свойство «МГц» в нужное значение в мегагерцах.

По умолчанию тактовая частота установлена в 20 МГц для Pic-контроллеров и 16 МГц для Avr и Arduino.

Теперь вы готовы включить моделирование и увидеть симуляцию.

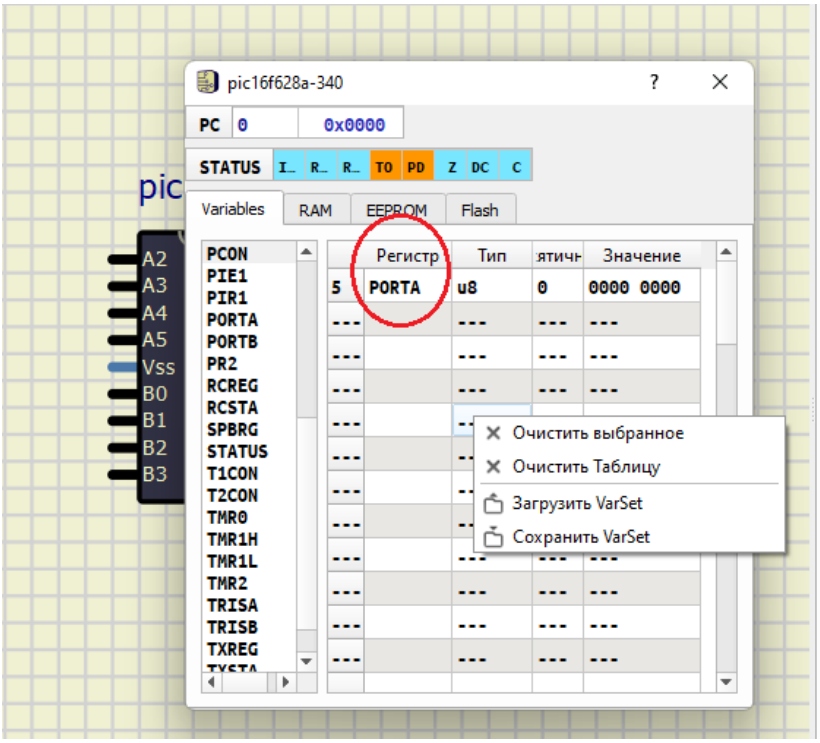


Просмотр регистров и переменных:

Во время работы прошивки вы можете наблюдать за регистрами и переменными:

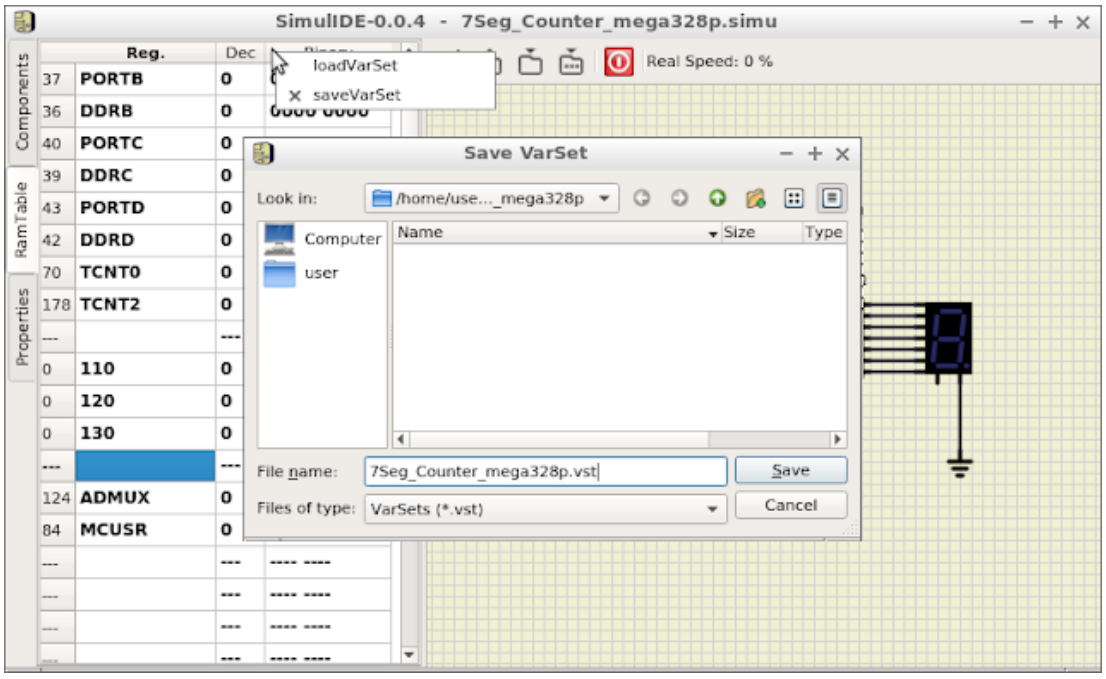
- Щёлкните правой клавишей мышки по микроконтроллеру, выберите раздел «Open Mcu Monitor» и щёлкните левой клавишей мышки по нему.
- В появившемся окне на закладке «Variables» введите (в верхнем регистре) имя регистра.

Вы увидите регистровые значения в десятичном и двоичном виде в полях «Десятичное» и «Значение». Если ввести имя регистра, адрес будет отображаться в левом поле только для чтения в виде десятичного значения.



Можно сохранить список регистров, которые вы просматриваете, чтобы вы могли снова открыть его в другом сеансе вместо того, чтобы вводить все снова:

- Щелкните правой клавишей мышки в любом поле (кроме поля «Регистр») окна наблюдения.
- Выберите «Загрузить VarSet» или «Сохранить VarSet», то есть, загрузить или сохранить набор переменных (используется расширение .vst).
- Появится диалоговое окно Загрузка или Сохранение.
- Выберите varset, который вы хотите загрузить, или введите имя файла varset для сохранения.



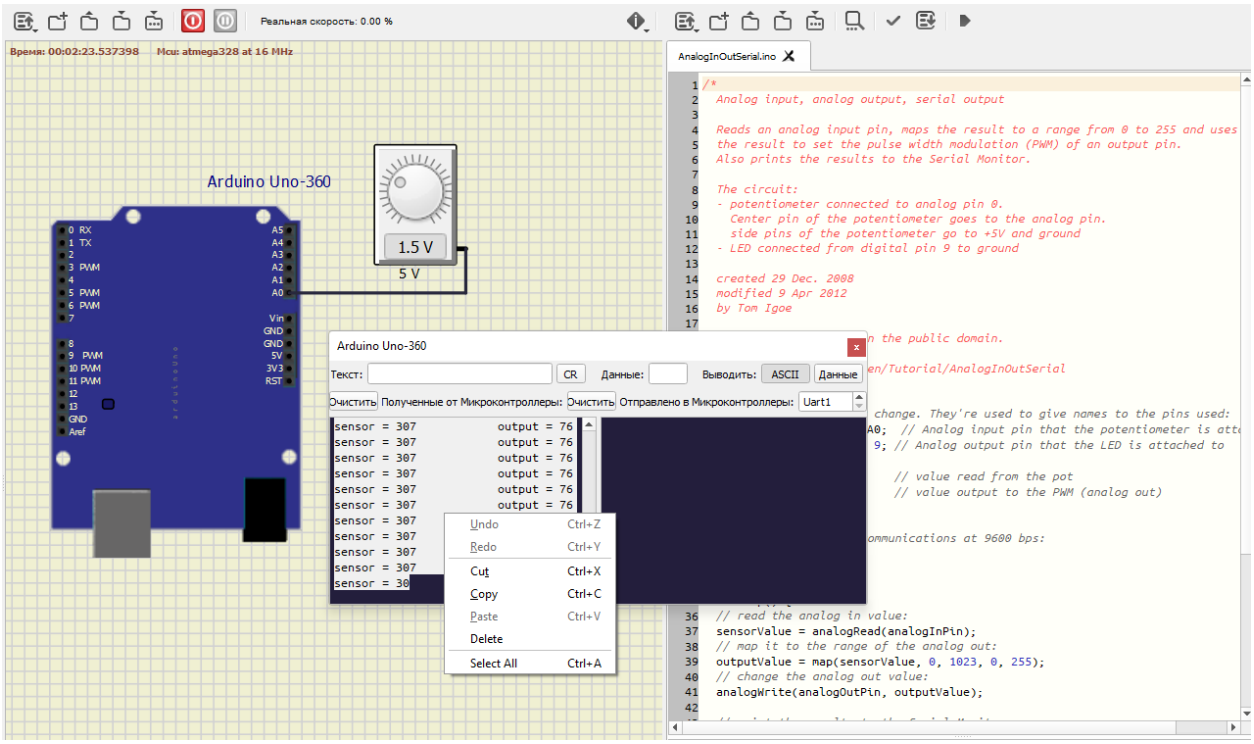
Последовательная коммуникация

- **Мониторинг uart**:

Вы можете контролировать первый встроенный модуль последовательного обмена данными микроконтроллера, uart. Щелкните правой клавишей мышки по микроконтроллеру и выберите «Открыть Serial Monitor», появится новый виджет. Последовательный монитор разделен на две части:

- Одна сторона показывает данные, отправленные микроконтроллером.
- Другая сторона показывает данные, полученные микроконтроллером.

Вы можете распечатать десятичное значение отправленных/полученных данных или связанное с ним значение ascii.
Существует поле «Текст» для отправки символов ascii на микроконтроллер и поле «Данные» для отправки 0-255 значений на микроконтроллер:



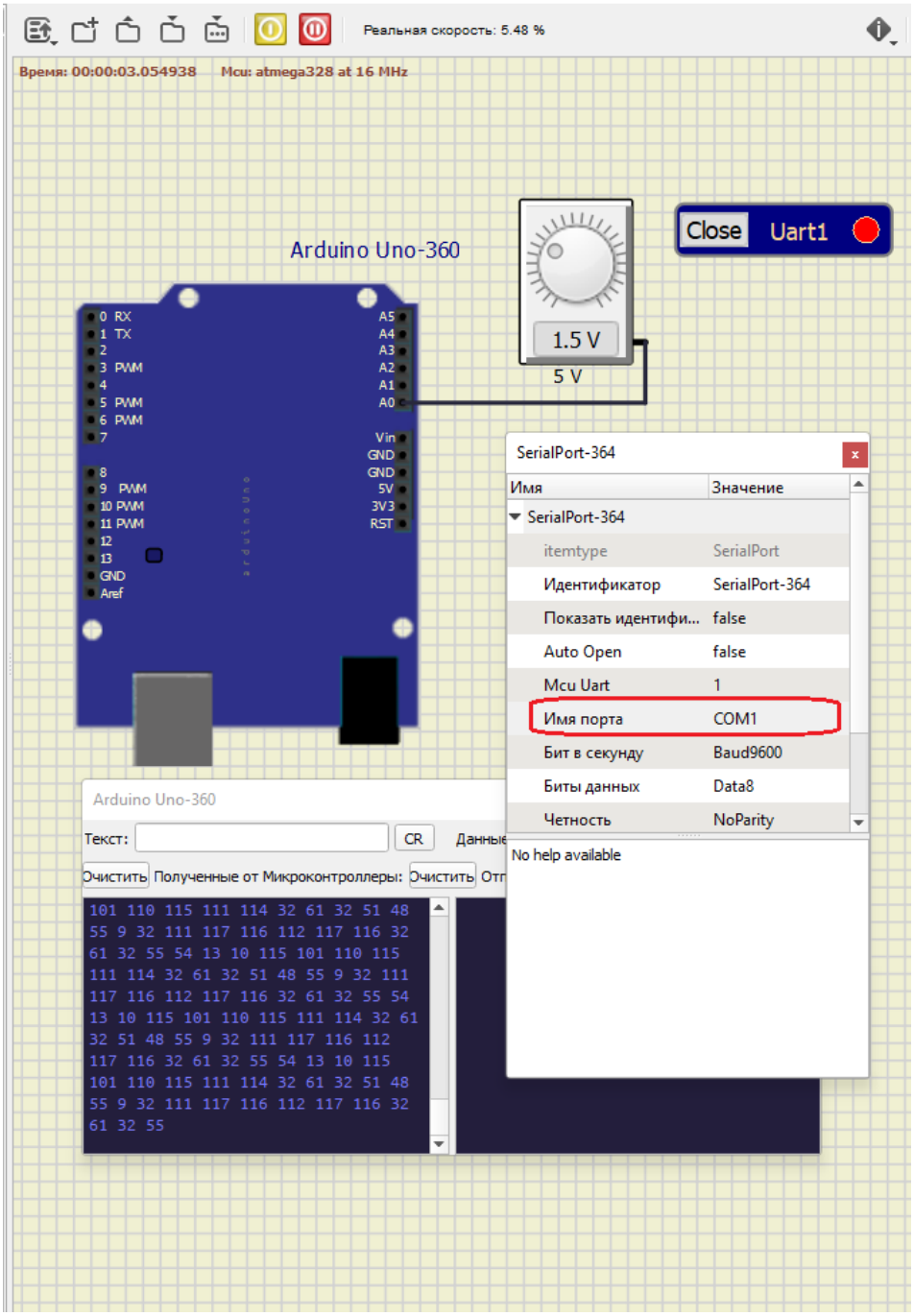
-Подключение к последовательному порту:

Возможно подключение микроконтроллера к последовательному порту на вашем компьютере:

- Щелкните правой клавишей мышки по микроконтроллеру.
- Выберите "Открыть Serial Port".
- Появится новый виджет.

Это панель подключения к виртуальному последовательному порту, где вы можете настраивать, подключать и отключать любой последовательный порт, который вы хотите, будь то реальное оборудование или виртуальное.

Вы по-прежнему можете использовать последовательный монитор, чтобы увидеть данные:



В оригинальном руководстве далее есть отсылка и к компилятору, и к отладчику. В данном руководстве сошлёмся на разделы «Редактор-Компилятор» и «Редактор-Отладчик» в описании правой панели, которые вы найдёте выше.

Подсхемы

Создание подсхемы

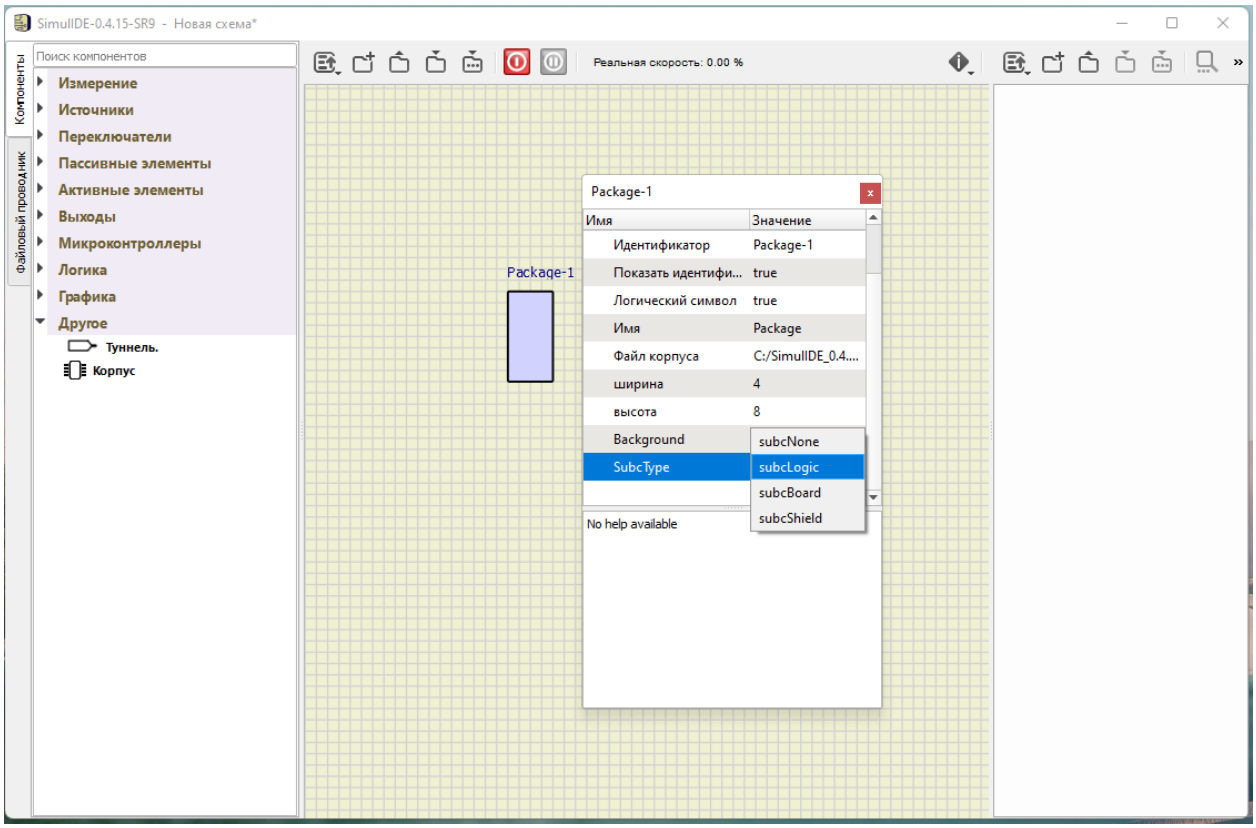
Для того чтобы создать подсхему нам понадобятся:

- Файл пакета (ов).
- Файл подсхемы.
- Добавить запись в .xml файл.

Поместите файл пакета(ов) и подсхему в папку, указанную в записи xml.

Перезапустите SimulIDE, и новая подсхема появится в категории, указанной в xml-файле.

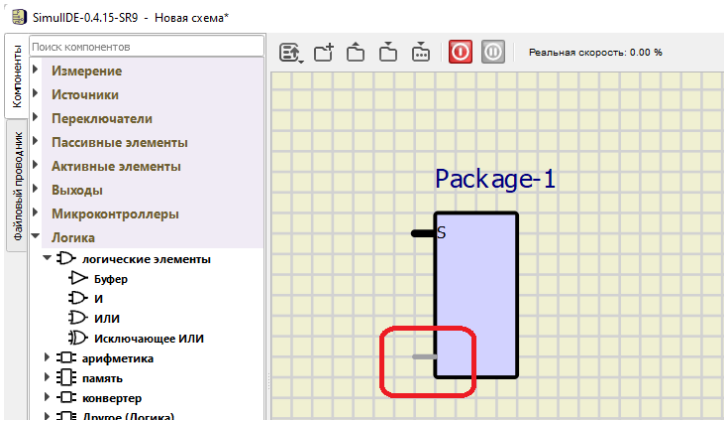
Создание файла (ов) пакета (ов):
В разделе «Другое» есть компонент «Корпус», в конце списка компонентов.
Когда вы добавляете его в схему, есть только синее поле, это пустой пакет, готовый к настройке:



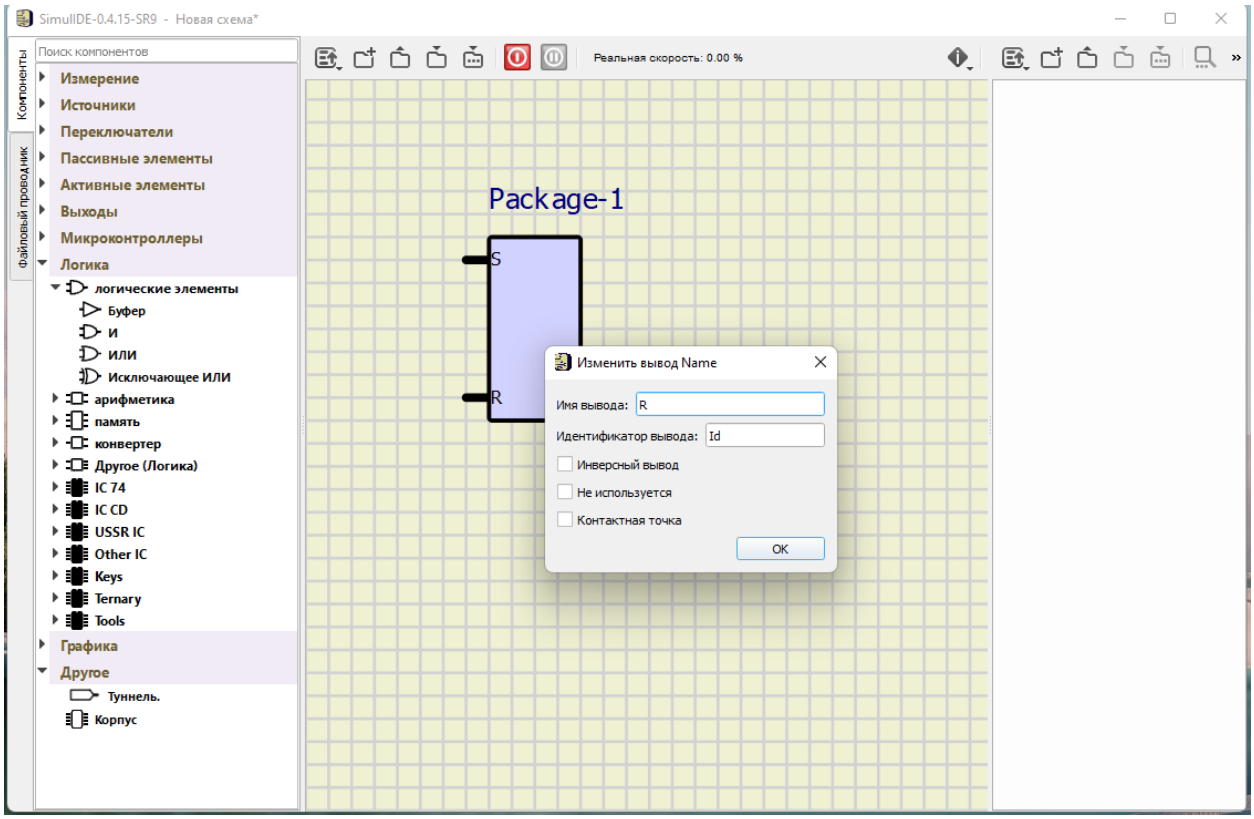
Здесь вы можете загрузить существующий файл пакета, чтобы отредактировать его, или создать новый пакет. В свойствах компонента можно задать размер пакета в ячейках схемы. Размер ячейки по умолчанию составляет 4x8.
Существуют также свойства для установки типа пакета, показанные на рисунке выше (двойной щелчок левой клавишей мышки в поле!).

Создание новых выводов:

Если вы наводите указатель мыши на края корпуса при нажатии на клавиатуре клавиши «Shift», вы увидите серую полоску вывода.



Щелкните левой клавишей мышки в нужном месте, появится диалоговое окно для задания свойств вывода.



После нажатия OK в этой позиции будет создан новый вывод с заданным именем и свойствами.

- **Имя вывода:** имя, которое отображается в пакете.
- **Идентификатор вывода:** уникальный идентификатор контакта.
- **Инвертировать вывод**: вывод отображается как перевернутый.
- **Не используется:** неактивный вывод.
- **Контактная точка:** делает длину вывода = 0.

Редактирование выводов:

Вы можете редактировать существующие контакты, щелкнув правой клавишей мышки по краю пакета в месте контакта.

Появится контекстное меню с параметрами закрепления:

- **Переместить вывод:** щелкните и переместите контакт.
- **Изменить вывод:** открывает виджет «Изменить вывод».
- **Удалить вывод:** удаляет вывод.

Типы корпусов:

> Примечание: Описанное ниже относится к предыдущей версии программы, но остаётся, видимо, полезно и для последующих версий.

Существует два возможных типа пакетов: Чип и Логический символ.

Можно иметь оба варианта для одного и того же компонента и переключаться между ними. Есть некоторые соображения, которые следует учитывать при создании пакетов Chip и Logic

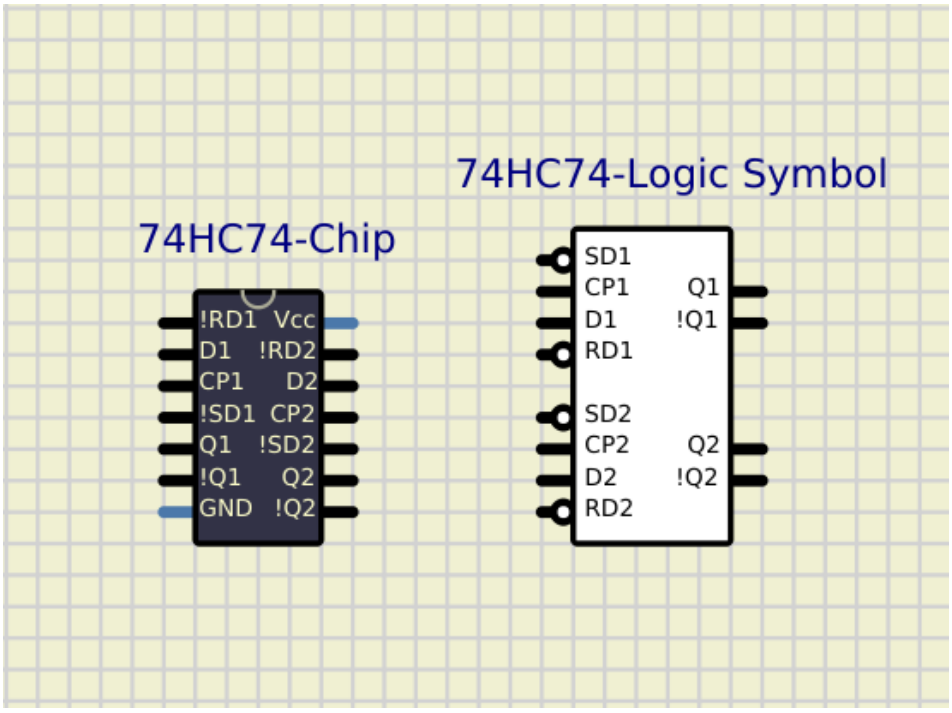
Symbol для одного и того же компонента. Оба пакета (корпуса) должны быть Pin-совместимыми для работы с одинаковым файлом подсхем:

- Все активные выводы (pins) должны существовать в обоих пакетах (не относится к неиспользуемым контактам).
- Один и тот же pin-код должен иметь одинаковый id в обоих пакетах.

Обратите внимание, что для каждого контакта есть «Id» и «Label», поэтому контакт может иметь различную метку (label) в обеих упаковках, в то время как Id должен быть одинаковым.

В качестве примера на изображении ниже, для 74HC74 некоторые контакты имеют немного разные метки: например, !RD1 против RD1.
Ряд этих различий в каком-то роде автоматизированы при создании пакета, например, дополнительные пространства и перевернутые контакты.

Для перевернутых контактов вы можете либо использовать «!» в качестве первого символа метки, либо установить контакт как перевернутый, в обоих случаях simulide установит Id как: «!PinName». Рекомендуется использовать «! PinName» для пакетов chip и «PinName»+invert-Pin для пакетов логических символов, но в некоторых случаях вы можете использовать «! PinName» для обоих, например «! Qn» в корпусе 74HC74 ниже.

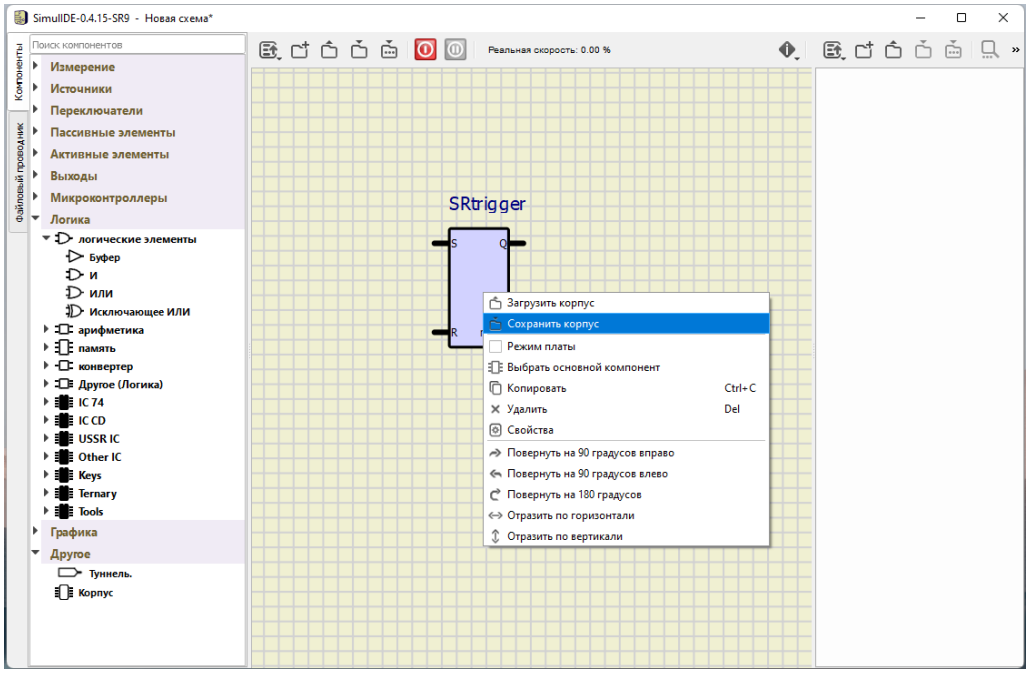


Создание файла подсхем:

> Примечание: эта часть руководства рассматривается в версии SimulIDE-0.4.15 -SR9. Описание будет несколько отличаться от представленного на сайте разработчиков.

Разберём создание подсхемы на примере SR-триггера. Выше мы начали создавать корпус для нашего компонента. Завершим его, добавив два выхода. В свойствах последнего «not Q» поставим метку «Инверсный вывод». А в свойствах самого корпуса обозначим его как «SRtrigger». Удобно, наверное, было бы назвать его SR-trigger, но это можно оставить для последующих экспериментов.

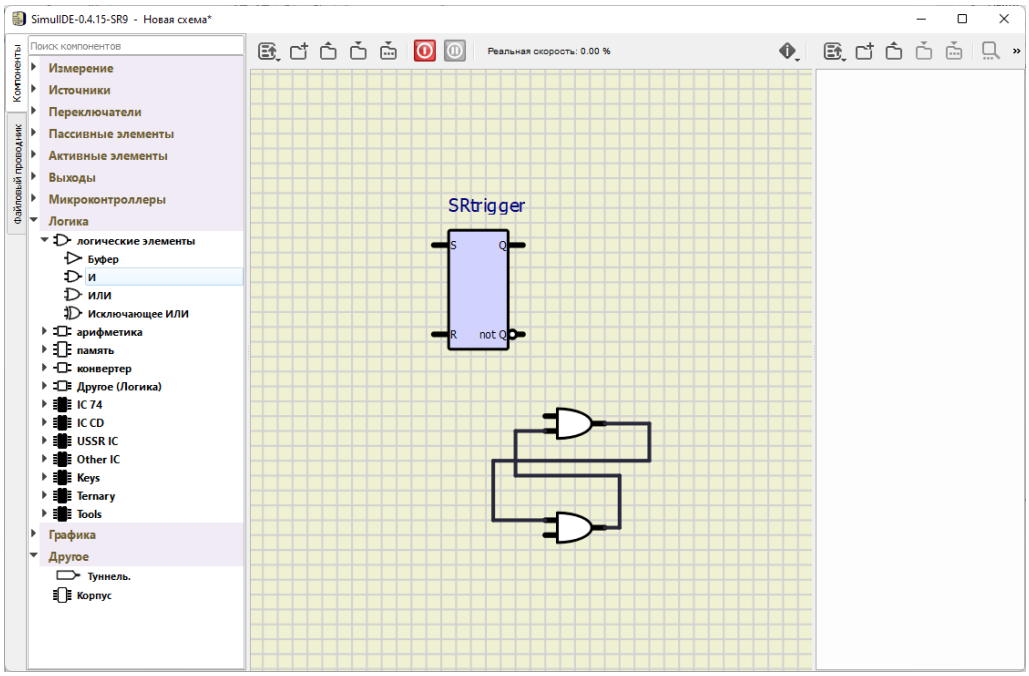
Щелчок правой клавишей мышки по корпусу открывает меню, в котором есть пункт сохранения корпуса.



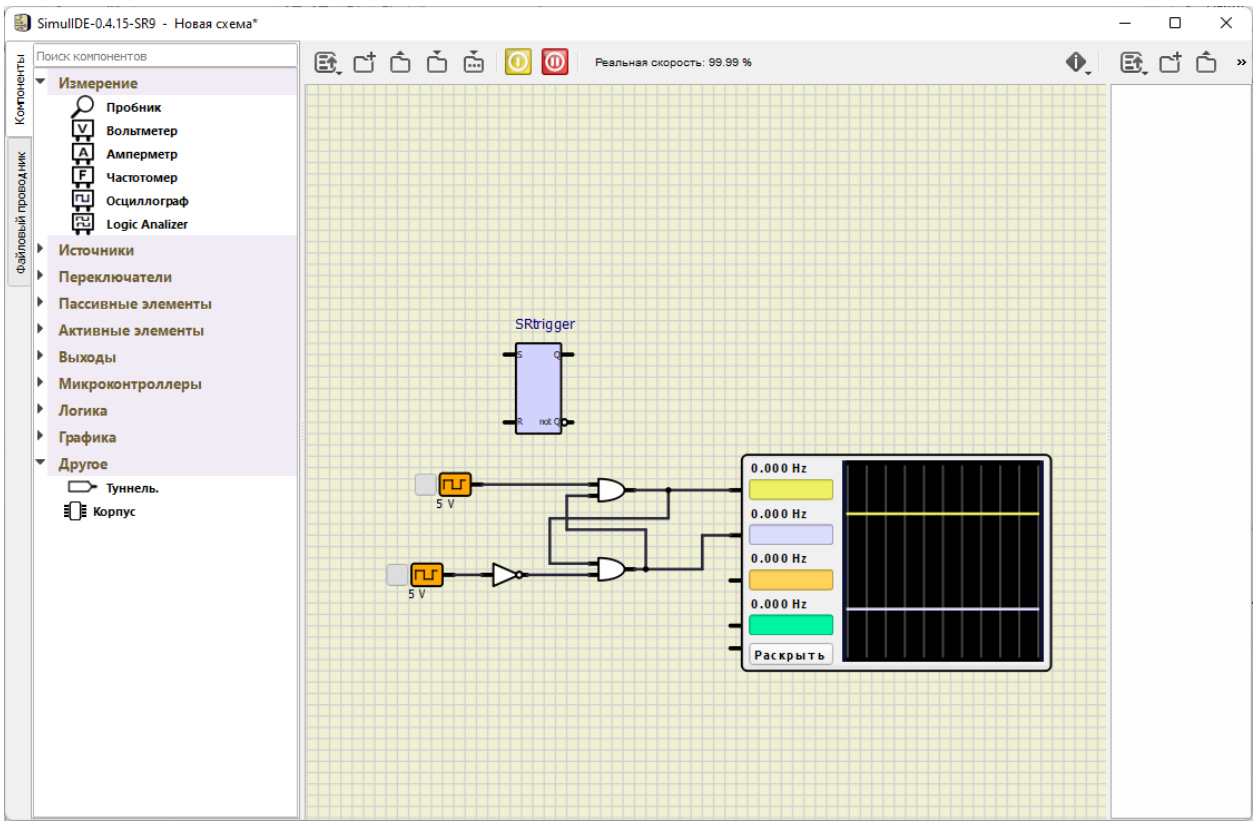
Для сохранения я, например, создал папку на рабочем столе с именем SRtrigger, где и намерен хранить всё необходимое. Файл сохранится с расширением «.package».

Теперь соберём схему (подсхему) этого триггера из логических компонентов.

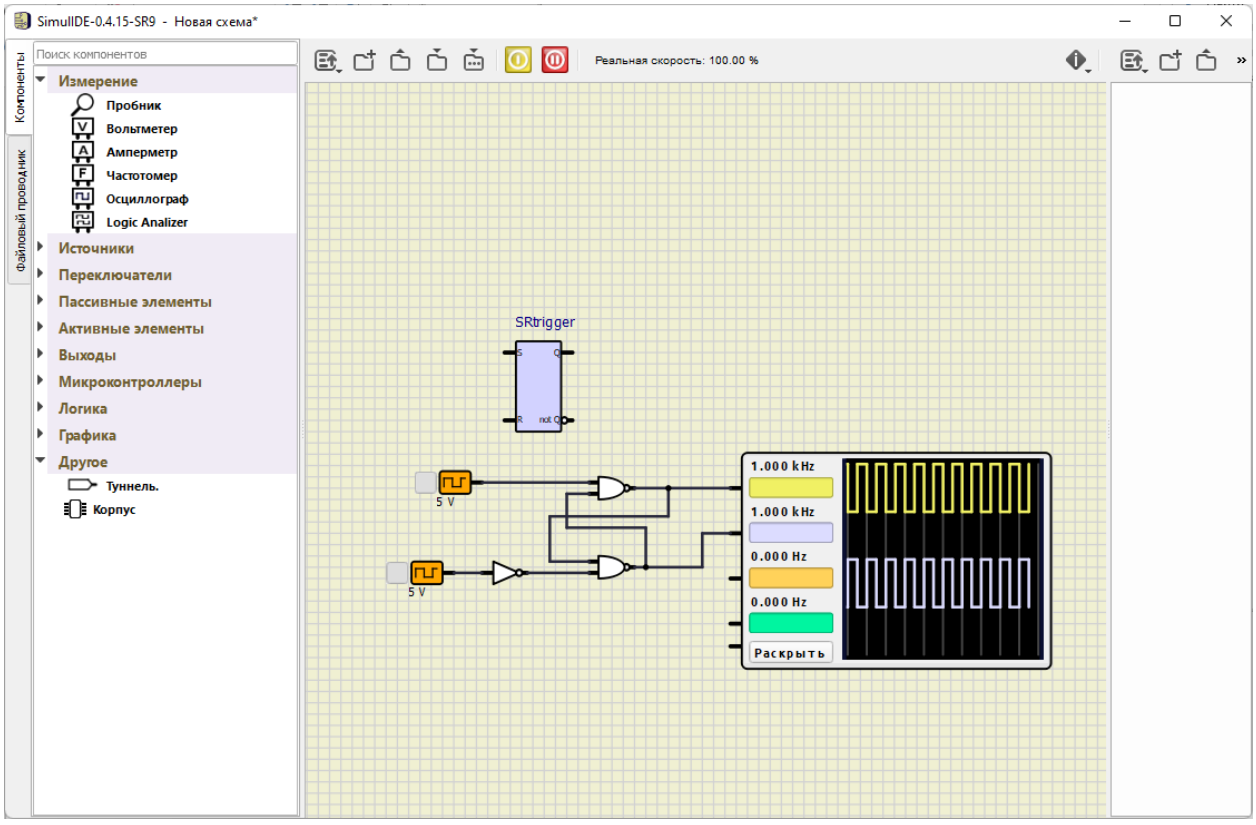
Пока добавим только два двухвходовых элемента «И», соединив их должным образом.



Чтобы проверить правильность сборки используем два тактовых генератора, работающих в противофазе. Поскольку я не нашёл в настройках генератора возможности изменить фазу, поступим проще – добавим инвертор, выполненный из буфера, у которого в свойствах указано, что выход инверсный.

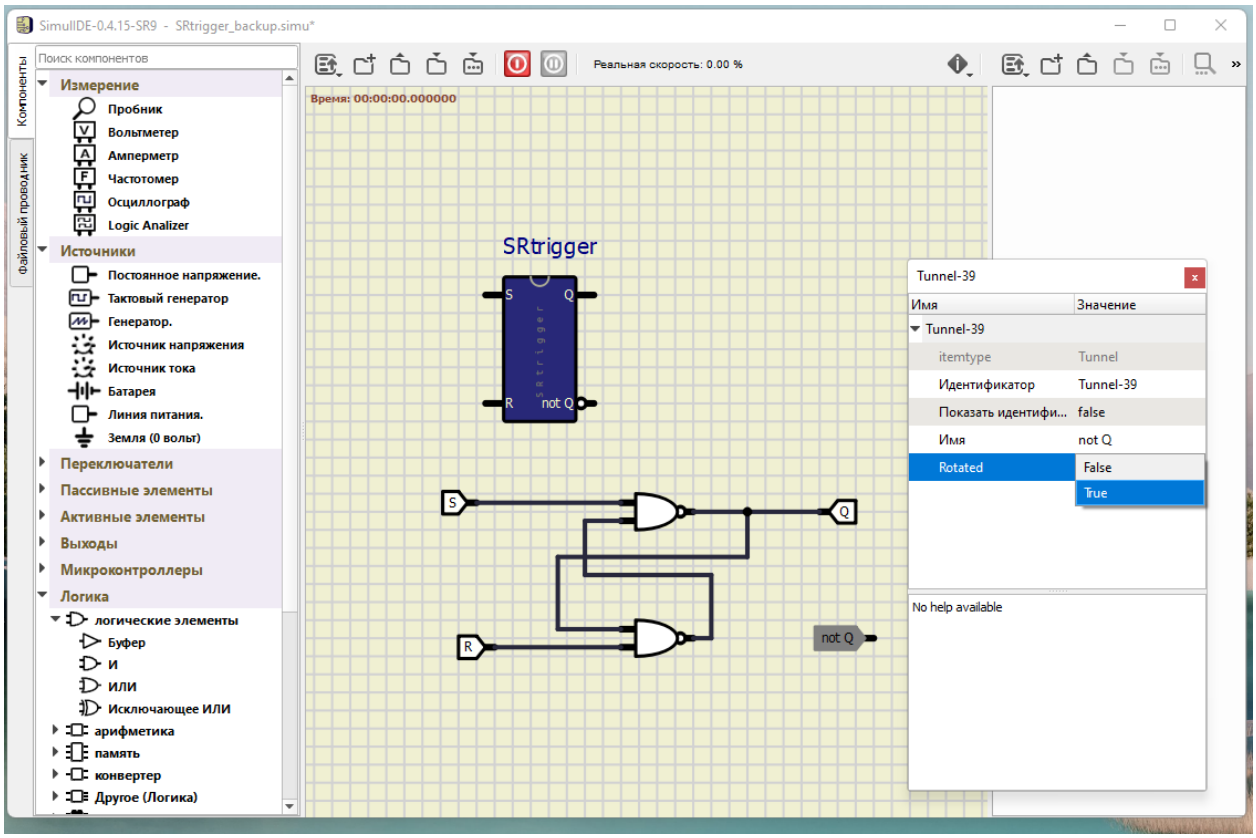


На выходах отсутствуют сигналы, что и следовало ожидать, поскольку нам были нужны компоненты «И-НЕ». Инвертируем выходы у нашей сборки.



Теперь на выходах триггера можно увидеть выходные сигналы, которые, как и следовало ожидать, находятся в противофазе.

Уберём всё лишнее, а на входы и выходы добавим соединители, туннели из раздела «Другое». Выходные соединители нужно «развернуть»: задавая имя соединителя, установим в *true* свойство «Rotated». Можно отразить их по горизонтали, но тогда надписи будут перевёрнуты.

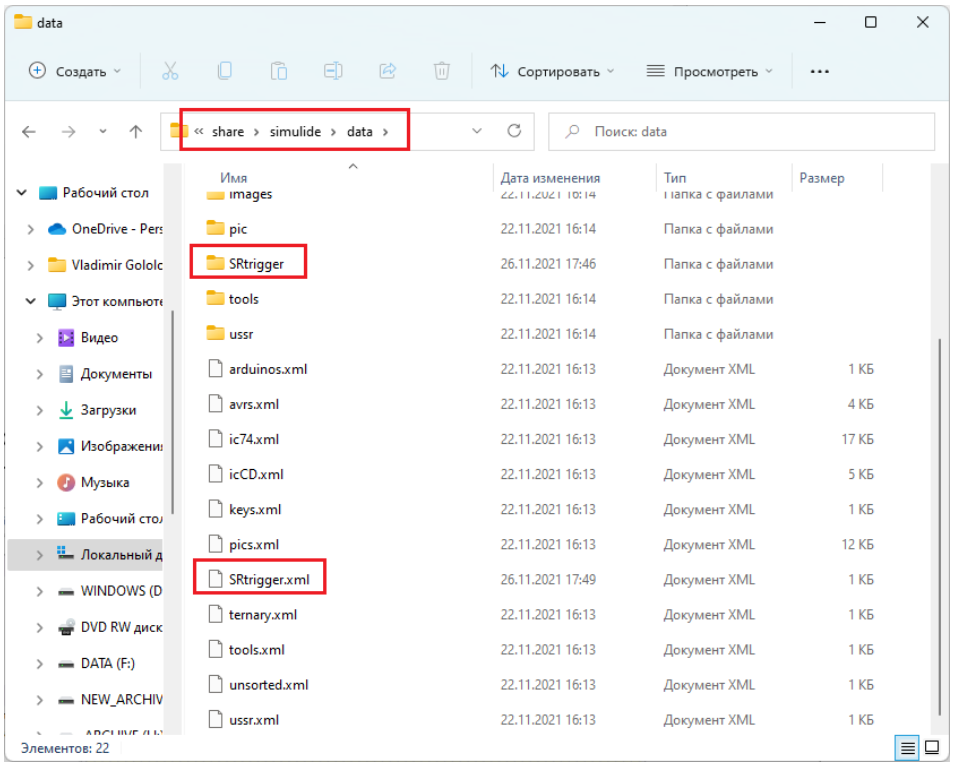


Сохраним полученный результат в формате simu в качестве subsircuit в папке «SRtrigger». Осталось написать (или отредактировать готовый) файл xml.

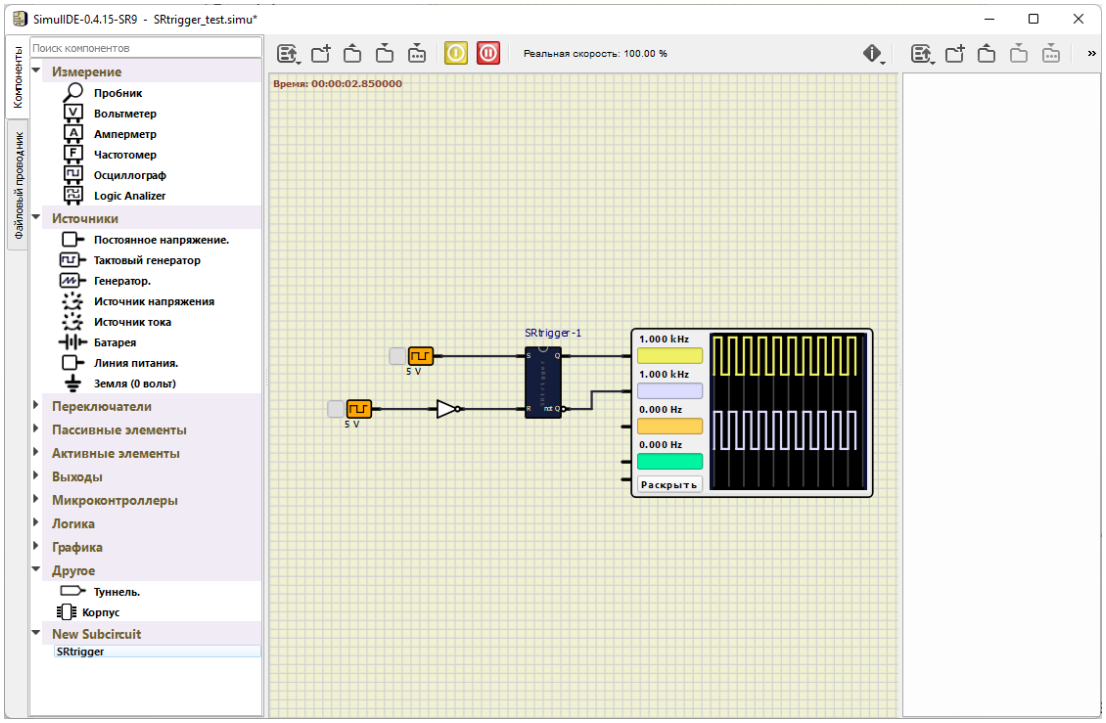
```
<!DOCTYPE SimulIDE>

<itemlib>

    <itemset category="New Subcircuit" type="Subcircuit">

    <item name="SRtrigger" package="SRtrigger/SRtrigger.package"
subcircuit="SRtrigger/SRtrigger.simu"></item>

</itemset>

</itemlib>
```

Я записываю этот файл в блокноте, где, на всякий случай, выбираю кодировку UTF-8.

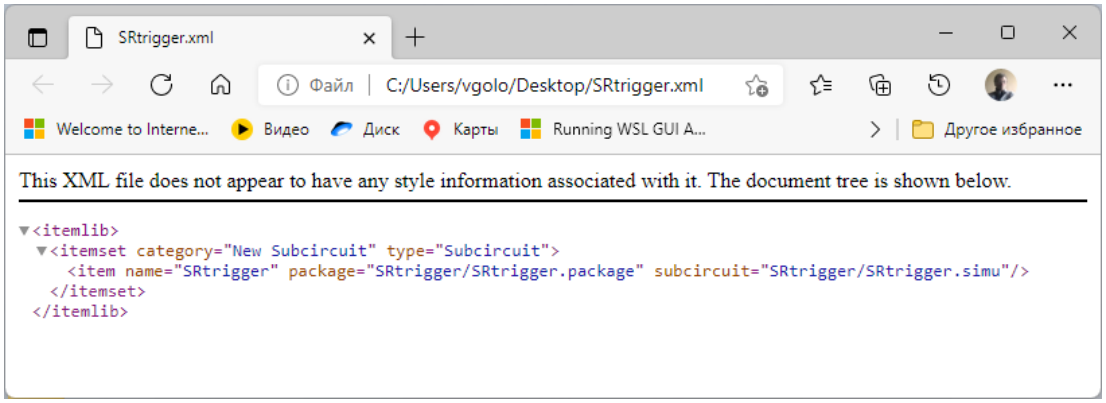
После добавления полученных «заготовок», папки с корпусом и схемой и файла xml, к компонентам программы SimulIDE…



Сознаюсь не сразу у меня всё получилось, но теперь можно запустить программу, чтобы выбрать вновь созданный компонент и провести опыт для проверки его работы.



В какой-то момент при переделке файла xml у меня возникли проблемы, файл не читался при попытке вставить в программу новый компонент. Для проверки правильности написания файла в ОС Windows 11 оказалось удобно использовать просмотр файла в Internet Explorer.



Если всё правильно, файл открывается, если нет, появляется сообщение об ошибке. И ещё, при написании файла я, например, допустил опечатку, пропустив букву в имени папки, эта ошибка вскрылась при попытке добавить новый компонент в схему – программа выдала сообщение, что не может открыть файл xml. В разные моменты времени появлялись разные ошибки, с которыми я толком не разобрался. Поэтому ниже я возвращаюсь к описанию с сайта разработчиков.

Добавить запись в .xml файл:

Для того, чтобы ваша подсхема была включена в список компонентов SimulIDE, вам необходимо добавить запись в один из существующих xml-файлов или создать новый.

Эти файлы находятся по адресу: share/simulide/data

Пожалуйста, посмотрите на любой существующий файл .xml в папке данных, чтобы получить представление.

Структура этих xml-файлов выглядит следующим образом:

```
<! DOCTYPE SimulIDE>

<itemlib>

  <itemset category="New Subcircuits" type="Subcircuit">
  <item name="lcdShield" package="test/lcdshield_LS.package" subcircuit="test/lcdshield-board.simu" />
  <name= "Arduino Uno B" package="test/arduino_board.package"  subcircuit ="test/arduino_uno_board.simu" />

  </itemset>

</itemlib>
```

Поясним по уровням:

<itemset/> представляется набор компонентов.
category имя категории, в которую будут включены все компоненты этого набора.
Это может быть существующая категория или новая.
Возможно обращение к подкатегории: «Category/Subcategory» (Категория должна существовать).
type является типом компонента, в данном случае «Подсхема».
<item/> представляет компонент, добавляемый в список.
name имя, которое будет отображаться в списке компонентов.
package путь к файлу пакета относительно папки данных (где находится .xml файл).
subcircuit путь к файлу подсхемы относительно папки данных (где находится файл .xml).
info (необязательно) дополнительная информация, которая будет отображаться помимо имени компонента.

Существует возможность упростить эти записи, указав папку, в которой находятся файлы подсхем и пакетов. Чтобы этот параметр работал, все имена должны быть одинаковыми, а файлы должны находиться в папке с одинаковым именем.

Примером может быть 74 серия (в версии 0.5.15):

Например, файлы 74HC00 находятся в папке с именем 74HC00 внутри папки «ICs».

ICs (папка)
  74HC00 (папка)
      74HC00.simu
      74HC00. package
      74HC00_LS. package

Тогда запись xml-файла выглядит следующим образом:

```
<itemset category="IC 74/7400-7499" type="Subcircuit" folder="ICs">
<name= "74HC00" info=" quad2-input NAND gate" >
</item>
```

В этом случае атрибут **folder** применяется ко всем компонентам в этом **itemset.** Но для каждого **item** можно использовать разные варианты.

И можно задать точный путь к файлам **package** и **sucircuit:**

```
<itemset category="IC 74/7400-7499" type="Subcircuit" folder="ICs">
<name= "74HC00" info=" quad2-input NAND gate" ></item>
<name= "Comp_X" folder="XX" info=" Мои файлы в папке XX" ></item>
```

<item name="Comp_Y" package="blah/cY.package" subcircuit="bla2/compY.simu" info=" Мои файлы в папках blah & bla2" ></item>

Файлы подсхем и пакетов расположены в папках внутри папки данных, но это не обязательно. Путь к этим файлам находится относительно папки «data».

Создание плат

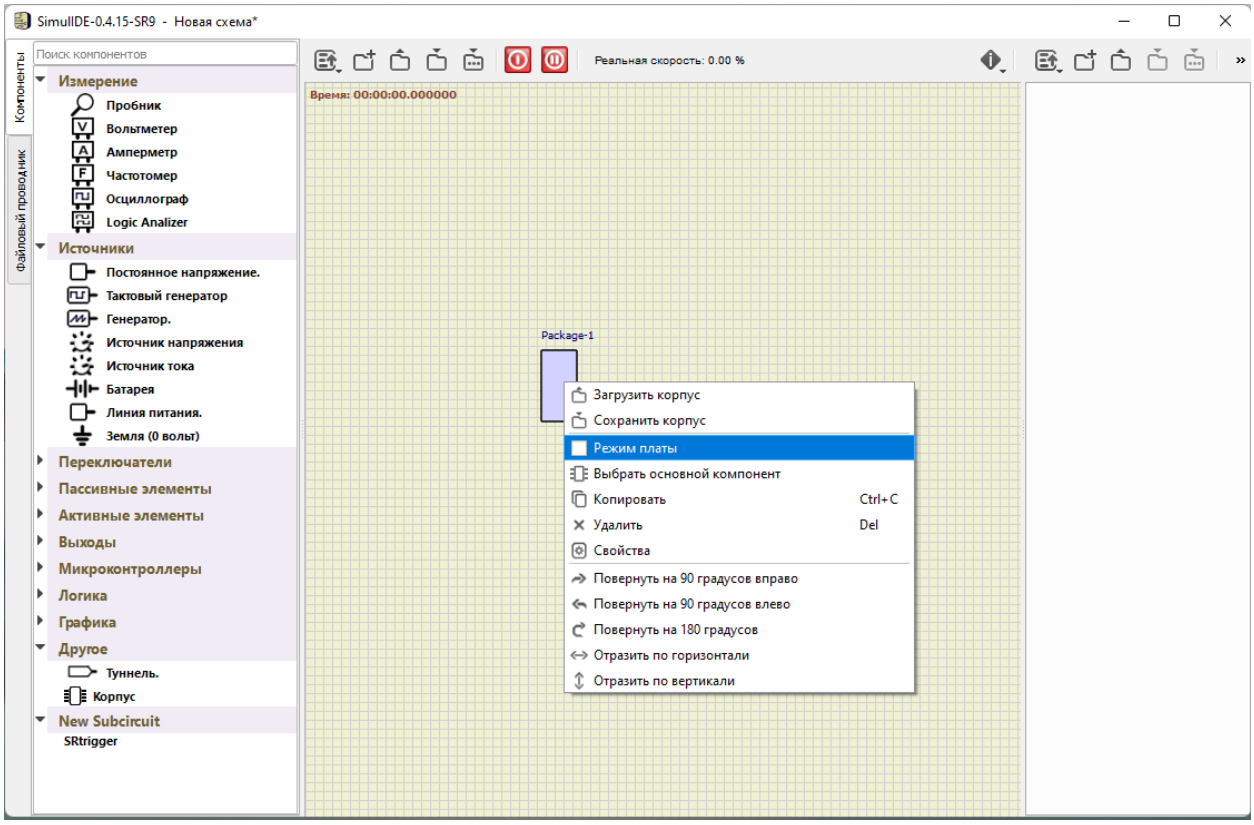
На сайте этот раздел сопровождается «кино», которое лучше посмотреть. Итак.

Платы представляют собой особый тип подсхем, содержащих «графические» компоненты.

Графические компоненты – это те, которые предназначены для взаимодействия с пользователем: дисплеи, двигатели, кнопки, циферблаты и т. д. Основное различие в том, что эти графические компоненты показаны, в то время как остальные компоненты скрыты.

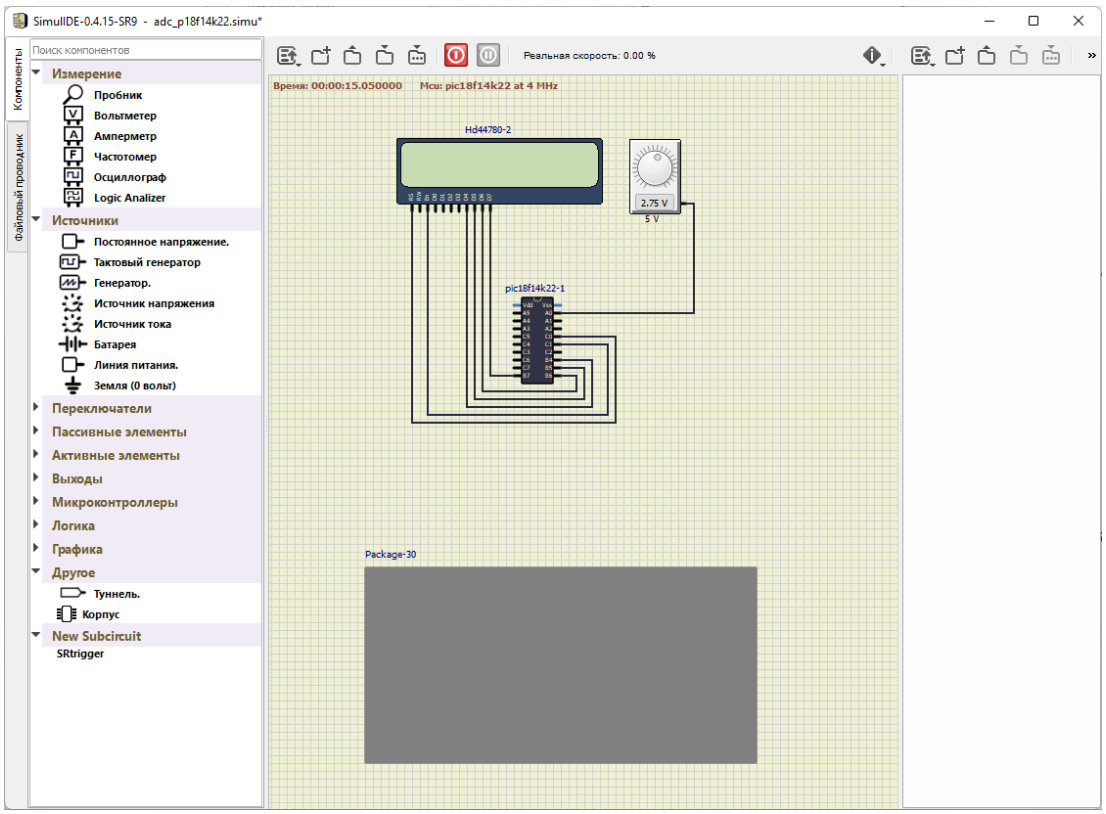
Для этого требуется дополнительный шаг: размещение графических компонентов внутри пакета.

Чтобы решить эту проблему, компонент пакета имеет контрольную запись «Режим платы» в контекстном меню.

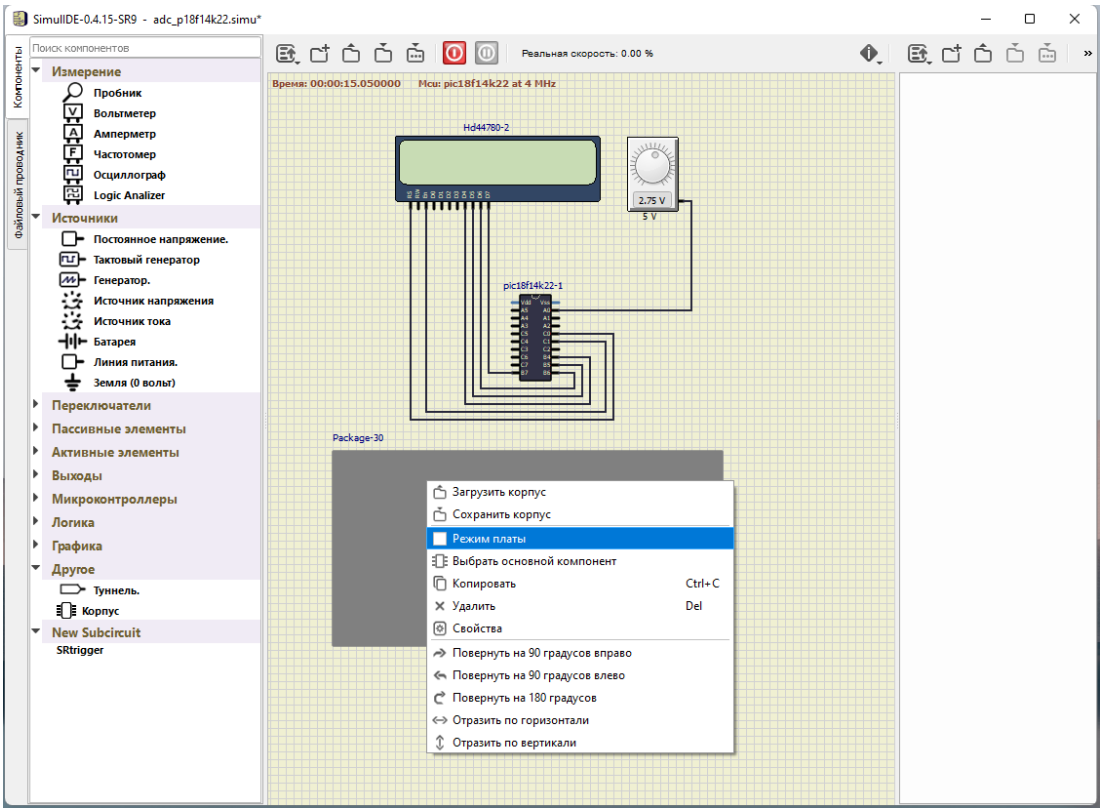


Проверка этой записи скрывает все неграфические компоненты в подсхеме и показывает графические, позволяя поместить их в положение, которое будет отображаться при использовании этой подсхемы.

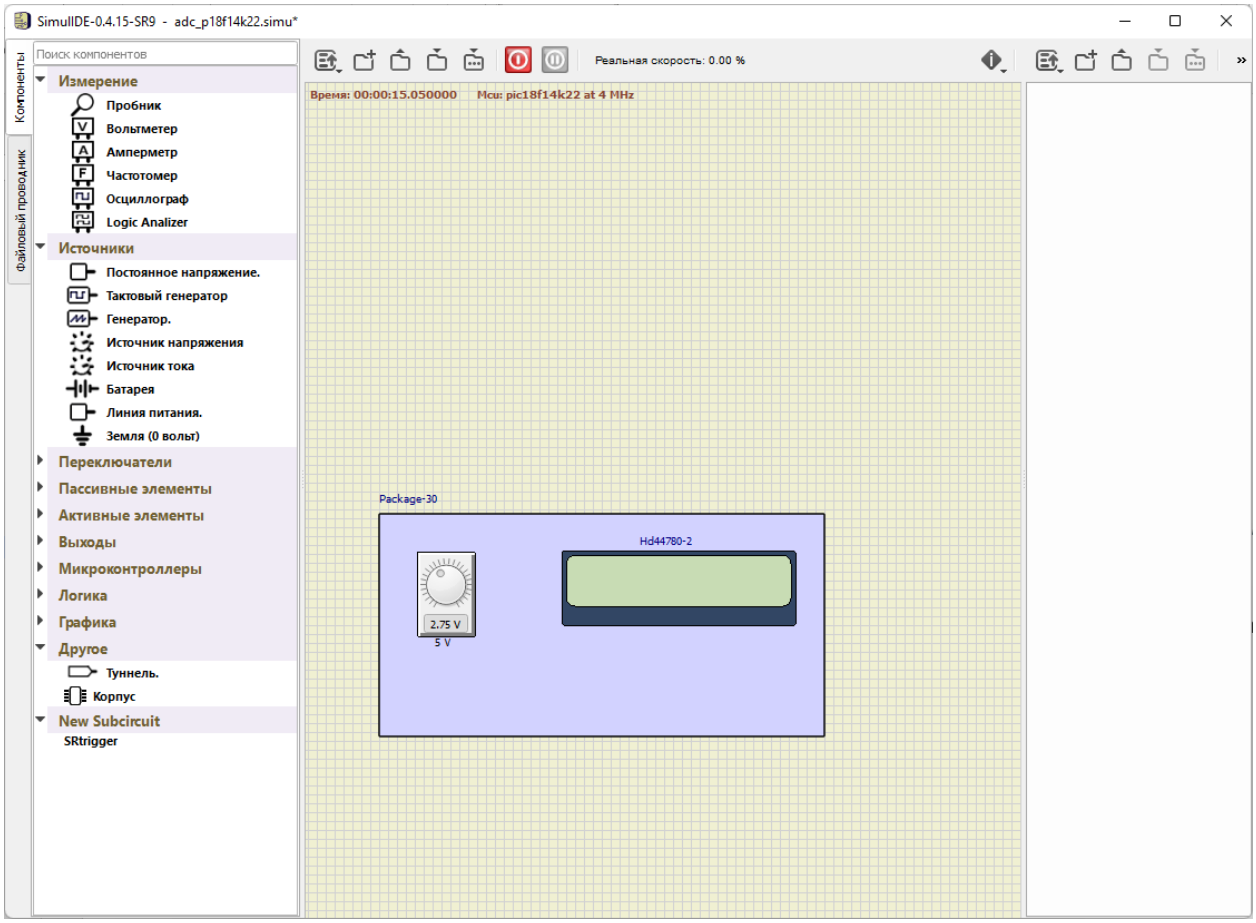
Выберем подходящую схему, добавим корпус, в свойствах которого изменим размер к удобному для использования вида.



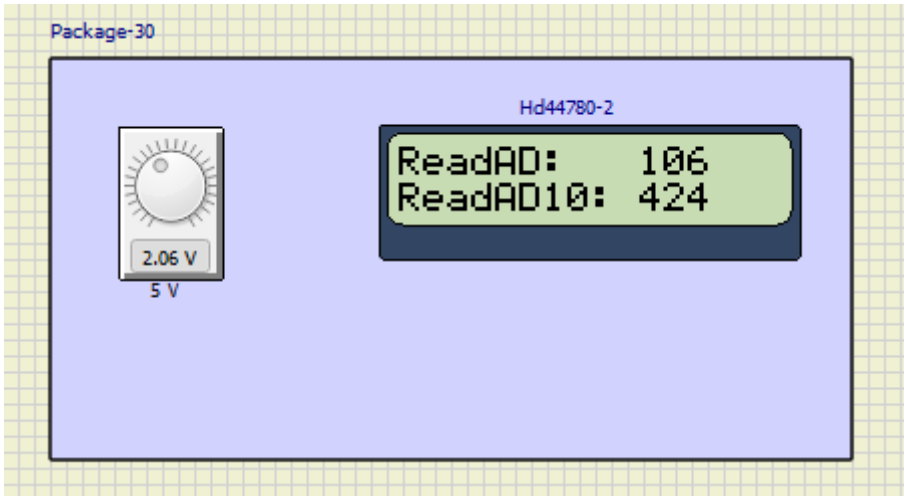
Щёлкнем правой клавишей мышки по выделенному пакету, чтобы выбрать режим платы.



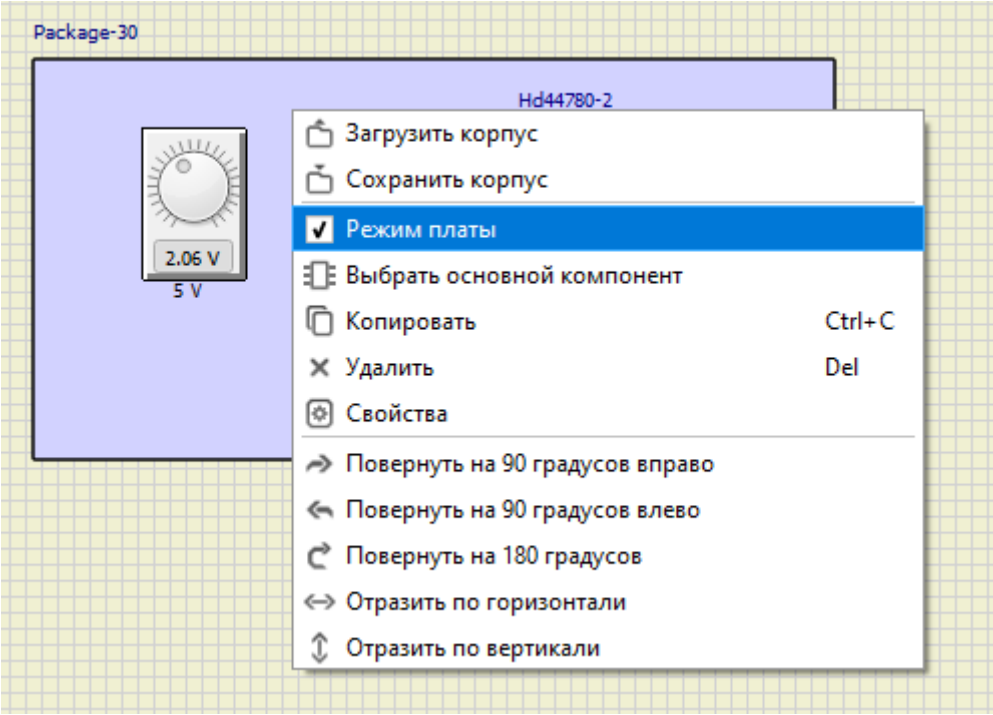
Отметим этот раздел и получим:



В этом режиме, можете проверить, схема работает – запустите моделирование:



Когда вы снимаете флажок «Режим платы», всё возвращается к «обычной жизни», показывая все компоненты в исходном положении.



Положение компонентов в одном режиме не влияет на позиции в другом режиме.

Кроме того, есть еще одна проблема, которую необходимо решить: свойства компонентов внутри подсхем недоступны.

Это может быть проблемой для таких компонентов, как микроконтроллеры, потому что вы не можете выбрать загрузку прошивки, последовательные порты и т. д.

Предлагаемое решение также находится в контекстном меню компонента пакета: «Выбрать основной компонент».

Выбрав эту опцию и щелкнув по любому компоненту в подсхеме, вы добавите контекстное меню и свойства этого компонента к подсхеме.

Основной компонент будет выделен желтым цветом.

Другое

О скорости моделирования

SimulIDE стремится стать симулятором схем в реальном времени.

Чтобы достичь этого, нам нужно найти компромисс между скоростью и точностью, чтобы моделирование можно было выполнять в режиме реального времени и поддерживать точность для правильного моделирования поведения цепи. Но это зависит от схемы, которую вы моделируете:

- Количества компонентов.
- Количества подключений.
- Типа компонентов.

Существует 3 типа компонентов, которые могут очень интенсивно загружать процессор компьютера и замедлять моделирование:

- **Нелинейные**: диоды, транзисторы, операционные усилители.
- **Реактивные**: конденсаторы, индукторы.

- **Светодиоды**: будучи диодами, они добавляют дополнительную перегрузку, вычисляя яркость.

Чтобы узнать, как ускорить светодиодные симуляции, загляните в раздел «Моделирование светодиодов».

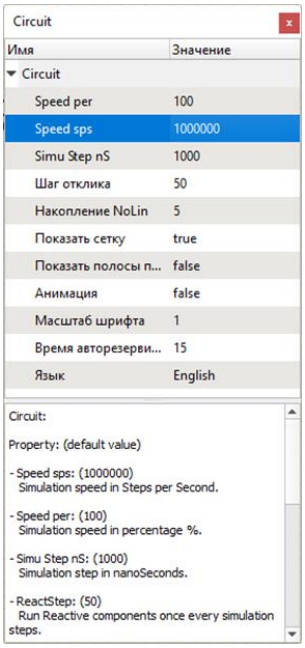
Теперь давайте сосредоточимся на скорости моделирования: основной цикл моделирования работает на частоте 1 МГц, 1e6 шагов в секунду, это эталонная скорость для всего.

Имеется два основных подцикла для нелинейных и реактивных компонентов.

Эти два подцикла выполняют каждый n основных шагов цикла. По умолчанию они выполняются:

- **Реактивный**: запускается один раз в 50 основных циклов (20 кГц).
- **Нелинейный**: запускается один раз в 10 основных циклов (100 кГц).

Обратите внимание, что они всегда выполняются относительно основного цикла. Щелкните правой клавишей мышки на любой пустой части схемы. Выберите из выпадающего меню раздел «Свойства».



Вы можете установить следующие три параметра:

- **Speed sps**: Скорость основного цикла с шагом в секунду: значение по умолчанию: 1.000.000, максимальное значение: 1.000.000, Мин Значение: 1

Если вы измените это значение, симуляция будет работать быстрее или медленнее.

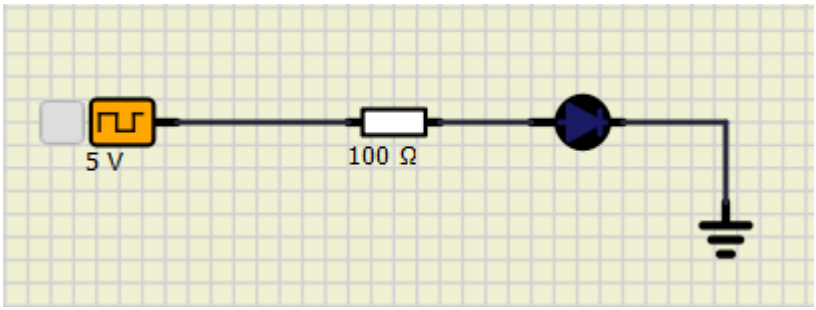
1.000.000 означает реальное время, 100.000 означает «замедленное движение», 10% от реального времени.

Чтобы понять это лучше: если у вас есть схема генератора, которая производит прямоугольные импульсы с частотой 1 Гц:

- При 1.000.000 вы увидите, что выход меняется каждые 1 секунду.
- При 500.000 вы увидите, что выход меняется каждые 2 секунды.
- При 100.000 вы увидите, что выход меняется каждые 10 секунд.

Таким образом, изменение этого значения полезно для просмотра моделирования, работающего в «замедленном движении».

Чтобы проверить это, можно собрать простую схему.



Частота генератора выбрана 1 Гц. Меняя параметр Speed sps, о чём написано выше, можно получить представление о поведении схемы при разных значениях этого параметра.

- **Шаг отклика**: количество шагов основного цикла для запуска реактивного подцикла, значение по умолчанию: 50, максимальное значение: 100, минимальное значение: 1.

Чем ниже это число, тем точнее моделирование реактивных компонентов, но потребуется больше процессорного времени, что может снизить производительность.

Но все зависит от количества реактивных компонентов в цепи и мощности вашего компьютера.

Вы должны изменить это значение при малой величине емкости или индуктивности.

Например: для генератора 50 кГц, использующего конденсатор, вы должны установить это

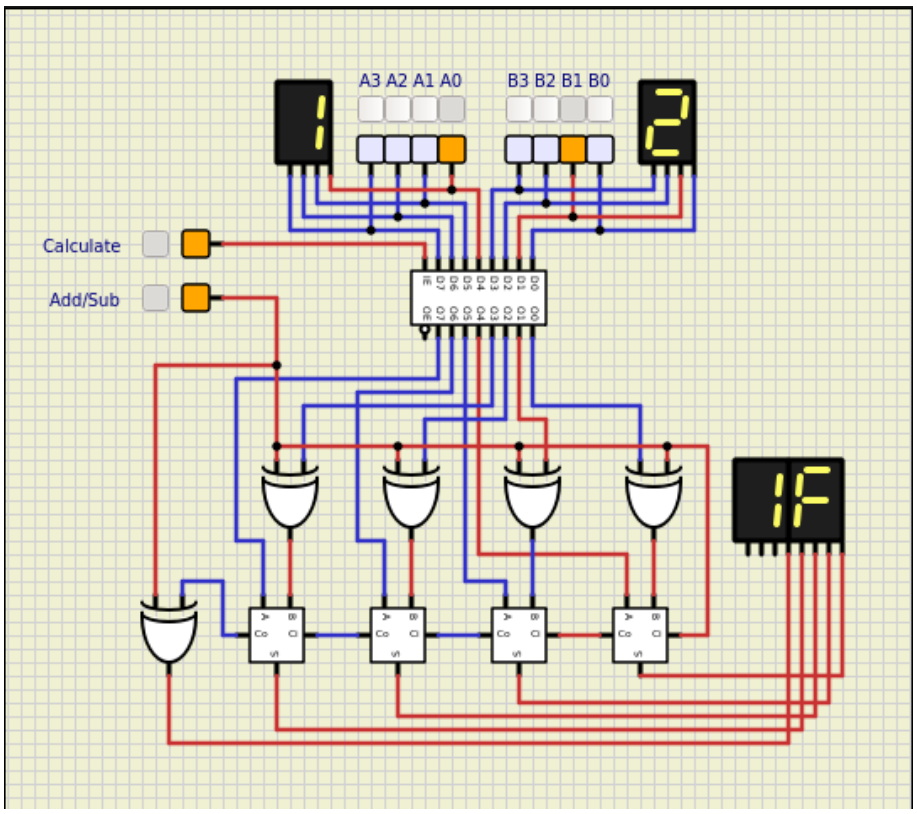
значение ниже 10, запуская реакцию каждые 10 шагов = 100 кГц.

- **Накопление NoLin**: точность нелинейных компонентов. В основном это наименьшая допустимая ошибка: NoLin = 5 означает ошибку +- 1*10e-5. Чем выше это значение, тем точнее моделирование, но может замедлить моделирование. Значение по умолчанию: 5, максимальное значение: 14, минимальное значение: 3.

- **Показать сетку**: отображается или скрывается сетка. Скрытая сетка может немного улучшить скорость моделирования

- **Анимация**: раскрасит провода, показывающие цифровые состояния (Полезно для цифровых схем). Красный для высокого состояния. Синий для низкого состояния.

Анимация схемы может замедлить скорость моделирования.



Диагональные соединения

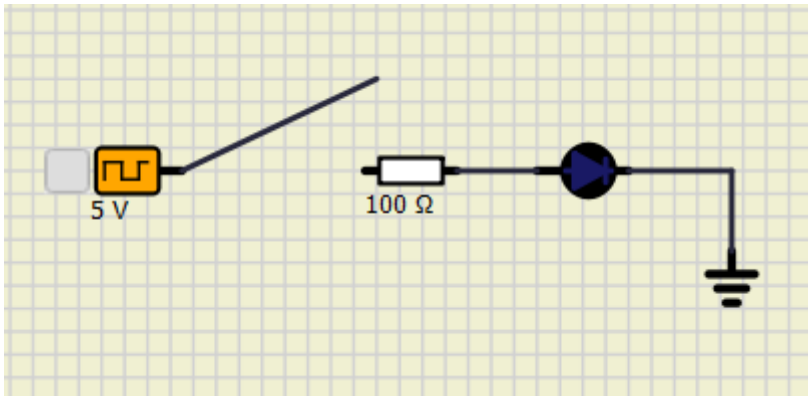
Это экспериментальная возможность, представленная в 0.3.11.

Можно рисовать любые виды диагональных проводов.

Как это сделать?

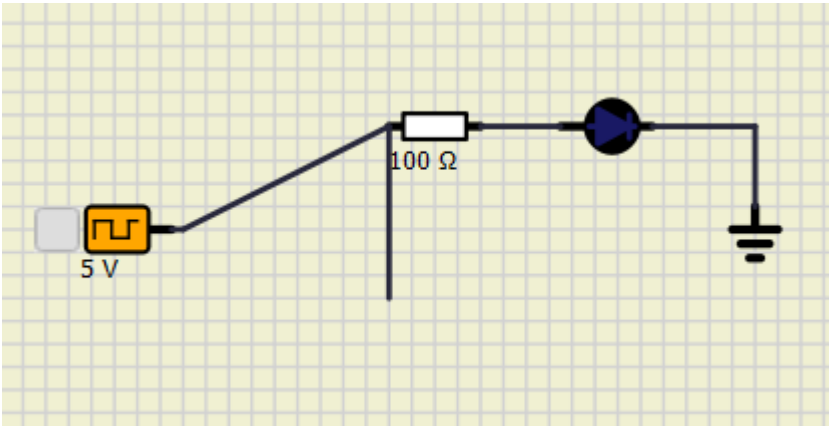
- **Отредактируйте** существующий провод:

- Поместите указатель мыши в угол провода.
- Нажмите клавишу «Shift» и переместите его.



- Сделайте диагональный провод, что удобно **при рисовании** разъема:

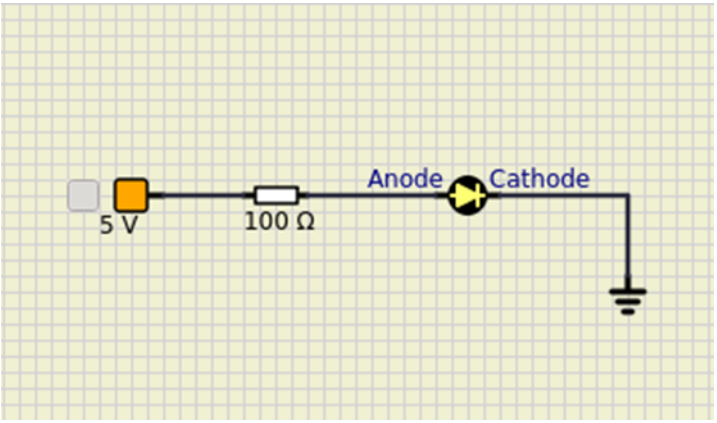
- При рисовании разъема нажмите клавишу "Shift".
- Теперь этот провод будет «свободен» от горизонтальных или вертикальных ограничений.
- Создайте нормальный угол, чтобы закончить диагональный режим.



Моделирование светодиодов

Простой светодиод:

Светодиоды представлены символом диода для идентификации анода и катода.

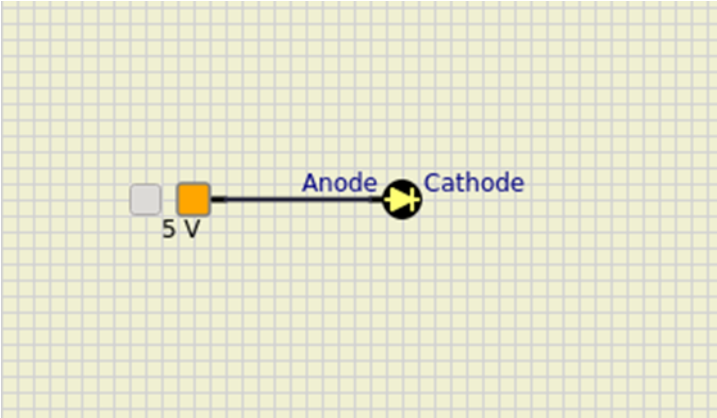


Выбрав компонент и открыв вкладку «Свойства», можно задать свойства светодиода:

- Цвет: цвет светодиода – желтый, красный, зеленый, синий, оранжевый или фиолетовый.
- **Порог**: пороговое напряжение диода, ниже этого напряжения он не будет светиться.
- **Максимальный ток:** поддерживаемый ток, выше этого значения светодиод сгорит.
- **Сопротивление: в**нутреннее сопротивление, в качестве трюка можно заменить внешний резистор.
- **Заземленный:** установка этого свойства позволяет избежать подключения катода к земле, что делает схему проще и чище.

Если для этого свойства задано значение true, контакт катода исчезает, а также соединение, если таковое имеется.

Например, если мы зададим значение «Сопротивление 100» и «Заземление» переведём в истинное значение, мы сможем сделать предыдущую схему проще, что может быть очень полезно в схемах с большим количеством светодиодов.



Светодиодная сборка:

Светодиодная сборка содержит массив из 8 небольших светодиодов, которые смещены к катоду. Те же свойства, что для светодиода, доступны для светодиодных сборок, например, с Сопротивление = 100 и Заземленный = true: