**Руководство пользователя RudironIDE**

**Содержание**

[**0.** **Введение** 2](#_Toc204644331)

[**1. Установка** 3](#_Toc204644332)

[1.1 Установка на Windows 3](#_Toc204644333)

[1.2 Установка на Linux 6](#_Toc204644334)

[**2. Подключение к плате** 9](#_Toc204644335)

[2.1 Подключение через USB порт 9](#_Toc204644336)

[2.2 Подключение через Bluetooth 11](#_Toc204644337)

[**3. Работа с файлами** 13](#_Toc204644338)

[Создание нового файла 13](#_Toc204644339)

[Открытие файла 14](#_Toc204644340)

[Сохранение файла 16](#_Toc204644341)

[Работа с вкладками 17](#_Toc204644342)

[**4. Категории блоков** 18](#_Toc204644343)

[1. Конструкции 18](#_Toc204644344)

[2. Операторы 19](#_Toc204644345)

[3. Переменные и значения 22](#_Toc204644346)

[4. Электроника 23](#_Toc204644347)

[5. Датчики 24](#_Toc204644348)

[**5. Рабочая область** 25](#_Toc204644349)

[Поле для блоков 25](#_Toc204644350)

[Масштабирование 26](#_Toc204644351)

[Центрирование 27](#_Toc204644352)

[Миникарта 28](#_Toc204644353)

[Горячие клавиши: 28](#_Toc204644354)

[Контекстное меню 28](#_Toc204644355)

[**6. Работа с блоками** 30](#_Toc204644356)

[Добавление блоков на рабочую область 30](#_Toc204644357)

[Cоединение и вложенность блоков 31](#_Toc204644358)

[Перемещение и отсоединение блоков 32](#_Toc204644359)

[Удаление блоков 33](#_Toc204644360)

[Копирование и вставка блоков 34](#_Toc204644361)

[Настройка параметров блоков 35](#_Toc204644362)

[Работа с переменными 35](#_Toc204644363)

[Расширение условий через шестерёнку 38](#_Toc204644364)

[Дополнительные действия через контекстное меню 40](#_Toc204644365)

[**7. Визуализация платы** 41](#_Toc204644366)

[Интерактивность 41](#_Toc204644367)

[Цветовая индикация 43](#_Toc204644368)

[**8. Монитор порта** 44](#_Toc204644369)

[Управление 45](#_Toc204644370)

[**9. Выполнение программ** 46](#_Toc204644371)

[Пошаговая инструкция запуска: 47](#_Toc204644372)

[**10. Примеры программ** 48](#_Toc204644373)

# **Введение**

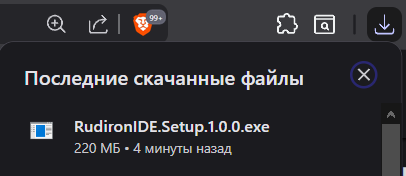
Rudiron IDE — это интегрированная среда разработки для визуального программирования микроконтроллера Рудирон от компании Aquarius. Приложение предоставляет интуитивно понятный интерфейс для создания программ с помощью блочного программирования для взаимодействия с аппаратной платформой Рудирон, обеспечивая полный цикл программирования от создания кода до его выполнения на контроллере.

# **1. Установка**

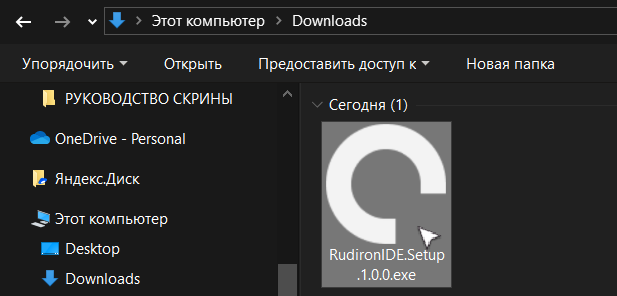
## 1.1 Установка на Windows

1. Перейдите на страницу релизов проекта на GitFlic: https://gitflic.ru/project/akvarius-rudiron/aqrudiron-ide/release

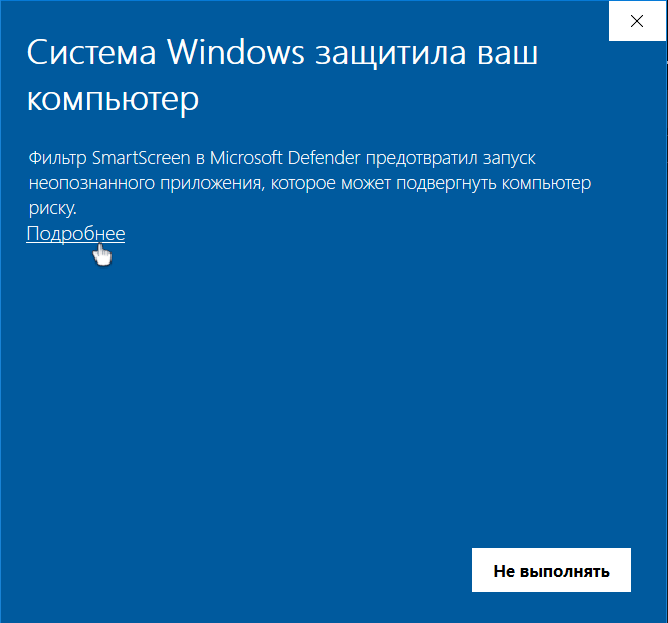
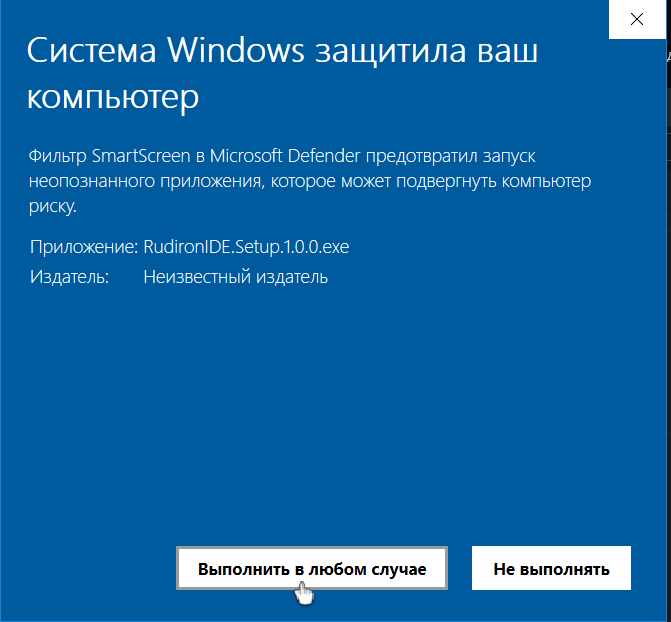
2. Найдите последнюю версию приложения и скачайте установочный файл с расширением .exe (например, RudironIDE-Setup.exe).



3. Запустите скачанный файл двойным щелчком мыши.



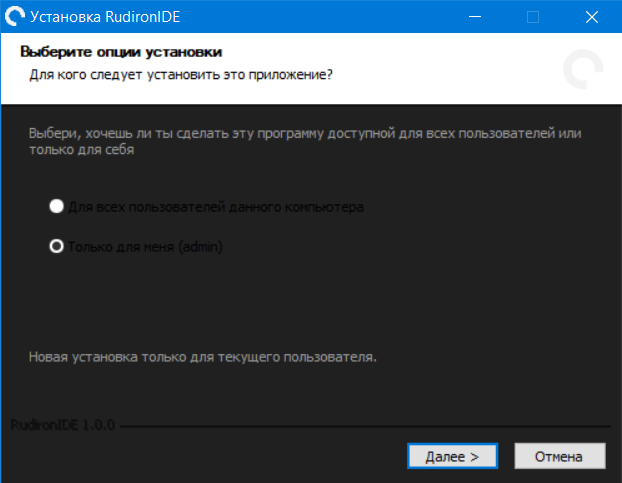
4. Если при запуске появляется предупреждение SmartScreen, выберите «Подробнее» → «Выполнить в любом случае».



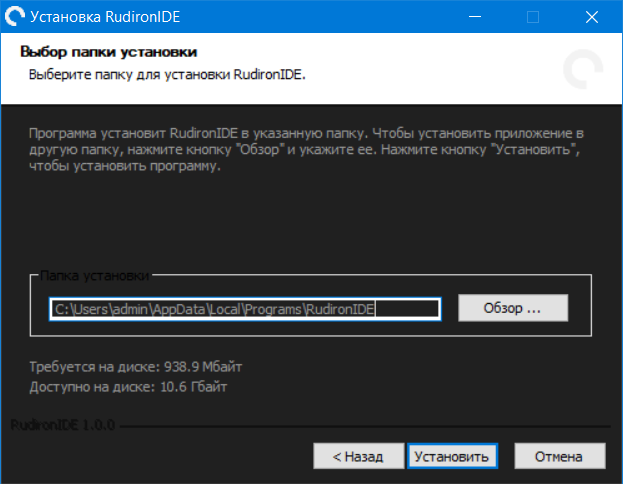


5. В появившемся окне следуйте инструкциям установщика:

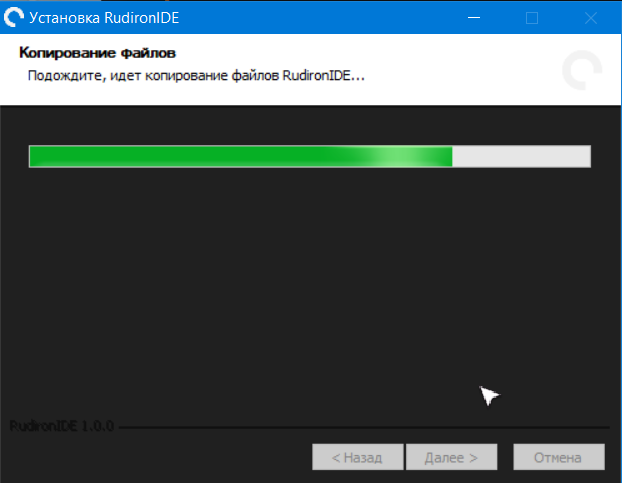
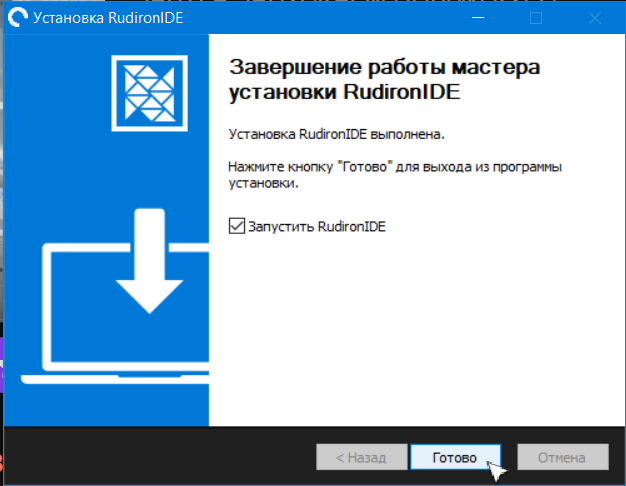
* Укажите пользовательские права доступа



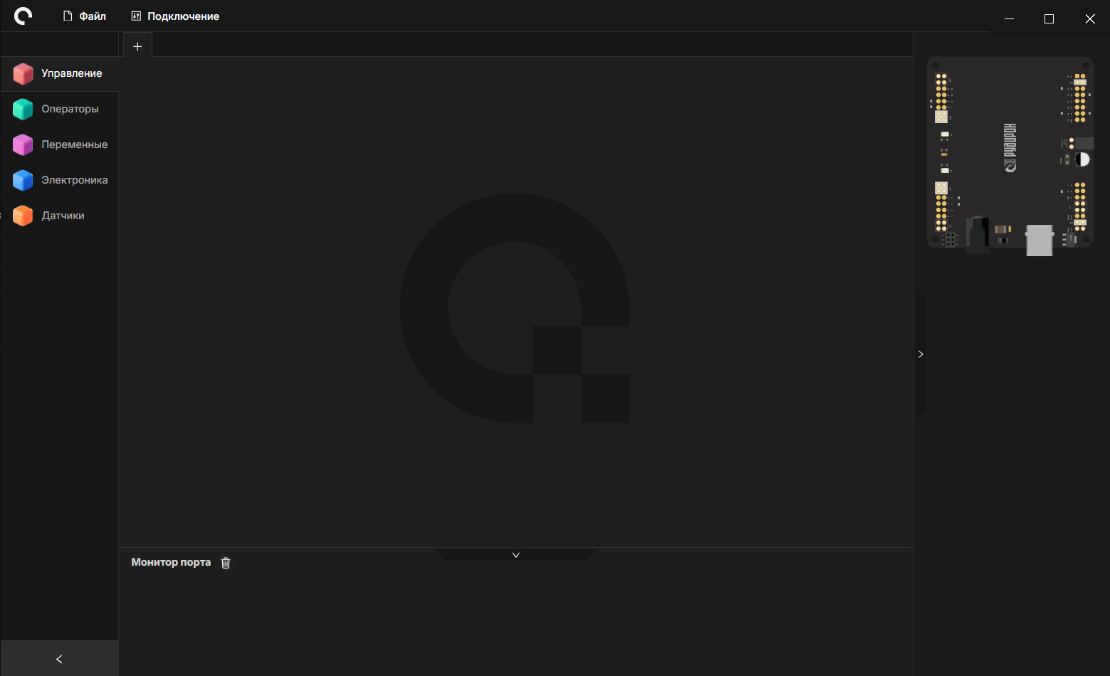
* При необходимости выберите папку для установки



* Дождитесь завершения процесса установки



5. После завершения установки на рабочем столе появится ярлык Rudiron IDE (если вы выбрали соответствующую опцию).

6. Запустите приложение через ярлык или из меню «Пуск».

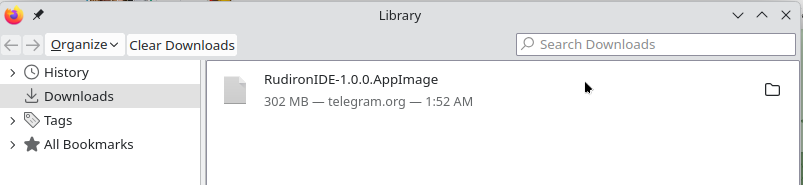
Примечание:

Для работы программы не требуется установка дополнительных драйверов для большинства плат Rudiron. Если плата не определяется, скачайте и установите драйверы с официального сайта производителя платы.

## 1.2 Установка на Linux

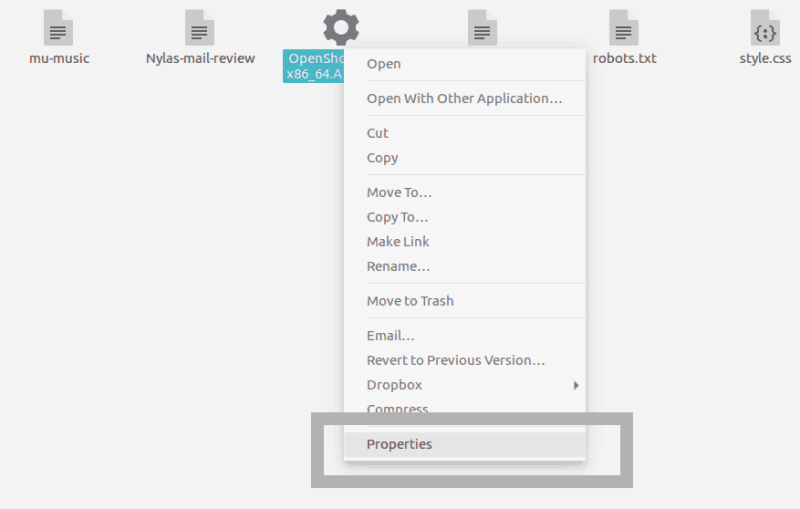
1. Перейдите на страницу релизов проекта RudironIDE на GitFlic: <https://gitflic.ru/project/akvarius-rudiron/aqrudiron-ide/release>

2. Найдите последнюю версию приложения и скачайте файл с расширением .AppImage (например, RudironIDE-1.0.0.AppImage). AppImage — это универсальный формат, который позволяет запускать приложение без установки на всех дистрибутивах Linux.

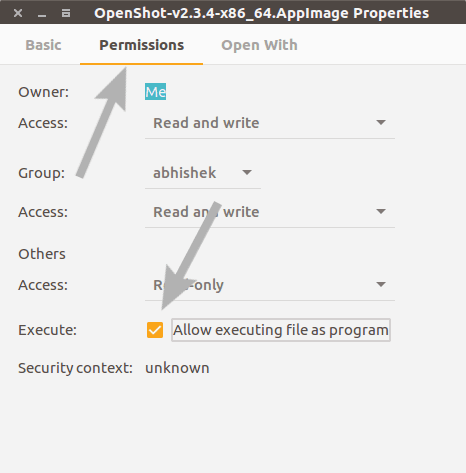


3. После окончания загрузки файла по умолчанию он не имеет разрешения на выполнение. Вам нужно будет изменить права доступа к файлу, чтобы сделать его исполняемым. Для этого вам не нужны права root.

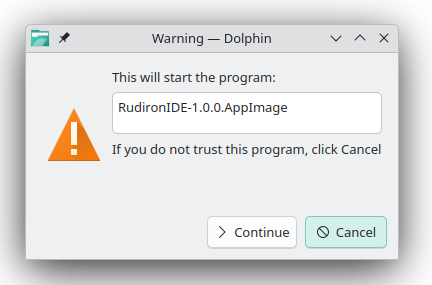
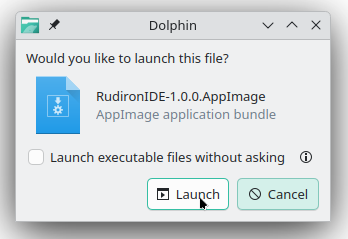
Если вы предпочитаете графический способ, просто щелкните правой кнопкой мыши на загруженный файл .Appimage и выберите «Свойства» из контекстного меню.



На следующем экране перейдите на вкладку «Разрешения» и установите флажок «Разрешить выполнение файла как программы».



В качестве альтернативы, если вы предпочитаете командную строку, вы можете просто использовать chmod +x RudironIDE-1.0.0.AppImage в bash терминале, чтобы сделать его исполняемым.

Примечание: инструкция может отличаться в зависимости от ОС, например, в Mangaro при двойном клике на файл, сразу появляется модальное окно для выбора исполнения, без необходимости копания в параметрах.

4. После этого достаточно просто запустить файл — дважды щелкнуть по файлу или же запустить из командной строки, например:

*```bash*

*./RudironIDE-1.0.0.AppImage*

*```*

5. Решение проблем на linux

Если AppImage не запускается:

1. Проверьте права доступа

*```bash*

*ls -la RudironIDE-1.0.0.AppImage*

*# Должно быть: -rwxr-xr-x или -rwxr--r--*

*```*

2. Установите необходимые библиотеки

*```bash*

*sudo apt update*

*sudo apt install libfuse2*

*#Ubuntu*

*```*

3. Запустите с отладкой

*```bash*

*./RudironIDE-1.0.0.AppImage --verbose*

*```*

4. Проверьте архитектуру

*```bash*

*uname -m*

*file RudironIDE-1.0.0.AppImage*

*# Должно быть x86\_64 для 64-bit*

*```*

# **2. Подключение к плате**

Рекомендуется перед запуском приложения подключить плату Рудирон к компьютеру, но это не обязательно.

## 2.1 Подключение через USB порт

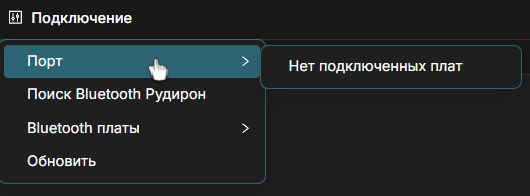
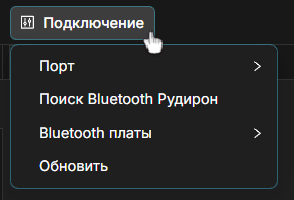
1. Подготовка платы

* Убедитесь, что плата Рудирон подключена к компьютеру через USB кабель
* Проверьте, что на плате горят красные светодиоды питания
* Убедитесь, что плата находится в режиме питания по USB



2. Открытие меню подключения

В заголовочной панели приложения найдите меню "Подключение" и кликните на него левой кнопкой мыши

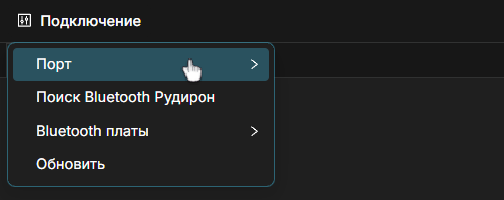
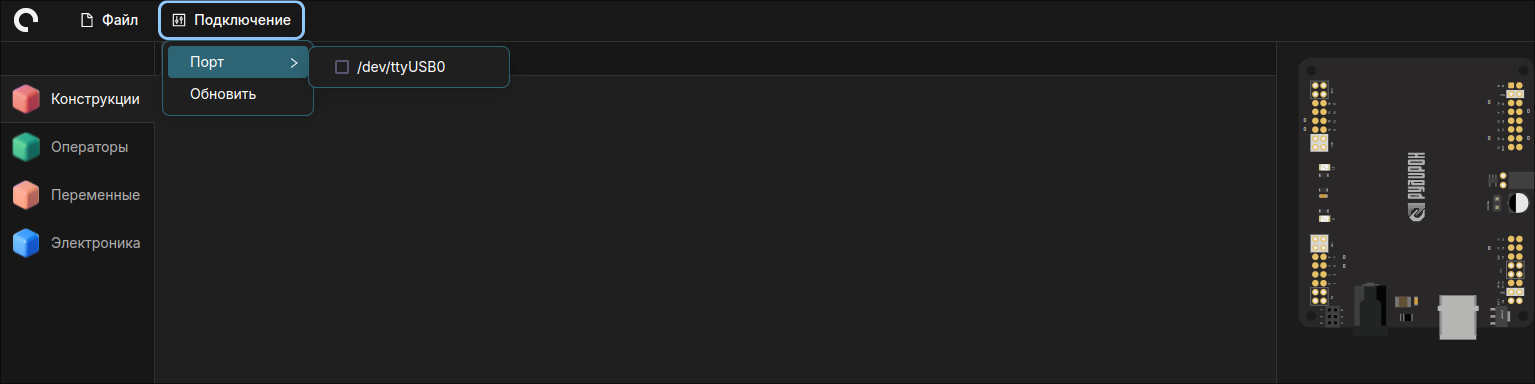




1. Выбор порта

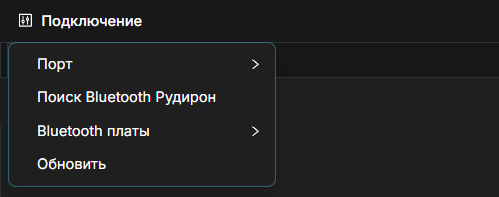
В выпадающем списке вы увидите доступные порты:

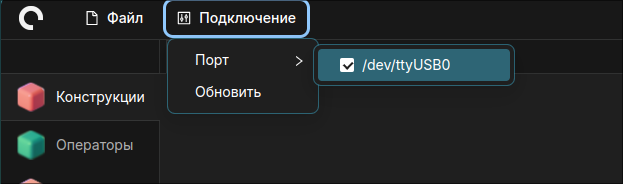
* На Windows порты имеют названия вида `COMx` (например, COM3, COM4)
* На Linux порты имеют названия вида `/dev/ttyX` (например, /dev/ttyUSB0)



4. Подключение к плате

Выберите порт, к которому подключена ваша плата Рудирон, кликнув на него в списке. При успешном подключении рядом с портом появится галочка:





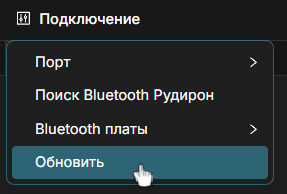
5. Проверка подключения

В мониторе порта (нижняя панель) появится сообщение о подключении:” [IDE] Подключено к плате Рудирон на порту: XXXXX”.

Если плата не отображается:

1. Обновите список портов

* Нажмите кнопку "Обновить" в меню подключения
* Подождите несколько секунд



2. Проверьте физическое подключение

* Убедитесь, что USB кабель надежно закреплен
* Попробуйте другой USB порт
* Проверьте, что кабель исправен

3. Убедитесь, что плата в нормальном рабочем режиме и не находится в режиме JTAG отладки



4. Проверьте, что плата действительно определяется вашей системой. На Windows - через диспетчер устройств (тип устройства `COM` порты). На Linux через терминальную команду `lsusb`. Рудирон платы будут иметь следующие `VID=1a86` и `PID=55d4`

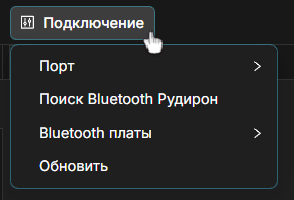
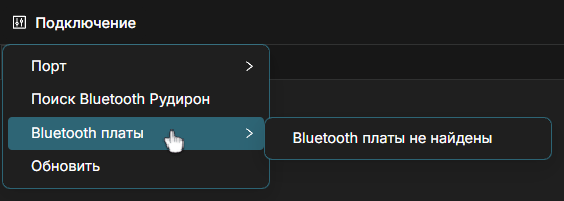
## 2.2 Подключение через Bluetooth

1. Подготовка Bluetooth модуля

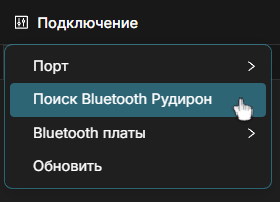
Убедитесь, что к плате подключен Bluetooth модуль HC-06 и он готов к работе.

2. Открытие меню подключения

В заголовочной панели приложения найдите меню "Подключение" и кликните на него левой кнопкой мыши



3. Поиск Bluetooth устройств

В меню подключения выберите пункт "Поиск плат по Bluetooth"

В мониторе порта появится информация о процессе поиска

Примечание: в *Linux* перед поиском устройств необходимо ввести команду в терминал для работы Bluetooth (ровно один раз, только при первом подключение)

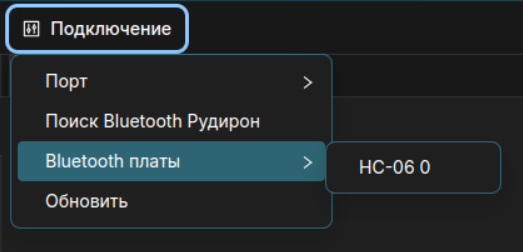
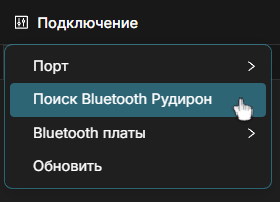
```bash

sudo setcap cap\_net\_raw+eip $(eval readlink -f `which node`)

```

4. Выбор платы

В разделе "Bluetooth платы" появится список найденных устройств, найдите плату с модулем Bluetooth HC-06 и кликните по ней



5. Подключение

После выбора платы в мониторе порта появится информация о подключении:

# **3. Работа с файлами**

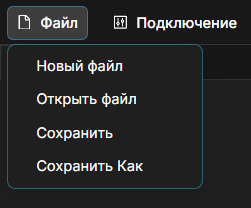
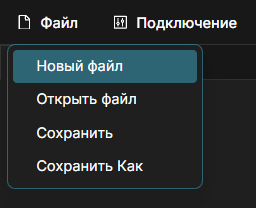
## Создание нового файла

1 способ:

1. Нажмите на меню "Файл" в заголовочной панели

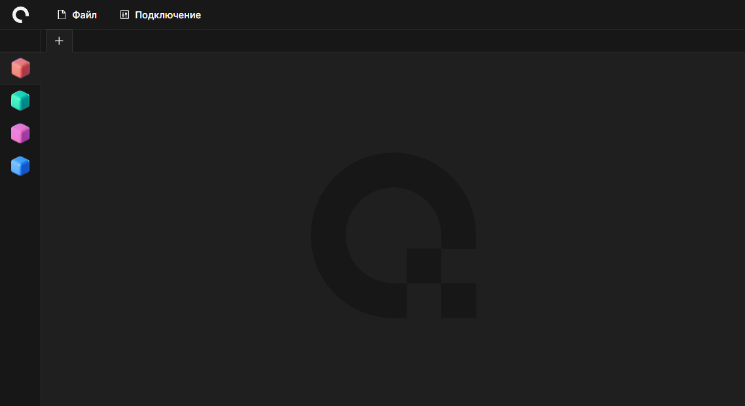
****

2. Выберите "Новый файл"

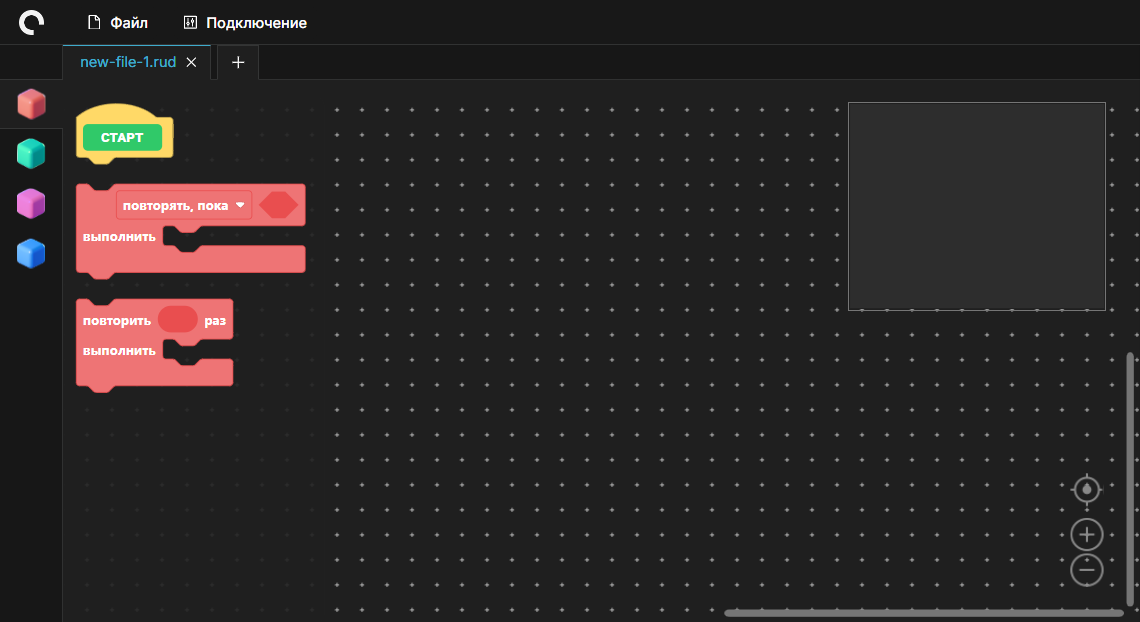


🡪

3. Откроется новая вкладка с пустой рабочей областью с базовым названием new-file-Число.rud

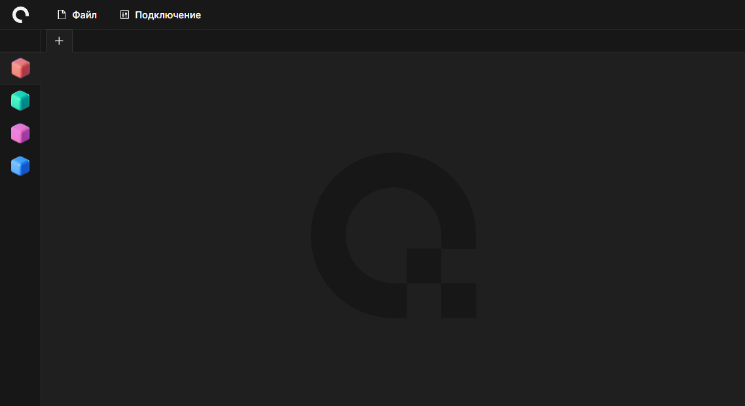


⭣

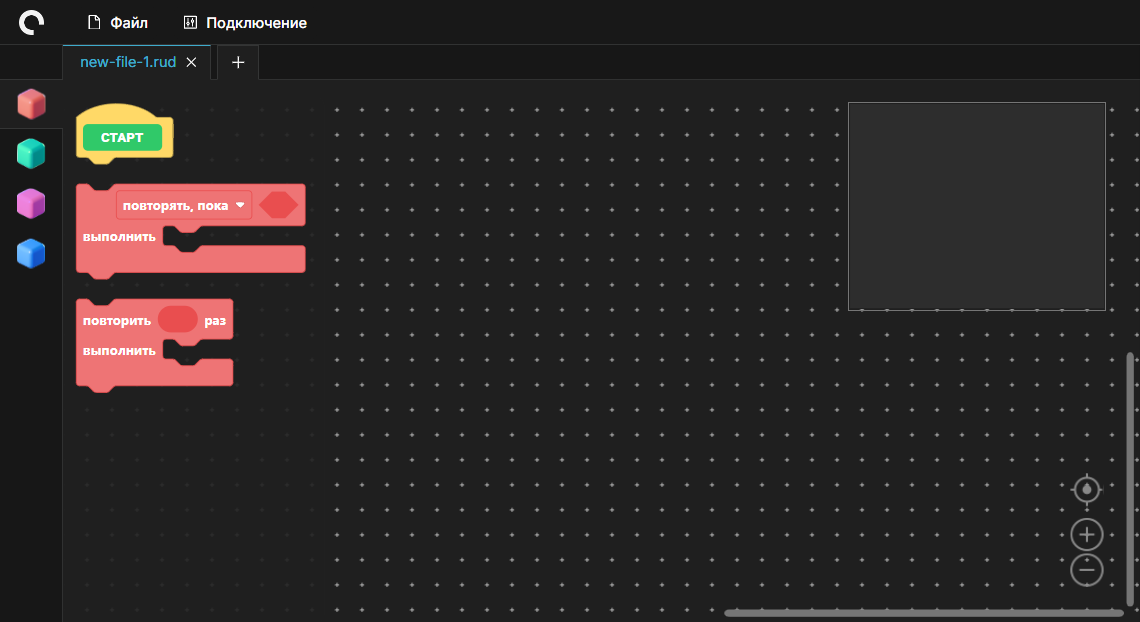


2 способ:

1. Нажмите на кнопку плюс в панели файлов
2. Откроется новая вкладка с пустой рабочей областью с базовым названием new-file-Число.rud



⭣

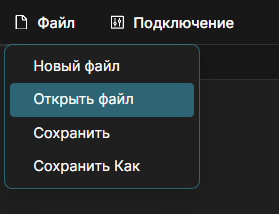


## Открытие файла

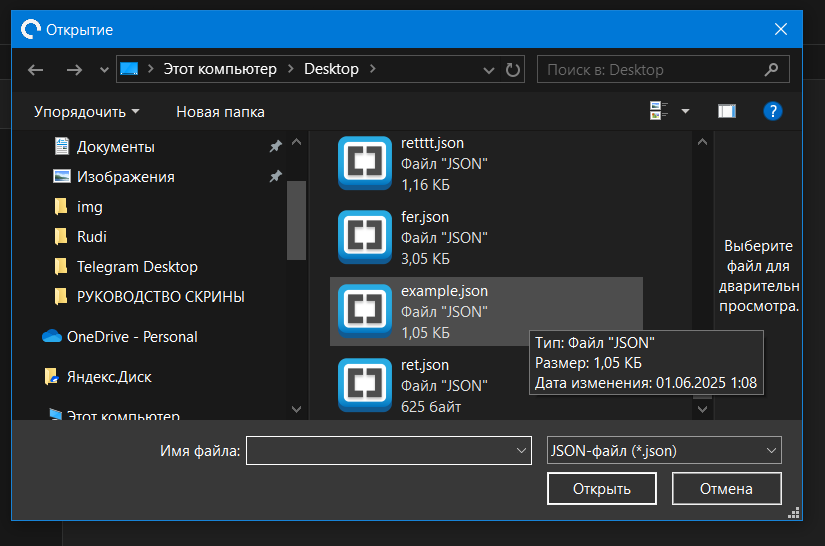
1. Нажмите на меню "Файл" в заголовочной панели

****

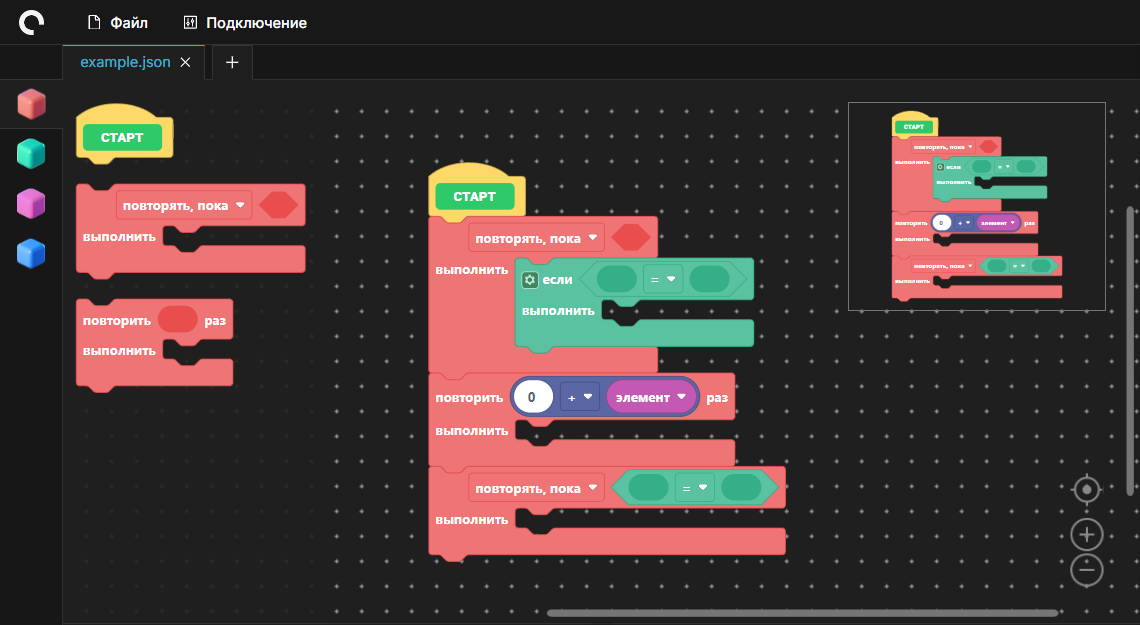
2. Выберите "Открыть файл"



3. В открывшемся диалоговом окне выберите файл проекта (.json)

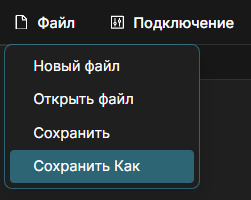


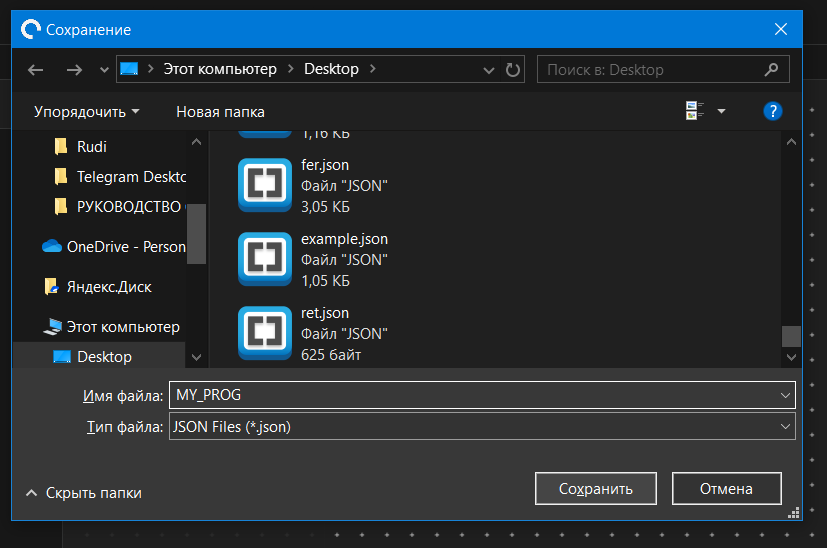
4. Проект загрузится в новую вкладку с сохраненным именем файла



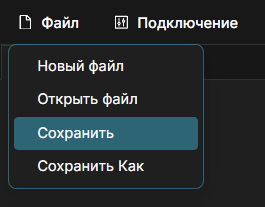
## Сохранение файла

****1. Нажмите на меню "Файл"

2. Выберите "Сохранить как", если хотите указать директорию сохранения и имя файла, для этого появиться диалоговое окно проводника

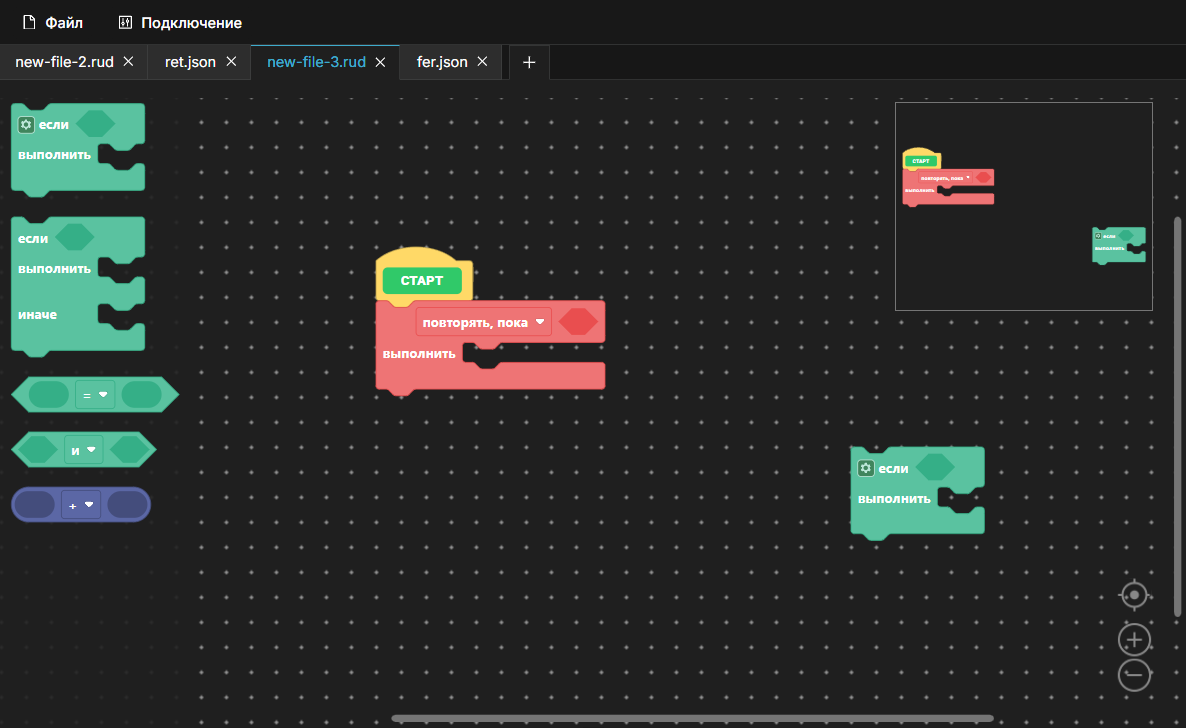
⭣

3. Выберите “Сохранить”, если хотите сохранить обновления, которые внесли в код.



Примечание: две опции будут работать как “Сохранить как” при первом сохранение файла.

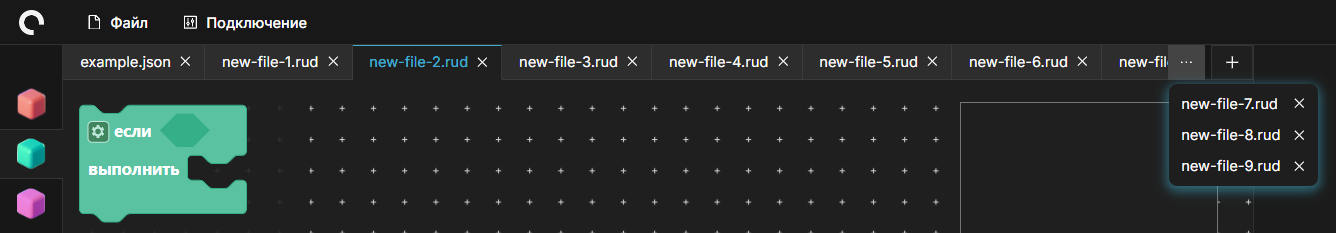
## Работа с вкладками

* Каждый открытый файл отображается в отдельной вкладке
* Для переключения между файлами кликните на нужную вкладку





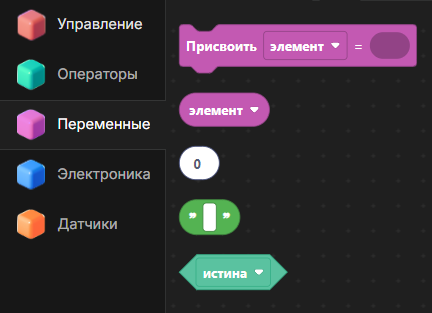
* Для закрытия вкладки нажмите на крестик справа от названия
* При переполнении панели файлов (при большом кол-во открытых вкладок) появляется кнопка троеточие, наведя на которую, можно посмотреть все названия скрытых файлов в отдельном модальном окне и перейти к ним.

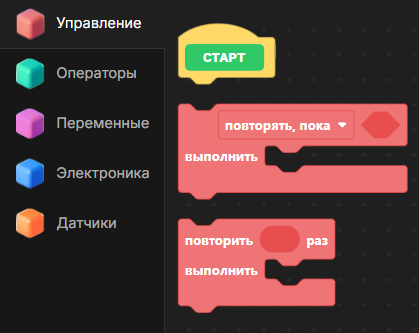
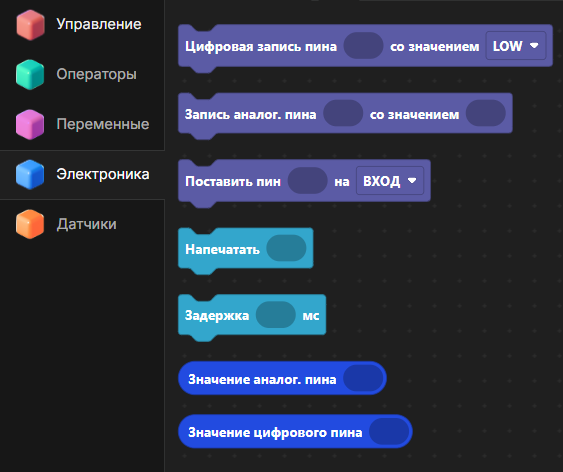


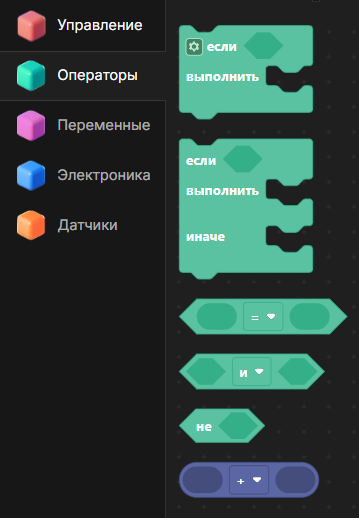
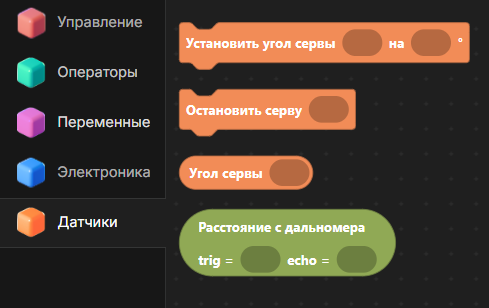


Альтернативный вариант, использовать прокрутку вкладок через колесико мыши, прежде наведясь в эту область:

# **4. Категории блоков**

В левой панели доступны 5 категории блоков:

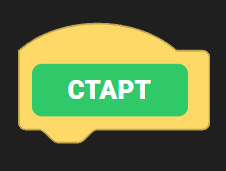




## 1. Конструкции

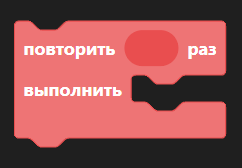
Содержит блоки для управления потоком выполнения

* *Старт* — блок запуска и остановки программы



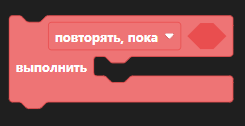
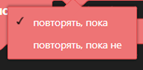
Входные данные: не принимает входных данных, служит началом программы. Если удалить этот блок, то все блоки, присоединённые к нему, также будут удалены.

* *Повторить N раз* — цикл с заданным количеством повторений



Входные данные: числовое значение (количество повторов). Поддерживает вложенность в тело цикла. Используется для повторяющегося выполнения блока кода определенное количество раз.

* *Повторять, пока* / *Повторять, пока* не — цикл с условием

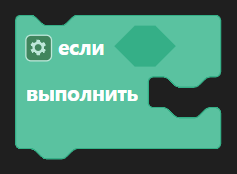


Входные данные: логическое выражение (условие). Поддерживает вложенность в тело цикла. Используется для выполнения блока кода до тех пор, пока условие не станет ложным / не станет истинным.

## 2. Операторы

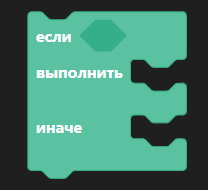
Содержит логические и математические операторы

* *Если* — условный оператор



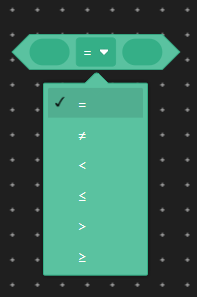
Входные данные: логическое выражение (условие). Поддерживает вложенность в тело оператора. Используется для выполнения блока кода только в случае, если выражение истинно.

* *Если-иначе* — условный оператор с альтернативной веткой



Входные данные: логическое выражение (условие). Поддерживает вложенность в обе ветви (действия при выполнении и невыполнении условия). Используется для выполнения определенного блока кода в случае, если выражение истинно, и альтернативного блока кода в случае, если ложно. Информация о дополнительных ветках описана в главе “Расширение условий через шестерёнку”

* *Сравнение* — сравнение двух значений (=, ≠, >, <, ≥, ≤)



Входные данные: два числовых, строковых или логических значения для сравнения. Используется для проверки отношений двух значений. Возвращаемое значение: истина или ложь.

* *Логические операторы* — И, ИЛИ



Входные данные: два логических значения. “И” используются для проверки, истинны ли обе операции одновременно, “ИЛИ” для проверки, истинно ли хотя бы одно из логических выражений. Возвращаемое значение: истина или ложь.

* Логический оператор НЕ



Входные данные: одно логическое значение. Используется для инвертирования логического значения — если на вход подаётся "истина", то результат будет "ложь", и наоборот. Возвращаемое значение: истина или ложь.

* *Математические операторы* — сложение, вычитание, умножение, деление

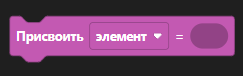


Входные данные: два числовых значения. Используется для выполнения математических операций с ними. Возвращаемое значение: Числовое значение

## 3. Переменные и значения

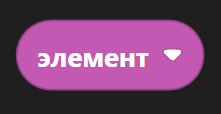
Содержит блоки для работы с данными

* *Присвоить* — присваивания значения переменной.



Входные данные: первое поле - имя переменной (выбор из списка), второе поле - значение (число, текст, логическое значение).

* *Переменная* — использование значения переменной



Входные данные: имя переменной (выбор из списка). Возвращаемое значение: число, текст, логическое значение (значение переменной)

* *Число*

**

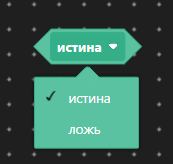
Входные данные: числовое значение (ввод вручную).

* *Текст*

**

Входные данные: строковое значение (ввод вручную).

* *Логическое значение*

**

Входные данные: Истина или ложь (выбор из списка).

## 4. Электроника

Содержит блоки для работы с платой

* *Цифровая запись* — установка значения на цифровом пине



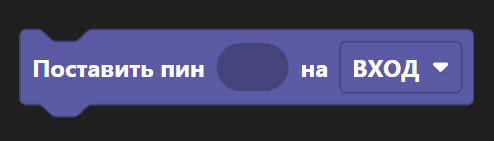
Входные данные: числовое значение (номер пина), значение HIGH или LOW (выбор из списка).

* *Аналоговая запись* — установка значения на аналоговом пине



Входные данные: числовое значение (номер пина), значение (число от 0 до 255).

* *Поставить пин* — настройка режима пина



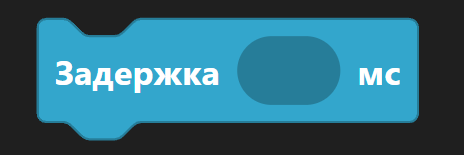
Входные данные: числовое значение (номер пина), режим - ВХОД, ВЫХОД, ПОДТЯГИВАНИЕ (выбор из выпадающего списка).

* *Напечатать* — вывод текста в монитор порта



Входные данные: число, строка, логическое выражение или переменная

* *Задержка* — пауза в миллисекундах



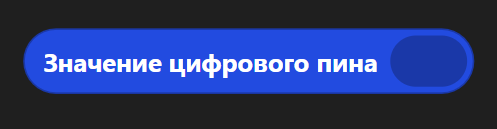
Входные данные: числовое значение

* *Значение аналог. пина* — чтение значения с аналогового пина

Входные данные: числовое значение (номер пина)

Возвращаемое значение: Значение аналог. пина

* *Значение цифр. пина* — чтение значения с цифрового пина



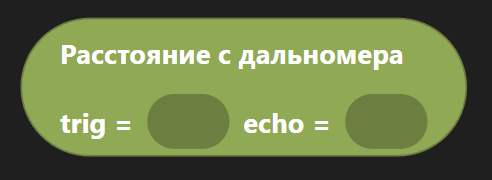
Входные данные: числовое значение (номер пина)

Возвращаемое значение: Значение цифр. пина

## 5. Датчики

Содержит блоки для работы с датчиками и сервоприводами

* *Расстояние с дальнометра* — получение расстояния с дальнометра



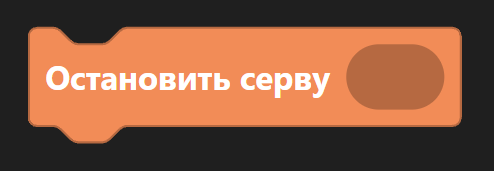
Входные данные: числовое значение (trig пин) и числовое значение (echo пин), подключённые к плате. Используется для измерения расстояния с помощью ультразвукового датчика. Возвращаемое значение: числовое значение (расстояние)

* *Угол серво* — получение текущего положения (угла) на сервоприводе



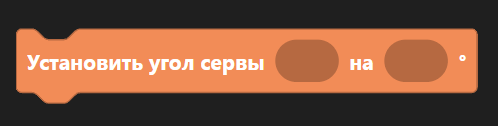
Входные данные: числовое значение (пин сервы). Возвращаемое значение: числовое значение (угол в градусах).

* *Остановить серву* — остановить сервопривод



Входные данные: числовое значение (пин сервы).

* *Установить угол сервы* — повернуть сервопривод на заданный угол.



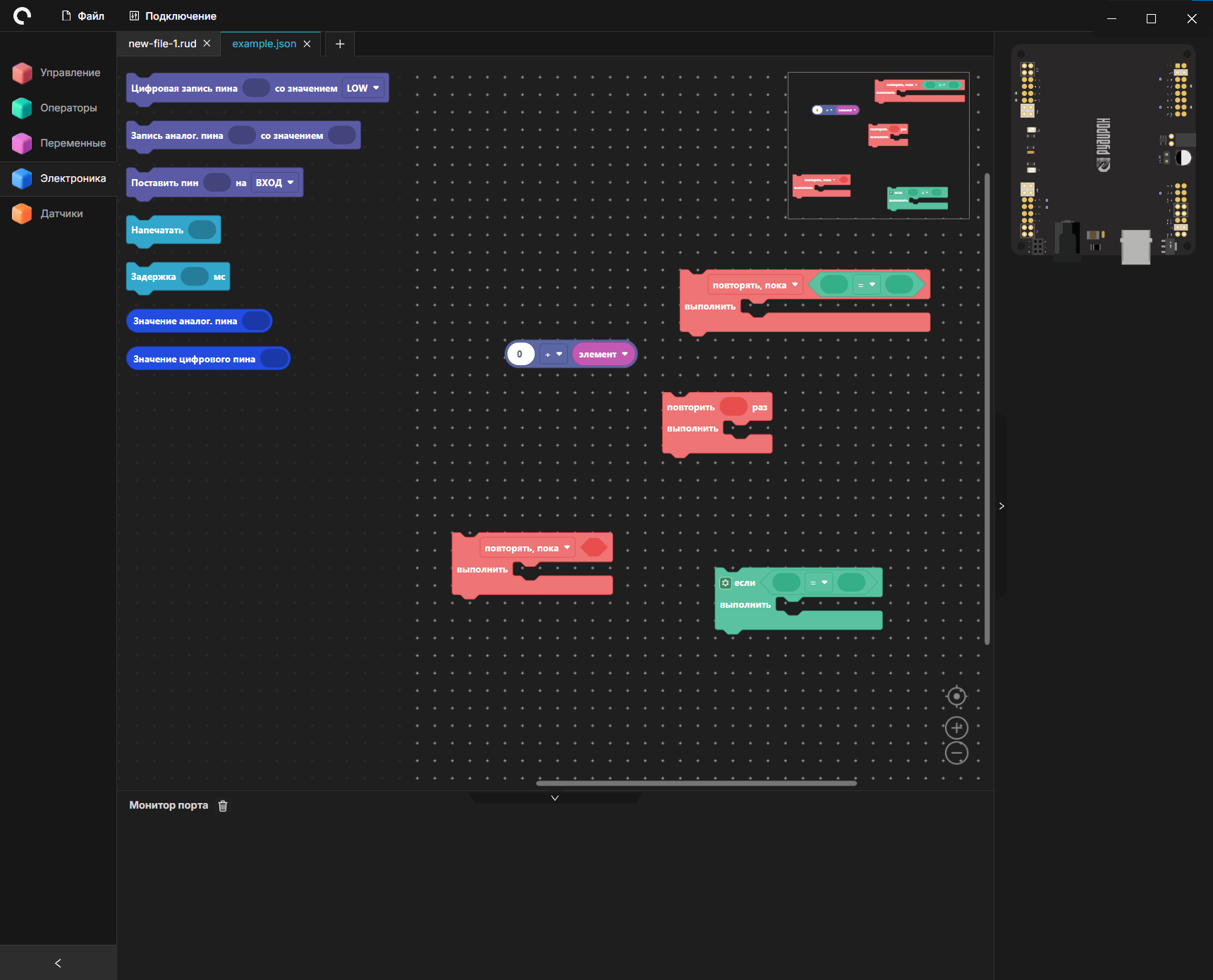
Входные данные: числовое значение (пин сервопривода) и числовое значение (угол в градусах).

# **5. Рабочая область**

Рабочая область — это центральная часть интерфейса, где происходит основная работа с блоками. Здесь вы можете создавать, соединять, перемещать и удалять блоки, формируя алгоритм вашей программы. Основные элементы рабочей области:

## Поле для блоков

Именно сюда перетаскиваются блоки из панели категорий. Их можно свободно перемещать по всей рабочей области — они не ограничены размерами экрана и могут располагаться в любом месте поля. Если блоки приближаются к границам видимой области, рабочая область автоматически расширяется, обеспечивая бесконечное пространство для размещения новых блоков. Передвигаться по полю можно двумя способами: с помощью горизонтального и вертикального ползунка или зажав левую кнопку мыши на пустом месте рабочей области и перемещая курсор.

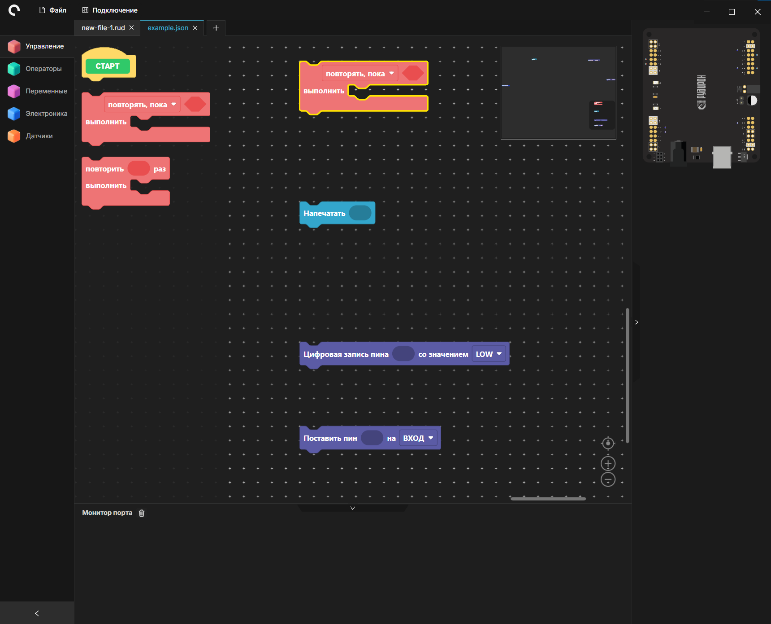
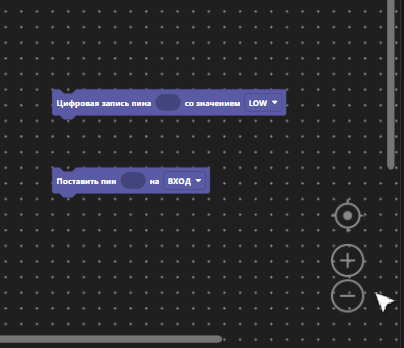




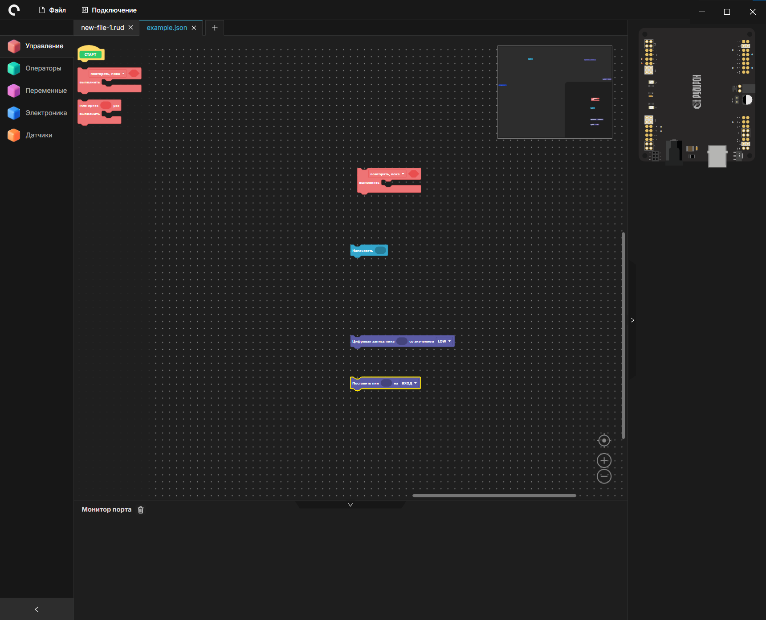
Позиция скролла сохраняется для каждого файла: если вы переключитесь на другую вкладку и вернётесь обратно, то будет видна ровно та часть рабочей области, которую вы оставили.

## Масштабирование

В правом нижнем углу расположены кнопки и для увеличения и уменьшения масштаба рабочей области и блоков левой панели. Также его можно изменять с помощью колесика мыши, наведясь на поле.



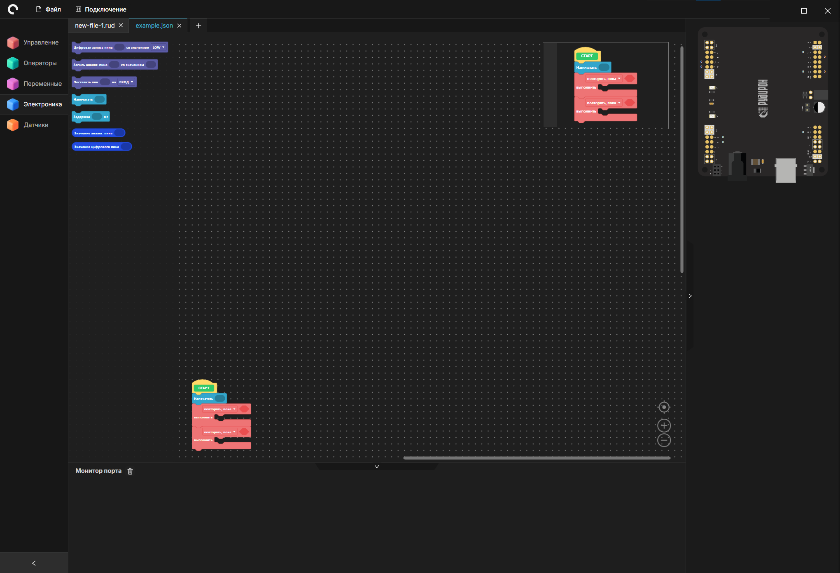






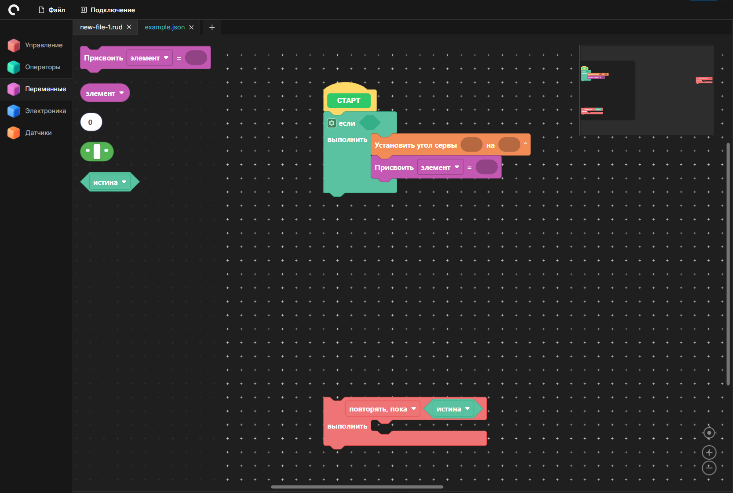
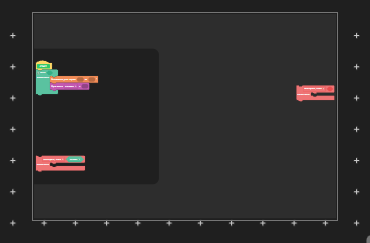
Масштаб, выставленный для каждого файла, сохраняется автоматически: при переключении между вкладками для каждого из них он будет восстановлен.

## Центрирование

Кнопка "Центрировать" перемещает область видимости в центр рабочей области и возвращает ее базовый масштаб.

## Миникарта

В правом верхнем углу отображается миникарта, на которой видно расположение всех блоков. В ней выделена рамка (темно-серая) — это область видимости, которая показывает, в какой части большого рабочего поля вы сейчас находитесь. Это помогает быстро сориентироваться в масштабных проектах.





Для передвижения по рабочей области можно зажать левую кнопку мыши на миникарте и перемещать рамку — видимая часть рабочего поля будет следовать за движением курсора.

## Горячие клавиши:

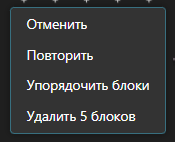
Кликните ЛКМ на рабочую область и нажмите клавиши:

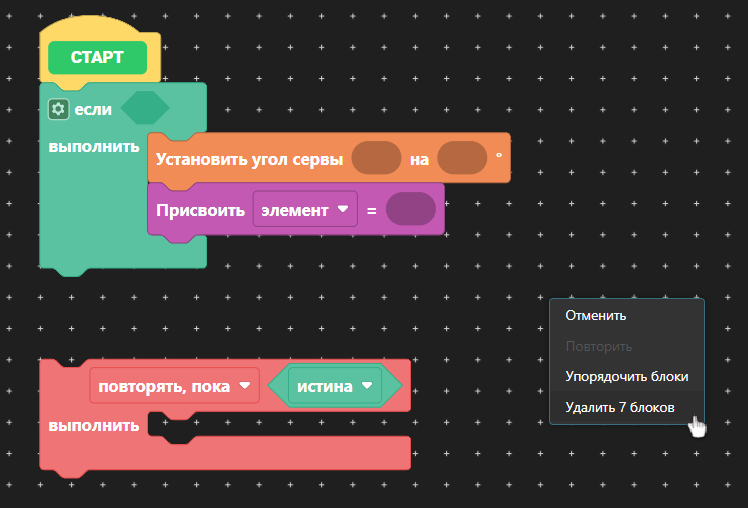
Ctrl+Z — отменяет последнее действие

Ctrl+Y — возвращает отменённое действие

## Контекстное меню

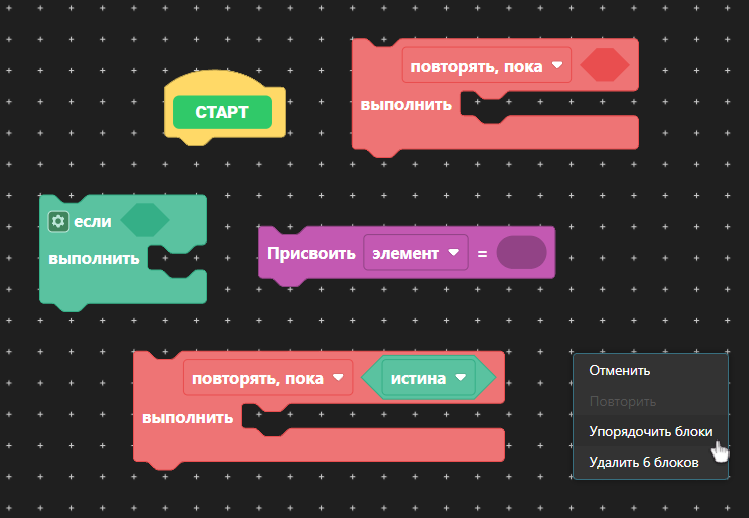
При нажатии правой кнопкой мыши (ПКМ) на пустую область рабочей зоны открывается контекстное меню с дополнительными функциями:



1. "Удалить все блоки" – полностью очищает рабочую область от всех блоков.

2. "Упорядочить все блоки" – выстраивает блоки друг под другом в один столбец, если они были разбросаны по рабочей области.



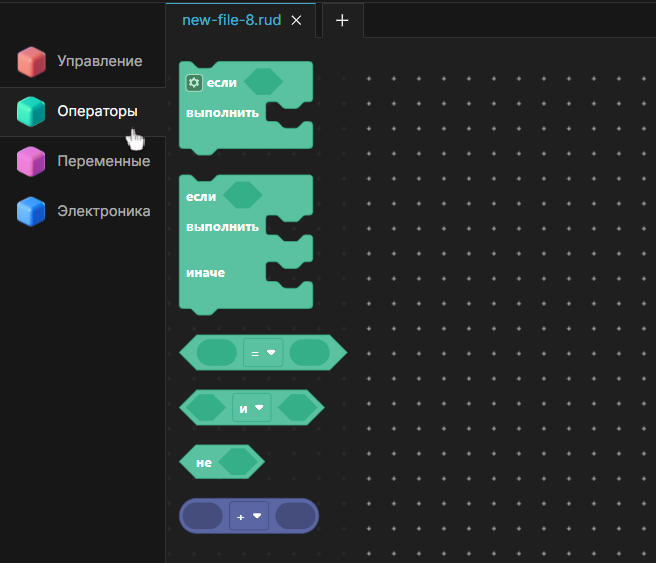


3. "Отменить" – возвращает рабочую область на один шаг назад, отменяя последнее действие (аналог Ctrl+Z)

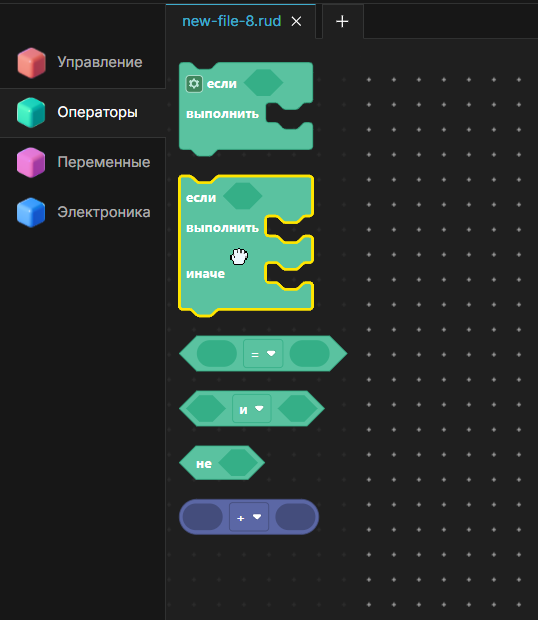
4. "Повторить" – Возвращает отменённое действие, двигаясь на шаг вперёд (аналог Ctrl+Y)

# **6. Работа с блоками**

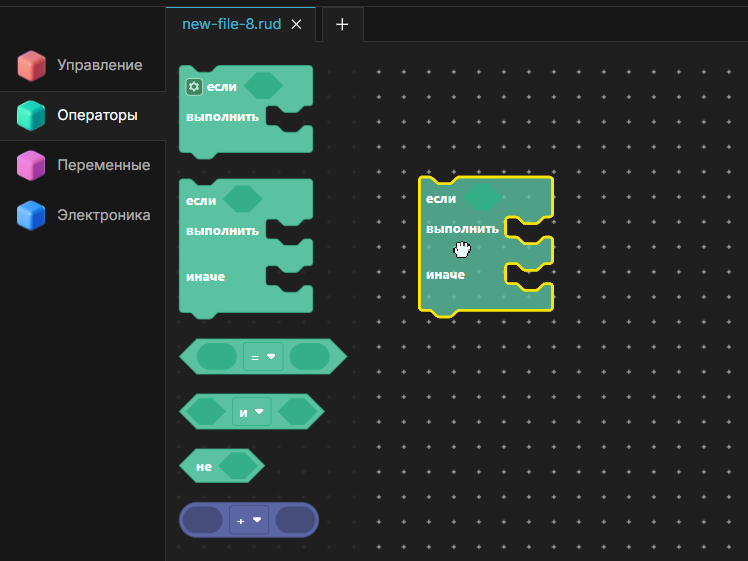
## Добавление блоков на рабочую область

1. Выберите нужную категорию в левой панели.



2. Нажмите и удерживайте нужный блок.

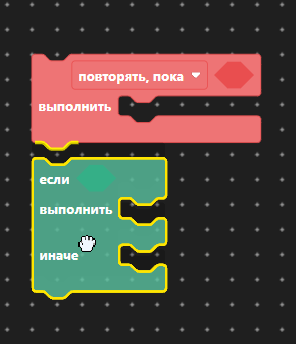
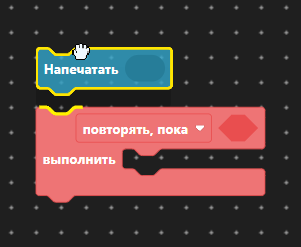


3. Перетащите блок на рабочую область и отпустите кнопку мыши.

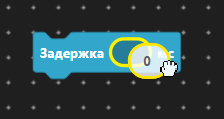
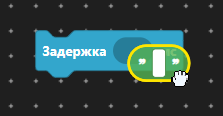


## Cоединение и вложенность блоков

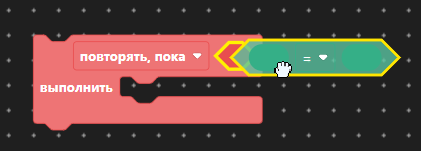
Блоки соединяются между собой с помощью «зубчиков» автоматически при приближении друг к другу сверху и снизу, образуя цепочки команд, которые выполняются последовательно.



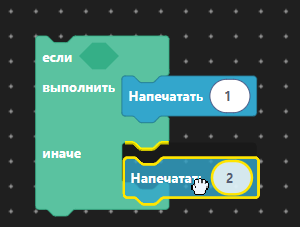
У многих блоков есть входные параметры — специальные поля, куда можно вставлять другие блоки подходящего типа. Например, в поле блока «Задержка» можно вставить только числовой блок, а строку — нельзя. Если тип блока не подходит для данного поля, вставка невозможна — желтая цветовая подсветка не появится.





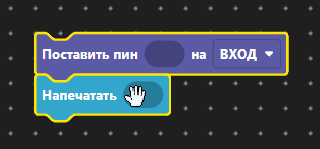


🡪

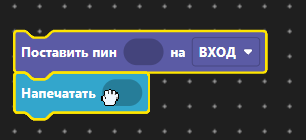
Вложенность означает, что один или несколько блоков помещаются внутрь тела другого (родительского) блока и выполняются только в его контексте: например, блоки внутри цикла выполняются на каждой итерации, а блоки внутри условия — только если условие истинно.

## Перемещение и отсоединение блоков

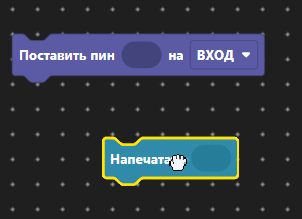
1. Внутри рабочей области наведитесь на нужный блок мышью.



2. Нажмите и удерживайте блок.



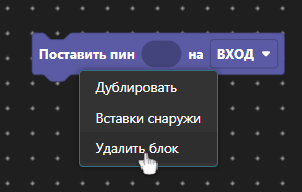
3. Перетащите блок в новое место, путем перемещения курсора и отпустите кнопку мыши.



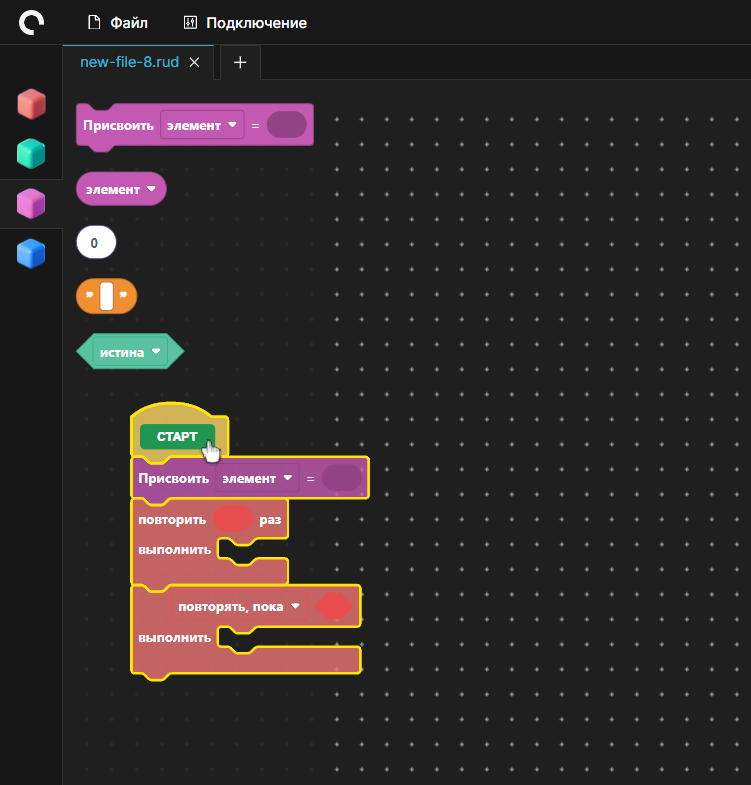
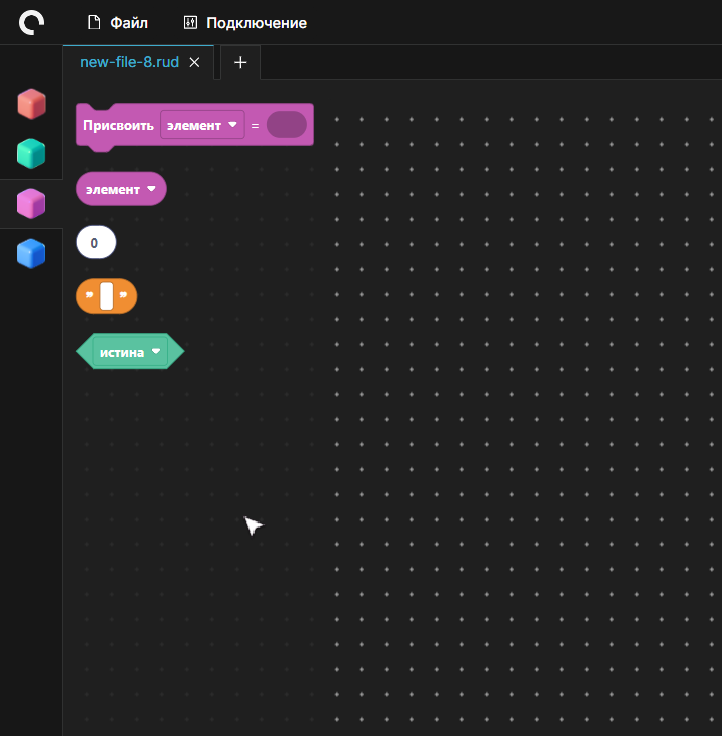
Примечание: если блок снизу соединён с другими блоками (образует цепочку), при перемещении этого блока также перемещаются все связанные с ним снизу блоки — он выступает в роли главного для всей цепочки. Чтобы отсоединить блок от цепочки, нужно сначала отсоединить снизу все связанные с ним блоки, а потом уже отсоединить сам блок.

## Удаление блоков

1) Чтобы удалить блок, выделите его (кликните левой кнопкой мыши) и нажмите клавишу Delete.

****2) Можно вызвать контекстное меню (ПКМ по блоку) и выбрать пункт «Удалить».

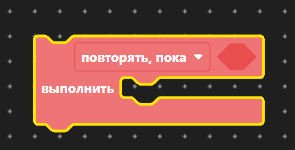
3) Также удаление происходит если перетащить блок или цепочку блоков обратно в панель категорий, где выбираются блоки. Этот способ особенно удобен для удаления сразу нескольких связанных блоков.



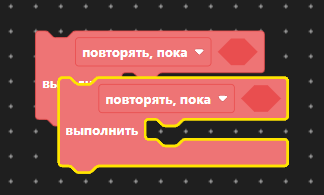
🡪

- При удалении родительского блока автоматически удаляются все вложенные в него блоки.

## Копирование и вставка блоков

1. Для копирования блока выделите его (кликните левой кнопкой мыши) и нажмите Ctrl+C.

2. Для вставки используйте Ctrl+V — блок появится рядом с копируемым оригиналом.

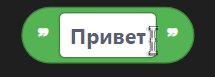


## Настройка параметров блоков

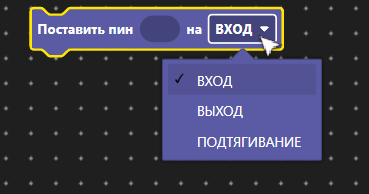
Числовые значения: кликните внутри числового блока и введите новое значение.



Текстовые поля: кликните внутри текстового блока и введите новое значение.

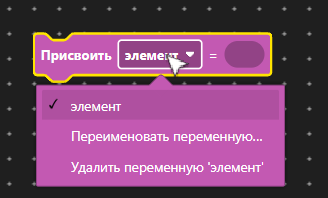
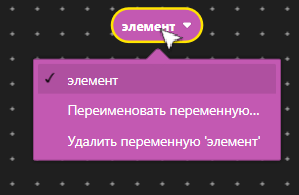


Выбор заготовленных вариантов: кликните на стрелку внутри блока и выберите нужное значение из выпадающего списка.



## Работа с переменными

В Rudiron IDE есть два типа блоков, связанных с переменными:

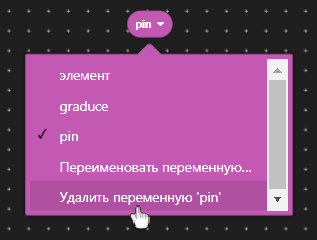
Блок «Переменная» — можно вставлять в другие блоки, он хранит в себе значение переменной. Блок «Присвоить» — используется для присваивания значения переменной. В обоих блоках присутствуют инструменты для работы с ними - одинаковый набор опций:

Переменные по умолчанию имеют название «Элемент» — это базовое имя, которое всегда остается доступным в выпадающем списке, даже если вы создадите и переименуете другие переменные.

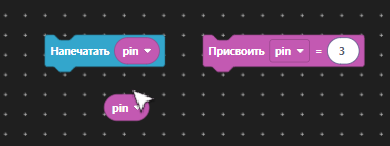
**Удаление переменной:**

1. Кликните по блоку «Переменная» или по выпадающему списку в блоке «Присвоить».

2. В выпадающем меню кликните на название переменной, около нее появиться галочка, далее выберите «Удалить переменную».

****

3. Все блоки с этим именем будут удалены из рабочей области.

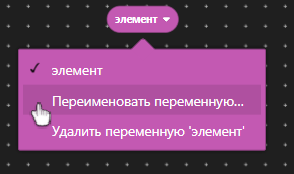


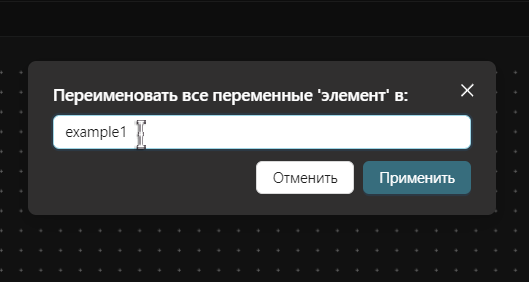
🡪

**Переименование переменной:**

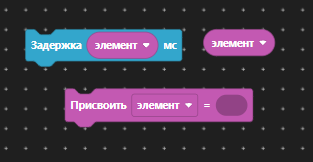
1. Кликните по блоку «Переменная» или по выпадающему списку в блоке «Присвоить».

2. В выпадающем меню кликните на название переменной, около нее появиться галочка, далее выберите «Переименовать переменную».



3. Введите новое имя во всплывающем окне и нажмите «Применить».

1. Имя изменится у всех блоков с этим названием.



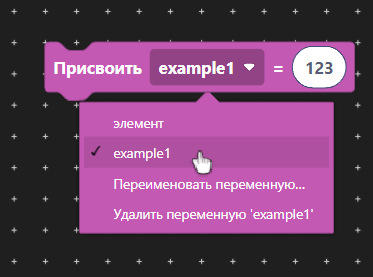
🡪

**Изменение значения переменной:**

1. Используйте блок «Присвоить».

2. В первом поле выберите нужную переменную, около нее появиться галочка.

3. Во втором поле укажите новое значение (число, строку или другую переменную).

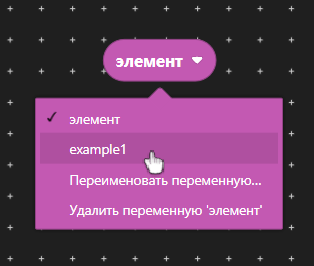


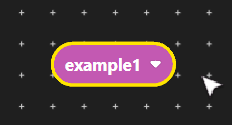
**Использование существующей переменной:**

1. Кликните по блоку «Переменная» или по выпадающему списку в блоке «Присвоить».

2. В выпадающем списке выберите нужное имя переменной, около нее появиться галочка

3. Блок будет ссылаться на выбранную переменную.





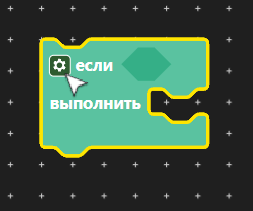
🡪

## Расширение условий через шестерёнку

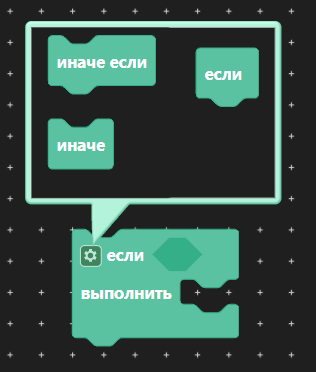
У некоторых блоков (например, «Если», «Если-иначе») есть кнопка-шестерёнка. При нажатии на неё открывается меню настроек блока, где можно добавлять дополнительные ветви («Иначе если», «Иначе»), а также менять их порядок с сохранением вложенностей. Это необходимо для построения сложных ветвлений прямо внутри одного блока, вместо их повторений.

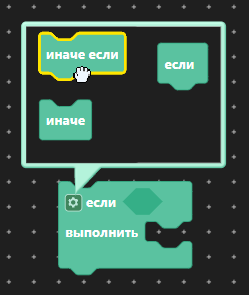
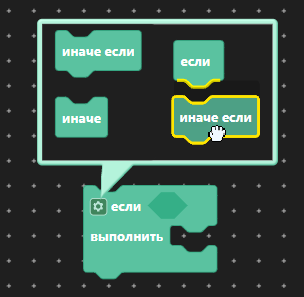
**Инструкция:**

1. Нажмите на шестерёнку на блоке условия («Если» или «Если-иначе»).



1. Откроется отдельное окно с доступными блоками ветвей.

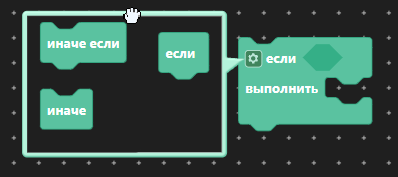


1. Перетащите нужные блоки «Иначе если» или «Иначе» из левой части окна в правую, соединяя их с блоком «ЕСЛИ».



Примечание: работа с блоками внутри этого окна аналогична работе с блоками в файле — их также можно перемещать, соединять и менять порядок.

Окно настроек остаётся открытым до тех пор, пока вы снова не нажмёте на шестерёнку, в связи с этим его можно перетаскивать в любое место рабочей области, зажав левую клавишу мыши и перемещая курсор. Функция позволяет не загораживать код, который вы редактируете.



## Дополнительные действия через контекстное меню

Кликните правой кнопкой мыши по блоку, чтобы открыть контекстное меню.

Доступные функции:

- Дублировать

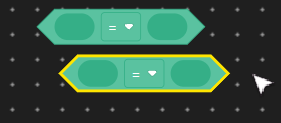
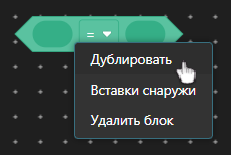
- Удалить

- Вставка внутри /Вставка снаружи

1. **Дублирование блоков**



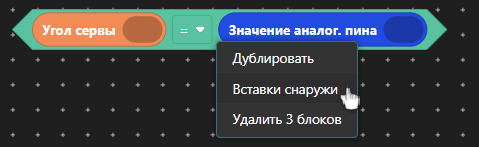
Для создания копии блока выберите пункт «Дублировать» в контекстном меню. Дублируется как сам блок, так и все вложенности.

****

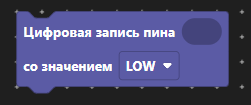
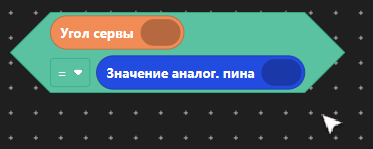


1. **Сворачивание и разворачивание блоков**

Некоторые блоки (например, условные операторы или работа с пинами) можно свернуть для компактного отображения. Для этого используйте пункт «Вставка внутрь» или «Вставка снаружи» в контекстном меню. Свёрнутый блок отображается в двухстрочном виде, одно поле - одна строка.

****



****



# **7. Визуализация платы**

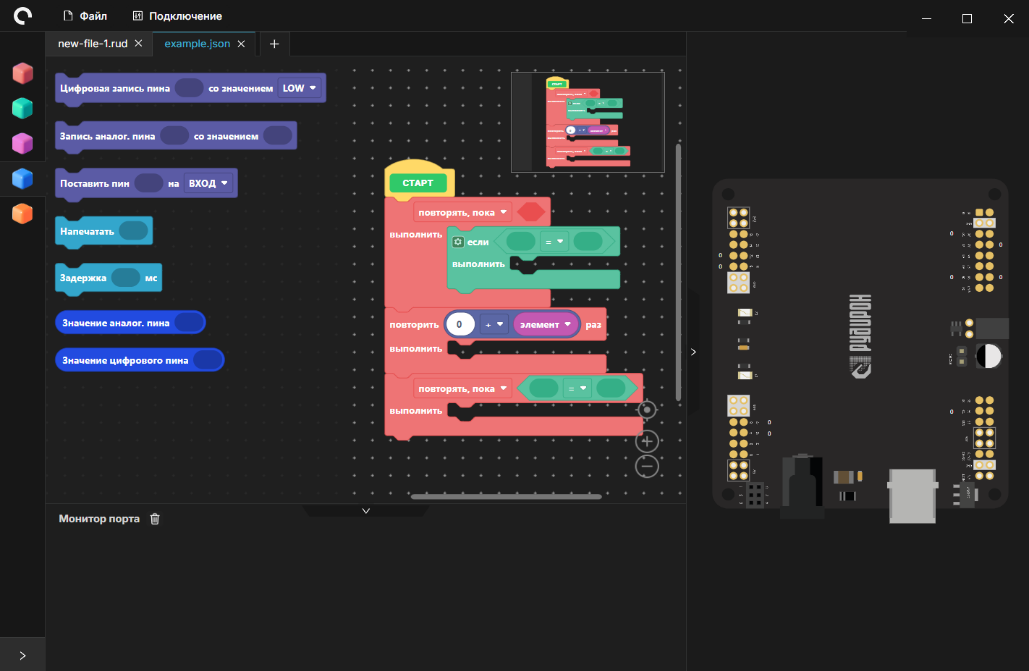
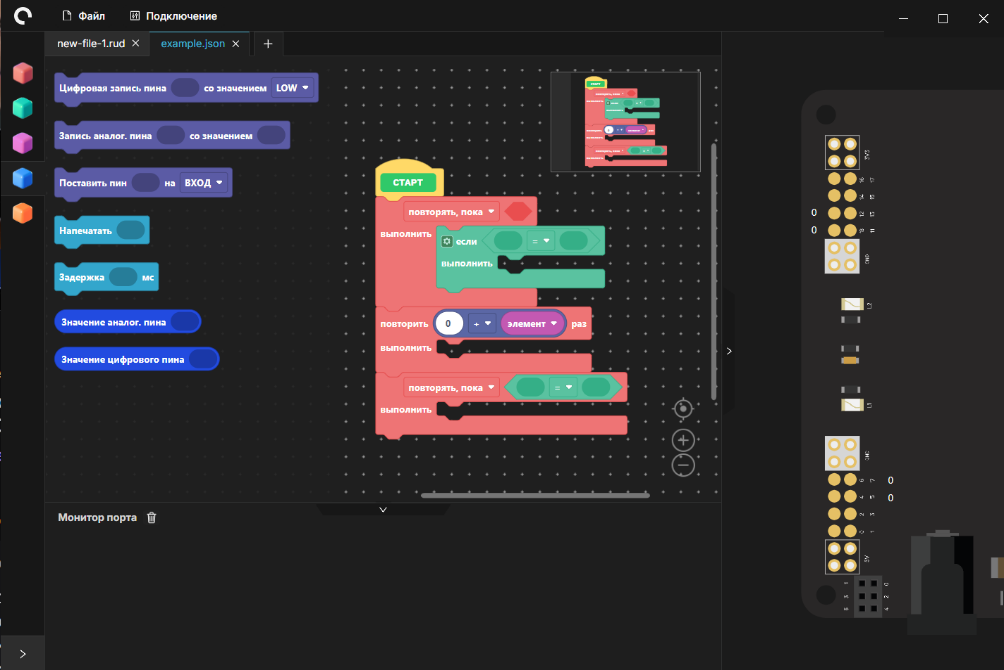
Справа от рабочей области находится панель с изображением платы Рудирон. На ней отображаются:

* Все доступные пины
* Состояние цифровых пинов (HIGH/LOW)
* Значения аналоговых пинов
* PWM-пины с текущими значениями

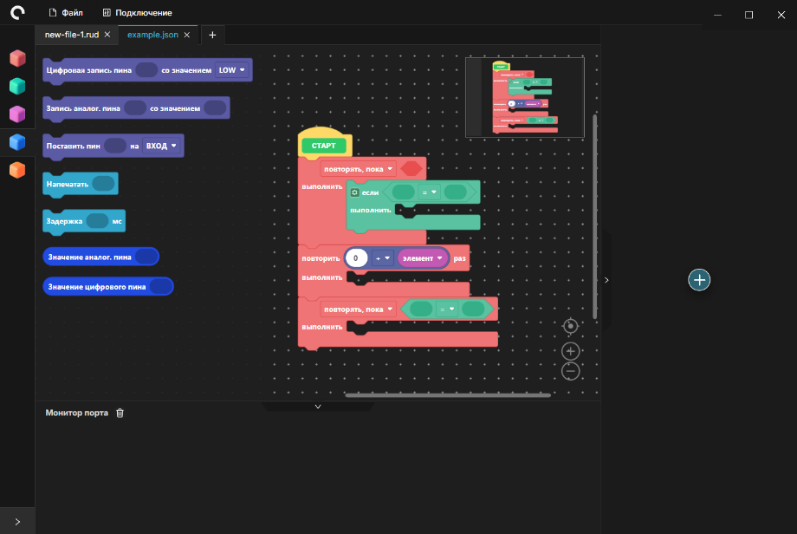
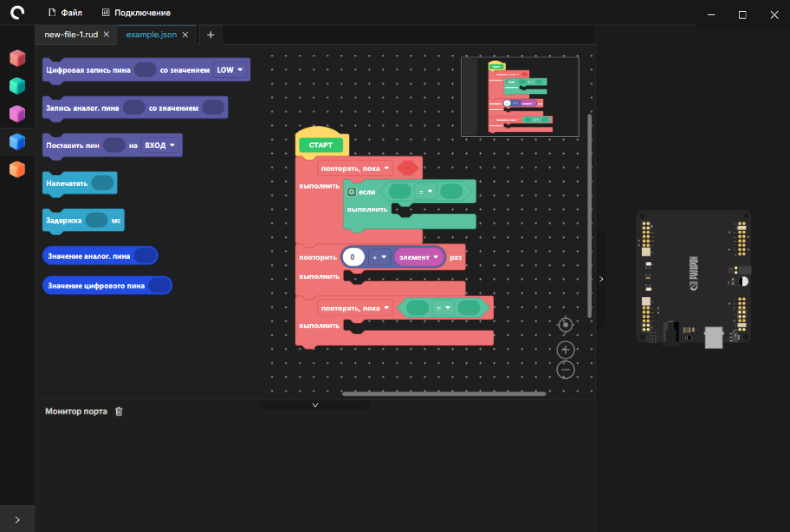
## Интерактивность

1) Плату можно двигать, зажав мышку в пустой области или на самой плате и перемещая курсор внутри панели. Это удобно, если вы работаете только с частью платы, например, с левыми пинами — можно сдвинуть плату так, чтобы видеть только нужную часть.

2) Масштабирование осуществляется колесиком мыши — это позволяет лучше рассмотреть значения на аналоговых пинах.

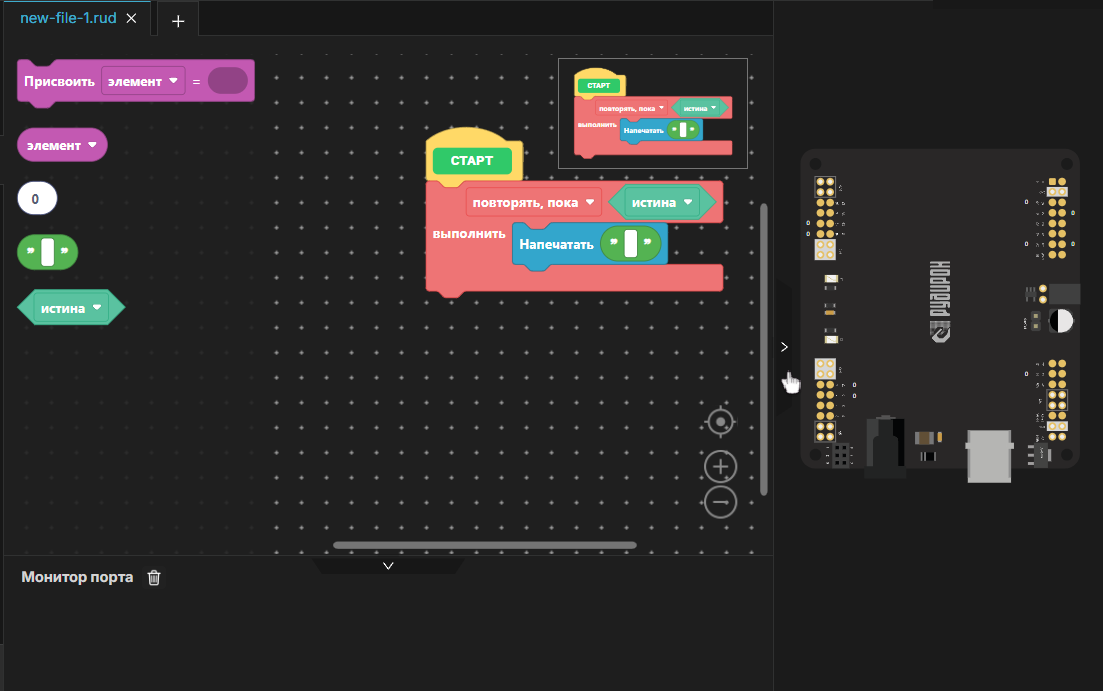


3) Если плата уходит за пределы области видимости, появляется кнопка возвращения, которая ставит плату в центр панели.

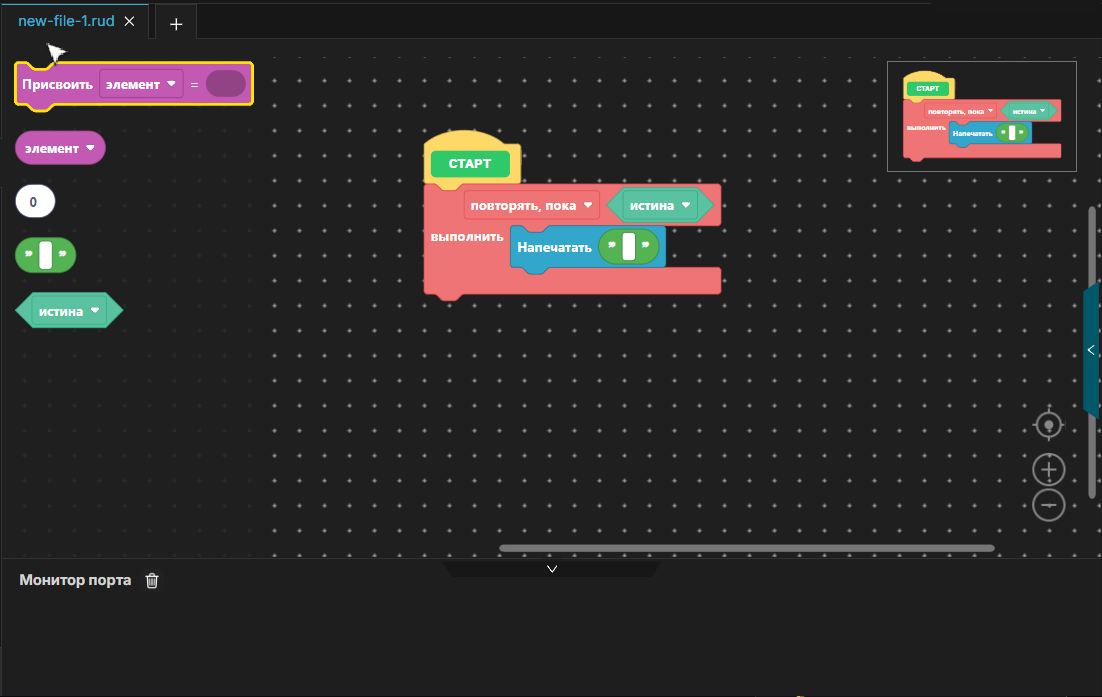


🡪



****4) Сворачивание: на границе между рабочей областью и визуализацией платы расположена кнопка-свернуть. При нажатии на неё панель полностью скрывается, освобождая место для рабочей зоны. Повторное нажатие на эту кнопку разворачивает монитор обратно, возвращая ему прежние размеры.

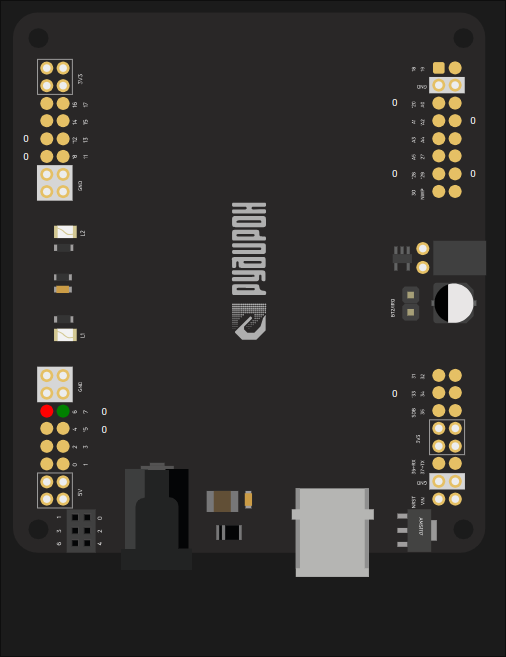


****

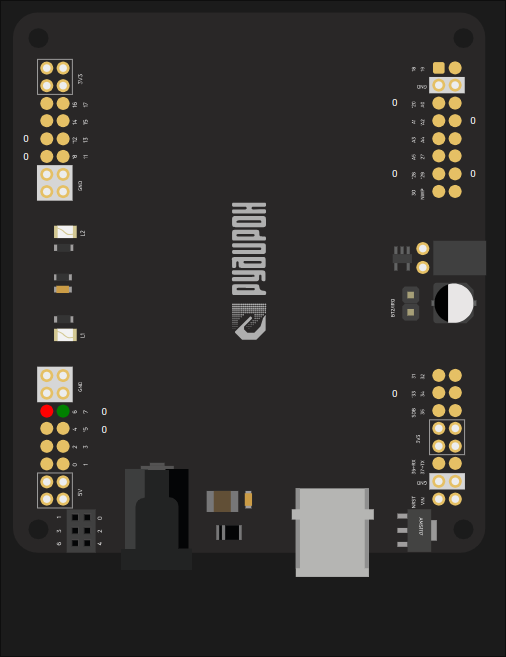


## Цветовая индикация

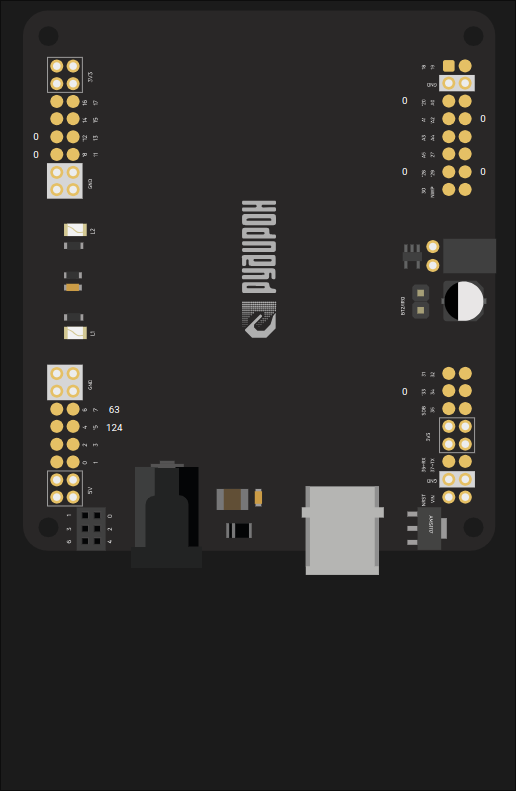
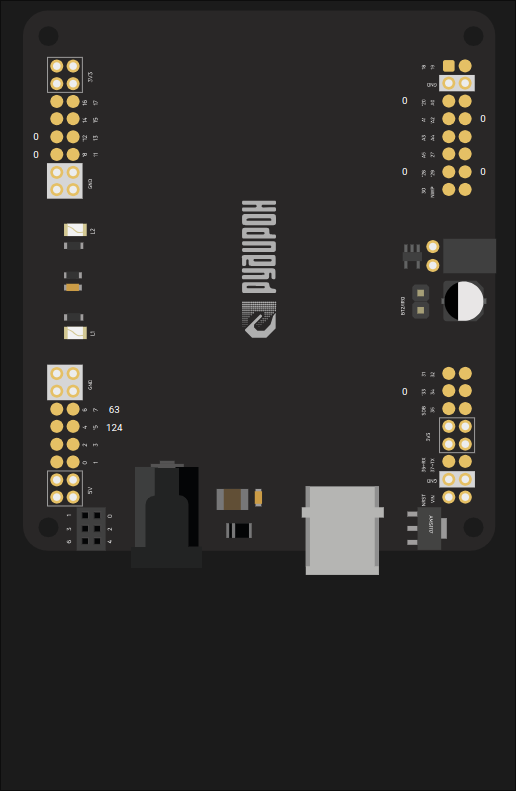
* Желтый — пин никак не настроен



* Красный — пин в цифровом режиме, на нём сейчас 0
* Зелёный — пин в цифровом режиме, на нём сейчас 1



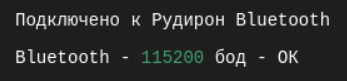
* Если пин в аналоговом режиме (вход, выход, ШИМ, ЦАП и т.д.), то он будет цвета как на официальной распиновке, но рядом с ним будет отображаться число (от 0 до 255), показывающее текущее значение на этом пине



# **8. Монитор порта**

Монитор порта — это динамическая панель в нижней части приложения, предназначенная для отображения всей информации, поступающей с платы во время работы программы. Здесь выводятся:

* Сообщения о подключении и отключении платы



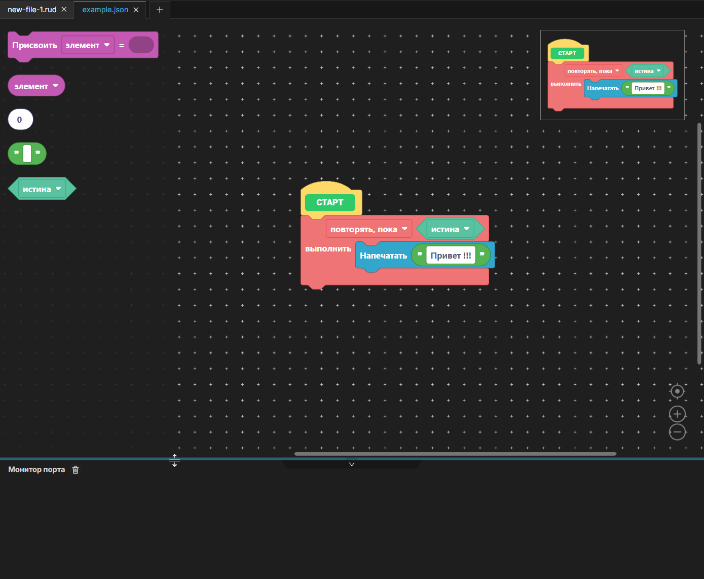
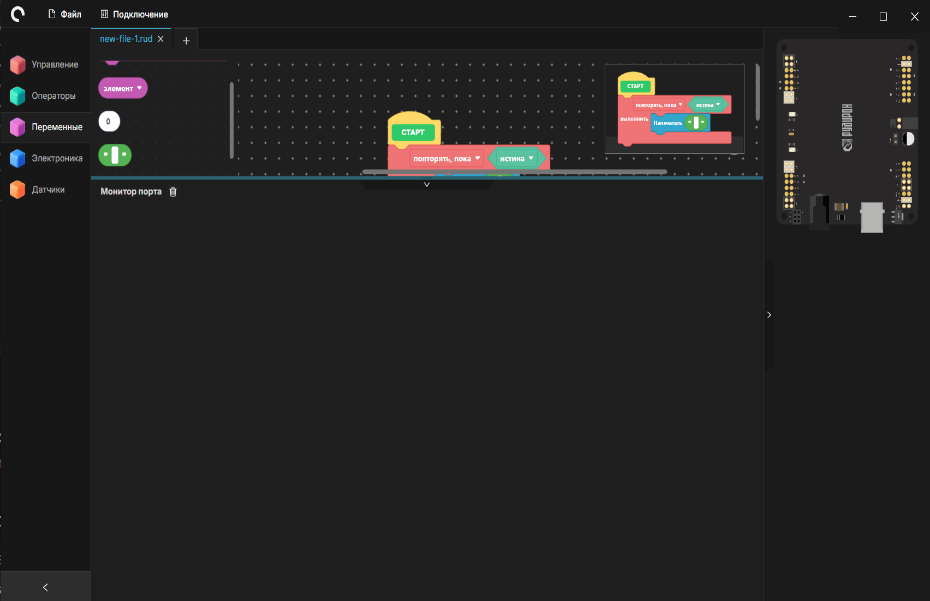
* Результаты работы блока "Напечатать"



* Другие сообщения, отправленные с платы

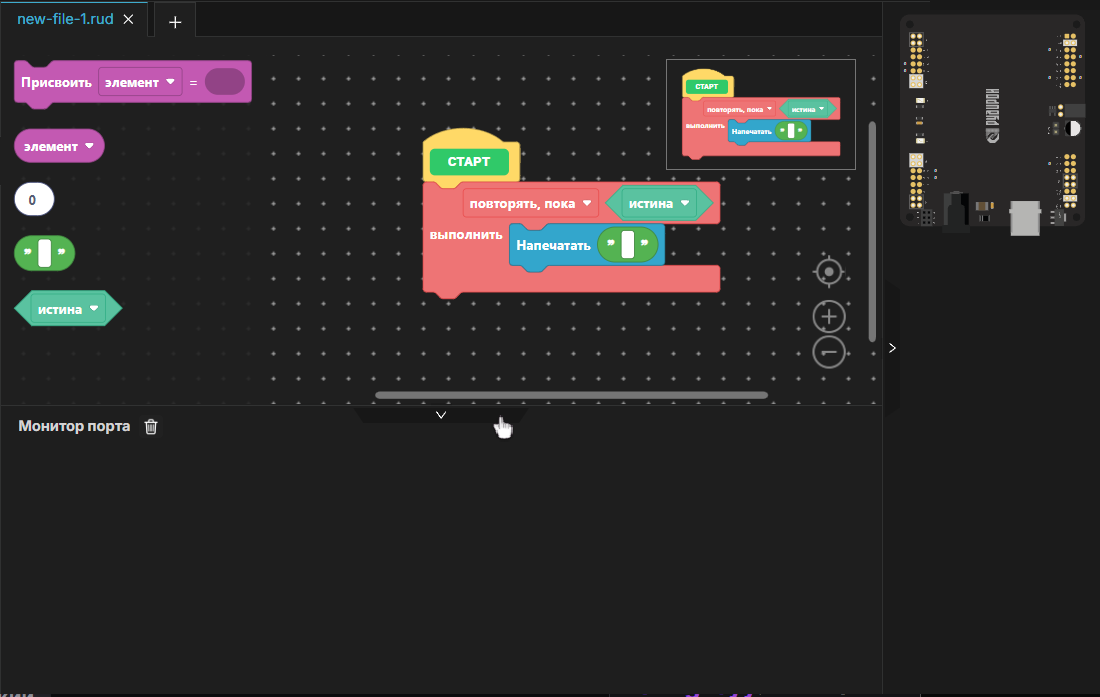


## Управление

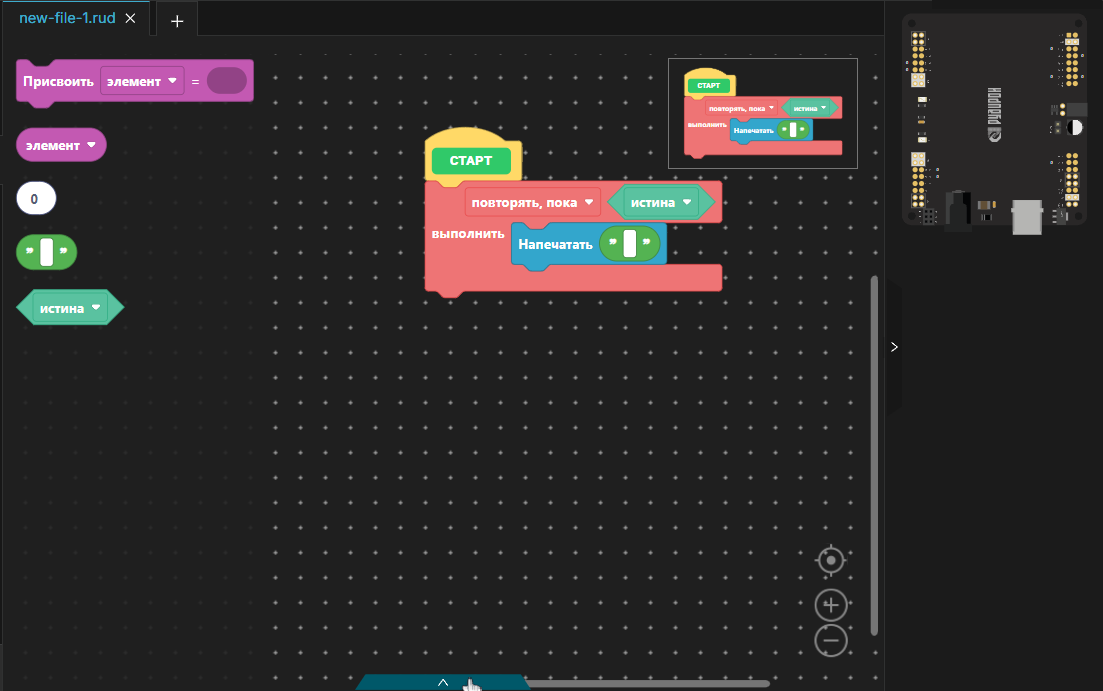
1) Динамическое изменение размера: Панель можно увеличивать или уменьшать по высоте. Для этого наведите курсор на границу между рабочей областью и монитором порта, зажмите левую кнопку мыши и перемещайте границу вверх или вниз.



2) Сворачивание: на границе между рабочей областью и монитором порта расположена кнопка-свернуть. При нажатии на неё панель полностью скрывается, освобождая место для рабочей зоны. Повторное нажатие на эту кнопку разворачивает монитор обратно, возвращая ему прежние размеры.







3) Очистка: кнопка с иконкой корзины позволяет очистить все сообщения в мониторе порта.

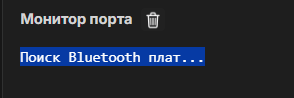


4) Автопрокрутка: Новые сообщения автоматически прокручивают панель вниз, чтобы всегда была видна самая последняя информация.





5) Выделение и копирование: Любой текст в мониторе порта можно выделить мышкой и скопировать для дальнейшего использования.



# **9. Выполнение программ**

Блок "СТАРТ" является главным блоком любой программы в Rudiron IDE. Он служит точкой входа и началом выполнения программы.

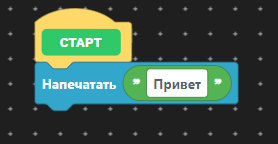
## Пошаговая инструкция запуска:

1. Подготовка

* + Убедитесь, что плата Рудирон подключена к компьютеру
  + Проверьте, что в мониторе порта отображается сообщение о подключении
  + Убедитесь, что визуализация платы показывает состояние пинов

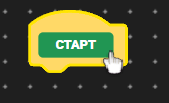
2. Поиск блока "СТАРТ

* + Найдите желтый блок с кнопкой в категории “Управление”
  + Блок "СТАРТ" должен быть в начале цепочки программы
  + Убедитесь, что к нему подключены все необходимые блоки



3. Запуск программы

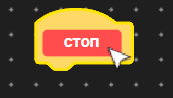
* 1. Кликните на кнопку "СТАРТ" на блоке



* 1. Кнопка изменится на "СТОП", что означает начало выполнения
  2. В мониторе порта появится сообщение о запуске программы

4. Остановка программы:

1. Нажмите на кнопку "СТОП" на блоке "СТАРТ"



1. Программа немедленно остановится
2. Кнопка вернется в состояние "СТАРТ"

5. Мониторинг выполнения

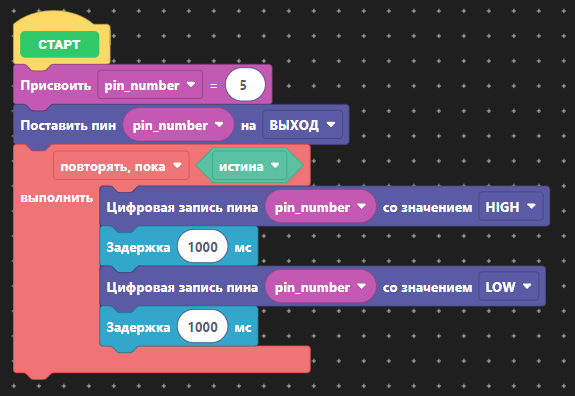
* + Следите за визуализацией платы — состояние пинов будет меняться
  + В мониторе порта будут появляться сообщения о выполнении команд и при использовании блока "Напечатать"

Примечание: в одном файле можно создать несколько независимых цепочек блоков, но только одна может быть активной одновременно. Пример: Цепочка 1: СТАРТ → Настройка пинов → Основная программа. Цепочка 2: СТАРТ → Тестовая программа

# **10. Примеры программ**

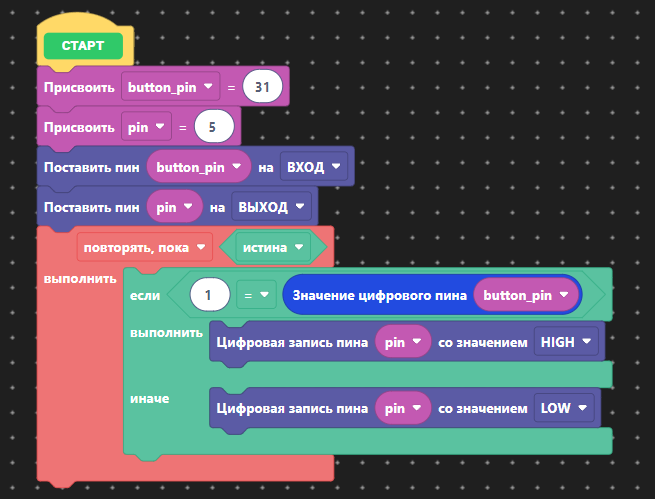
1. Работа со светодиодами

Светодиод на пине 5 будет мигать каждую секунду.



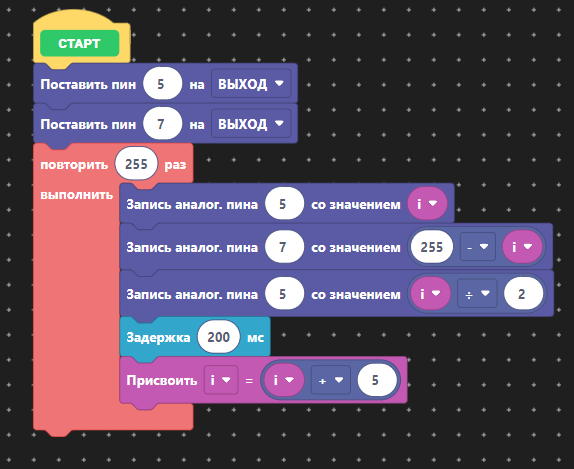
2.Обработка нажатия кнопки

Светодиод загорается при нажатии кнопки.



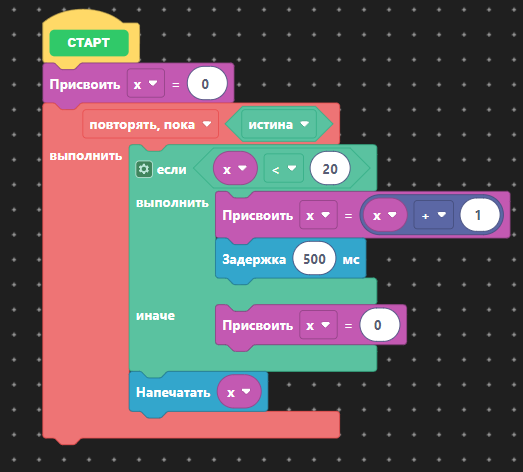
3. Работа с циклами

Три светодиода создают эффект радуги, плавно меняя цвета.



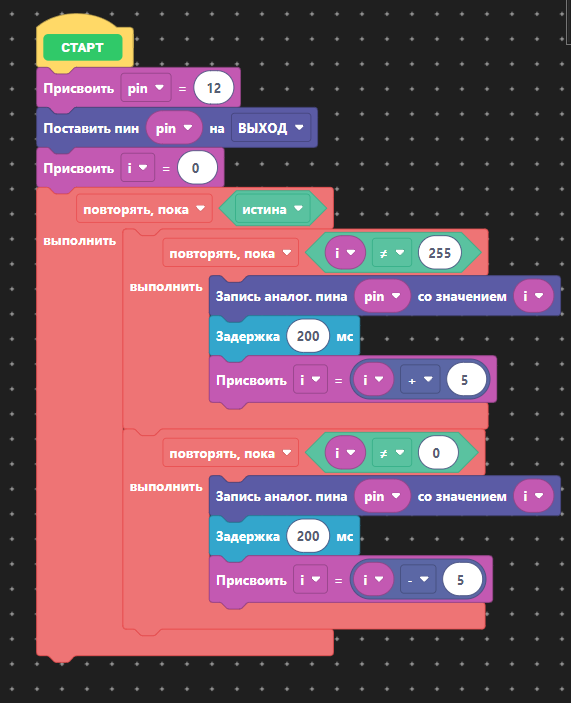
4.Работа с переменными

Переменная x увеличивается, а при достижении 20 сбрасывается.

****

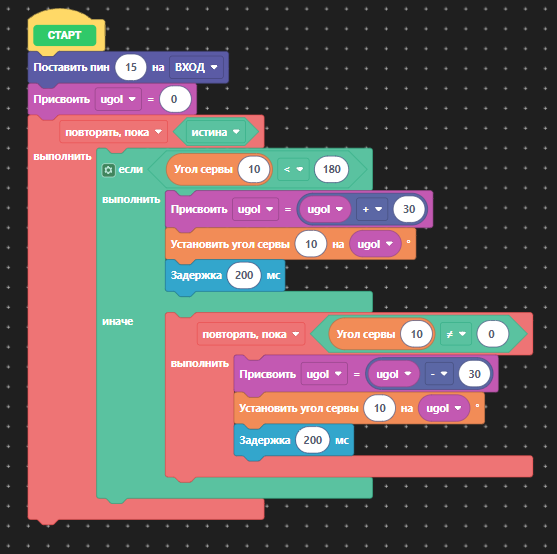
5. Работа с ШИМ

Яркость светодиода плавно увеличивается и уменьшается.



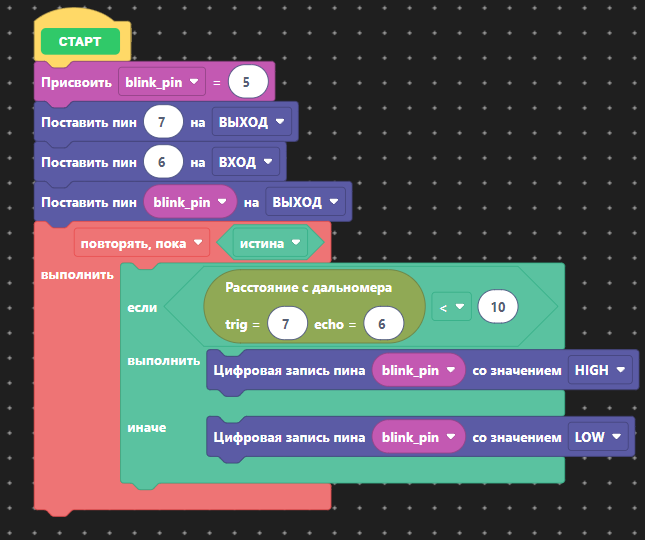
6.Работа с сервоприводом

Серво плавно двигается до 180 градусов и обратно к 0.



7. Работа с ультразвуковым дальномером

Светодиод загорается, если препятствие ближе 10 см.



8.Работа с моторами

Машинка 3 секунды едет прямо, а потом столько же задним ходом

