НАУЧИМ ИЗОБРЕТАТЬ
БУДУЩЕЕ

TRIK

Алгоритмические структуры и элементарные действия

ООО «КиберТех»
Санкт-Петербург, 2020

Цели урока

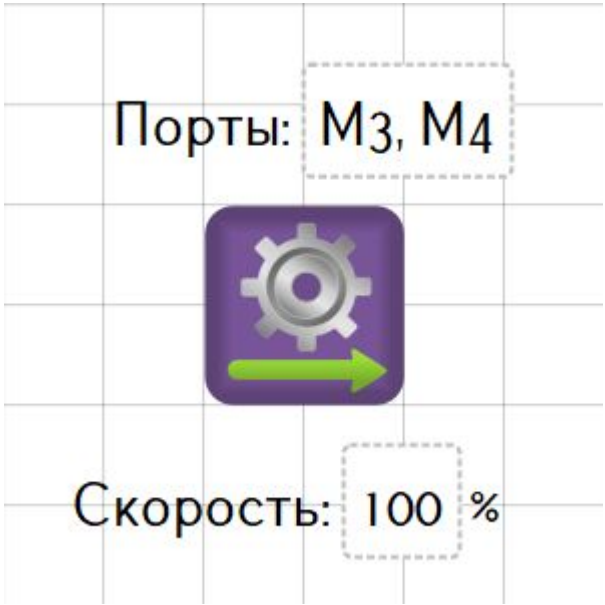
TRIK

- Научиться реализовывать алгоритмы для элементарных действий мобильного робота
- Познакомиться с базовыми алгоритмическими структурами

Движение вперед

Движение вперед базовой тележки задается подачей на левый и правый мотор одинаковой скорости.

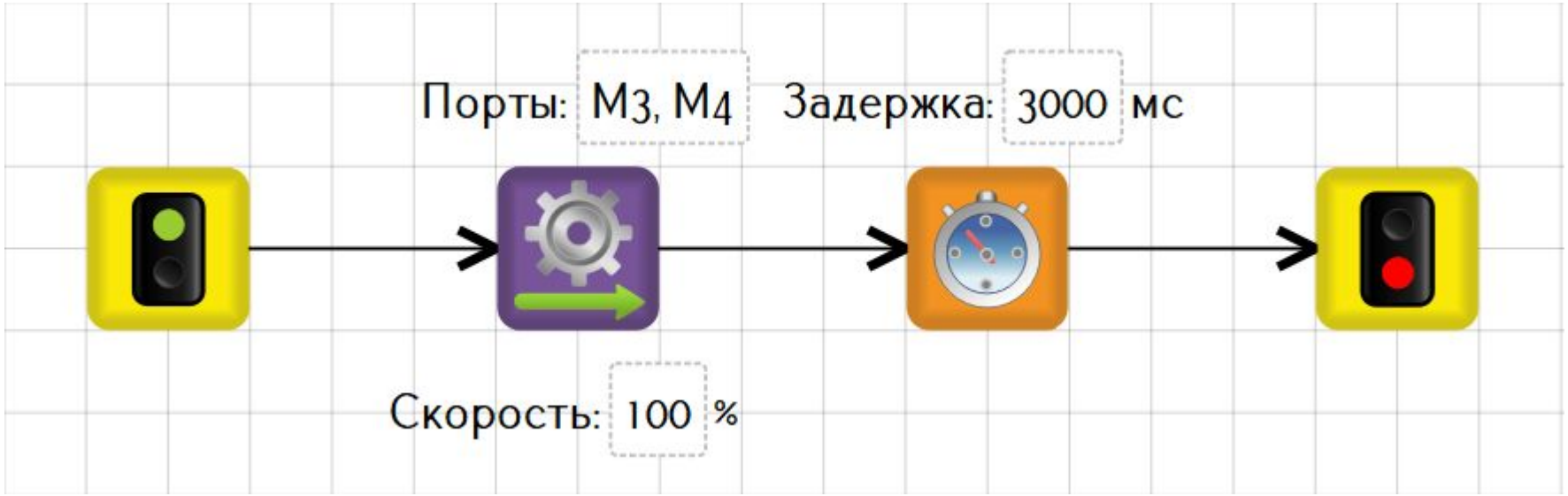
В TRIK Studio для подачи мощности на мотор существует отдельный блок **«Моторы вперед»**.



У этого блока два свойства:

1. Порты
2. Скорость.

Движение вперед

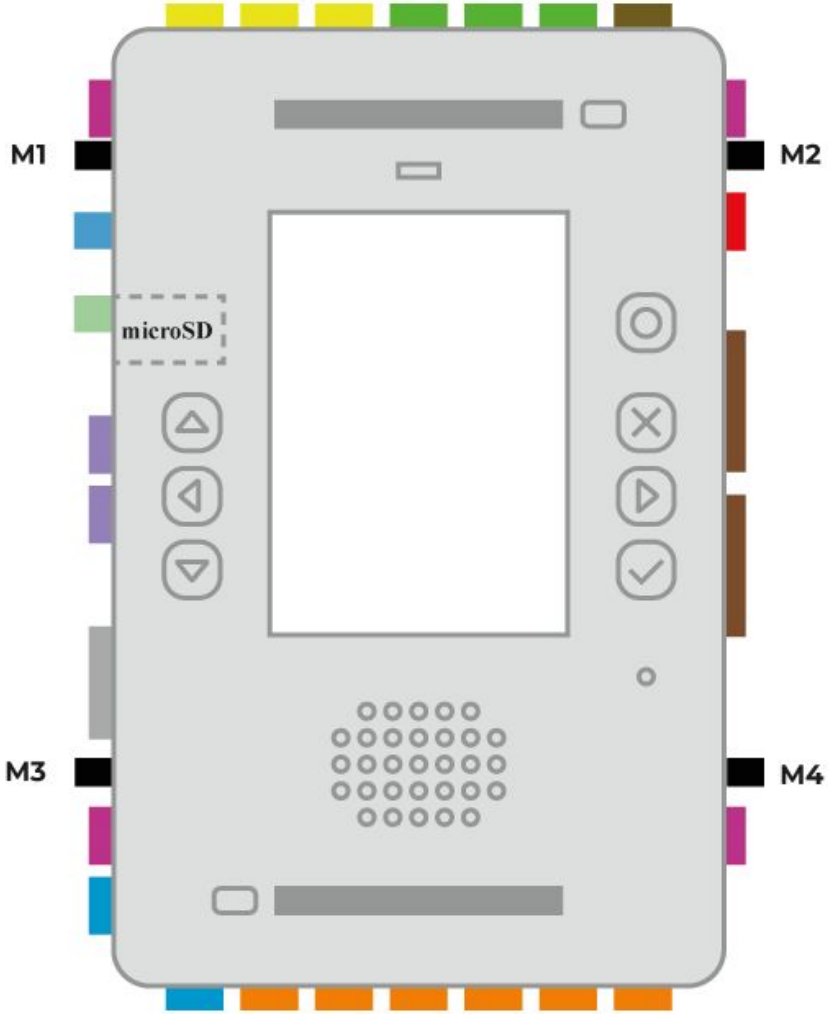


Подключение моторов

TRIK

У контроллера ТРИК четыре порта для подключения силовых моторов:

M1, **M2**, **M3** и **M4**.



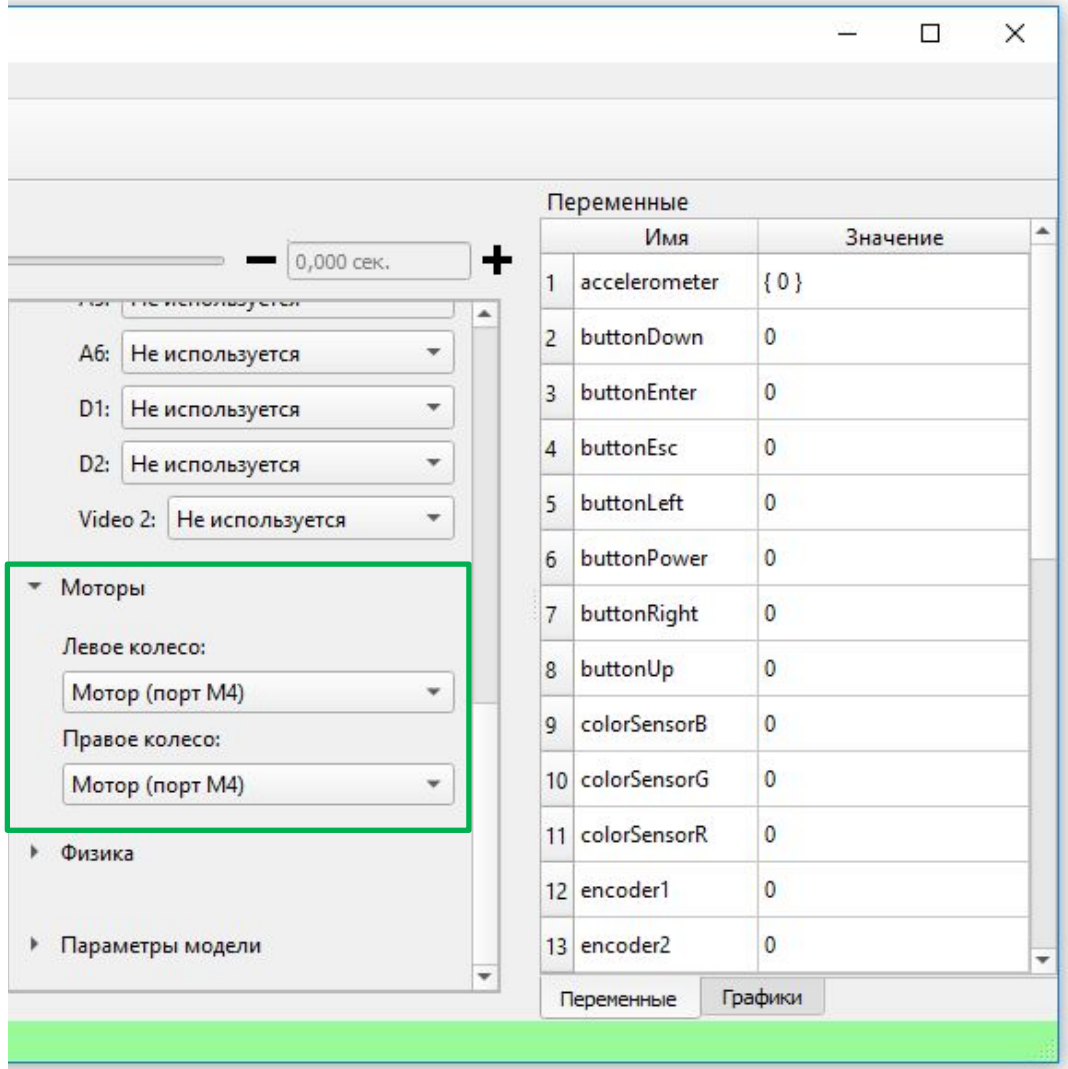
Подключение моторов

TRIK

Подключение моторов в 2D-модели по умолчанию:

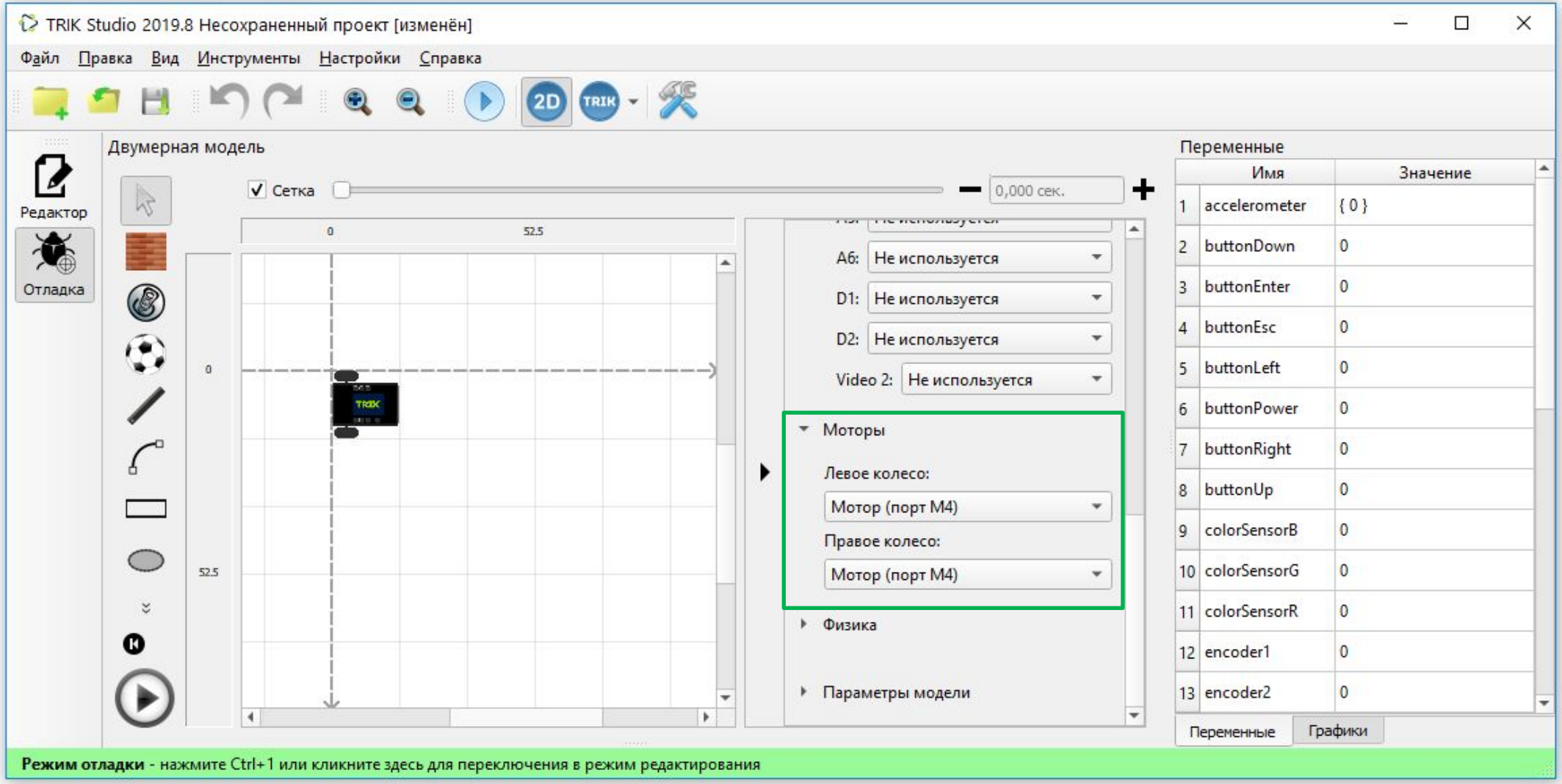
- левый — к порту **M3**
- правый — к порту **M4**.

Настройку подключения моторов можно изменить в режиме отладки на центральной панели в разделе «Моторы».



Подключение моторов

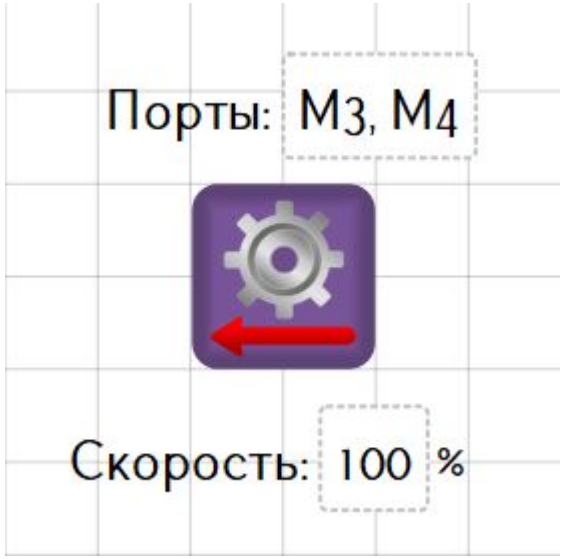
TRIK



Движение назад

Движение назад выполняется аналогично.

Используем блок
«Моторы назад».



Движение назад

Движение назад

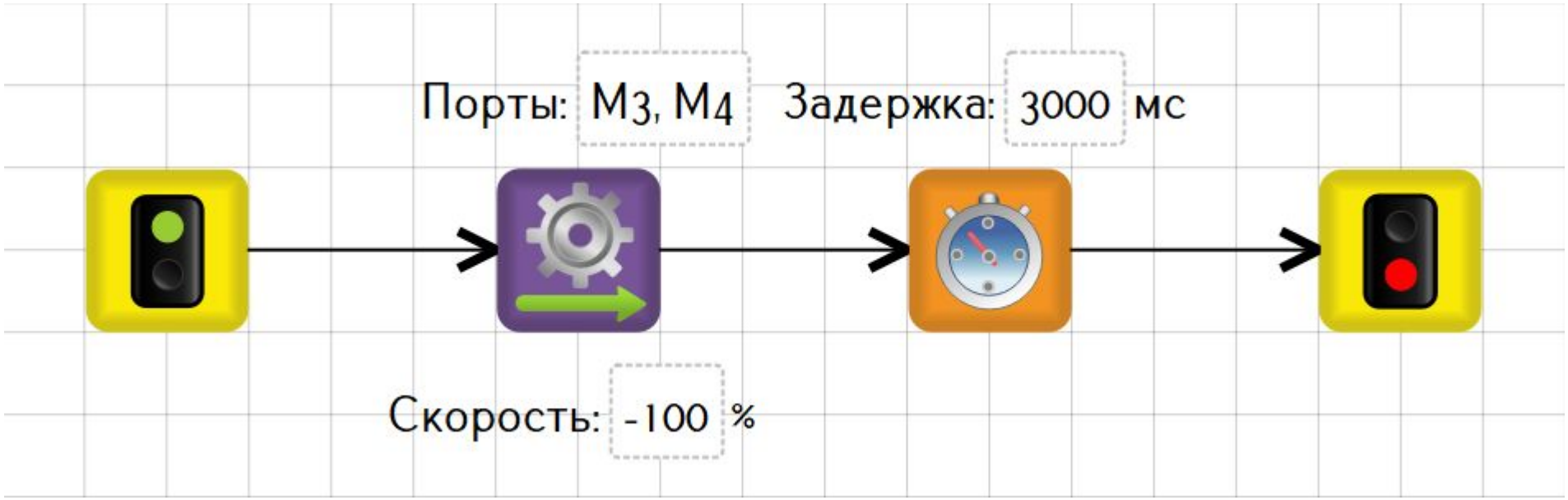
Но! Диапазон подаваемой мощности: от -100 до 100 %.

То есть для движения назад можно использовать и блок «Моторы вперед», подав мощность -100 %.

Движение назад

TRIK

Но! Диапазон подаваемой мощности: от -100 до 100 %.



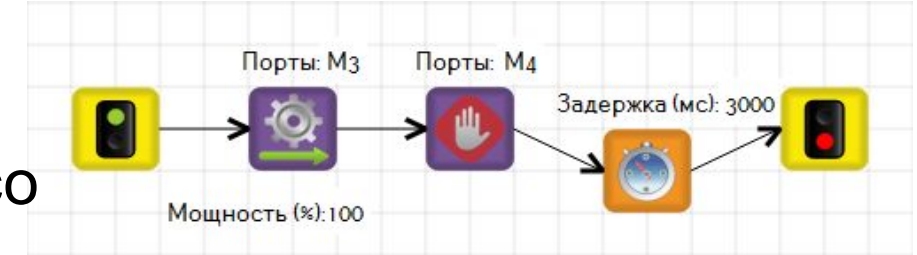
То есть для движения назад можно использовать и блок «Моторы вперед», подав мощность -100 %.

Повороты

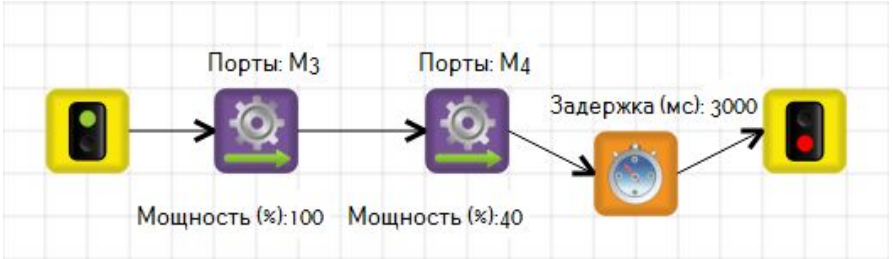
TRIK

Повороты можно разделить на 3 типа:

- **резкий поворот**
  мощность подается только на одно колесо



- **плавный поворот**
  мощность подается на два колеса, но на одно больше



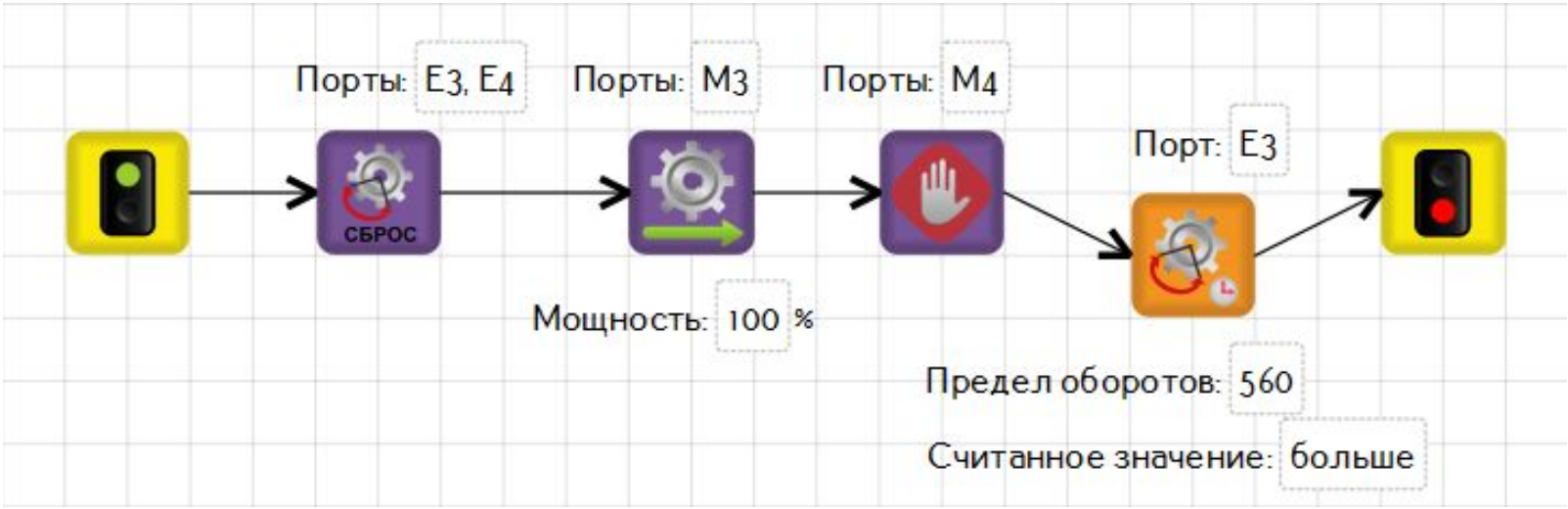
- **поворот на месте**
  одинаковая мощность с разными знаками на два колеса

Модели алгоритмов

TRIK

Представленные выше алгоритмы – **тайм-модели**. Движение осуществляется по таймеру. Это «плохой» подход, так как в этом случае выполняемое действие зависит от заряда аккумулятора.
Правильно будет использовать **ожидание значения энкодеров**.

В этом случае перед элементарным действием необходимо сбросить значения энкодеров.

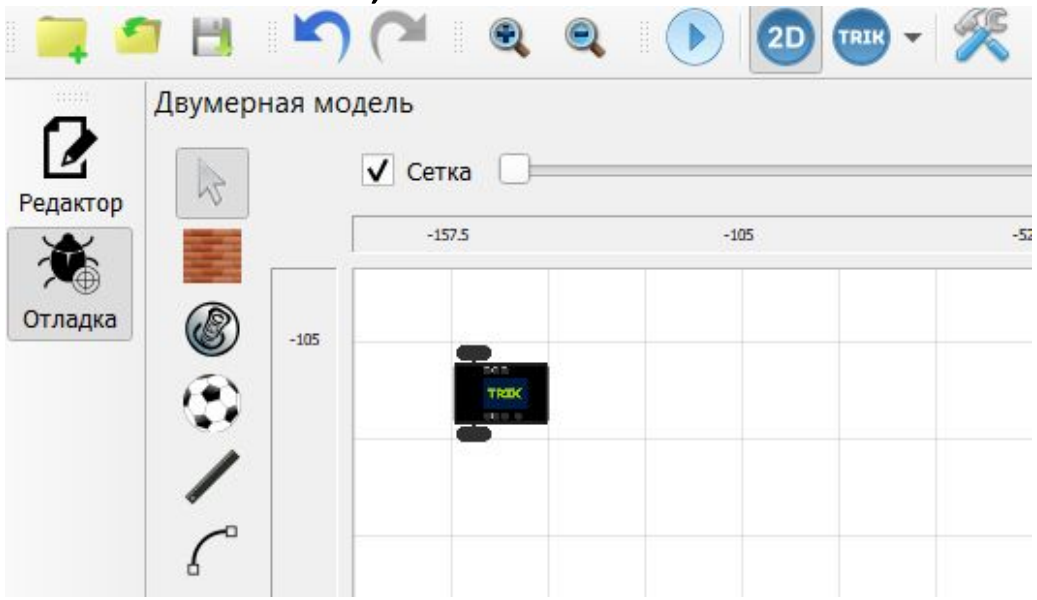


Остальные элементарные действия (движение назад, повороты) реализуются аналогично.

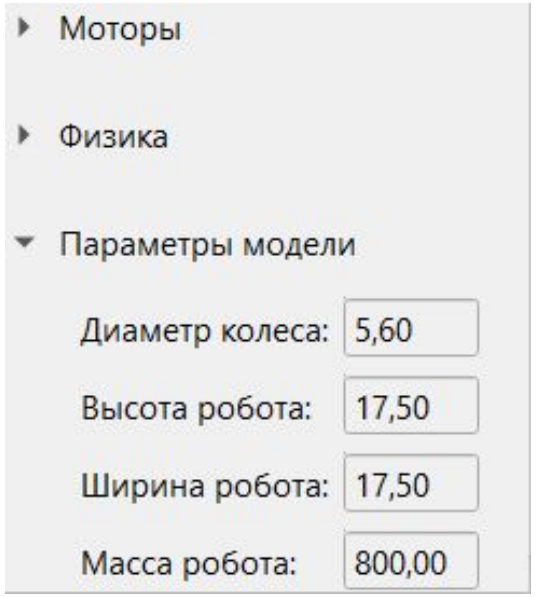
Точные перемещения

Поставьте галочку **«Сетка»**. Теперь вы можете отслеживать точные перемещения модели.

1 клетка = 17,5 см

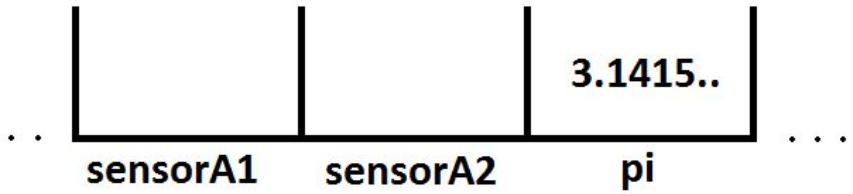


Также, в режиме «отладка» всегда можно посмотреть параметры визуальной модели Для удобства длина и размер базы робота совпадают с размером клетки (17,5 см)



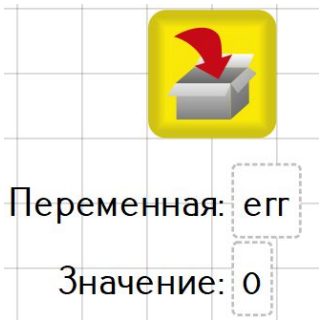
Переменные

Переменная — поименованная область памяти.

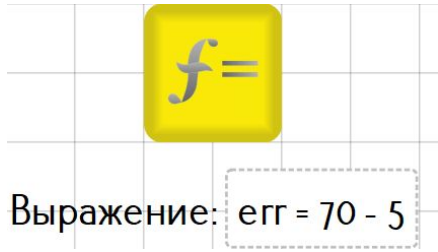


В TRIK Studio можно ввести свои переменные, используя блоки:

«Инициализация переменной»



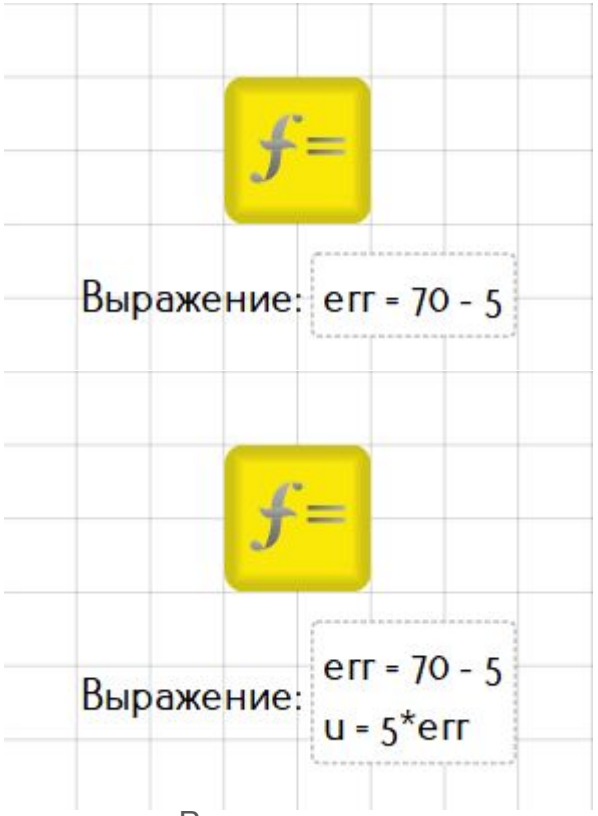
«Выражение»



Переменные

TRIK

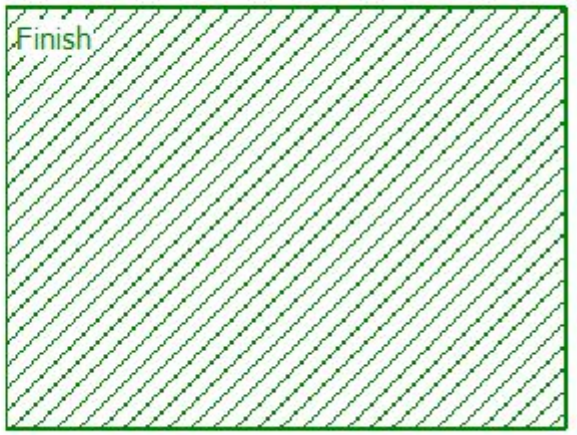
В блоке **«Выражение»** можно как создавать новые переменные, так и записывать выражения.



- Нецелые числа пишутся через точку. Например: **1.75**
- Для перехода на новую строку используйте **«Shift» + «Enter»**
- Созданным переменным можно присваивать другие переменные, если последние были объявлены и инициализированы ранее. Например: **u** = 5***err**

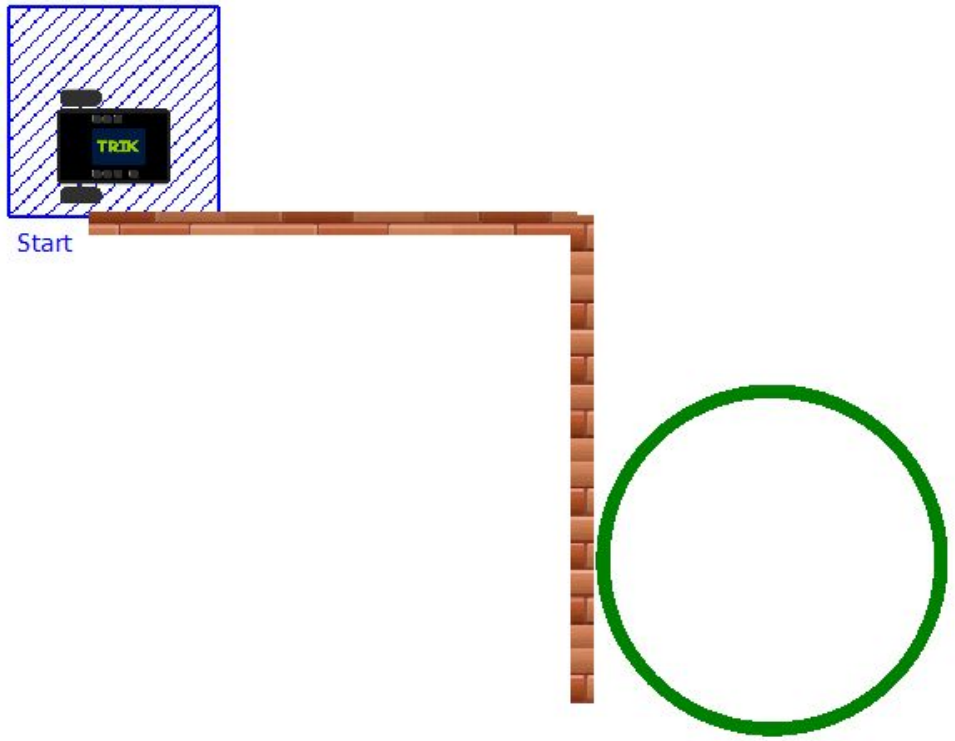
Энкодеры. Задача

Задача 2.1.1 Робот находится в синей зоне старта. Робот должен проехать вперед, развернуться на 180°между зонами старта и финиша, проехать задом и остановиться в зеленой зоне финиша. Использовать энкодерную модель.



Энкодеры. Задача

Задача 2.1.2. Обогнуть угол. Робот должен проехать вперед со скоростью 60, повернуть на 90°, проехать вперед с максимальной скоростью и остановиться в зеленом круге. Использовать энкодерную модель.



Точные перемещения. Задача

Задача 2.1.3. Проехать вперед ровно на 1 метр и 5 сантиметров. Использовать энкодерную модель.

Вам пригодятся следующие параметры:

- d = 5,6 см (диаметр колеса)
- CPR = 360 (полный оборот колеса)

Точные перемещения. Решение

TRIK

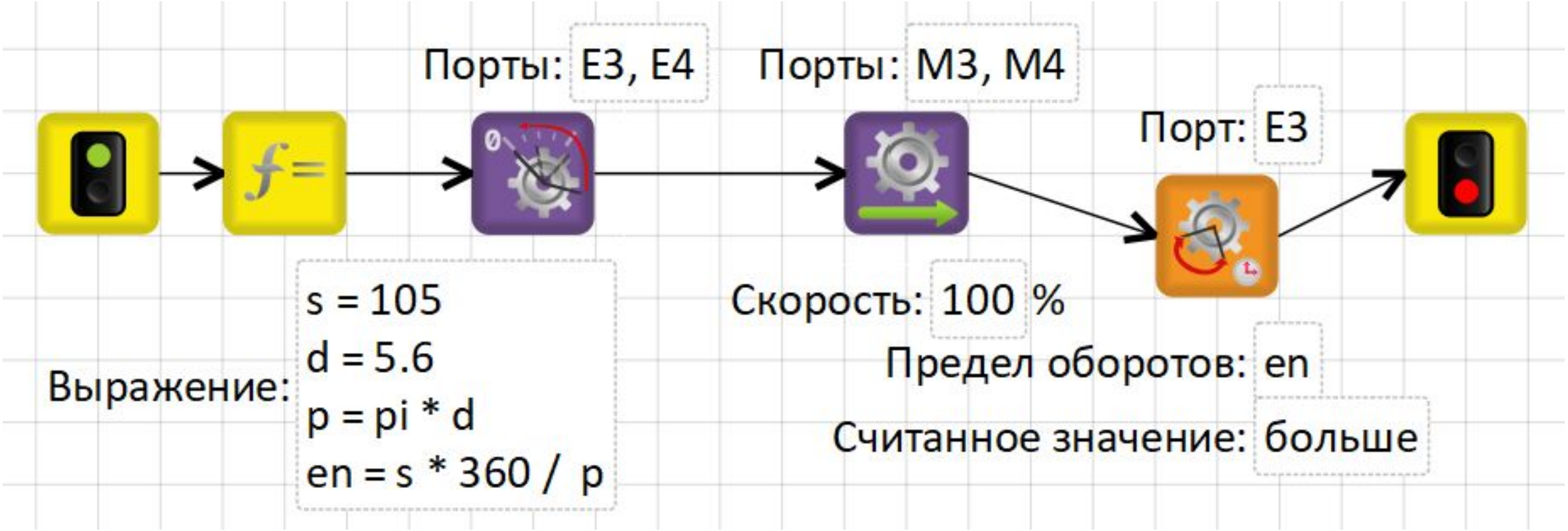
Решение.
Для решения этой задачи необходимо вспомнить элементарные формулы из курса школьной математики: расчет длины окружности и угла поворота.

Введем следующие переменные:

d — диаметр колеса робота
dist — расстояние, которое необходимо проехать роботу
cpr — один оборот колеса в градусах (количество сигналов на оборот)
p — длина окружности
en — количество энкодеров

Точные перемещения. Решение

TRIK



Точные перемещения. Задача

TRIK

Задача 2.1.4. (самостоятельно) Развернуться на месте ровно на 90 градусов. Использовать энкодерную модель.

Для решения вам понадобится дополнительный параметр:

- b = 17.5 см (ширина колеи робота)

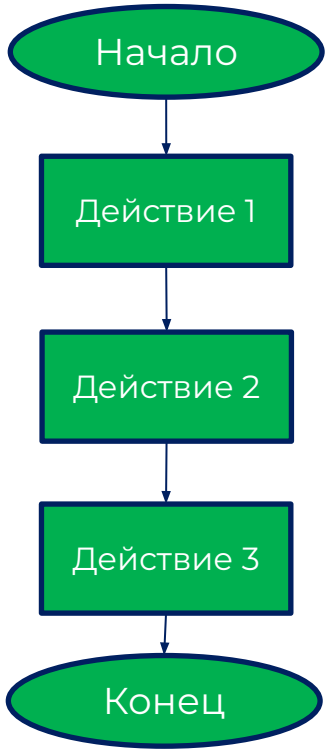
Алгоритм

Алгоритм — набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное число действий, при любом наборе исходных данных.

Исполнитель: робот или любое другое устройство

Инструкции: включить мотор, ждать 3 секунды, повернуть серводвигатель на 80 градусов, включить диод и т.д.

Блок-схема — распространенный тип **схем** (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде **блоков** различной формы, соединенных между собой линиями, указывающими направление последовательности.



Алгоритмические структуры

TRIK

Следование (последовательность) — однократное выполнение операций в том порядке, в котором они записаны в тексте программы.

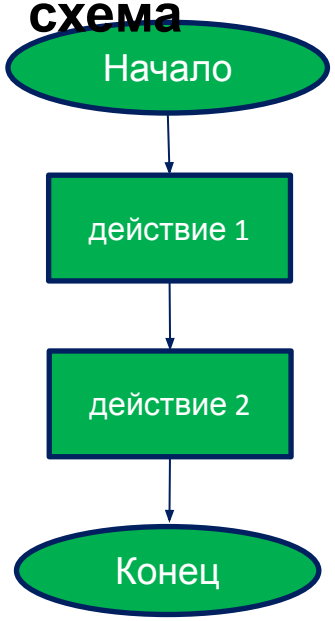
Ветвление — однократное выполнение одной из двух или более операций, в зависимости от выполнения заданного условия.

Цикл — многократное исполнение одной и той же операции до тех пор, пока выполняется заданное условие

Следование

Следование (последовательность) — однократное выполнение операций в том порядке, в котором они записаны в тексте программы.

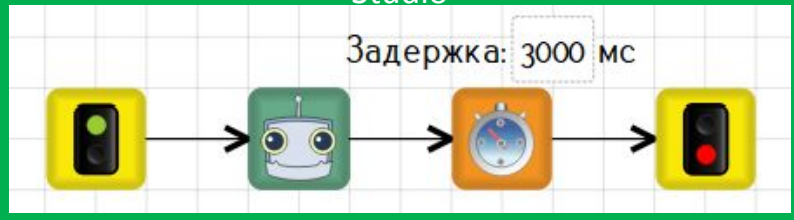
Блок-схема



Псевдокод

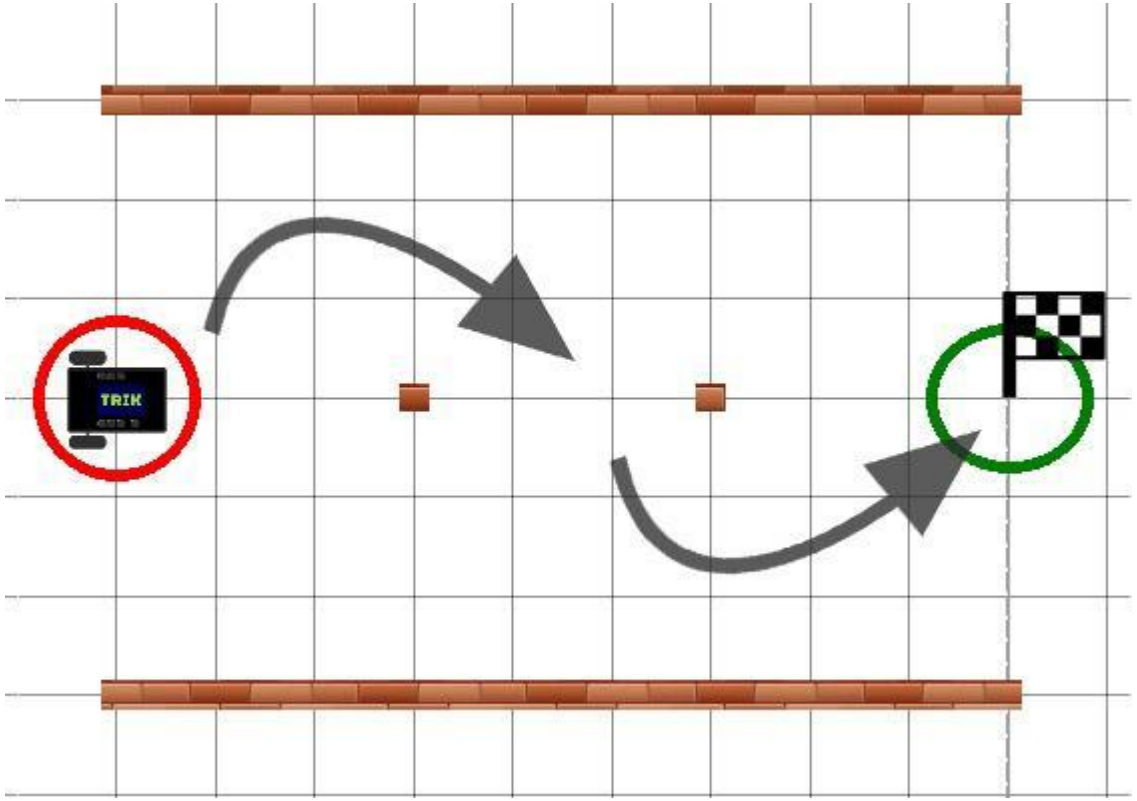
```
speed=-100;
robot.motor.[M2].setPower(100);
robot.motor.[M3].setPower(speed);
robot.wait(1500)
```

Пример в TRIK Studio



Следование. Задача

Задача 2.1.5. (самостоятельно) Написать алгоритм движения модели «змейкой». Использовать энкодерную модель.



Ветвление

TRIK

Выполнение программы идет по одной из двух, нескольких или множества ветвей. Выбор ветви зависит от условия на входе ветвления и поступивших сюда данных.

Существует две основные формы условной инструкции, встречающиеся в реальных языках программирования:

- условный оператор (оператор **if**)
- оператор многозначного выбора (оператор **switch**)

Ветвление. Условный оператор

TRIK

Условный оператор реализует выполнение одной последовательности (ветви) команд при условии, что некоторое логическое выражение (условие) принимает значение «истина», и другой последовательности (ветви), если выражение "ложно". Любая из этих последовательностей может быть "пустой", т.е. не выполнять никаких действий.

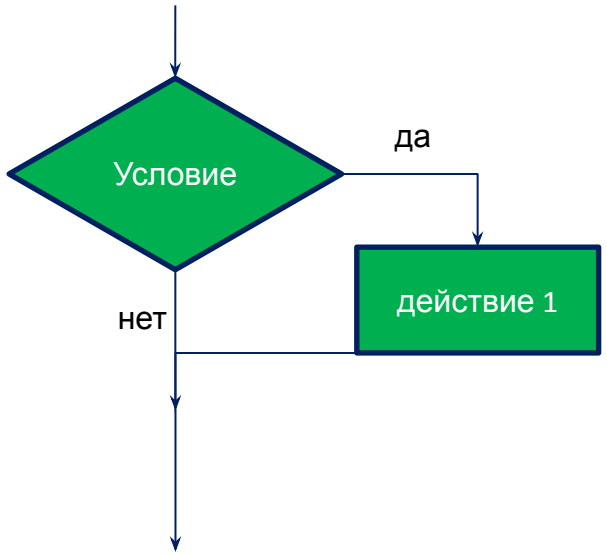
Встречаются следующие формы условного оператора:

- Условный оператор с одной ветвью
- Условный оператор с двумя ветвями
- Условный оператор с несколькими условиями

Неполное ветвление

TRIK

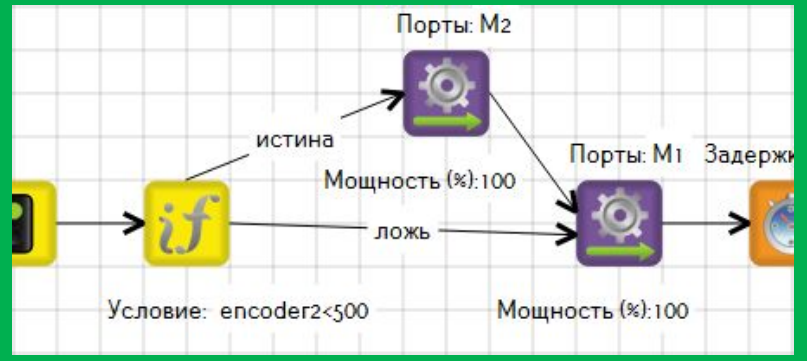
Блок-схема



Псевдокод

```
if (encoder.[E2].read() < 500)
      robot.motor.[M2].setPower(100);
robot.motor.[M1].setPower(100);
```

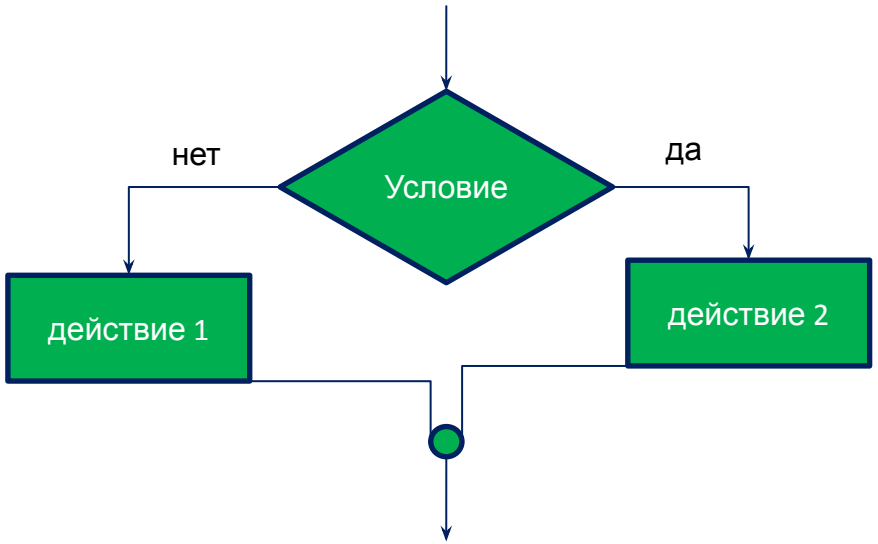
Пример в TRIK Studio



Полное ветвление

TRIK

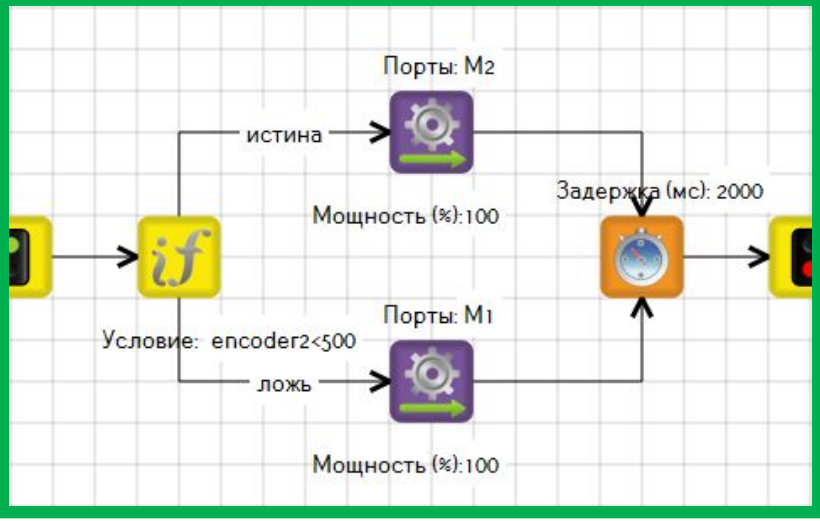
Блок-схема



Псевдокод

```
if (encoder.[E2].read() < 500)
    robot.motor.[M2].setPower(100);
else
    robot.motor.[M1].setPower(100);
robot.wait(2000);
```

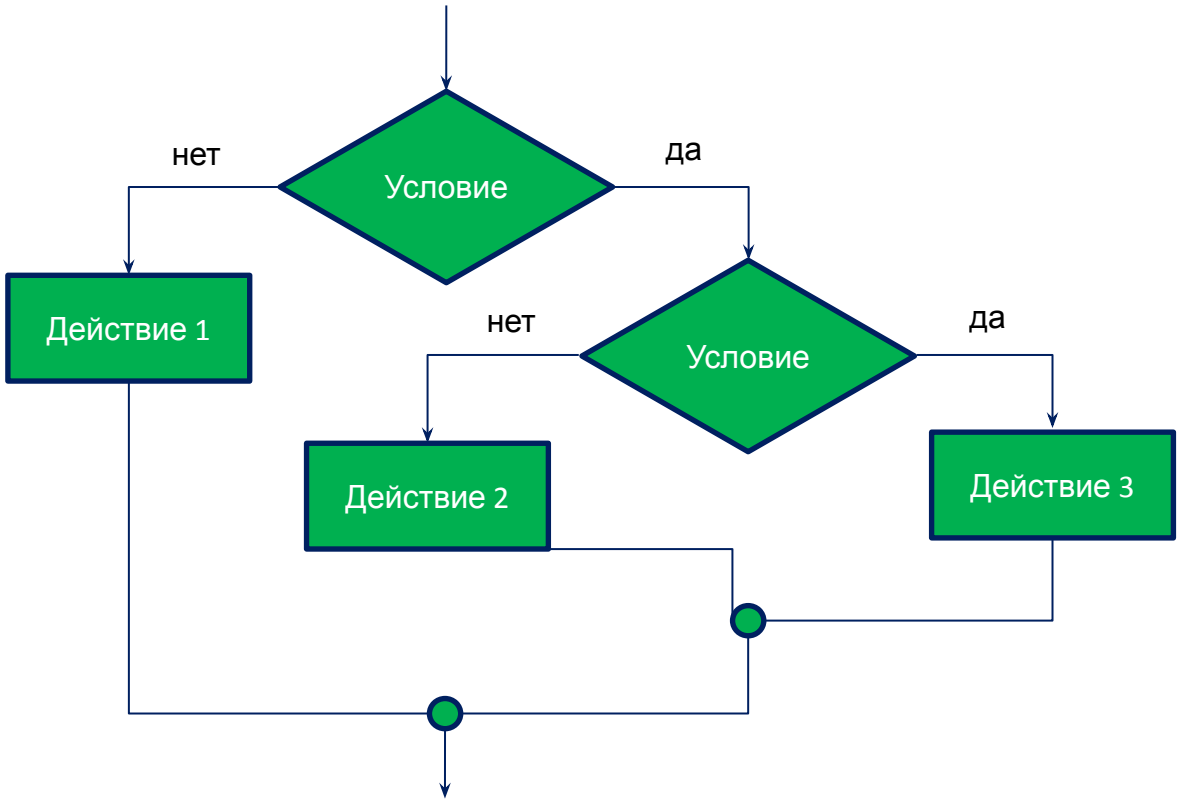
Пример в TRIK Studio



Каскадные условия

TRIK

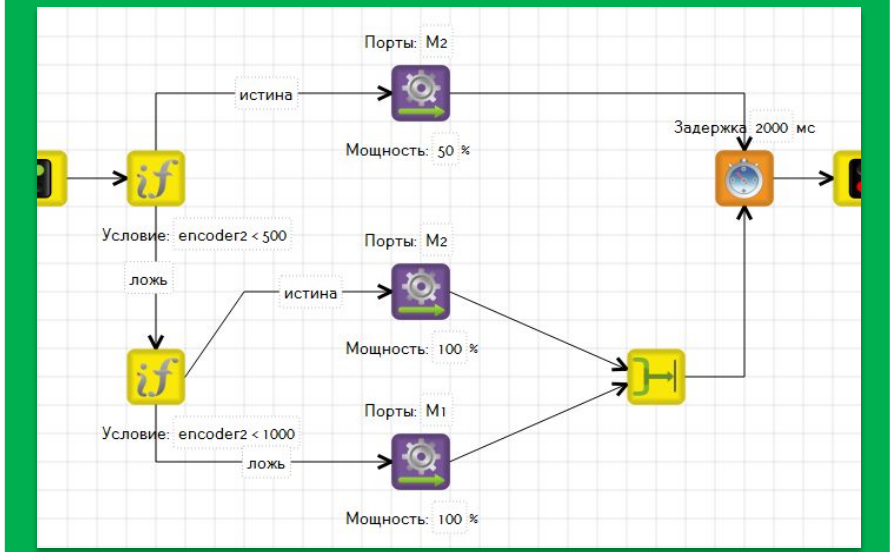
Блок-схема



Псевдокод

```
if (encoder.[E2].read() < 500)
    robot.motor.[M2].setPower(50);
elseif (encoder.[E2].read() < 1000)
    robot.motor.[M2].setPower(100);
else
    robot.motor.[M1].setPower(100);
robot.wait(2000);
```

Пример в TRIK Studio

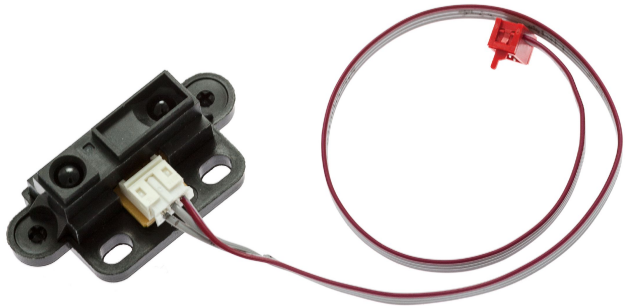


ООО «КиберТех»
Санкт-Петербург, 2020

Ветвление. Задача

Задача 2.1.6. вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко, на 3 секунды или дольше. За границу считать значение 50 ИК датчика.

Инфракрасный датчик расстояния — аналоговый датчик для измерения расстояния.
Рабочий диапазон: 10–80 см.

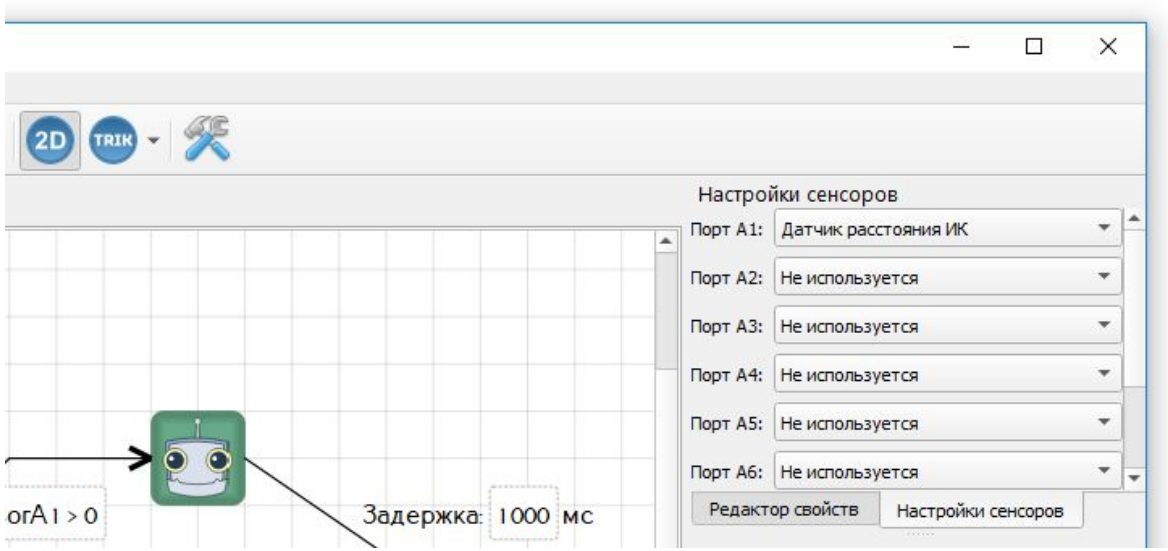


В TRIK Studio все датчики подключаются на панели **«Настройка сенсоров».**

Ветвление. Задача

Задача 2.1.6. вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко, на 3 секунды или дольше. За границу считать значение 50 ИК датчика.

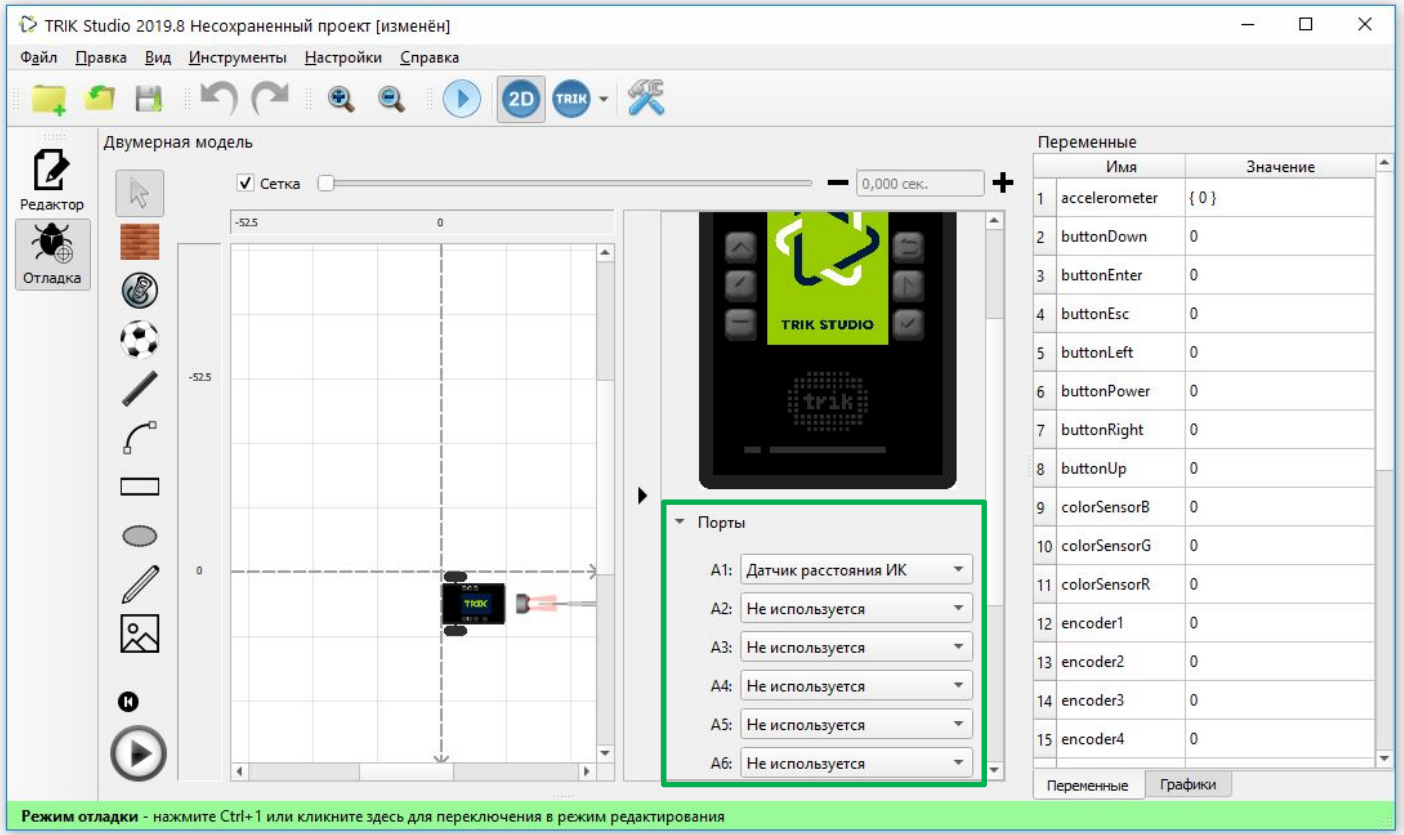
В TRIK Studio все датчики подключаются на панели **«Настройка сенсоров».**



Ветвление. Задача

Задача 2.1.6. вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко, на 3 секунды или дольше. За границу считать значение 50 ИК датчика.

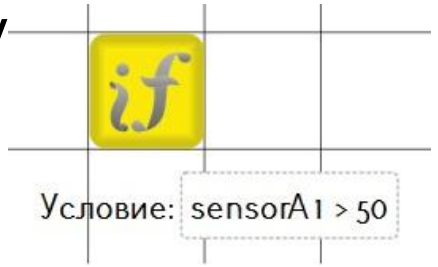
Или в режиме 2D-модели на панели **«Порты»**.



Ветвление. Задача

Задача 2.1.6. вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко, на 3 секунды или дольше. За границу считать значение 50 ИК датчика.

Для ветвления в TRIK Studio используется блок **«Условие»**, у которого имеется только одно свойство — само условие.



Использование значений датчика осуществляется в TRIK Studio через **сенсорные переменные**.

При подключении датчика:

- к порту A1 используется сенсорная переменная **sensorA1**
- к порту A2 — **sensorA2**
- и т.д.

Сенсорной переменной нельзя присвоить значение. В нее записывается регулярно показание с датчика

Операторы

TRIK

Для задания различных условий роботу необходимы операторы сравнения и логические операторы.

Операторы сравнения

Оператор	Синтаксис	Пример
равенство	==	enterButton == 1
неравенство	!=	rightButton != 0
больше	>	sensorA1 > 50
меньше	<	sensorA2 < 30
больше или равно	>=	sensorA3 >= 50
меньше или равно	<=	sensorA4 <= 50

Операторы

TRIK

Для задания различных условий роботу необходимы операторы сравнения и логические операторы.

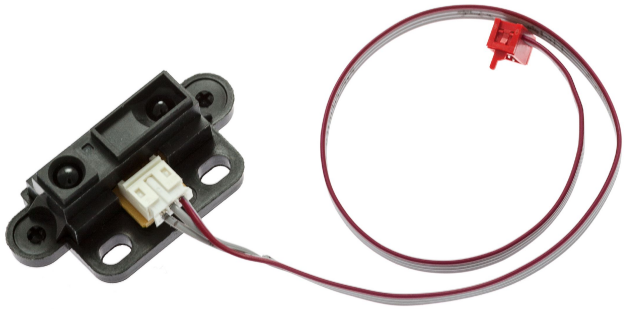
Логические операторы

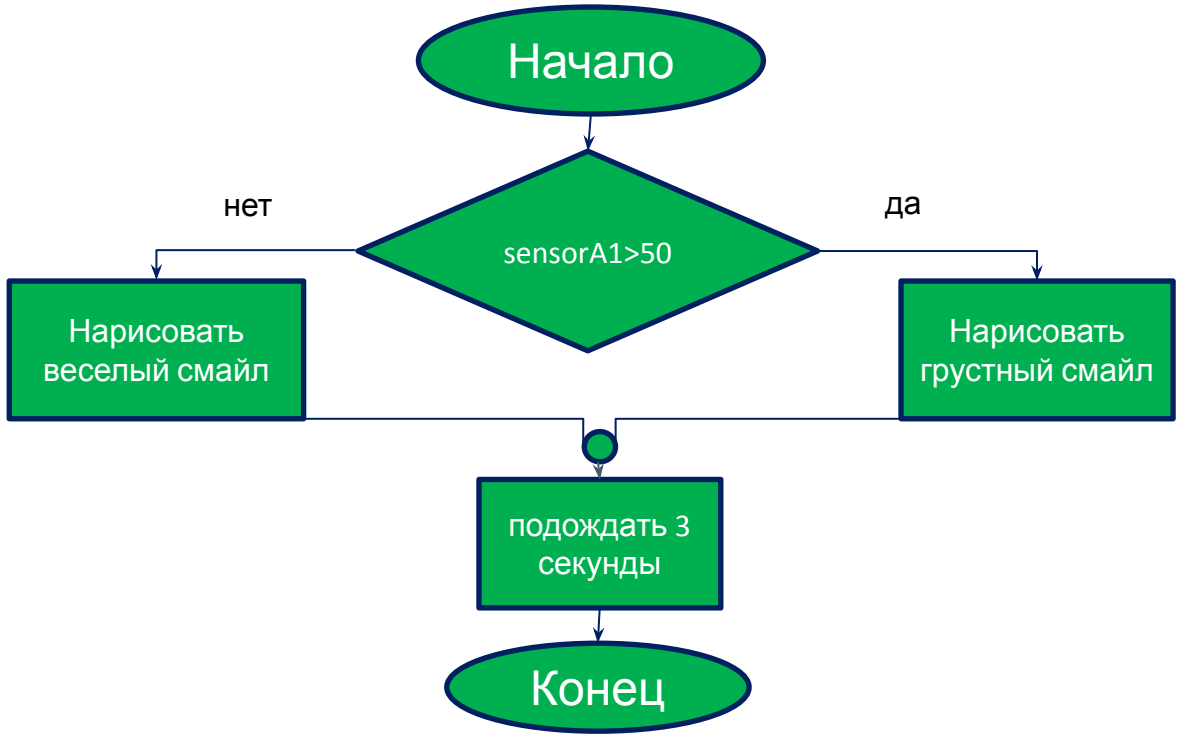
Оператор	Синтаксис	Пример
логическое отрицание, НЕ	!	!flag
логическое умножение, И	&&	(sensorA1>20) && (sensorA1<60)
логическое сложение, ИЛИ	||	(sensorA1<30) || (sensorA1>70)

Ветвление. Задача

Задача 2.1.6. вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко, на 3 секунды или дольше. За границу считать значение 50 ИК датчика.

Инфракрасный датчик расстояния — аналоговый датчик для измерения расстояния.
Рабочий диапазон: 10–80 см.





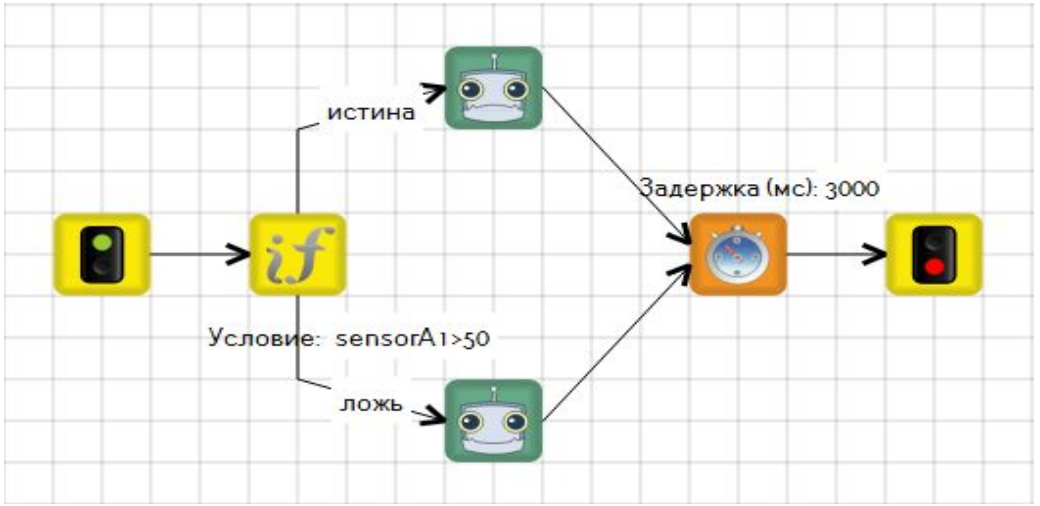
Ветвление. Задача

Задача 2.1.6. вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко, на 3 секунды или дольше. За границу считать значение 50 ИК датчика.

Псевдокод

```
if (robot.sensor.[A1].read() > 50)
    robot.sadSmile();
else
    robot.smile();
robot.wait(3000);
```

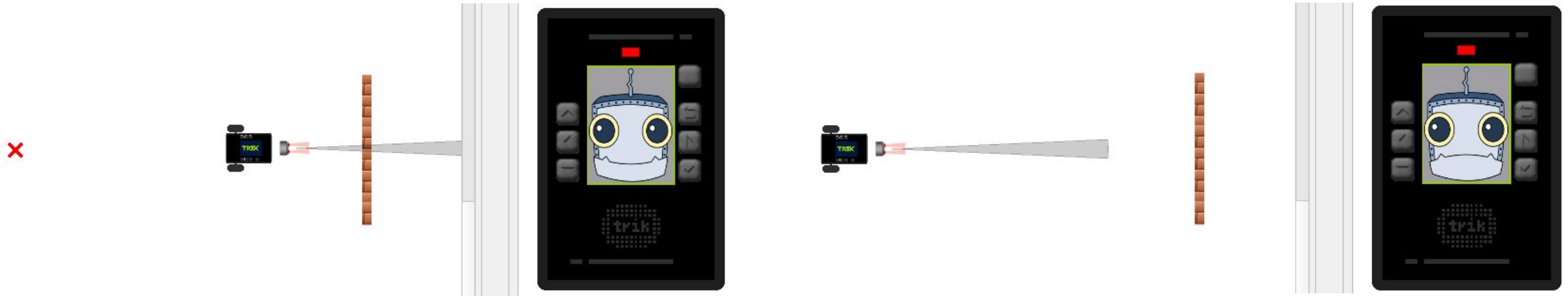
Решение в TRIK Studio



На связях, идущих от условия, указывается в свойствах **истина** и **ложь** для определения дальнейших действий, когда условие верно, и когда — нет.

Ветвление. Задача

Для проверки задачи используйте 2 разных поля: на одном стена близко к роботу, на другом - далеко.

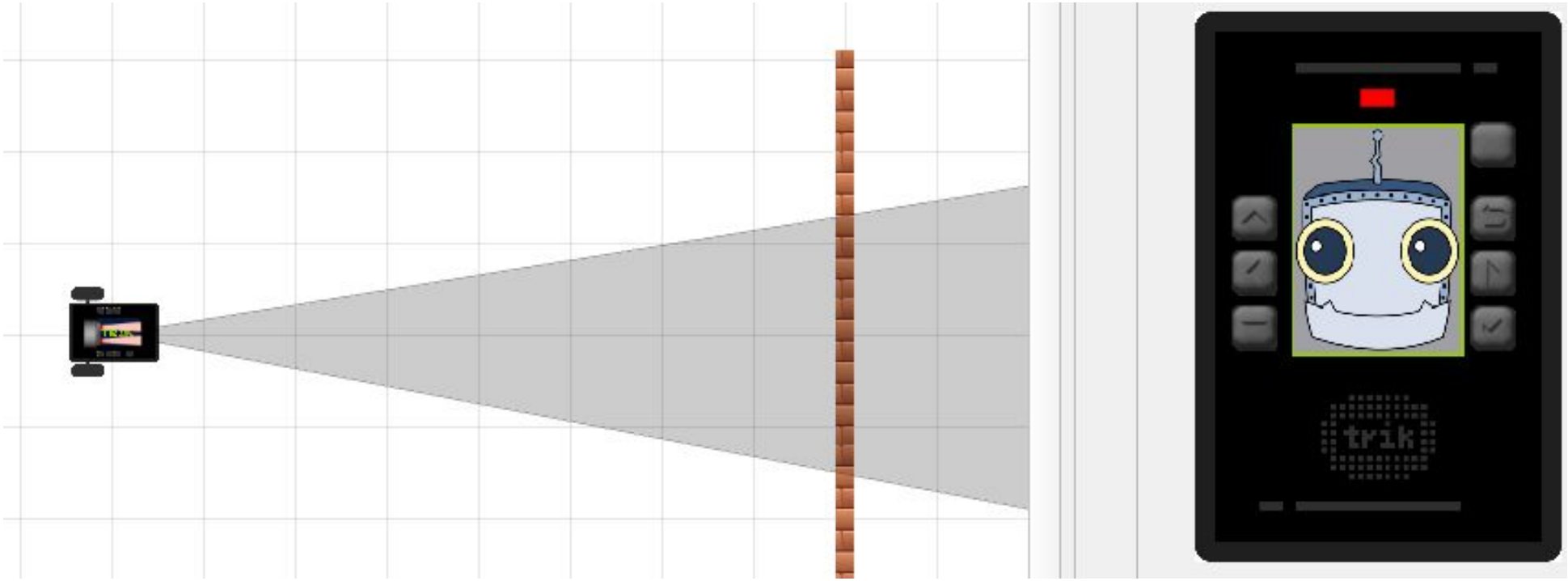


Для проверки можно использовать поля **2.1.6 – field1.xml**, **2.1.6 – field2.xml**

Ветвление. Задача

Задача 2.1.7 (самостоятельно) Вывести на экран:

- Веселый смайлик, если ИК датчик выдает до 40.
- Слово «неопределенность», если ИК датчик выдает от 40 до 60.
- Грустный смайлик — в противном случае.

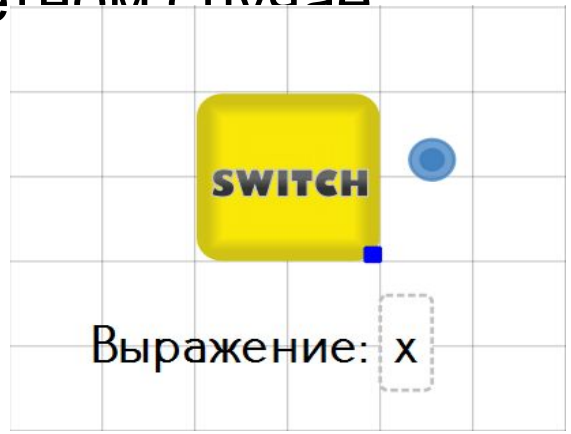


Switch

TRIK

Представляет собой структуру, построенную по принципу меню, и содержит все возможные варианты условий и инструкции, которые следует выполнить в каждом конкретном случае.

В TRIK Studio реализуется с помощью одноименного блока:

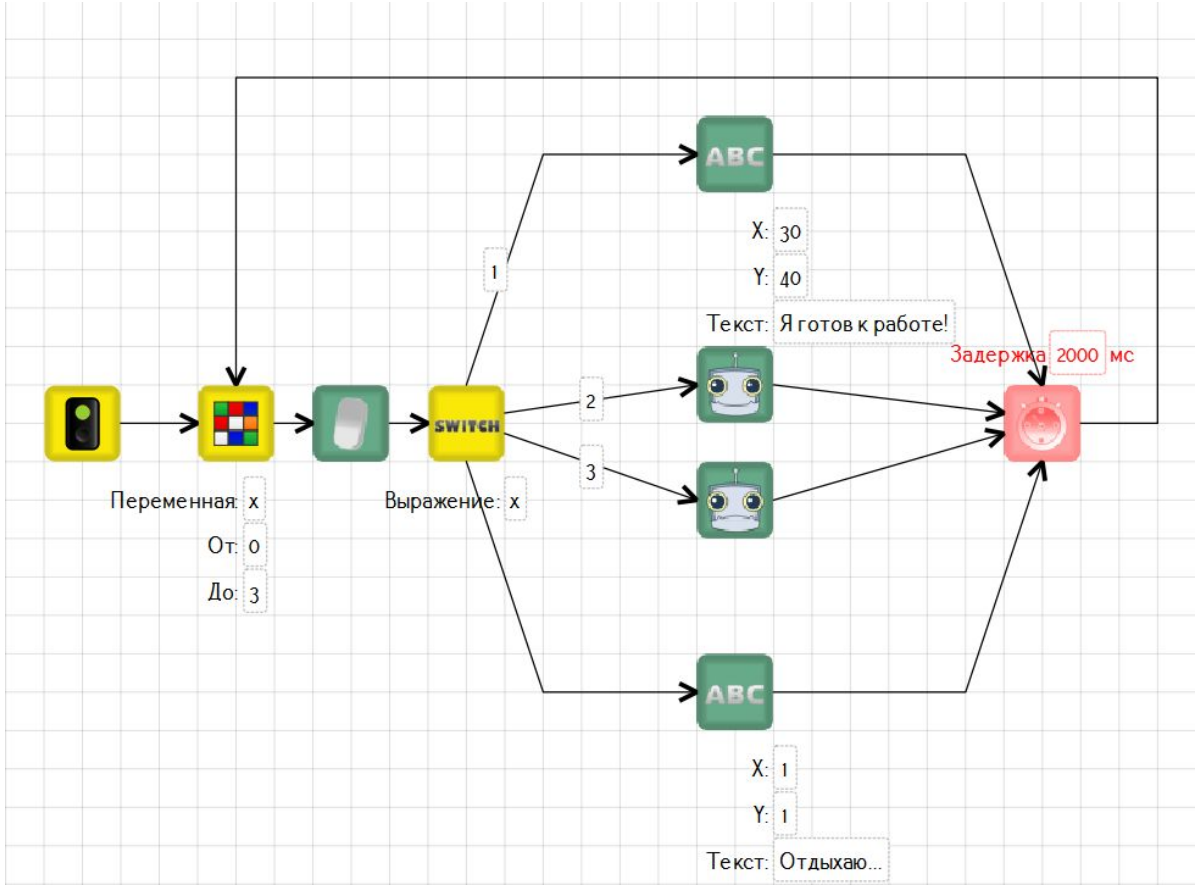


Блок проверяет выражение. От блока отводятся связи, на которых указываются возможные значения этого выражения (например, переменной). Одна связь **обязательно** должна быть пустая (“default”) - по ней алгоритм будет двигаться, если не выполнено ни одно из условий.

Switch

TRIK

Данный пример демонстрирует случайный выбор одного из четырех состояний робота: «Я готов к роботе», «Улыбаюсь», «Грущу», «Отдыхаю…»



Switch

TRIK

Написать программу, делающую случайный выбор одного из восьми состояний робота:
«Я готов к роботе»,
«Улыбаюсь»,
«Грущу»,
«Отдыхаю…»
Езда вперед 1 секунду
Езда назад 1 секунду
Поворот налево 1 секунду
Поворот направо 1 секунду

Switch

Задача 2.1.8. Выводить в цикле с задержкой минимум в 100 мс на экран робота в 2D модели коды кнопок контроллера ТРИК, по нажатию на них.

В TRIK Studio имеется блок **«Получить код кнопки»**, который записывает код нажатой кнопки в переменную. Все коды кнопок представлены в кодировке ASCII.

Зная коды кнопок, с помощью **switch** можно написать своё меню.

Switch

TRIK

Задача 2.1.9. (самостоятельно)

Выполнять в цикле действия по нажатию клавиш:
«вверх» (103) — крутить моторами вперед
«вниз» (108) — крутить моторами назад
«влево» (105) — поворачивать влево
«вправо» (106) — поворачивать вправо
«ввод» (28) — улыбаться и говорить «Привет»
«Esc» (1) — выход из программы

Цикл

TRIK

Цикл — управляющая конструкция в языках программирования для организации многократного выполнения набора инструкций.

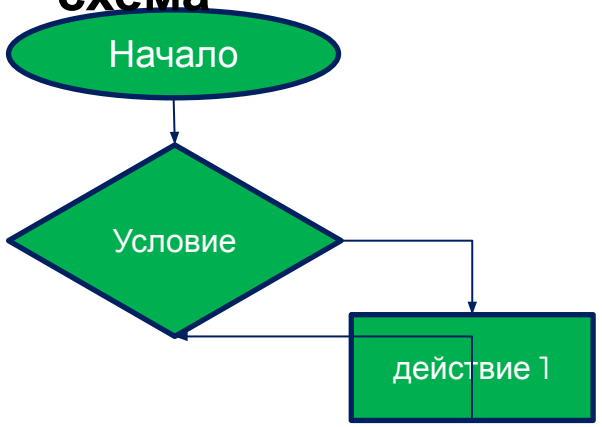
Существует 4 основных вида циклов

- Безусловные циклы
- Цикл с предусловием
- Цикл с постусловием
- Цикл со счетчиком

Цикл безусловный

TRIK

Блок-схема

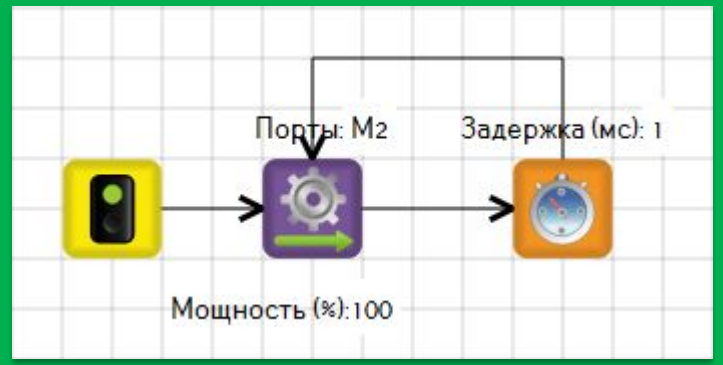


Псевдокод

```
while true do
    robot.motor.[M2].setPower(100);
```

В этом случае у программы может не быть конца!

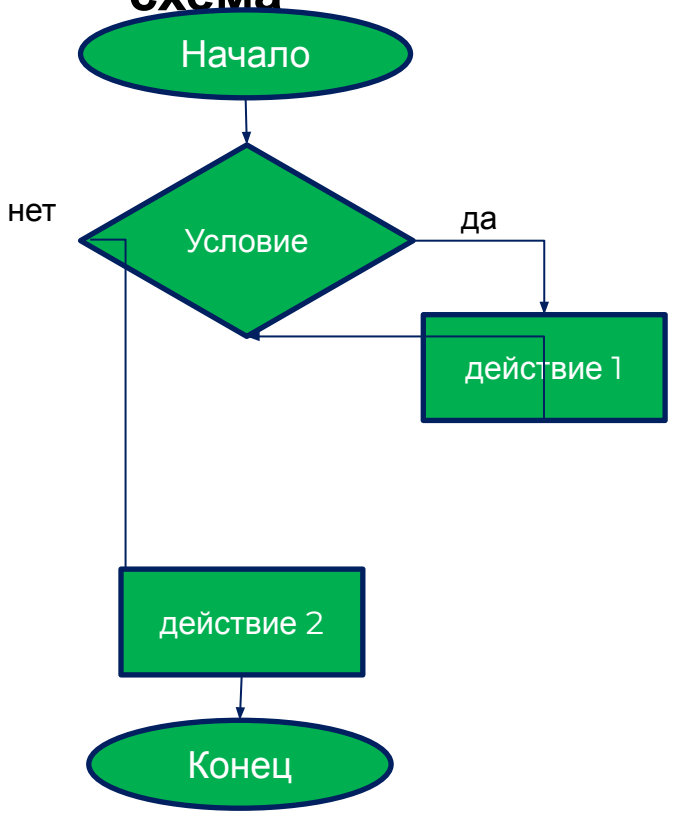
Пример в TRIK Studio



Цикл с предусловием

TRIK

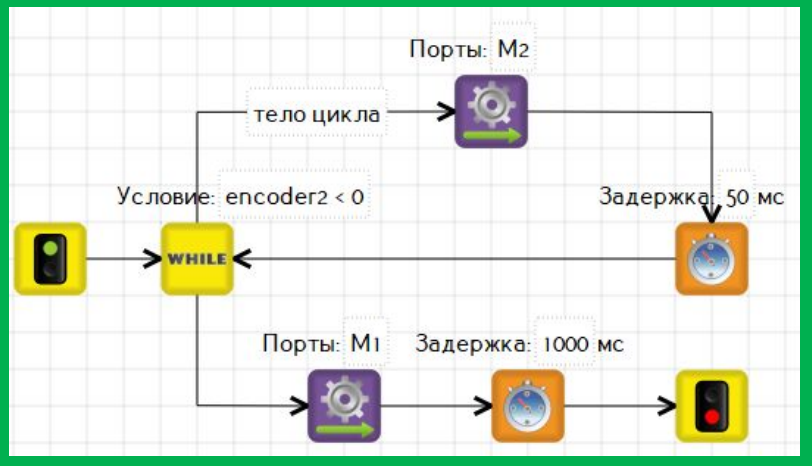
Блок-схема



Псевдокод

```
while encoder.[E2].read() < 500 do
        robot.motor.[M2].setPower(100);
robot.motor.[M1].setPower(100);
```

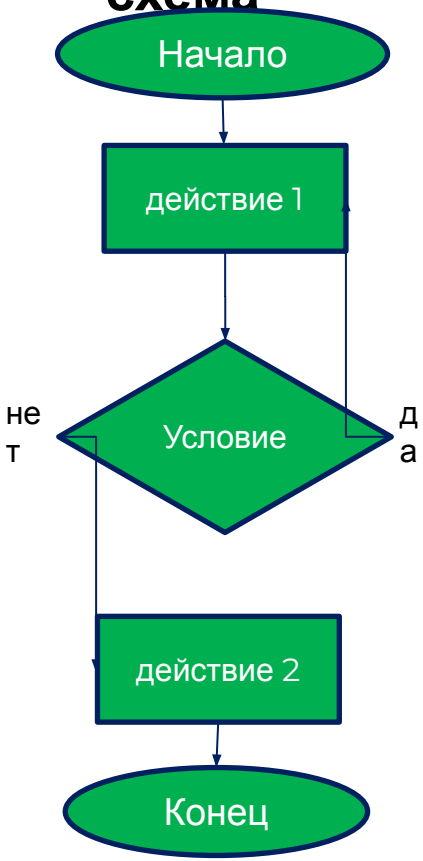
Пример в TRIK Studio



Цикл с постусловием

TRIK

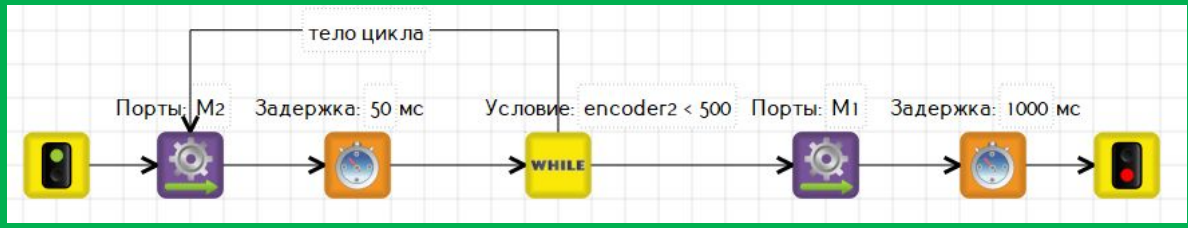
Блок-схема



Псевдокод

```
do
    robot.motor.[M2].setPower(100);
    robot.wait(1);
while encoder.[E2].read() < 500
robot.motor.[M1].setPower(100);
```

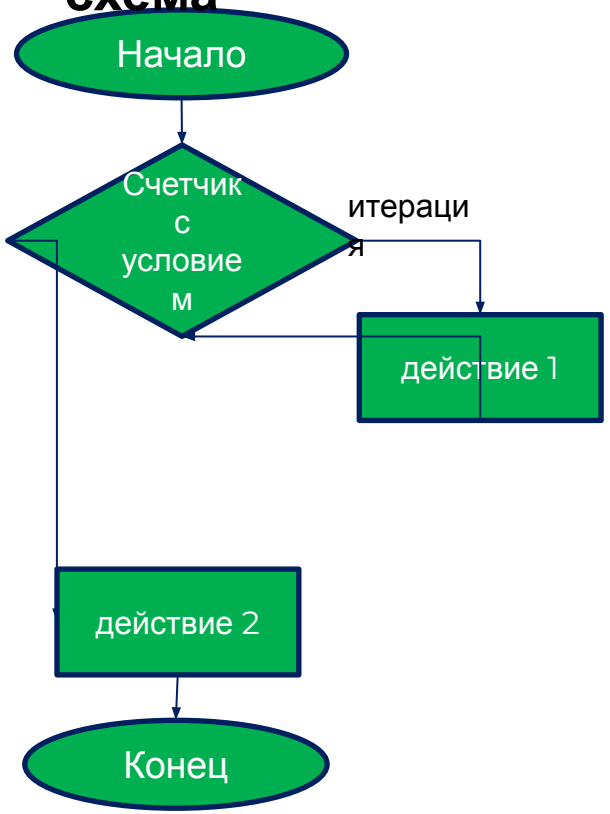
Пример в TRIK Studio



Цикл с итерациями

TRIK

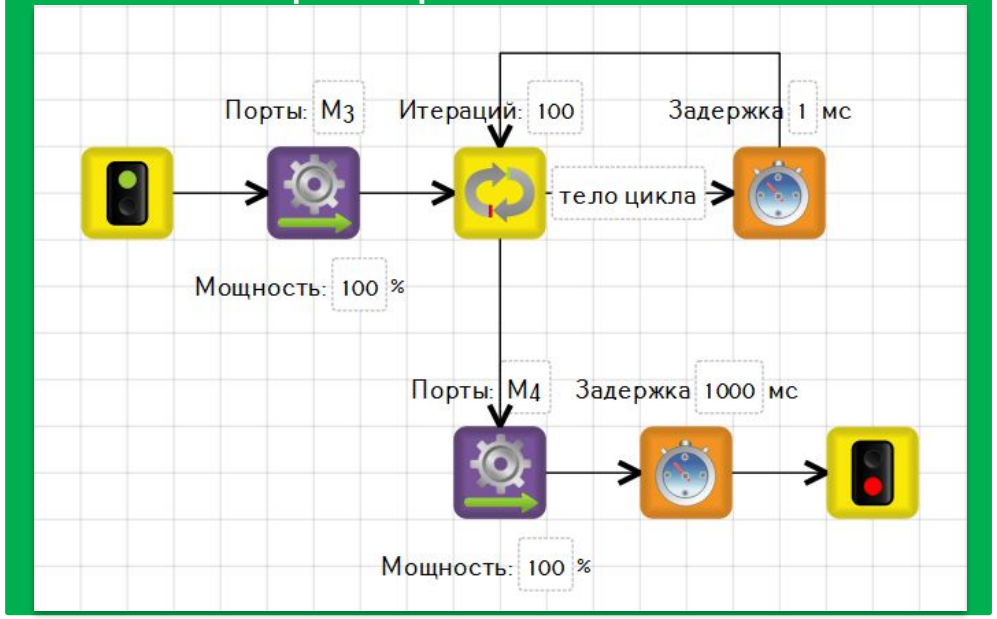
Блок-схема



Псевдокод

```
robot.motor.[M3].setPower(100);
for (i = 0; i < 1000; i++)
    robot.wait(1);
robot.motor.[M4].setPower(100);
```

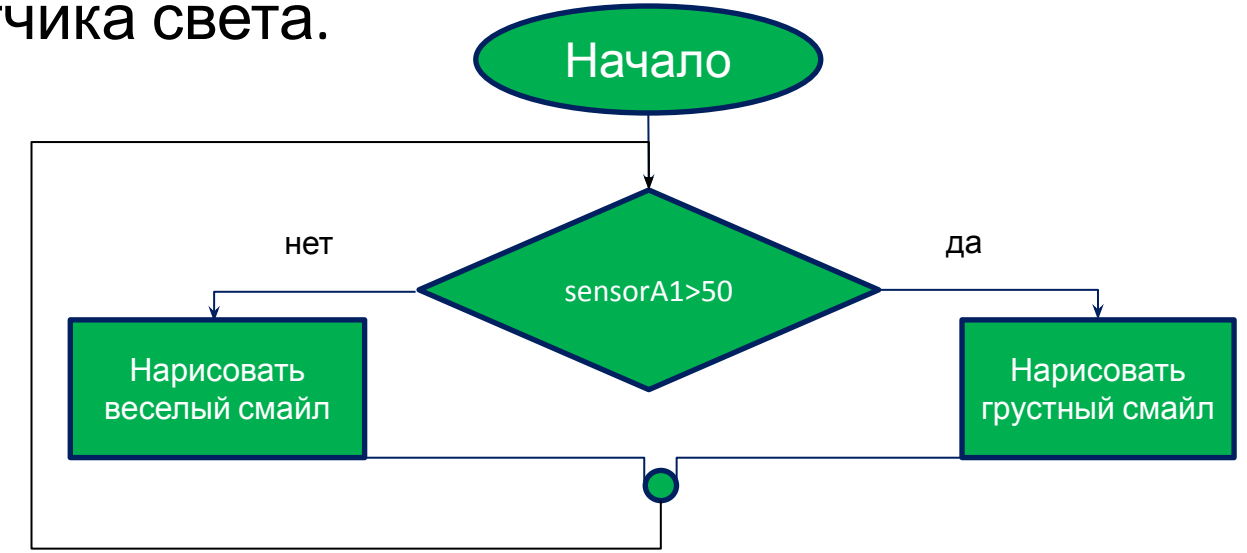
Пример в TRIK Studio



Цикл. Задача

TRIK

Задача 2.1.10. «Настроение робота» Робот двигается прямо через черные и белые поля. Непрерывно выводить на экран веселый смайлик, если робот на черном поле, и грустный, если на белом. За границу считать значение 50 датчика света.



Датчик освещенности – аналоговый датчик для измерения освещенности. Выдает значение от 0 до 100.

Цикл. Задача

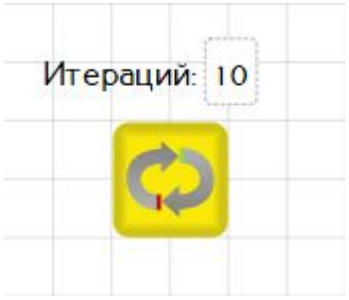
Задача 2.1.10. «Настроение робота» Робот двигается прямо через черные и белые поля. Непрерывно выводить на экран веселый смайлик, если робот на черном поле, и грустный, если на белом. За границу считать значение 50 датчика света.

Цикл. Задача

Задача 2.1.10. «Настроение робота» Робот двигается прямо через черные и белые поля. Непрерывно выводить на экран веселый смайлик, если робот на черном поле, и грустный, если на белом. За границу считать значение 50 датчика света.

Для циклов с условиями в TRIK Studio используется блок **«Цикл с предусловием»**.

А с итерациями — блок **«Цикл»**.



Бесконечные циклы реализуются путем соединения одного из блоков с каким-либо предыдущим.

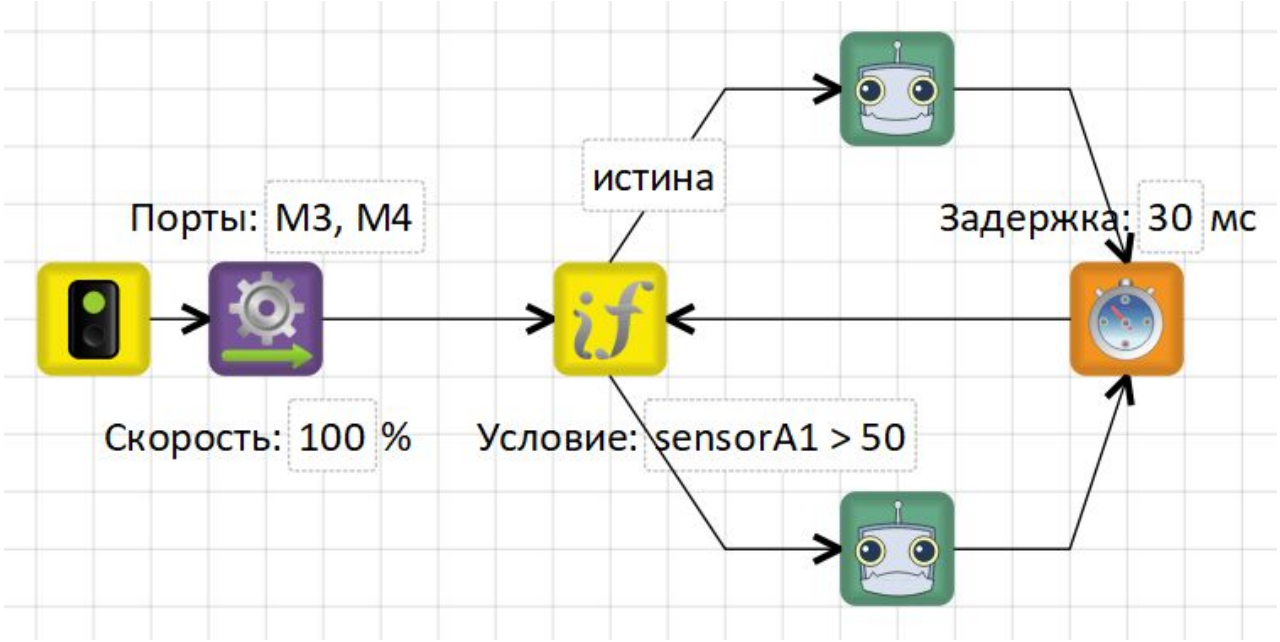
Цикл. Задача

Задача 2.1.10. «Настроение робота» Робот двигается прямо через черные и белые поля. Непрерывно выводить на экран веселый смайлик, если робот на черном поле, и грустный, если на белом. За границу считать значение 50 датчика света.

Псевдокод

```
robot.motor.[M3].setPower(100);
robot.motor.[M4].setPower(100);
while true do
    if (robot.sensor.[A1].read() > 50)
        robot.smile();
    else
        robot.sadSmile();
    robot.wait(30);
```

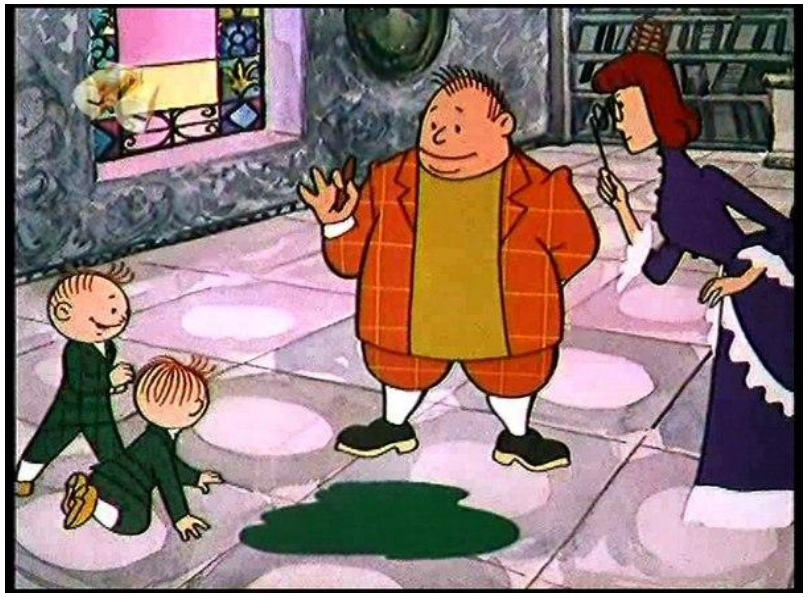
Решение в TRIK Studio



Цикл. Задача

TRIK

Задача 2.1.11 (самостоятельно): «Кентервильское привидение».



Кентервильский робот: привидение рисует каждую ночь лужи красной краской. Убедившись, что лужа красная, он довольный скрывается из виду. Когда красная краска заканчивается, он рисует лужи зеленым и расстроенный отключается.

Научите робота определять цвет лужи и выключаться, если лужа зеленая. В первый раз робот всегда в приподнятом настроении.

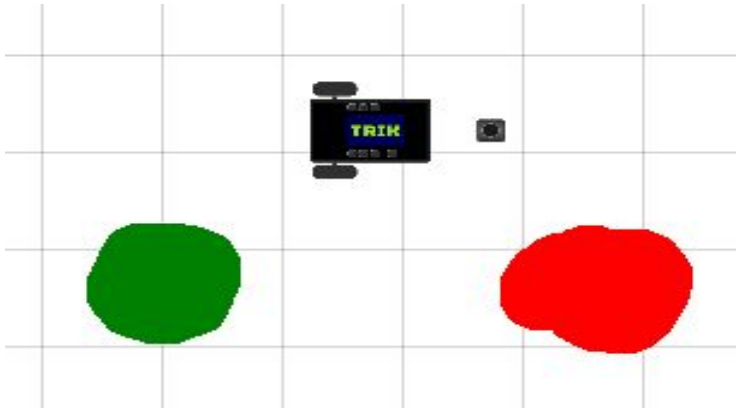
Цикл. Задача

TRIK

Задача 2.1.11 (самостоятельно): «Кентервильское привидение»

Пояснение.
Выводить на экран:

- Веселый смайлик, если робот видит красную лужу (больше 72) или пустой пол (меньше 5)
- Грустный смайлик (в течение 3 секунд) — в противном случае (зеленая лужа: от 59 до 69). И закончить выполнение программы.



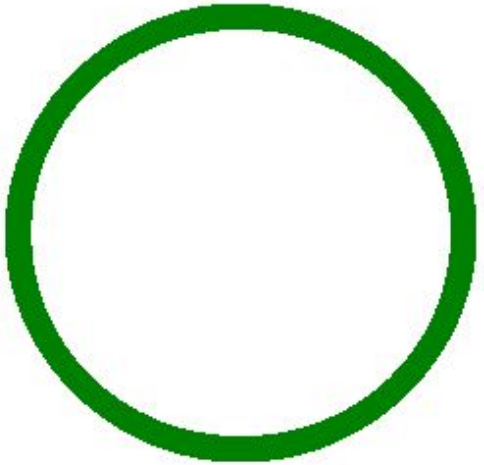
Считывать новое значение с датчика **каждую секунду.** Использовать блок **«Цикл с предусловием»**

Цикл. Задача

TRIK

Задача 2.1.12 (самостоятельно) «Разгон и торможение»

Напишите программу: плавный разгон робота от 0 до 100 в течение 2 секунд, а затем плавное торможение от 100 до 0 в течение 3 секунд. Использ

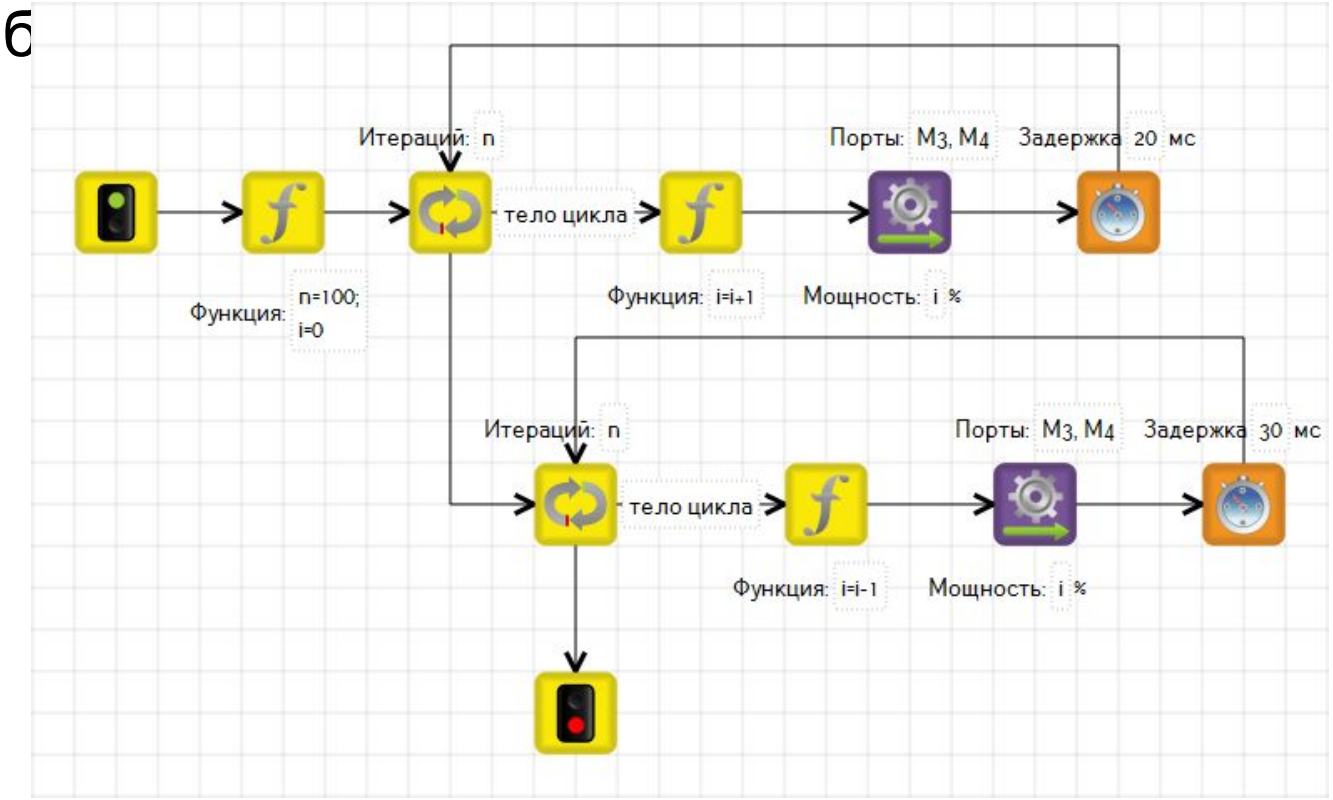


Цикл. Задача

TRIK

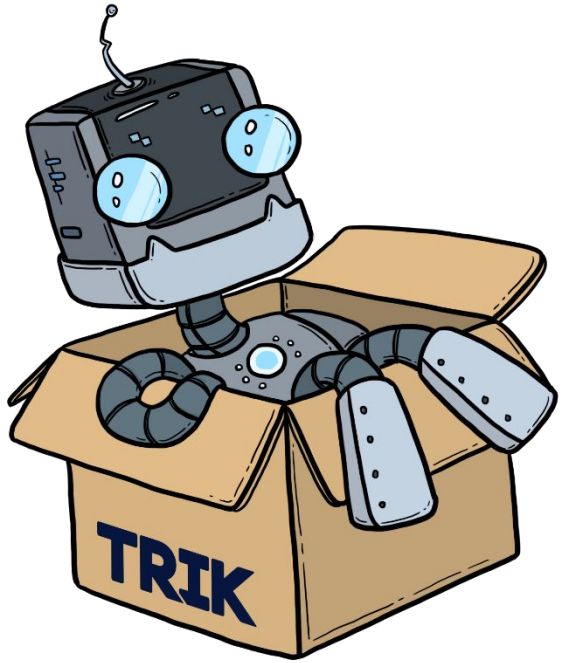
Задача 2.1.12 (самостоятельно) «Разгон и торможение»

Напишите программу: плавный разгон робота от 0 до 100 в течение 2 секунд, а затем плавное торможение от 100 до 0 в течение 3 секунд. Используйте б



Информация и контакты

trikset.com



Поддержка ТРИК:
support@trikset.com

Справочный центр ТРИК:
help.trikset.com

trikset