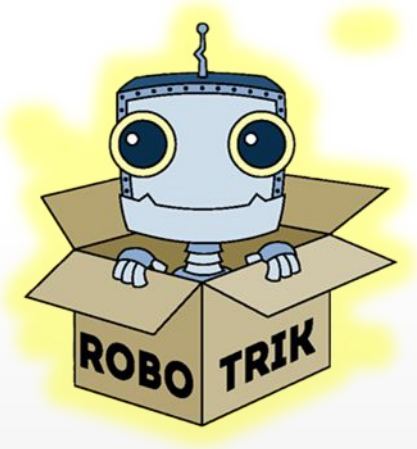
ТРИК

НАУЧИМ ИЗОБРЕТАТЬ БУДУЩЕЕ

Все материалы по TRIK Studio будут обновлены в версии 3.2.0

Элементарные действия
Алгоритмические структуры



Широколобов И. Ю.
ilya.shirokolobov@gmail.com

Эти материалы распространяются по лицензии Creative Commons «Attribution-NonCommercial-ShareAlike» («Атрибуция — Некоммерческое использование — На тех же условиях») 3.0 Непортированная. Чтобы ознакомиться с экземпляром этой лицензии, посетите http://creativecommons.org/licenses/by-nc-sa/3.0/ или отправьте письмо на адрес Creative Commons: 444 Castro Street, Suite 900, Mountain View, California, 94041, USA

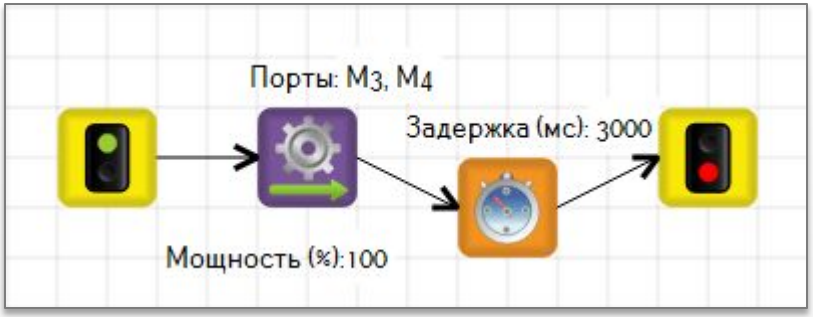
Санкт-Петербург, 2019

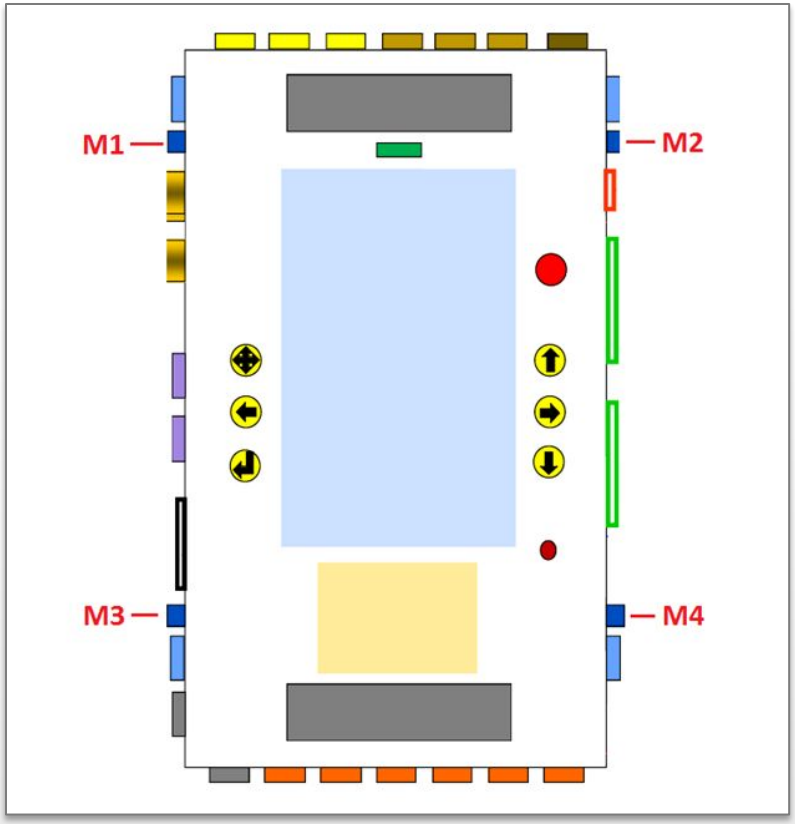
Движение вперед

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

В ТРИК Студии для подачи мощности на мотор существует отдельный блок «Моторы вперед». У этого блока есть два свойства: порт и мощность

У контроллера ТРИК есть четыре порта для подключения силовых моторов: M1, M2, M3 и M4





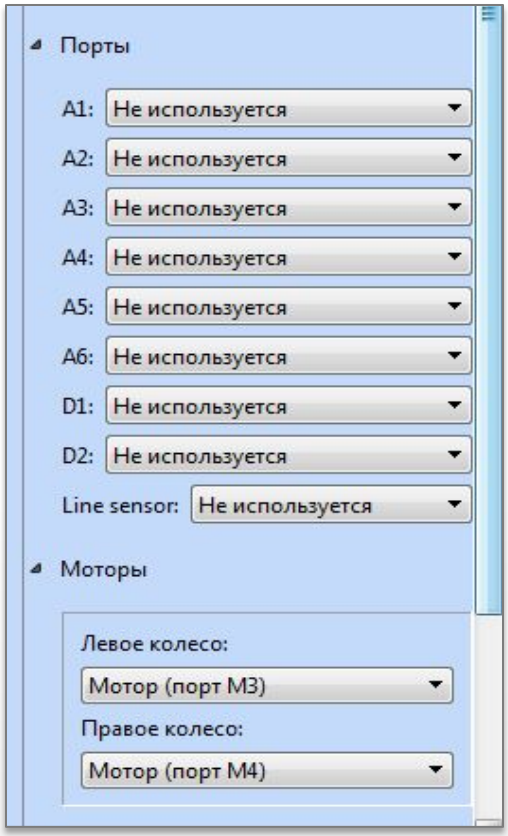
После элементарного действия всегда выставляется какой-либо блок ожидания: таймер, ожидание энкодера и т.д.

CC BY-NC-SA (C) ООО «КиберТех» 2019

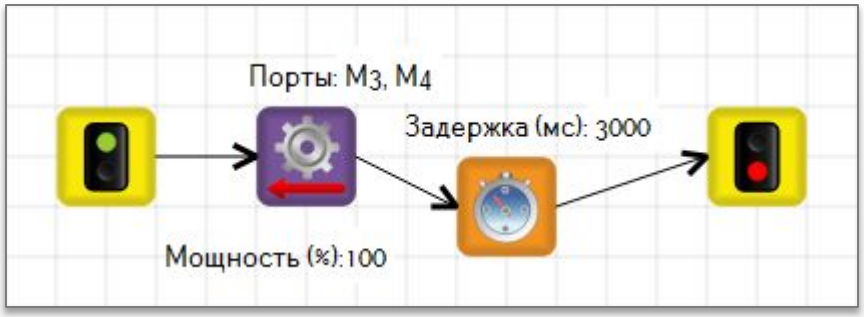
Движение назад

В 2D модели по умолчанию левый мотор подключен к порту М3, правый – М4

На выпадающем меню **Моторы** в 2D модели всегда можно изменить подключение моторов



Аналогично выполняется движение назад. Используется блок «Моторы назад»



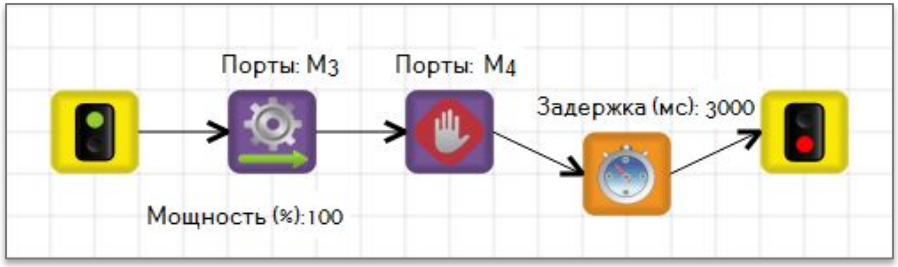
На самом деле, диапазон подаваемой мощности: от -100 до 100 процентов. Таким образом, для движения назад можно использовать и блок «Моторы вперед» (подав мощность -100 %)

CC BY-NC-SA (C) ООО «КиберТех» 2019

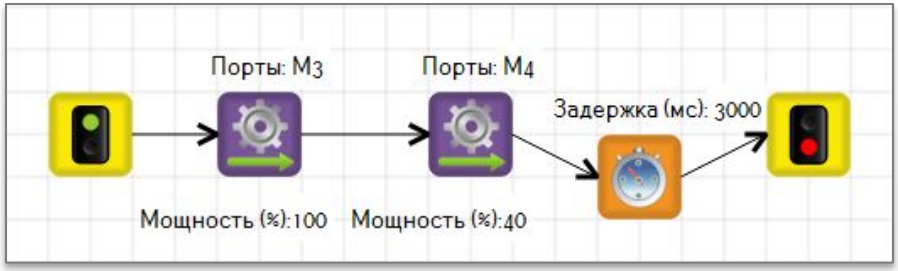
Повороты

Повороты можно разделить на 3 типа:

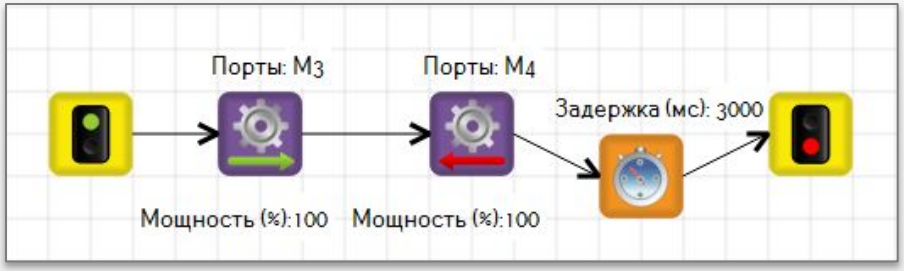
- **резкий поворот**
  мощность подается только на одно колесо



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



- **поворот на месте**

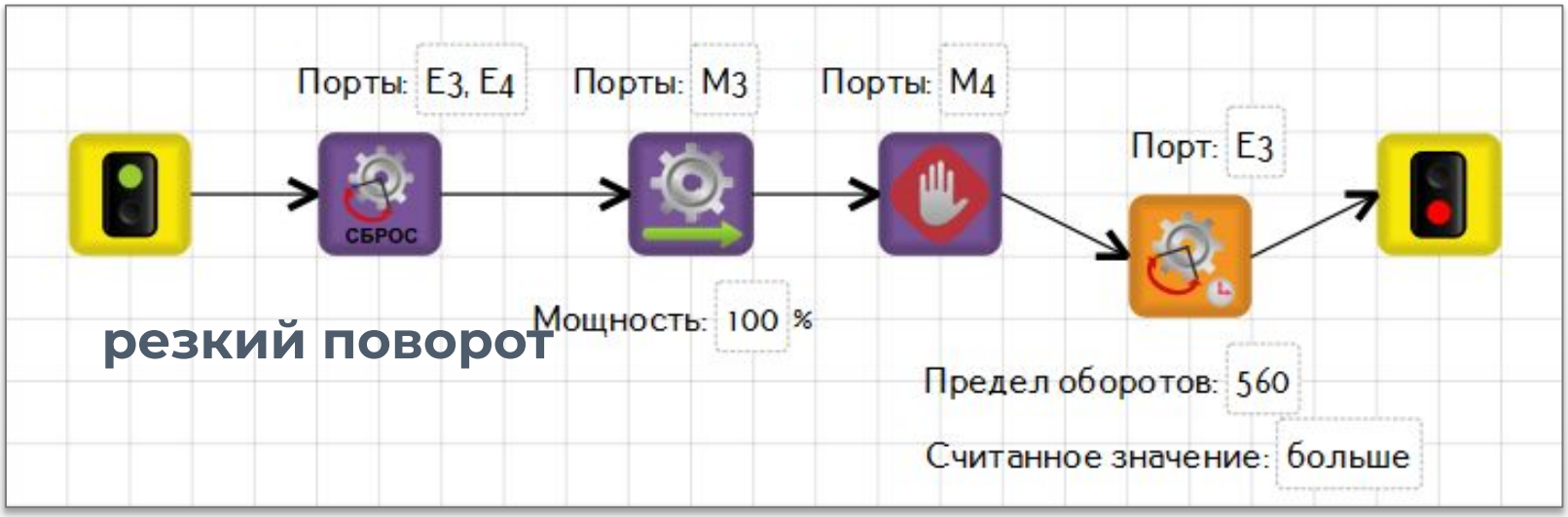


CC BY-NC-SA (C) ООО «КиберТех»
2019

Модели алгоритмов

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

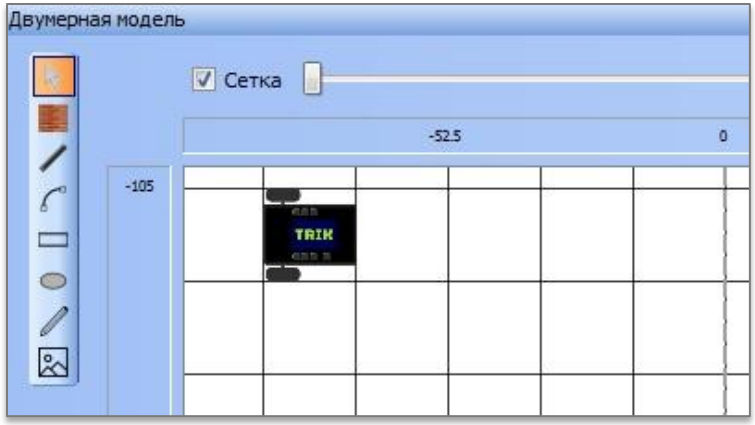


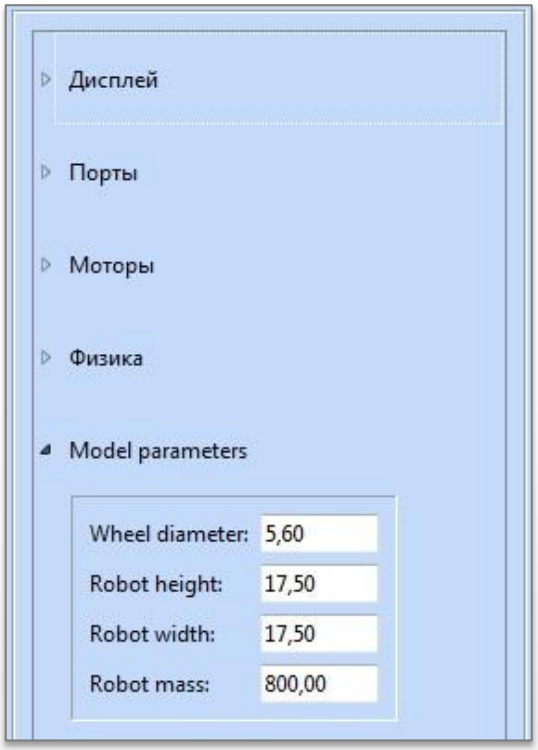
Остальные действия реализуются аналогично

CC BY-NC-SA (C) ООО «КиберТех»
2019

Точные перемещения

Поставьте галочку в окошко **«Сетка»**. Теперь Вы можете отслеживать точные перемещения модели.
(1 клетка = 17,5 см)





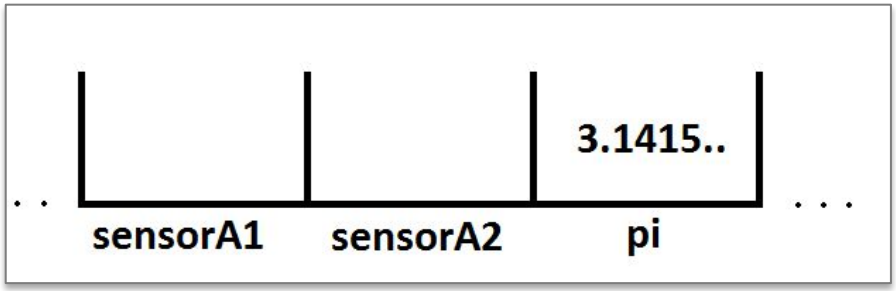
Также, в режиме «отладка» всегда можно посмотреть параметры визуальной модели (**Model parameters**)

Для удобства длина и размер базы робота совпадают с размером клетки (17,5 см)

CC BY-NC-SA (C) ООО «КиберТех» 2019

Переменные

Переменная — поименованная область памяти



В TRIK Studio можно ввести свои переменные, используя блок «Функция». Для объявления и инициализации новой переменной (например, **err**) просто введите в свойства этого блока: имя_переменной = значение (**err** = 70-5)

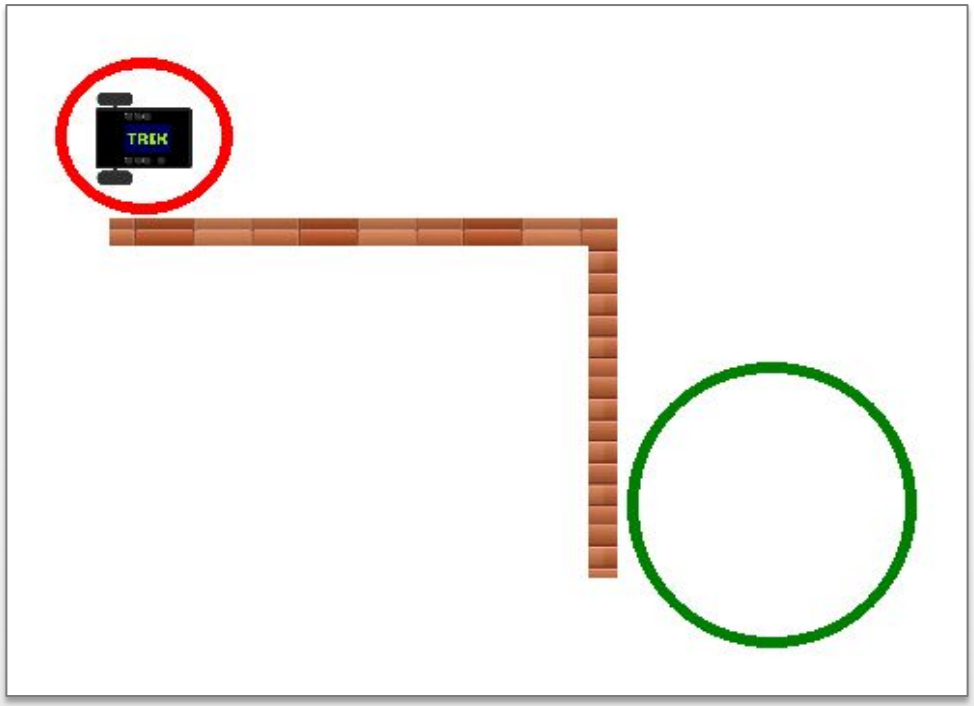
Созданным переменным можно присваивать другие переменные, если последние были объявлены и инициализированы ранее (**u**=5***err)**

CC BY-NC-SA (C) ООО «КиберТех»
2019

Энкодеры – задачи

Задача №1. Проехать вперед, развернуться на 180 градусов, проехать задом. Использовать энкодерную модель

Задача №2. Обогнуть угол. Проехать вперед со скоростью 60, развернуться на 90 градусов, проехать вперед с максимальной скоростью. Использовать энкодерную модель



CC BY-NC-SA (C) ООО «КиберТех»
2019

Точные перемещения – задача

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

Для этого Вам пригодятся следующие параметры:
d = 5,6 см (диаметр колеса), CPR = 360 (полный оборот колеса)

CC BY-NC-SA (C) ООО «КиберТех»
2019

Точные перемещения – решение

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 – периметр (длина) окружности;
en – количество энкодеров

CC BY-NC-SA (C) ООО «КиберТех» 2019

Точные перемещения – задача

Самостоятельная задача. Развернуться на месте ровно на 90 градусов. Использовать энкодерную модель

Для этого Вам пригодятся дополнительный параметр:
b = 17.5 см (база робота)

CC BY-NC-SA (C) ООО «КиберТех»
2019

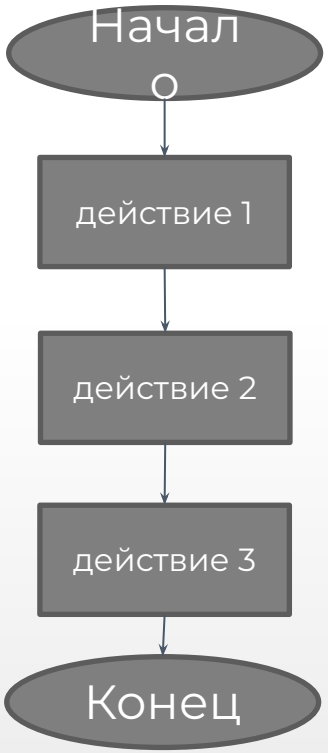
Алгоритм

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



CC BY-NC-SA (C) ООО «КиберТех» 2019

Алгоритмические структуры

Следование. Предполагает последовательное выполнение команд сверху вниз. Если алгоритм состоит только из структур следования, то он является линейным

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

Цикл. Предполагает возможность многократного повторения определенных действий. Количество повторений зависит от условия цикла

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

CC BY-NC-SA (C) ООО «КиберТех»
2019

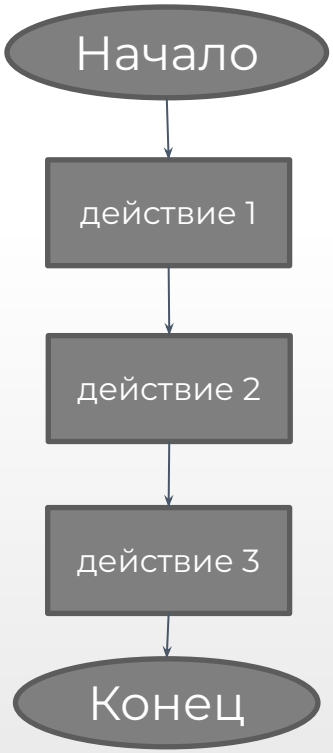
Следование

CC BY-NC-SA (C) ООО «КиберТех»
2019

Следование

Следование. Предполагает последовательное выполнение команд сверху вниз. Если алгоритм состоит только из структур следования, то он является линейным

Блок-схема



Псевдокод

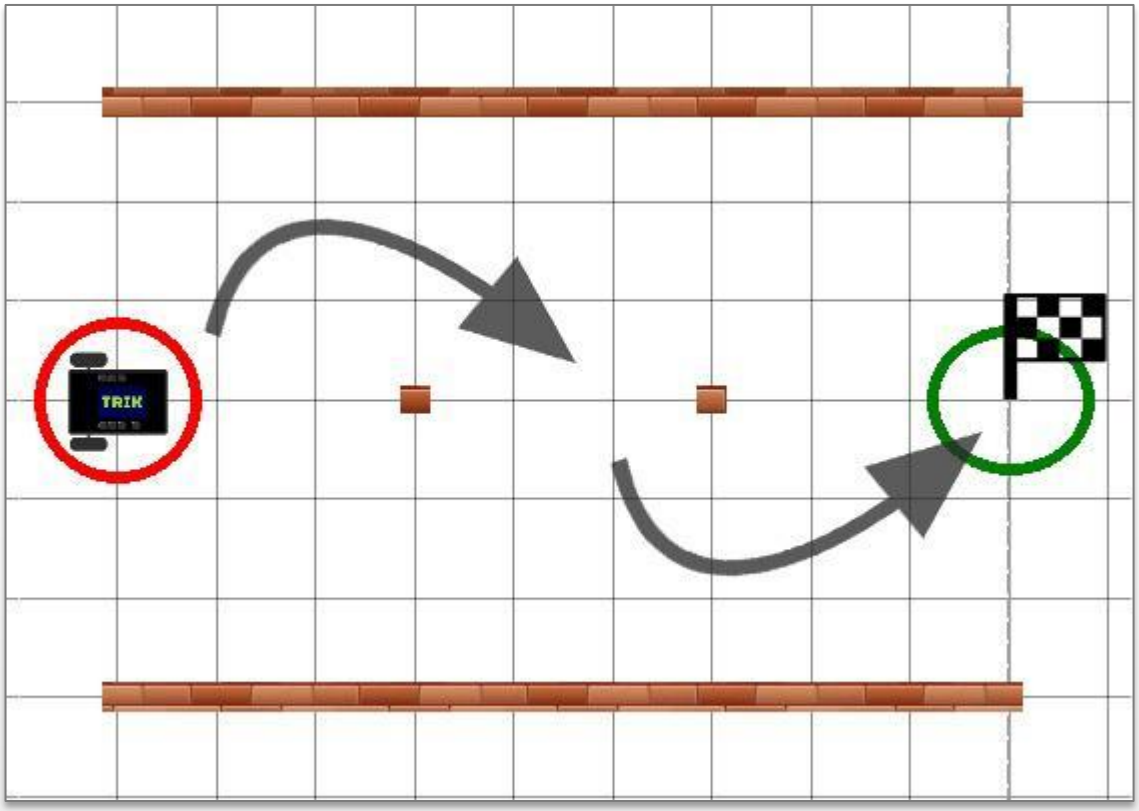
```
speed=-100;
robot.motor.[M2].setPower(100);
robot.motor.[M3].setPower(speed);
robot.wait(1500)
```

Например, любое элементарное действие — это структура следования

CC BY-NC-SA (C) ООО «КиберТех» 2019

Следование – задача

Задача. Написать алгоритм движения модели «змейкой». Использовать энкодерную модель



CC BY-NC-SA (C) ООО «КиберТех»
2019

Ветвление

CC BY-NC-SA (C) ООО «КиберТех»
2019

Ветвление

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

Существует 4 типа ветвления:

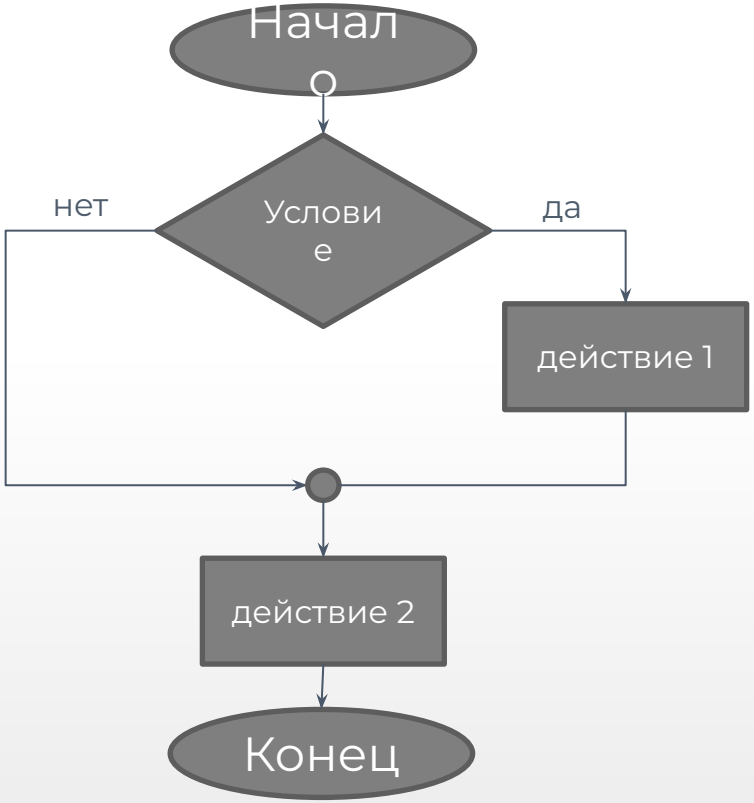
- если-то
- если-то-иначе
- выбор
- выбор-иначе

CC BY-NC-SA (C) ООО «КиберТех»
2019

Ветвление

Ветвление «если-то»

Блок-схема



Псевдокод

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

Пример в TRIK Studio

CC BY-NC-SA (C) ООО «КиберТех»
2019

Ветвление

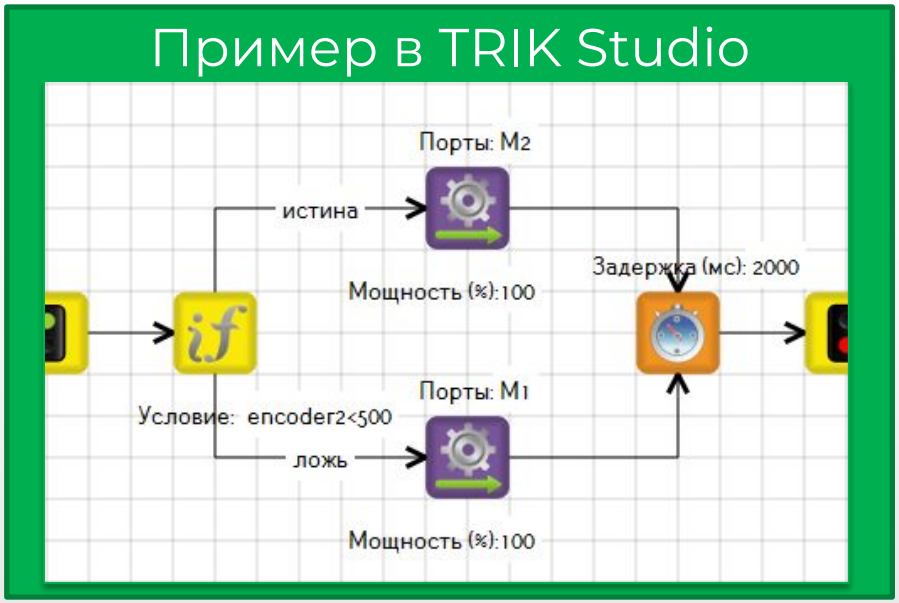
Ветвление «если-то-иначе»

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

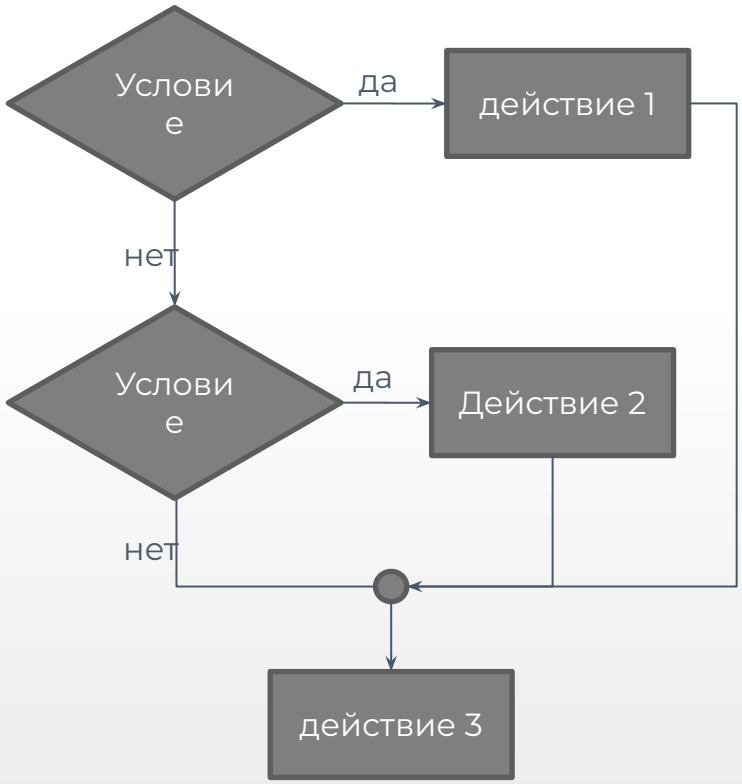


CC BY-NC-SA (C) ООО «КиберТех» 2019

Ветвление

Ветвление «выбор»

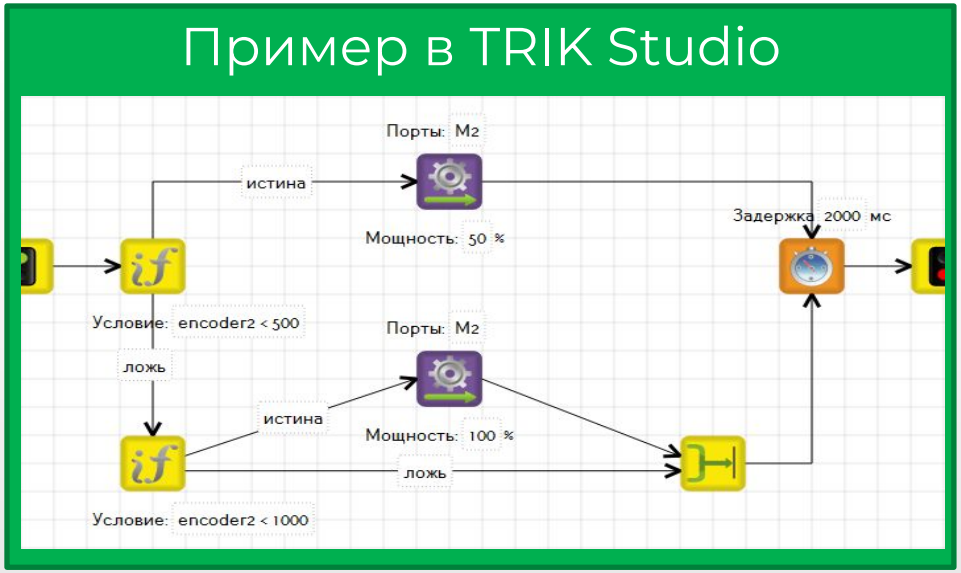
Блок-схема



Псевдокод

```
if (encoder.[E2].read() < 500)

robot.motor.[M2].setPower(50);
elseif

robot.motor.[M2].setPower(100);
robot.wait(2000);
```

Пример в TRIK Studio

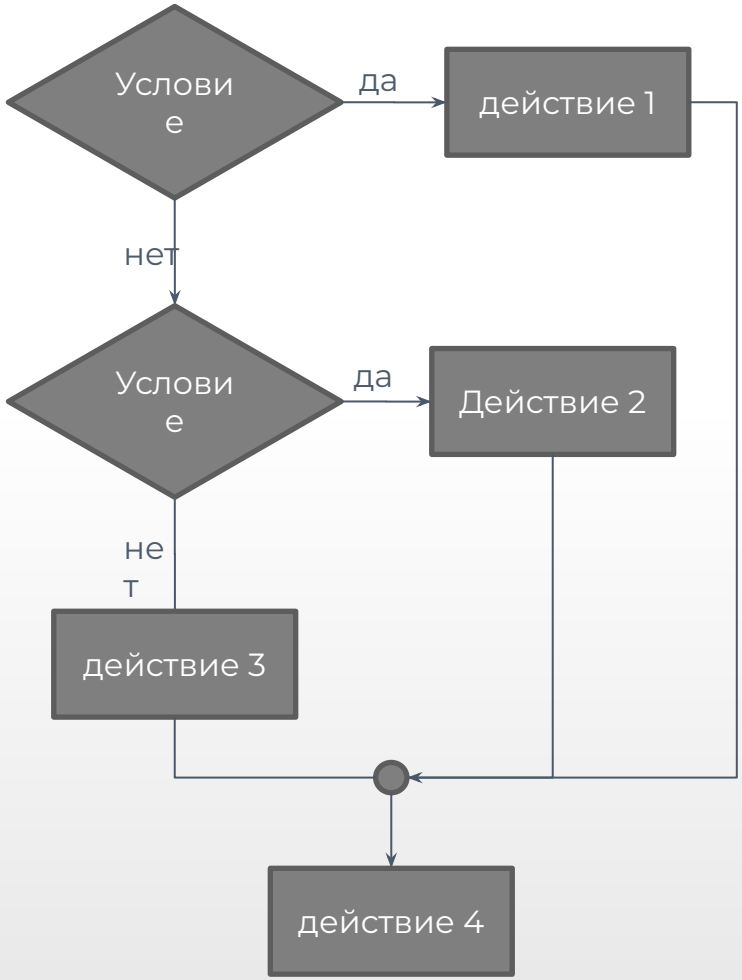


CC BY-NC-SA (C) ООО «КиберТех» 2019

Ветвление

Ветвление «выбор-иначе»

Блок-схема



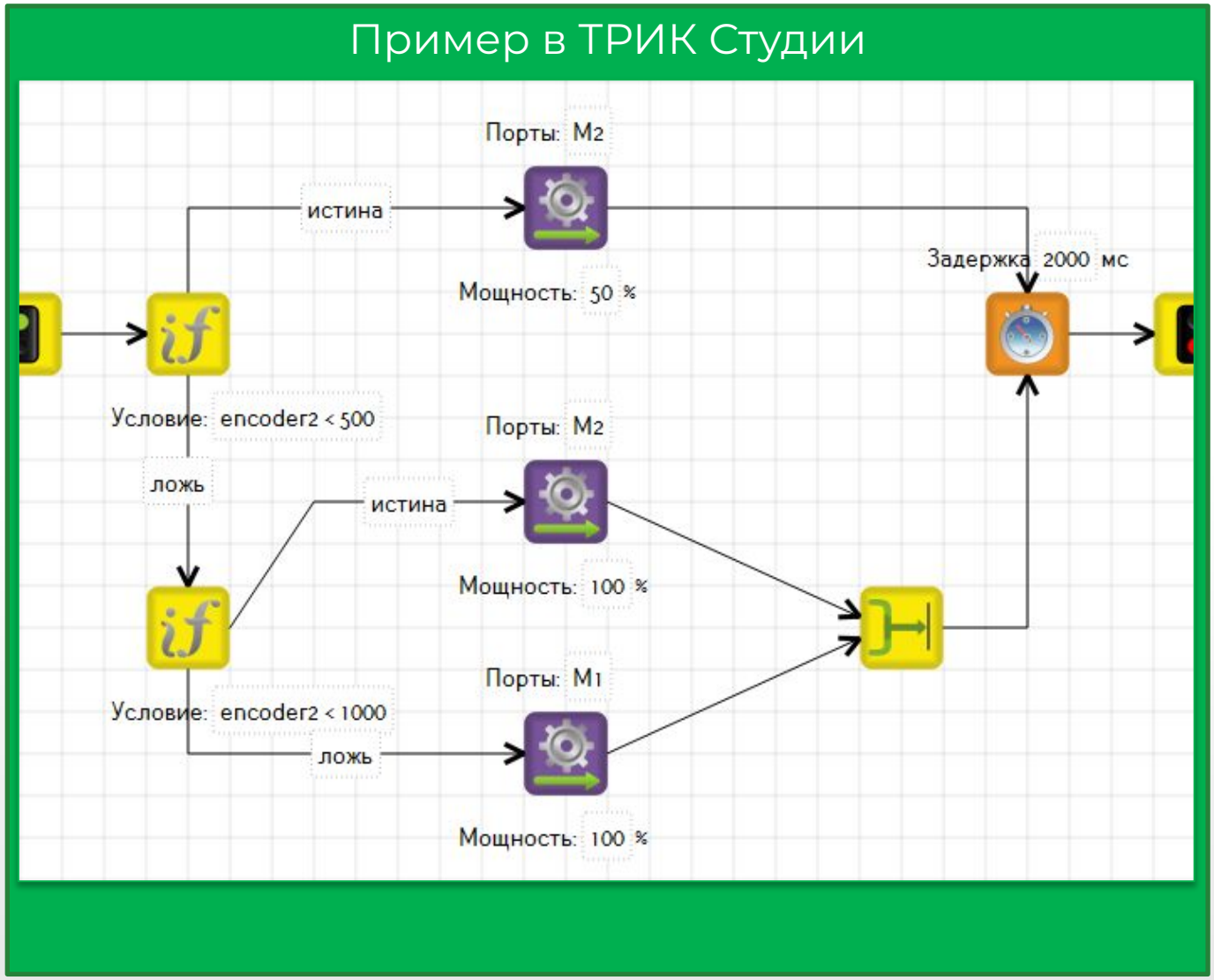
Псевдокод

```
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);
```

CC BY-NC-SA (C) ООО «КиберТех»
2019

Ветвление

Ветвление «выбор-иначе»



Ветвление

Задача: вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко; за границу считать значение 50 ИК датчика

Инфракрасный датчик расстояния — аналоговый датчик для измерения расстояния. Выдает значение расстояния. Рабочий диапазон от 10 до 80 см

Все датчики в ТРИК Студии подключаются на панели **«Настройка сенсоров»**

Для ветвления в ТРИК Студии используется блок «Условие», у которого имеется только одно свойство — само условие

Использование значений датчика осуществляется в ТРИК Студии также через переменные

При подключении датчика к порту A1 используется переменная **sensorA1**,
к A2 — **sensorA2** и т.д.

CC BY-NC-SA (C) ООО «КиберТех» 2019

Операторы

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

Логические операторы

оператор	синтаксис	пример
логическое отрицание, НЕ	!	!flag
логическое умножение, И	&&	(sensorA1>20) && (sensorA1<60)
логическое сложение, ИЛИ	||	(sensorA1<30) || (sensorA1>70)

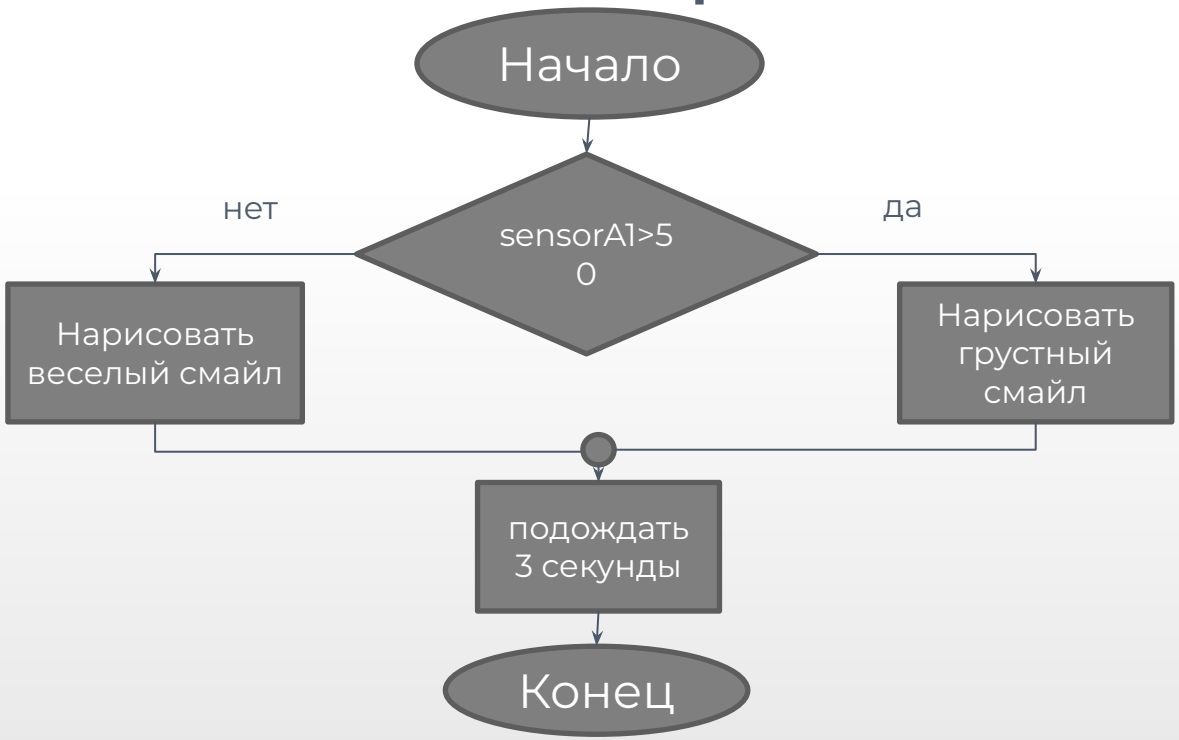
CC BY-NC-SA (C) ООО «КиберТех» 2019

Ветвление – задача

Задача: вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко. За границу считать значение 50 ИК датчика

Инфракрасный датчик расстояния — аналоговый датчик для измерения расстояния. Рабочий диапазон от 10 до 80 см

Блок схема алгоритма



CC BY-NC-SA (C) ООО «КиберТех» 2019

Ветвление – задача

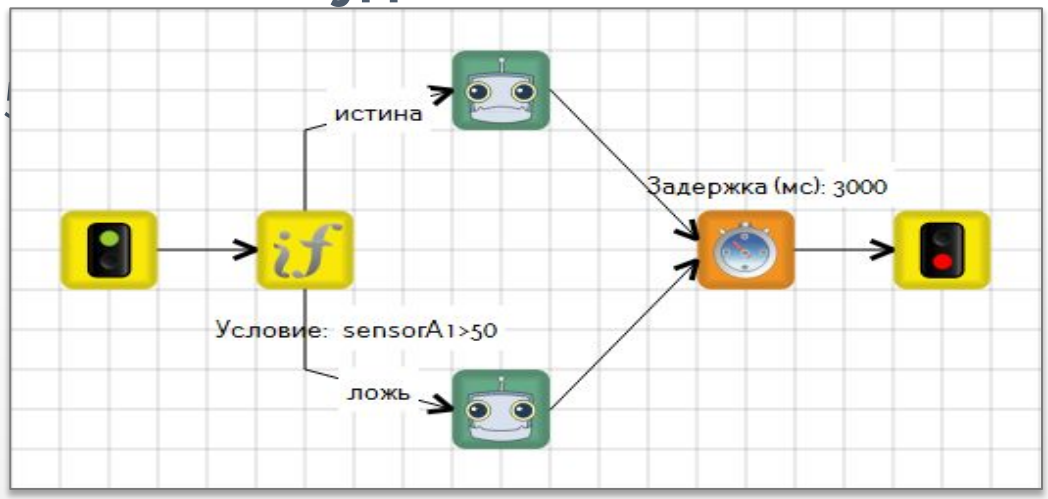
Задача: вывести на экран грустный смайлик, если робот далеко от стены, и веселый, если близко; за границу считать значение 50 ИК датчика

Инфракрасный датчик расстояния — аналоговый датчик для измерения расстояния. Рабочий диапазон от 10 до 80 см

Псевдокод

```
if (robot.sensor.[A1].read() > 5
    robot.sadSmile();
else
    robot.smile();
robot.wait(3000);
```

Решение в ТРИК Студии

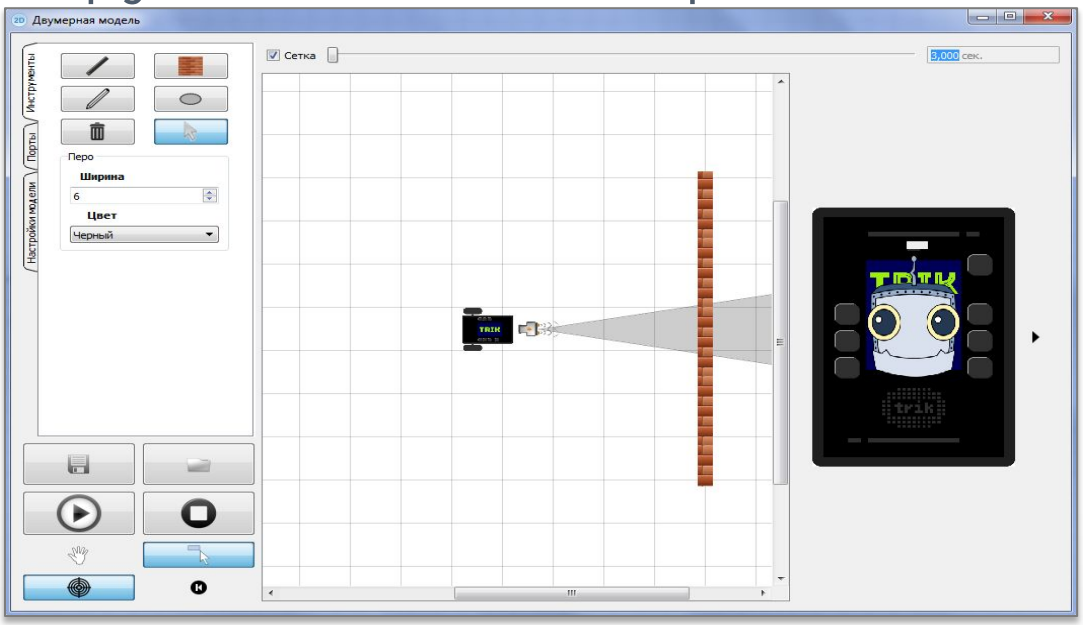


На связях, идущих от условия, указывается в свойствах **истина** и **ложь** для определения дальнейших действий, когда условие верно, и когда — нет

CC BY-NC-SA (C) ООО «КиберТех» 2019

Ветвление – задача

2D модель: для проверки задачи, нарисуйте при помощи инструмента «стена» препятствие перед роботом



Задача для самостоятельного решения: вывести на экран:

- веселый смайлик, если ИК датчик выдает до 40
- вывести слово «неопределенность», если ИК датчик выдает от 40 до 60 вывести грустный смайлик в противном случае

Какой в этом случае вариант ветвления лучше использовать?

CC BY-NC-SA (C) ООО «КиберТех»
2019

Цикл

CC BY-NC-SA (C) ООО «КиберТех»
2019

Цикл

Цикл. Предполагает возможность многократного повторения определенных действий. Количество повторений зависит от условия цикла

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

Существует 4 основных типа цикла:

- бесконечный (безусловный)
- с итерациями

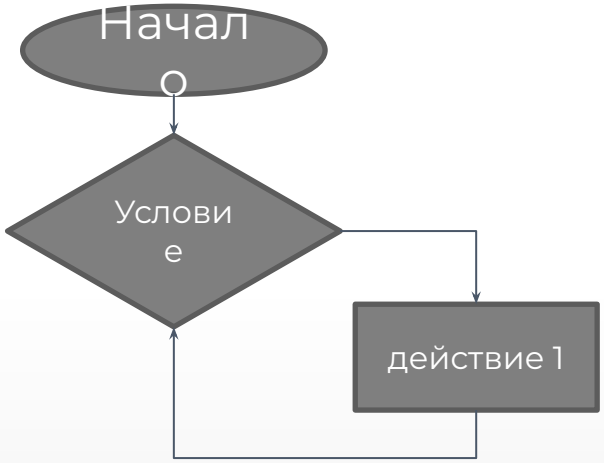
- с предусловием
- с постусловием

CC BY-NC-SA (C) ООО «КиберТех»
2019

Цикл

Цикл бесконечный (безусловный)

Блок-схема

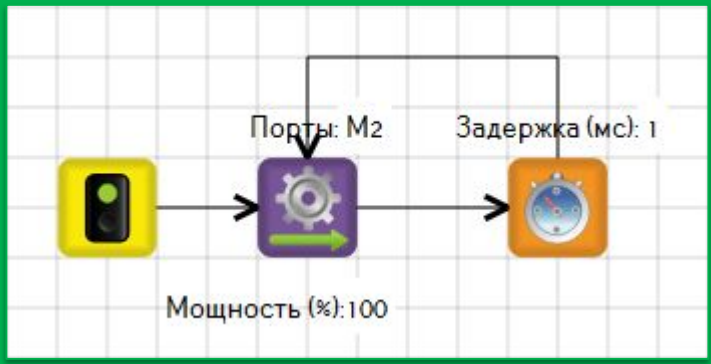


Псевдокод

```
while true do

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

В этом случае конца у программы может не быть

Пример в TRIK Studio



CC BY-NC-SA (C) ООО «КиберТех» 2019

Цикл

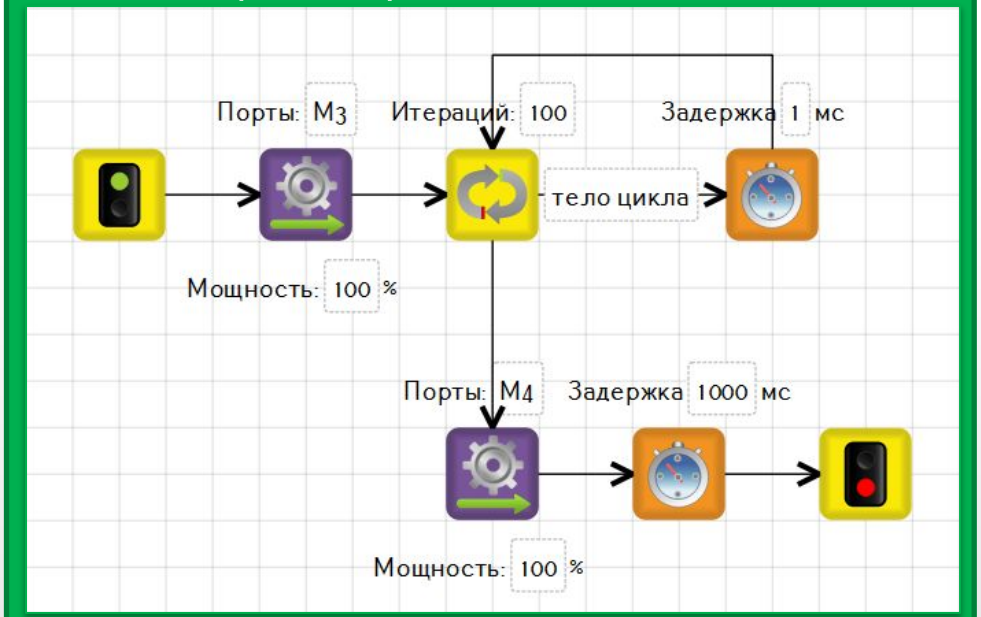
Цикл с итерациями

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

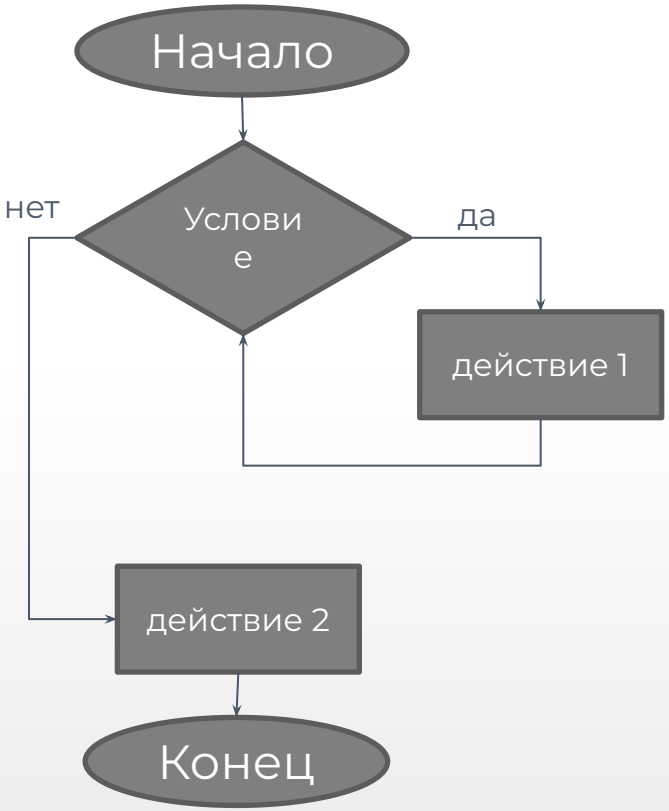


CC BY-NC-SA (C) ООО «КиберТех» 2019

Цикл

Цикл с предусловием

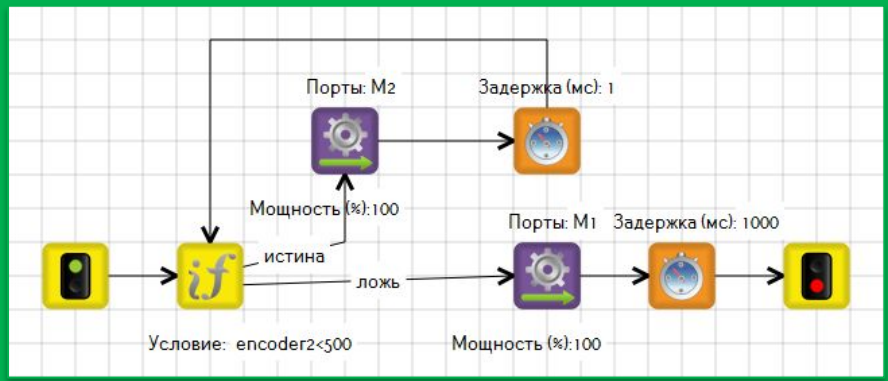
Блок-схема



Псевдокод

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

Пример в TRIK Studio

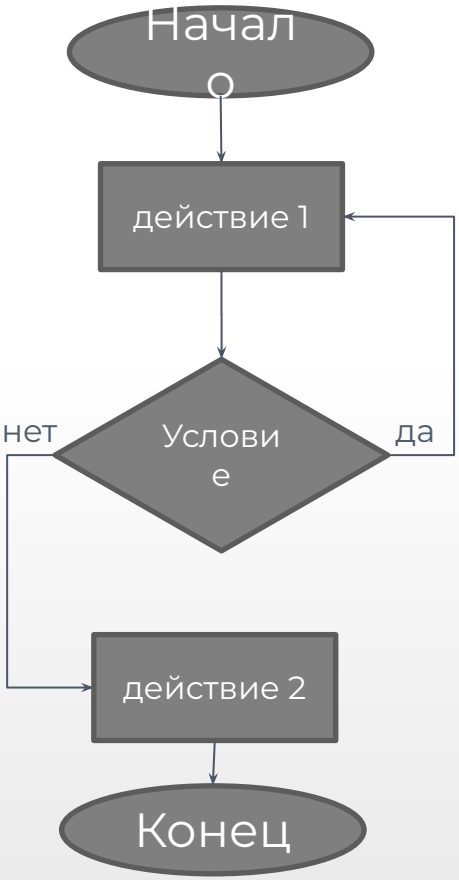


CC BY-NC-SA (C) ООО «КиберТех»
2019

Цикл

Цикл с постусловием

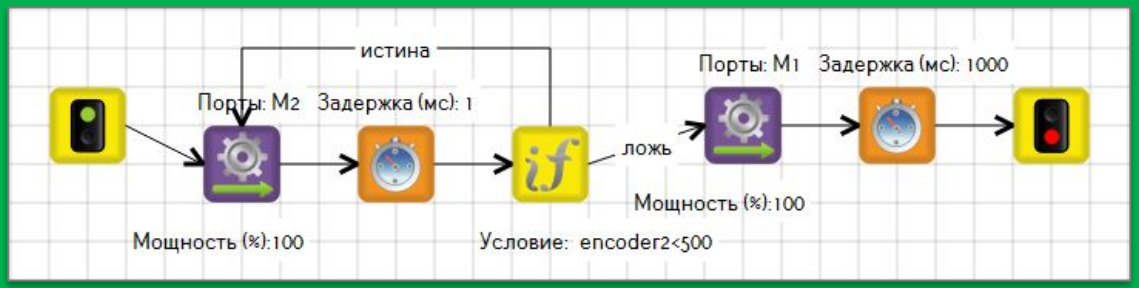
Блок-схема



Псевдокод

```
do

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

Пример в TRIK Studio



CC BY-NC-SA (C) ООО «КиберТех» 2019

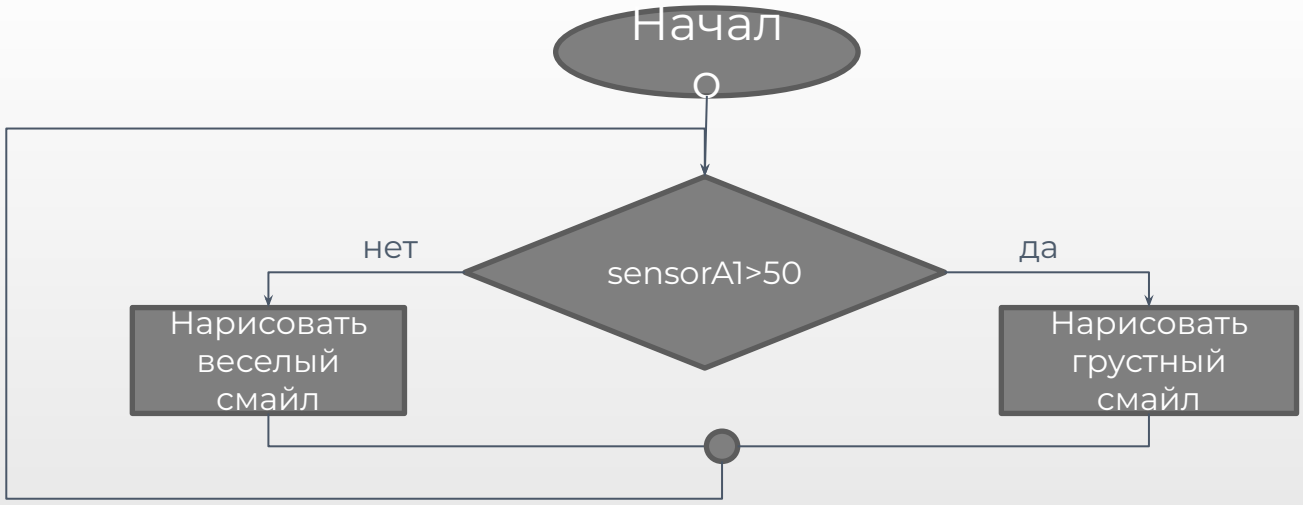
Цикл – задача

Задача «Настроение робота»: выводить на экран веселый смайлик, если робот на черном поле, и грустный, если на белом; за границу считать значение 50 датчика света

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

Для циклов с условиями в ТРИК Студии используется блок «Условие», а с итерациями – блок «Цикл». Бесконечные циклы реализуются путем соединения одного из блоков с каким-либо предыдущем

Блок схема алгоритма



CC BY-NC-SA (C) ООО «КиберТех» 2019

Цикл – задача

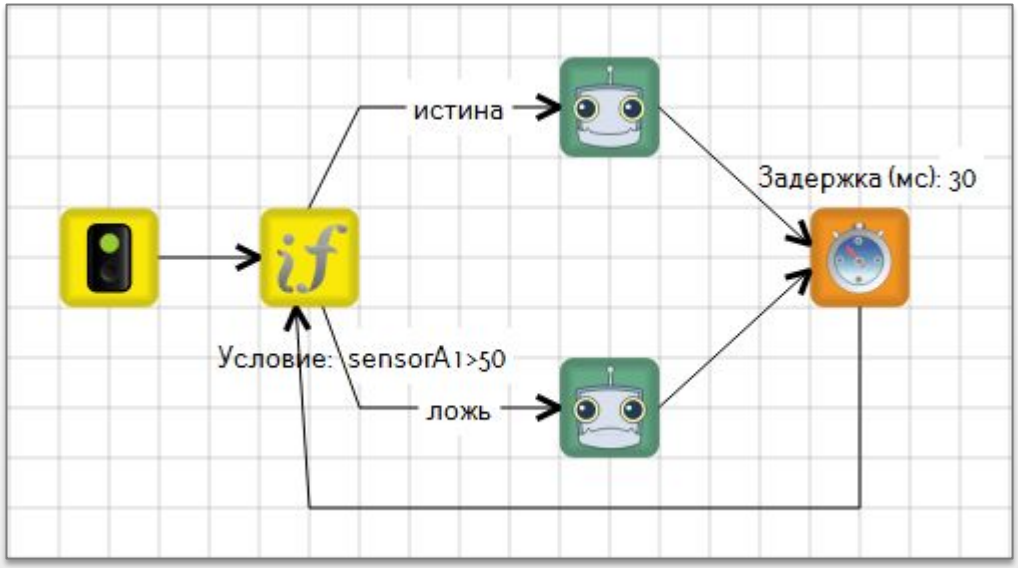
Задача «Настроение робота»: выводить на экран веселый смайлик, если робот на черном поле, и грустный, если на белом; за границу считать значение 50 датчика света

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

Псевдокод

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

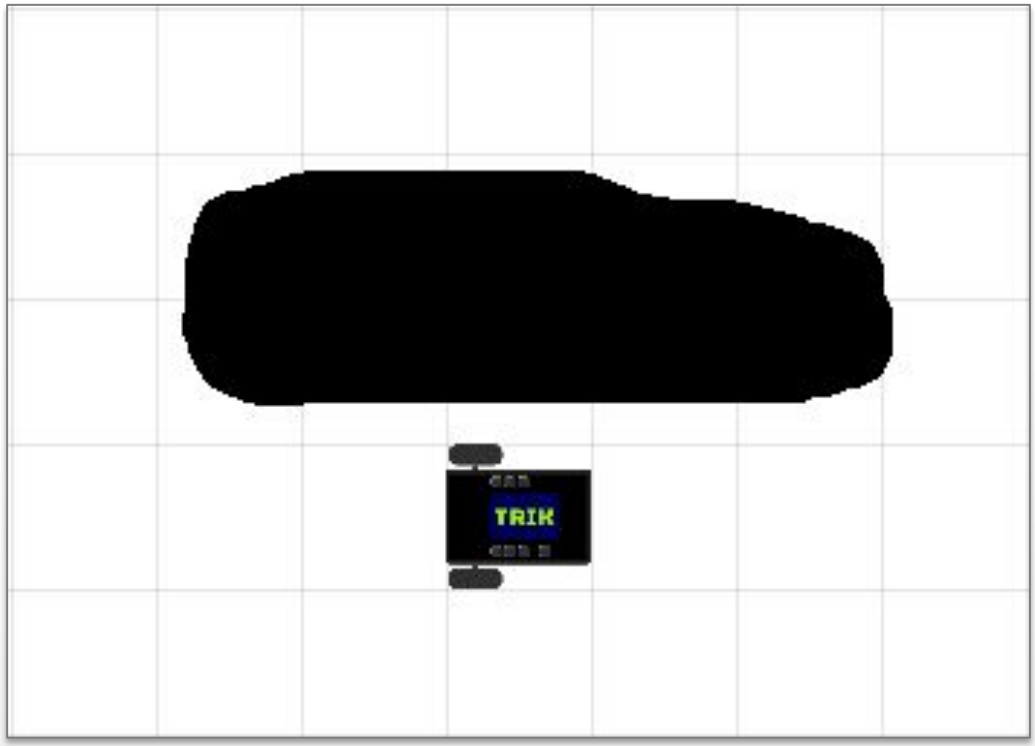
Решение в TRIK Studio



CC BY-NC-SA (C) ООО «КиберТех»
2019

Цикл – задача

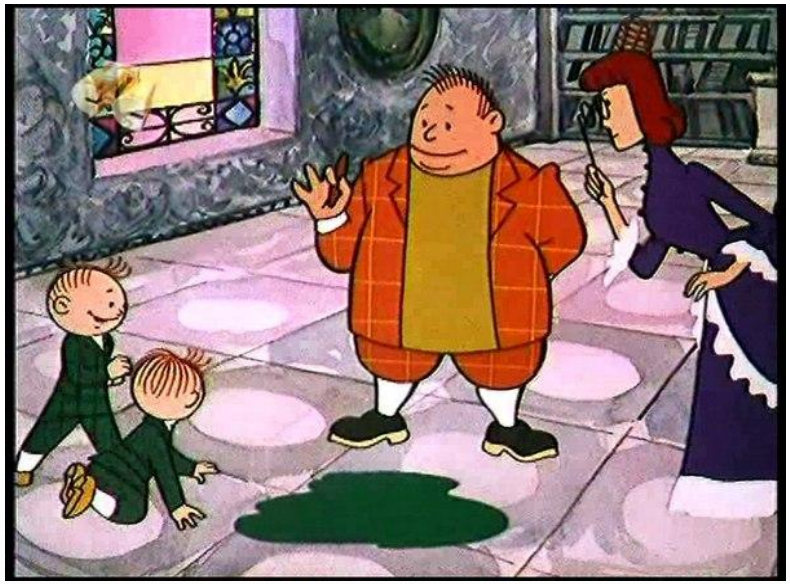
2D модель: нарисуйте масляное пятно при помощи инструмента «карандаш». Толщину карандаша задайте «30»



CC BY-NC-SA (C) ООО «КиберТех»
2019

Цикл

Задача для самостоятельного решения: 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

CC BY-NC-SA (C) ООО «КиберТех»
2019

Цикл

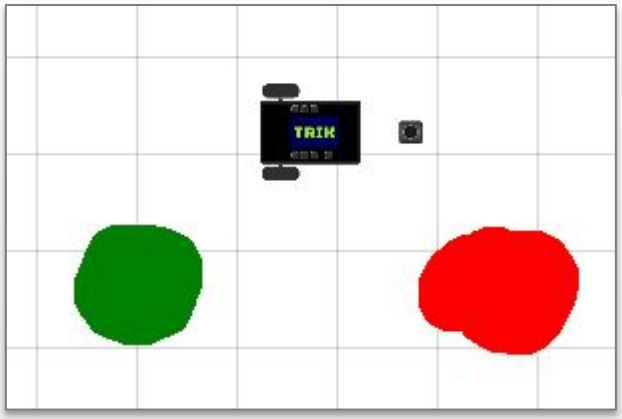
Задача для самостоятельного решения: Кентервильское привидение

Пояснение:

выводить на экран:

- веселый смайлик, если робот видит красную лужу (меньше 23) или пустой пол (больше 37)
- в противном случае (зеленая лужа) вывести на экран грустный смайлик (3 секунды) и закончить выполнение программы

Считывать новое значение с датчика **каждую секунду**

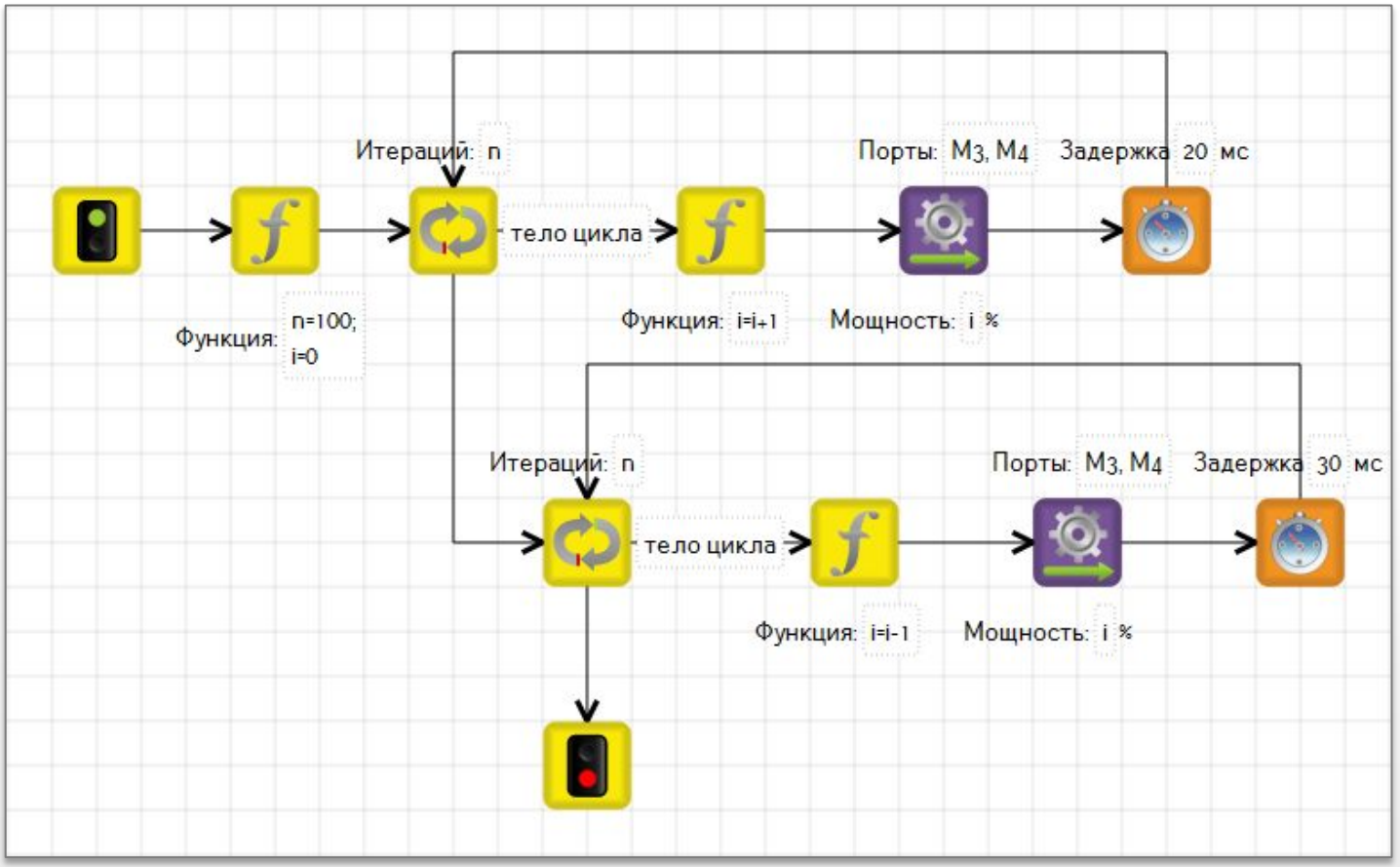


CC BY-NC-SA (C) ООО «КиберТех» 2019

Цикл – задача

Задача: Разгон и торможение

Напишите программу: плавный разгон робота в течение 2 секунд, а затем плавное торможение в течение 3 секунд



CC BY-NC-SA (C) ООО «КиберТех» 2019

Switch

CC BY-NC-SA (C) ООО «КиберТех»
2019

Switch

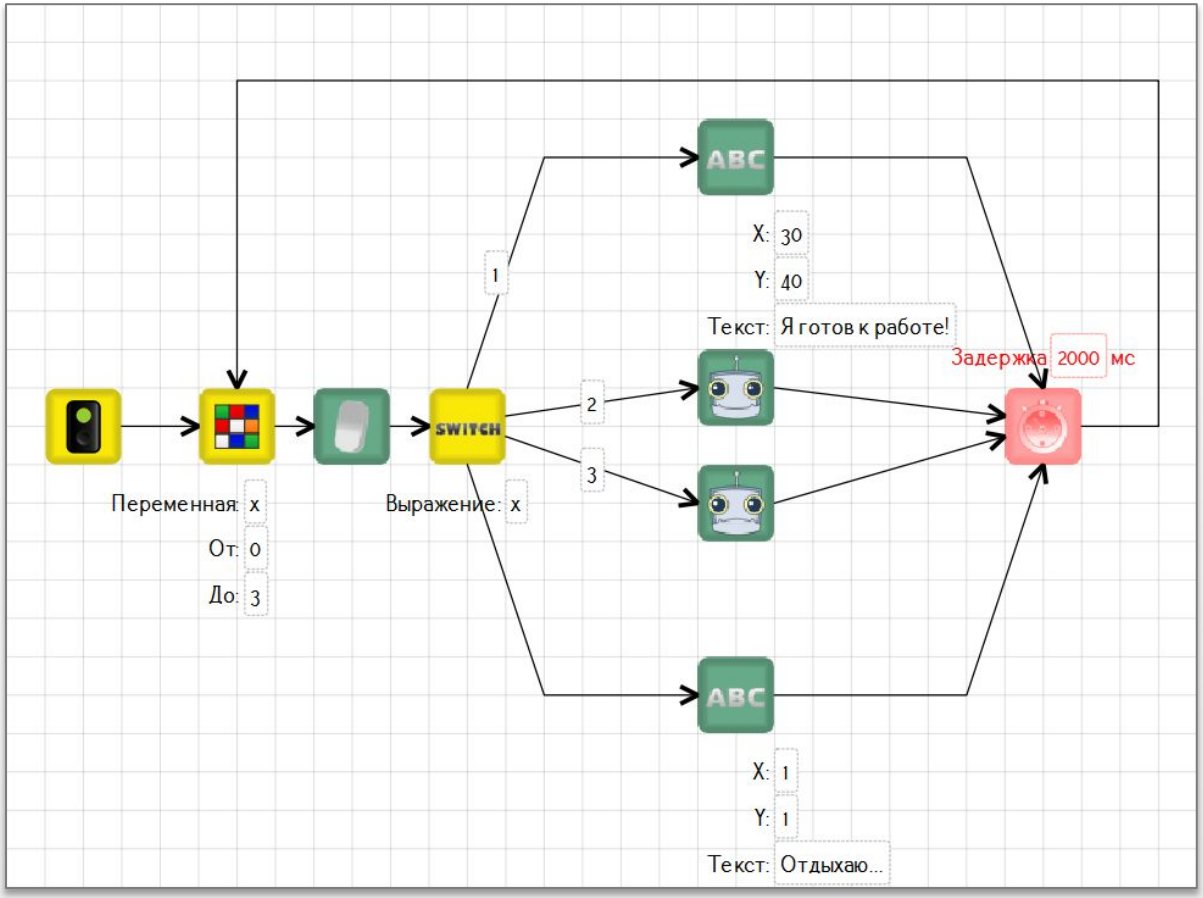
Представляет собой структуру, построенную по принципу меню, и содержит все возможные варианты условий и инструкции, которые следует выполнить в каждом конкретном случае

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”)

Switch

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



Switch – задача

Задача: выводить на экран робота в 2D модели по нажатию код кнопок контроллера ТРИК

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 – задача

Задача для самостоятельного решения:
Выполнять 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