

# **TRIK-Studio в примерах и задачах**

Методическое пособие  
по основам программирования в среде TRIK-Studio

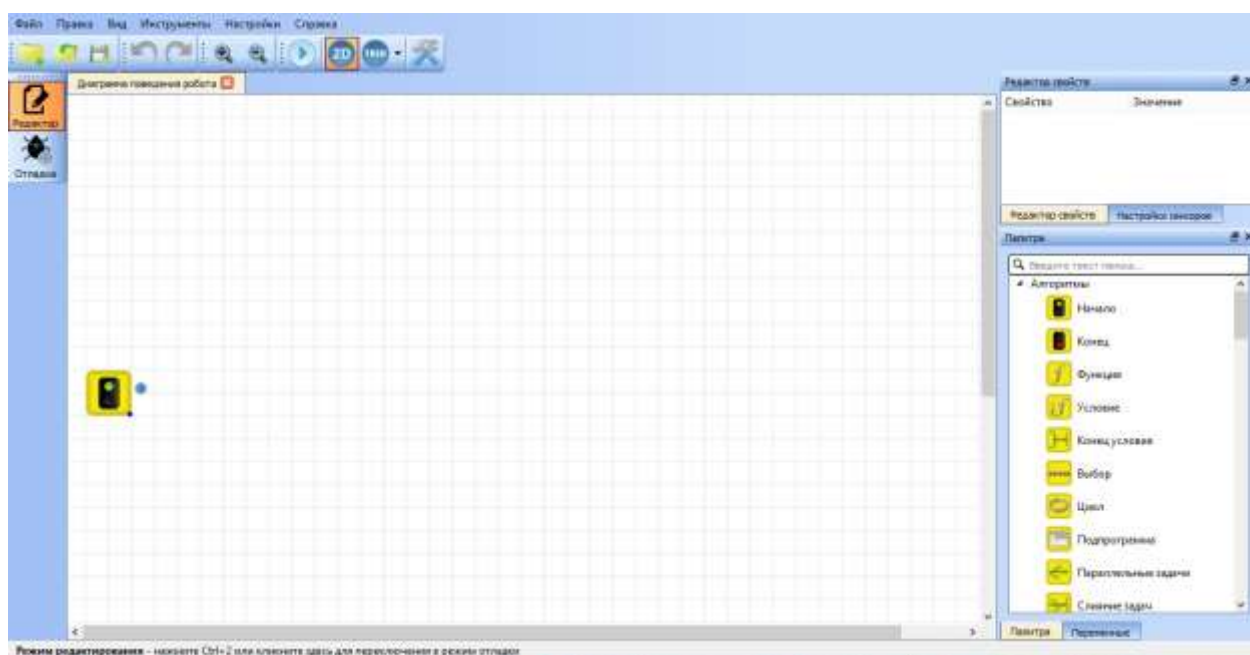
## Введение

Данное пособие предназначено для учащихся, желающих освоить основы программирования в среде TRIK-Studio. Особенность предлагаемой методики в том, что необходимые знания приобретаются в процессе решения большого числа достаточно простых и интересных задач, которые развивают творческий потенциал и алгоритмическое мышление учащихся. Учащиеся осваивают средства реализации линейных, ветвящихся и циклических алгоритмов, приемы структурного алгоритма: разбиение задачи на подзадачи и их реализация в виде процедур/функций.

Среда TRIK-Studio относится к визуальным языкам программирования, программы создаются при помощи пиктограмм. Интерфейс среды доступен для понимания даже учащимся начальных классов. Переход от визуального программирования к текстовому на языках RuC, C, C+, Java выполнен очень наглядно, поэтому ограничений по возрасту на использование TRIK-Studio не существует. TRIK-Studio поддерживает программирование контроллеров TRIK, LegoNXT, LegoEV3. Примеры разбираются для контроллера TRIK, для LegoNXT даются в сокращенном виде. Решения задач для закрепления материала прилагаются в электронном виде. Итак, оболочка TRIK-Studio установлена. Можно начинать:

## Занятие 1. Моторы вперед!

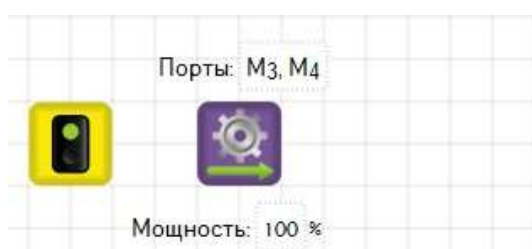
Запуск программы осуществляется с ярлыка, который лучше всего вынести на рабочий стол. Программы, с которыми работает TRIKStudio, называются проекты. Перед нами возникает окно с выбором проекта, над которым мы будем работать. Для начала нового проекта выбираем пункт меню «создать проект». Центральная часть окна называется сцена, на ней и будут строиться проекты.



Справа находится окно «палитра». На палитре располагаются блоки, с помощью которых реализуется управление роботом. Любой проект начинается с блока «Начало». Удерживая левую кнопку мыши, захватываем блок «Начало» и перетаскиваем его в окно «Диаграмма поведения робота» на сцене

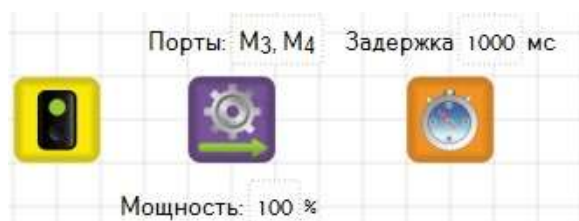


Начало положено. Попробуем заставить робота поехать вперед. Двигается наш робот при помощи моторов. Ищем в палитре блок «моторы вперед» и располагаем его справа от блока «Начало».



Отменить неправильное действие всегда можно при помощи стрелки «отменить» или в меню «Правка» пункт «Отменить».

Чтобы наши моторы заработали, необходимо задать условие их работы, например, время движения. Перетаскиваем из палитры блок «Таймер» и дополняем им наш проект.



Чтобы закончить программу, используем блок «Конец».

Укажем порядок выполнения блоков в проекте, связывая их линиями соединения. Чтобы соединить два блока, нужно, зажав правую кнопку мыши, провести линию от одного блока к другому. Должна появиться стрелка, указывающая направление выполнения команд. Повторим это действие для всех пар блоков. Программа «моторы вперед» приобрела законченный вид.



Посмотрим результаты ее работы. Для этого воспользуемся двумерной моделью робота. В меню «настройки» выбираем «роботы»- «двумерная модель» и нажимаем «ок». Запускаем программу кнопкой «отладка», расположенной слева от сцены.



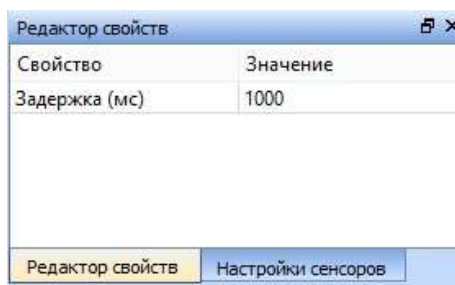
В окне «двумерная модель» робот двигается 1 секунду. В этом же окне расположены кнопки «пуск» и «стоп» для повторного запуска программы или остановки ее выполнения.



Начальное положение робота обозначено красным крестиком, что бы вернуться в начальное положение нужно нажать на кнопку «назад»



Рассмотрим нашу программу более подробно. Рассмотрим, что делает блок «таймер». Для этого выделим его, нажав по нему левой клавишей мыши. Слева в окне «редактор свойств» появится запись - задержка (мс) 1000. Это время, на которое установлен таймер. Поменять значение, можно щелкнув на нем и прописав нужное значение в миллисекундах.



У блока «моторы вперед» в «редакторе свойств» две записи.

Первая запись – порт. К контроллеру TRIK можно присоединить четыре мотора, которые обозначаются: M1, M2, M3 и M4. Разъемы на контроллере так и обозначены. Это и есть наши порты. В нашей программе в окне написано M3, M4 – то есть моторы подсоединены к разъемам M3 и M4. Порты можно поменять, щелкнув на записи M3, M4 мышкой и вручную прописать нужные нам. И последняя запись – это мощность, с которой работают моторы.

Попробуем изменить программу. Уберем один из портов, например, M4.



После запуска программы мы увидим, что робот совершает поворот вокруг одного из колес.

Сохраним результат нашей работы. Для этого выберем в меню «файл» вкладку «сохранить файл» или «сохранить файл, как». Определим папку, куда хотим сохранить наш проект, в поле имя файла вводим название для нашей программы. Нажимаем «сохранить». Программу TRIKStudio можно закрыть.

В отличие от контроллера TRIK к контроллеру NXT можно подключить 3 мотора, обозначаются они A, B и C.

Программа «моторы вперед» для NXT:



Программа для поворота робота:



Процессы сохранения проекта для TRIK и NXT идентичны.

**Задача:** робот движется вперед 2 секунды, затем поворачивает влево и едет 3 секунды.

Откроем сохраненный проект. В меню «файл» выберем «открыть». Находим папку, в которую была сохранена программа, выбираем нужный файл и нажимаем «открыть».

На сцену добавляем блоки «моторы вперед» и «таймер». У второго блока «моторы вперед» убираем порт M4, таймеры выставляем соответственно 2 и 3 секунды.



Попробуем совершить плавный поворот. Потребуется еще один блок «моторы вперед». В нем мы уберем порт M3 и изменим мощность на 50%. Получилось следующее:



Эта же программа для NXT:



### Задачи для закрепления материала

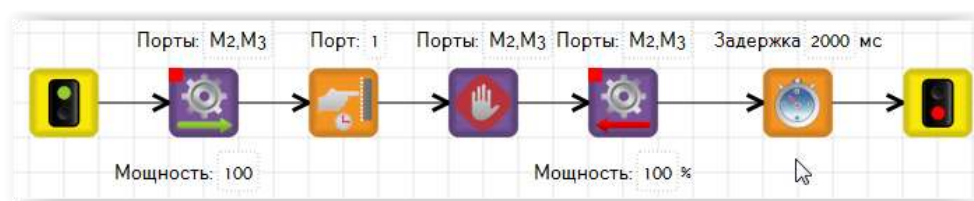
1. Движение вперед-назад
2. Движение по кругу
3. Движение по восьмерке
4. Движение по заданной линии. Линию, по которой будет двигаться робот, в окне «двумерная модель» можно нарисовать при помощи меню «инструменты».

## Занятие 2. Впереди стена

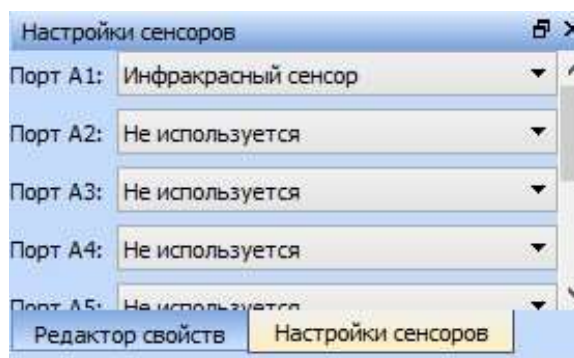
Мы уже знаем, что на контроллерах находятся выходы на моторы. На контроллере TRIK так же присутствуют порты A1-A6, D1, D2. Это порты входа для подсоединения датчиков. У контроллера NXT это порты 1, 2, 3 и 4

Попробуем решить следующую задачу: на пути движения робота находится стена, коснувшись ее, надо отъехать на некоторое расстояние назад.

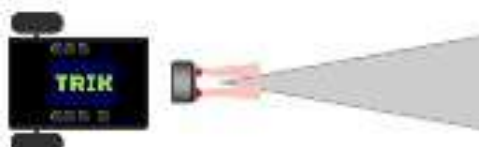
Для решения задачи на контроллере TRIK используем датчик касания. В палитре для него есть блок «ждать датчик касания». Блок «ждать датчик касания» является условием для осуществления движения, поэтому таймер нам не нужен, моторы будут работать, пока не сработает датчик касания. В свойствах блока необходимо указать порт, к которому он подсоединен. Итак, мы едем вперед, пока датчик не сработал, затем останавливаем моторы и едем назад две секунды.



Мы написали программу, но контроллеру еще надо сообщить, что к порту A1 присоединен сенсор касания. Для этого в настройках сенсоров выставляем для «порта A1» сенсор касания.



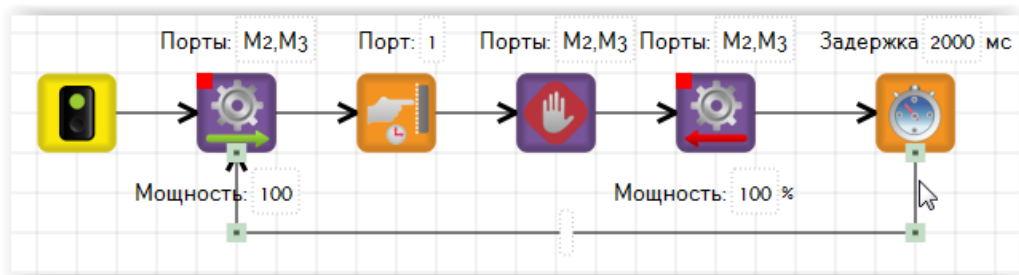
В окне «отладка» на роботе появится изображение датчика, который с помощью мыши надо перетащить к передней (как в нашем примере) или задней части робота.



Строим стену при помощи панели «инструменты» и запускаем программу:



Если несколько раз запустить нашу программу, покажется, что робот пытается проломить стену. Поможем ему в этих попытках. Уберем блок «конец» и соединим блоки «таймер» и «моторы вперед». Программа будет выполняться, пока мы не нажмем «стоп».



Для решения этой задачи на контроллере NXT используем датчик касания. Не забудем выставить в настройках сенсоров, что к «порту 1» присоединен датчик касания.



Точно также, как и на контроллере TRIК датчик нужно установить спереди



Что делать, если мы не хотим касаться стены, например, она окрашена? Изучим еще один датчик – датчик расстояния (сонар). Задача прежняя: доехав до стены, отъехать назад, но на этот раз без касания. В свойствах блока «ждать сонар» видим следующие свойства: «дистанция», «порт», «считанное значение».

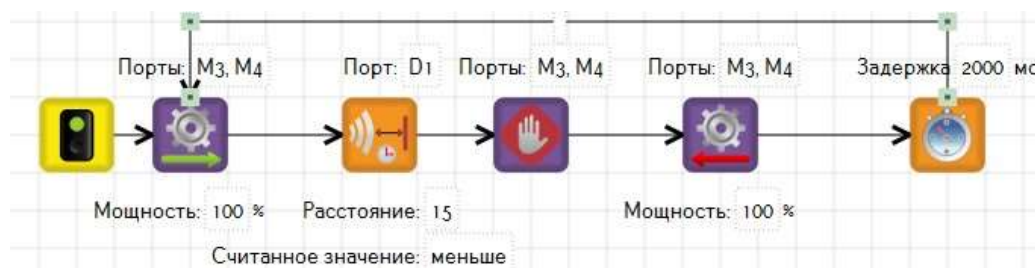


«Дистанция»: выставляем значение в сантиметрах до объекта от 0 до 255, в нашем случае не доедем до стены 15 сантиметров.

«Порт»: номер порта, к которому подключен датчик.

«Считанное значение» зависит от условия задачи, у нас оно должно стать меньше пятнадцати сантиметров.

Алгоритм решения задачи для TRIK:



Для NXT:



Не забудем выставить в настройках контроллера, что к «порту D1» (для NXT «1») присоединен сенсор расстояния.

**Задачи:**

1. Движение между стен с двумя датчиками.
2. Не упасть со стола (использовать бесконечный цикл)
3. Обнаружить предмет
4. Встреча

### **Занятие 3. Разноцветная дорога (только для контроллера NXT)**

В данном занятии мы рассматриваем только контроллер NXT, так как контроллер TRIK определяет цвета при помощи камеры (это будет рассмотрено позже).

Теперь наш робот сумеет не только передвигаться, но и видеть предметы. Еще одна способность робота – это определение цвета и света. Появляется она благодаря соответствующим датчикам. В одной из задач мы успешно избежали касания окрашенной стены. Но что делать, если окрашен пол. Попробуем использовать датчик света. Любая поверхность отражает свет, но количество отраженного света зависит от свойств поверхности. Допустим, что непокрашенный пол белый, а покрашенный черный.

Датчик света возвращает значения от 0 до 100, где 100 соответствует чисто белому свету и 0 – черному. Для датчика света в палитре имеется блок «ждать свет». В свойствах блока видим три записи: «проценты» – количество света (от 0 до 100), «порт к которому подключаемся» и «считанное значение», в нашем случае меньше 50 процентов. В настройках сенсоров контроллера выставляем сенсор света, подключенный к порту 1.



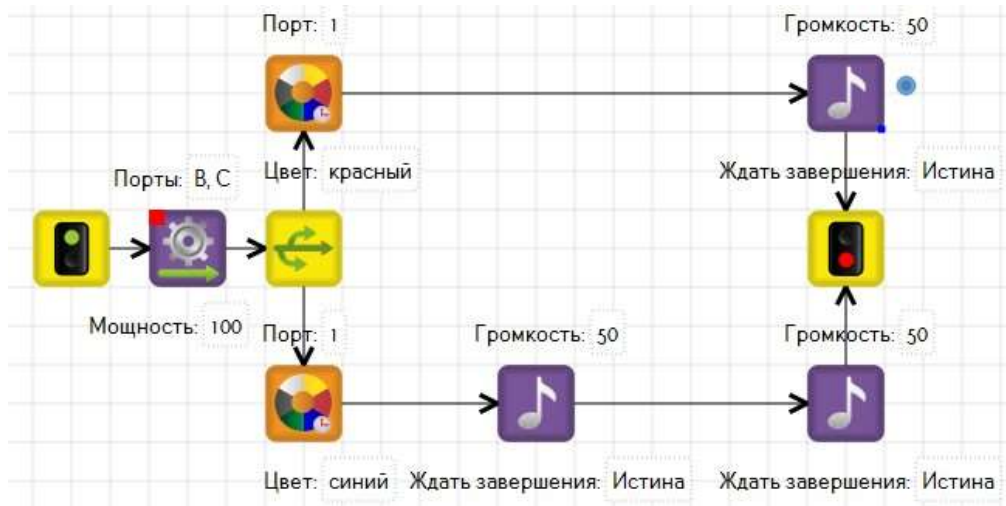
Теперь пятно на полу нам не страшно.

#### Задачи:

1. Начать движение, если свет включен.
2. Подать звуковой сигнал, если наехали на пятно.
3. Объехать пятно на полу.

Усложним задачу. Покрашенный пол синий, непокрашенный – красный. Роботу придется принимать решение, что делать, если он увидит синий цвет, и что делать, если красный.

Контроллер позволяет обрабатывать одновременно несколько задач благодаря блоку «параллельные задачи» (контроллер TRIK также может использовать этот блок). В нашем случае это две задачи – если красный цвет и, если синий. За определение цветов у нас отвечает датчик цвета, которому в палитре соответствует блок «ждать цвет».



В свойстве блока «ждать цвет» указываем цвет, а в свойстве «порт» – порт подключения датчика.

### Задачи:

1. Встретив синий предмет, запускаем мотор А на 2 сек.
2. Едем до светофора. Если красный, останавливаемся, ждем зеленый. На зеленый проезжаем.
3. Определяем цвет шара. На разные цвета различное количество сигналов.

### Занятие 4. Бег по кругу

Очень часто роботам приходится выполнять одну и ту же задачу неоднократно. Мы уже видели, как можно заставить программу выполняться бесконечное число раз. Но иногда достаточно, чтобы операция была исполнена 5, 10 или 100 раз. В программе TRIKstudio это возможно благодаря оператору «цикл» из палитры.

Вернемся к задаче, в которой мы «стучались» в стену. Постучимся три раза, развернемся и уедем. Из оператора «цикл» исходят две линии соединения: одна направлена на начало цикла, вторая отправляет нас дальше – на действие после цикла. Выделим блок цикл. В свойствах указаны итерации, то есть нужно определить сколько раз мы будем выполнять наше действие. Щелкаем на линию соединения, смотрящую на начало цикла, и в ее свойствах указываем условие – тело цикла. Последний блок исполняемого действия цикла соединяем с блоком «цикл».



Для NXT:

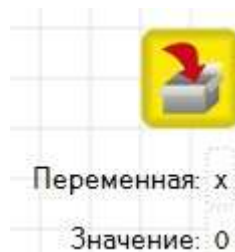


### Задачи:

1. Движение по квадрату
2. Обход пятна
3. Вытираем пятно

### Занятие 5.Познакомимся с неизвестными

Для объявления переменных и задания им определенных значений в TRIKstudio используется отдельный блок «инициализация переменной».



Перетащим блок «инициализация переменной» в окно «диаграмма поведения робота».

Выделим его. В редакторе свойств, в строке «Переменная» запишем R, а в значении 0. Это означает, что мы взяли какую-то переменную R и присвоили ей значение 0.

Для объявления сразу нескольких переменных или больших вычислений проще использовать блок функция:



В правой части строки «значение» мы можем использовать круглые скобки, числа, базовые арифметические операции, ранее определенные переменные, а также

зарезервированные переменные список, которых можно посмотреть в режиме «отладка» в правой части экрана

Уточним, что же такое зарезервированные переменные. Как мы уже знаем, к контроллеру TRIK можно присоединять датчики. В каждый момент времени датчики возвращают какое-то числовое значение на контроллер. Это и есть значение переменной sensorN.

Например, у нас к порту A1 подсоединен датчик расстояния. Робот находится вплотную около стенки. Значение переменной sensorA1=0. После перемещения робота на некоторое расстояние значение изменится. То есть значение переменной SensorN это то, что показывает датчик в данный момент времени

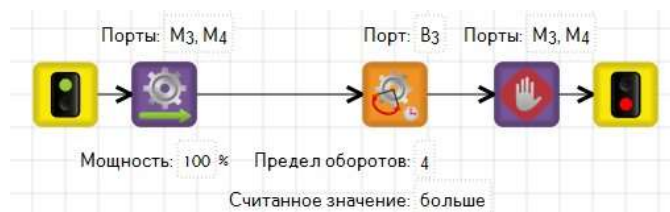


	Имя	Значение
4	buttonDown	0
5	buttonEnter	0
6	buttonEsc	0
7	buttonLeft	0
8	buttonRight	0
9	buttonUp	0
10	colorSensorB	0
11	colorSensorG	0
12	colorSensorR	0
13	encoder1	0
14	encoder2	0
15	encoder3	0
16	encoder4	0
17	gamepadButton1	0
18	gamepadButton2	0
19	gamepadButton3	0
20	gamepadButton4	0
21	gamepadButton5	0

Рассмотрим оставшиеся переменные Encoder. В конструкторах используются моторы со встроенным датчиком оборотов. То есть, мы всегда можем узнать, сколько оборотов совершил мотор, либо сами задать нужное количество оборотов.

Задача: проехать роботом 4 оборота колеса.

Для решения этой задачи на контроллере TRIK необходимо подключить моторы не только к портам M, но и к портам B – энкодерам. Номера портов M и B совпадают

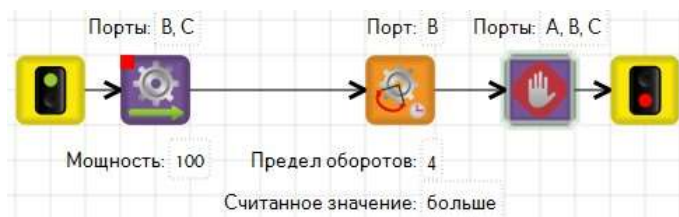


Датчику оборотов в палитре соответствует блок «ждать энкодер».

Для NXT также есть зарезервированные переменные, но их значительно меньше:

Переменные		
	Имя	Значение
1	buttonEnter	0
2	buttonEscape	0
3	buttonLeft	0
4	buttonRight	0
5	encoderA	0
6	encoderB	0
7	encoderC	0
8	pi	3.1415926535897931
9	sensor1	0
10	sensor2	0
11	sensor3	0
12	sensor4	0

Для решения задачи с энкодерами на NXT, в отличие от TRIK достаточно подключить моторы к соответствующим портам, у NXT нет выделенных портов для энкодеров.

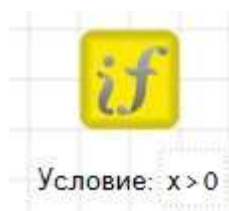


### Задачи:

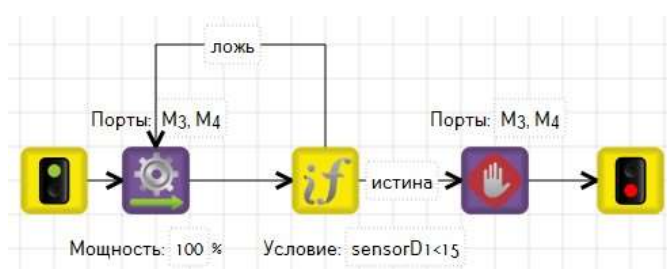
- 1) Подъехать к стенке с уменьшением скорости, используя датчик сонар.
- 2) Скорость увеличивается в зависимости от освещения. 3) Вернуться в исходную точку используя энкодер.

### Занятие 6. Что если?

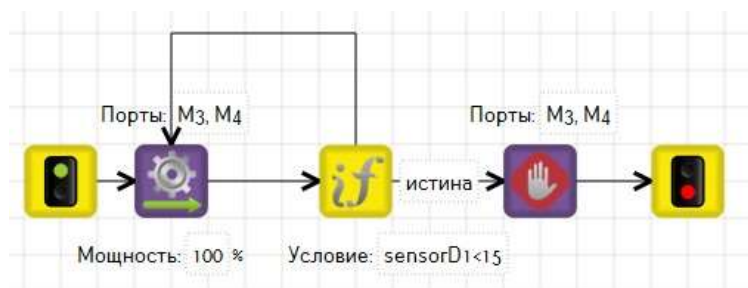
Что если нам необходимо задать какое-нибудь условие выполнения задачи? Здесь нам придет на помощь оператор перехода if, которому в палитре соответствует блок if. Вспомним задачу с окрашенным полом. Ее можно решить без распараллеливания процессов, используя оператор перехода.



**Задача:**Продолжать движение, пока расстояние до стены не станет меньше 15, не используя блок «ждать датчик расстояния»



Из оператора “if” обязательно должны выходить две связи «истина» и «ложь», причем не обязательно указывать обе. Программа все равно будет выполняться



**Задачи** 1.Едем, пока расстояние до объекта впереди больше какого-то значения.

2.Если проехали 2 метра, едем назад.

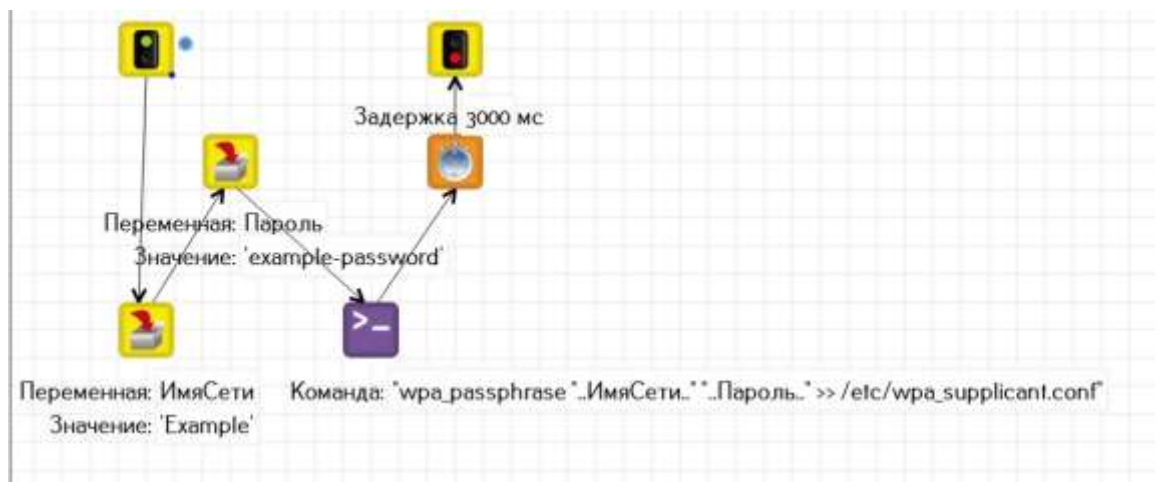
3.Если пол окрашен, вытираем.

## Занятие 7. Моторы вперед-2

Использовать 2D модель очень удобно, но гораздо интереснее увидеть, как же будет двигаться наш робот по-настоящему. В среде TRIK-Studio в папке с примерами имеется программа для настройки сети:

configureNetwork.qrs





Программа для конфигурирования WiFi-сети на роботе, позволяет задать подключение робота к WiFi-сети в режиме клиента.

Чтобы воспользоваться:

- включите робота в режиме точки доступа (меню "Сеть" -> "Wi-Fi точка доступа")
- переведите TRIK Studio в режим работы с реальным роботом (кнопкой на панели инструментов)
- подключитесь к роботу по WiFi (проверить корректность можно кнопкой "Подключиться" на панели инструментов)
- поменяйте значение переменной "ИмяСети", введя имя сети (SSID), в кавычках
- поменяйте значение переменной "Пароль", введя пароль сети
- запустите программу кнопкой "Загрузить и выполнить программу"
- после того, как программа отработает (появится меню на роботе), выберите режим "WiFi-клиент" в меню "Сеть"
- выберите вновь добавленную сеть и нажмите кнопку "Ввод" на роботе
- робот через некоторое время должен подключиться к сети и показать выданный ему IP-адрес в верхней части окна настройки сети- робот через некоторое время должен подключиться к сети и показать выданный ему IP-адрес в верхней части окна настройки сети.

Для NXT соберем робот-пятиминутку(инструкция в приложении). Эта модель наиболее удобна для проверки работоспособности программ.

Подключение роботаNXT к компьютеру возможно по USBинтерфейсу и по Bluetooth. Для подключения по USB необходим драйвер из поставки конструктора (скачать его можно по ссылке). В окне «Настройки» выбираем «роботы» – «легоNXT» – «USB»– «ок». Затем нажимаем на кнопку «подключиться». Если в нижнем окне появилась запись, что робот подключен, можно смело нажать «пуск». Для подключения по Bluetooth



никаких специальных драйверов не нужно. NXT является обычным Bluetoothустройством, пароль для соединения – 1234. В настройках, в разделе «дополнительно» необходимо указать com-порт подключения.

Теперь программы можно загружать непосредственно на робота с помощью кнопки



## Задачи

- 1.Едем по восьмерке.
- 2.Не упасть со стола.
- 3.Встреча.

## Занятие 8. Рисунки на экране

Дисплей контроллера TRIKпозволяет выводить на него различные рисунки, тексты и цифры.

Задача: вывести на дисплей контроллера веселый смайлик.



А теперь попробуем вывести на экран значения датчика расстояния



Как видим, рисование на дисплее иногда оказывается очень полезным

У NXT нет функции рисовать смайлики. Задача для вывода значения датчика выглядит так:



**Задачи:**

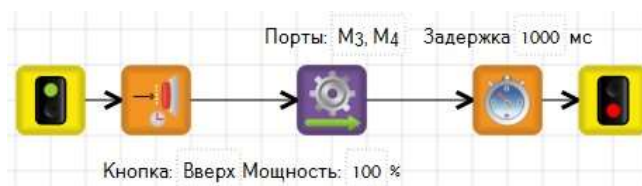
1. Вывести последовательность фигур
2. При обнаружении предмета вывести текст
3. Определить числовое значение цветов: красный, зеленый и синий

Рассмотрим еще один блок «ждать нажатия кнопки». В нашем случае, это означает нажатие кнопки на контроллере. В окне «двумерная модель» изображение контроллера спрятано за правым краем окна, на нем можно тоже нажимать кнопки.



**Задача:** движение робота начинается после нажатия кнопки.

**TRIK:**

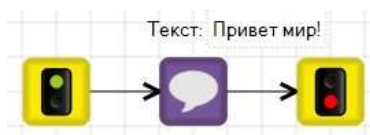


**NXT:**

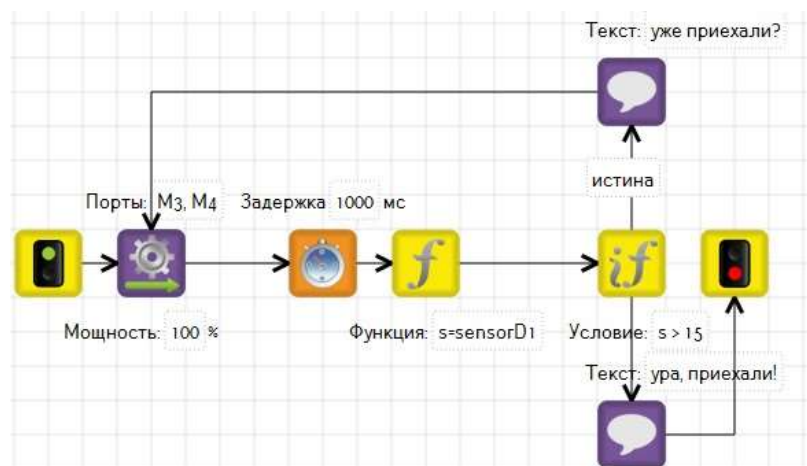


## Занятие 9. Привет Мир!(Только для контроллера TRIK)

В предыдущей главе мы познакомились с выводом данных на экран. Кроме этого контроллер TRIK способен выводить информацию при помощи звуков, например, он может генерировать устную речь. Напишем программу, состоящую из трех блоков «начало», «сказать» и «конец». В свойствах блока «сказать» напишем «Привет Мир!» - самый распространенный текст, используемый программистами при обучении для вывода информации.



Пусть наша тележка движется до стены, и пока во время движения задает вопрос «уже приехали?», как только расстояние до стены станет меньше 15-ти тележка должна сказать «ура, приехали!» и остановиться.



### Задачи:

1. Запустить сирену, после срабатывания датчика движения (звук нужно положить в корневую папку на контроллере, и указать в свойствах путь к звуку)
2. Когда робот датчиком касания упрется в стену сказать: «Обнаружена стена!»
3. После нажатия на кнопку на корпусе контроллера сказать: «Убери руки!»

### Отступление. Жесты мышью.

Разработчики программы TRIKStudio большое внимание уделяют usability - удобству пользования. Для того, чтобы каждый раз не искать нужный блок в палитре, достаточно изобразить в окне «диаграмма поведения робота» определенный жест мышью, зажав левую кнопку. Ознакомиться с жестами можно открыв меню «Инструменты» – «жесты мышью».

## Занятие 10.Исправление ошибок.

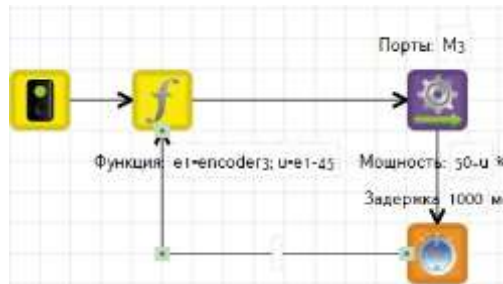
На практике мы неоднократно сталкивались с тем, что показания одинаковых приборов могут различаться. Это связано с рядом факторов, таких как условия окружающей среды, несовершенство изделий и т.п. В наборе TRIK датчики, используемые для конструирования, также не являются идеальными. Например, значения, возвращаемые двумя датчиками оборотов, могут не совпадать. Попробуем учесть эту разницу. Синхронизируем моторы.



В программе мы использовали метод релейного регулирования для исправления ошибки. В нашем тексте мы не будем вдаваться в подробности теории управления, но воспользуемся некоторыми ее результатами.

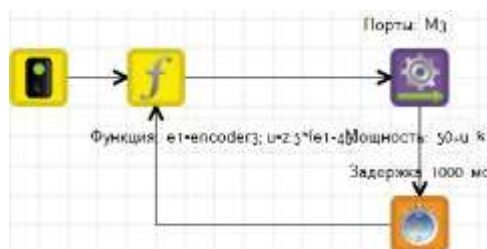
**Задача:** Стабилизировать мотор в положении  $45^0$ .

В этой задаче для исправления ошибки также воспользуемся релейным регулятором.

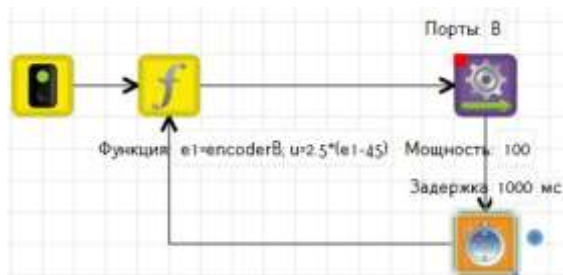


Мотор будет колебаться около положения  $45^0$ . Более точное исправление ошибки достигается использованием пропорционального регулятора. Подправим наш алгоритм, используя коэффициент усиления регулятора 2,5. Поправка  $u=2.5*(45-encoder1)$ .

Соответственно синхронизация моторов на П-регуляторе будет выглядеть так

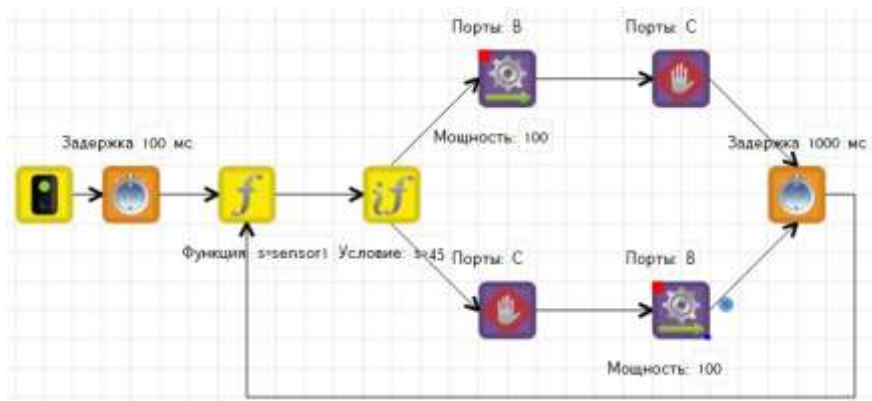
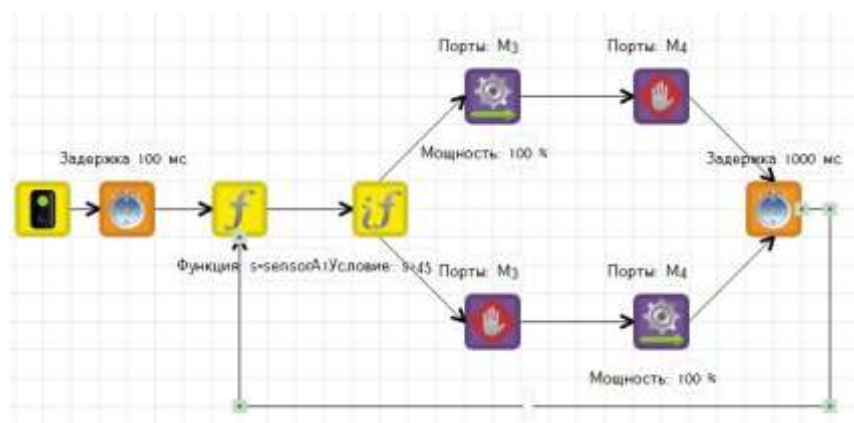


Для NXT программа выглядит следующим образом:



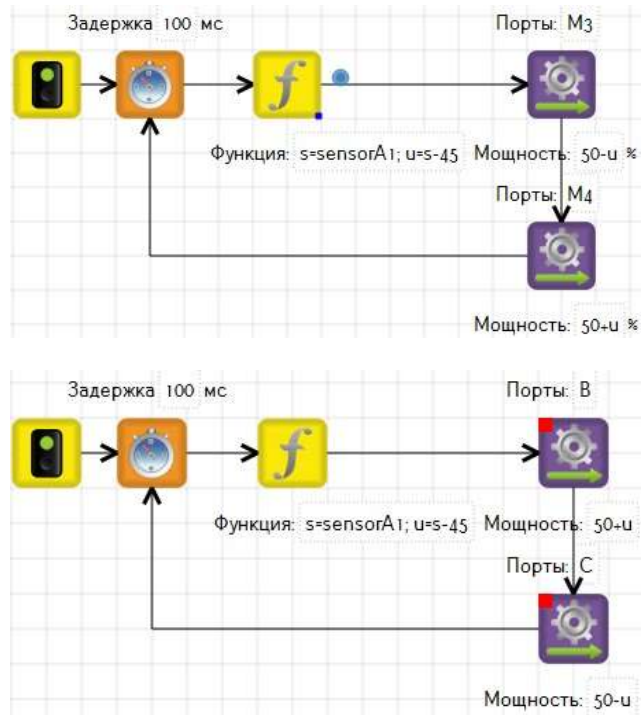
### Занятие 11. Движение по линии

Достаточно большое количество задач в робототехнике связано с движением вдоль линий. Рассмотрим пример движения с одним датчиком освещенности. Используем релейный регулятор.



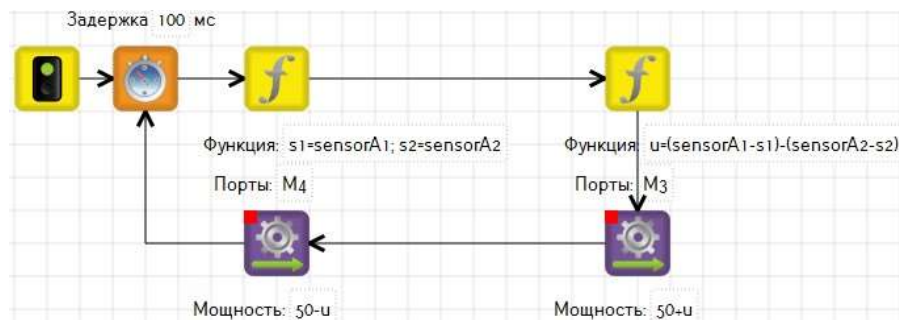
Робот движется по ломанной кривой, периодически наезжая на линию.

На П-регуляторе движения робота будут более осмысленными. Число 45 означает, что цвет серый.



Модернизируем конструкцию робота, добавив еще один датчик. На такую модель практически не влияют колебания освещенности в помещении.

Сравнивая начальные показания датчиков с текущими значениями на П-регуляторе, имеем



Задачи 1. Двигаться по линии, пока робот датчиком касания не упрется в стену

2. Двигаться по линии до перекрестка, на нем развернуться на  $180^\circ$  и двигаться по линии в обратном направлении

## Занятие 12. Движение вдоль стены. Объезд предметов

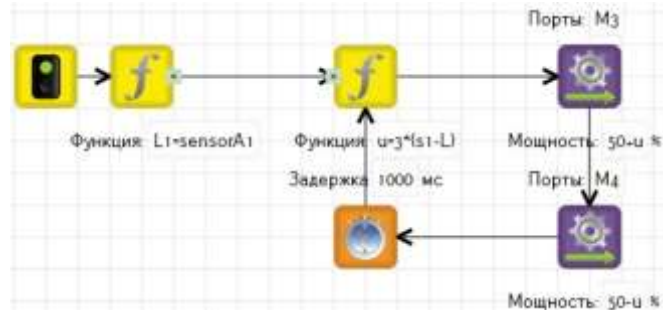
Умения видеть и объезжать препятствия, двигаться вдоль стены, являются основными для робота, находящегося в движении.

Решим такую задачу: Робот должен двигаться вдоль стены на определенном расстоянии  $L$ .

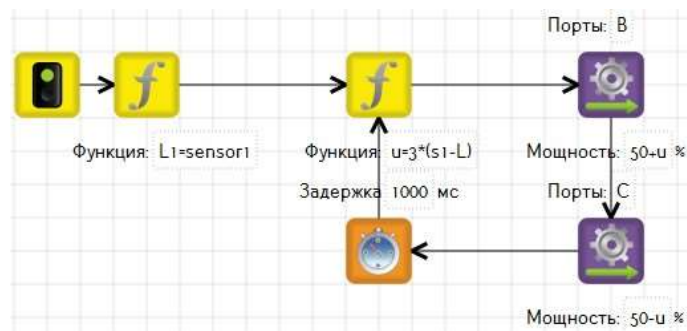
Из датчиков присутствует сонар, подключенный к порту A1.

Используем для исправления ошибки пропорциональный регулятор.  
 Поправка  $U=k*(Sensor1-L)$ .

TRIK:

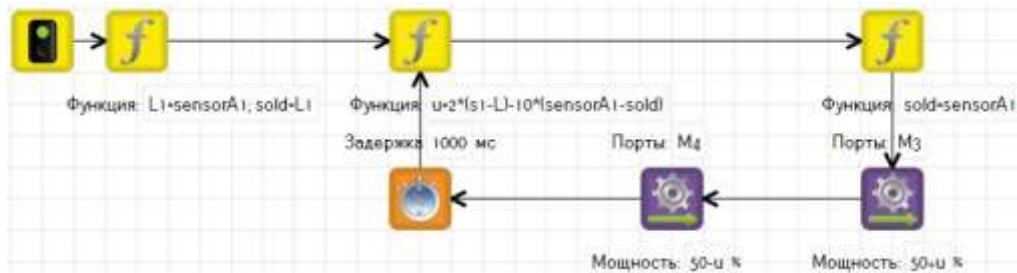


NXT:



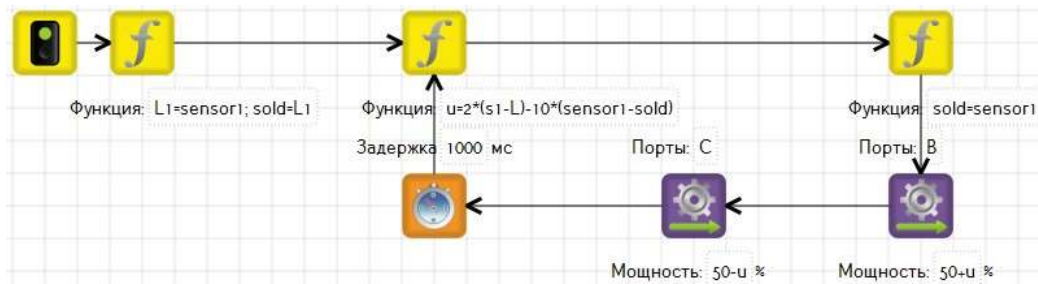
В некоторых ситуациях П-регулятор может вывести систему из устойчивого состояния. Например, если робот направлен от стены, но находится по отношению к ней ближе заданного расстояния, на моторы поступит команда еще больше повернуть от стены. Чтобы не попасть в такое положение, исправим ошибку при помощи пропорциональнодифференциального регулятора.

Поправка  $U=k1*(Sensor1-L)+k2*(Sensor1-Sold)$ , где Sold – расстояние до стены на предыдущем шаге измерения. TRIK:



NXT:





Задачи 1. Объехать вокруг коробки

2. Проехать вдоль стены, имеющей форму



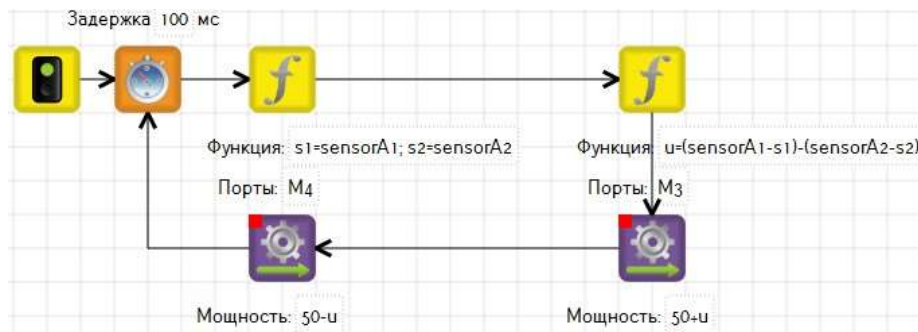
### Занятие 13. Подпрограмма – это маленькая программа

Часто бывает, что какие-то операции, выполняемые роботом, повторяются в различных частях программы. Их можно выделить и описать отдельно в блоке «подпрограмма». Перетащим блок «подпрограмма» в окно «диаграмма поведения робота». У каждой подпрограммы свое название. Щелкнув мышкой на блоке, откроем окно подпрограммы.

Начало любой подпрограммы – блок «начало». Заканчивается она блоком «конец». В отличие от программы, подпрограмма не может быть бесконечным циклом.

Пример: Пусть наш робот после каждого поворота едет по линии, что бы каждый раз не писать алгоритм движения по линии будем использовать подпрограмму





То есть, если подпрограмма зациклена, нам невозможно будет перейти к следующему шагу. Исправим ее.



Задачи:

1. Проехать по линии с тремя перекрестками и остановиться на 3-м
2. Проехать по линии с разрывом, вдоль которого установлена стена, во время разрыва ехать вдоль стены

## Занятие 14. Кегельринг

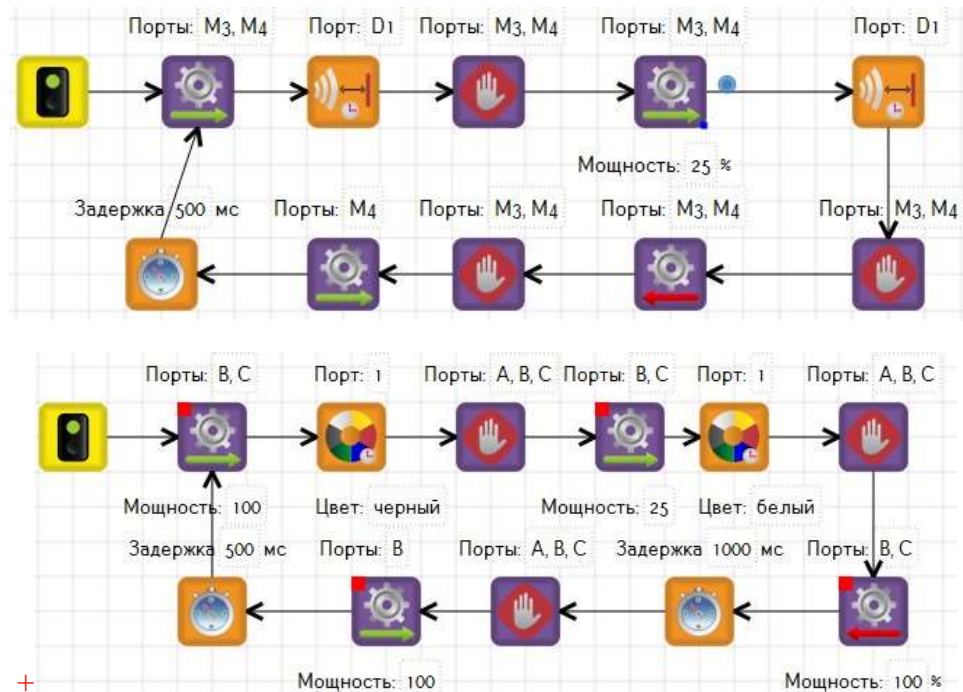
Кегельринг – это одно из состязаний, устраиваемое среди робототехников. Цель состязания – вытолкнуть за пределы очерченного круга диаметром 1 метр все расположенные внутри круга кегли (условия соревнований можно посмотреть здесь). Конструкция робота ограничена только линейными размерами и отсутствием специальных приспособлений.

Решим несколько задач для построения нужного алгоритма. «Танец

в круге»



«Танец в круге» с выездом точно за пределы круга.



«Танец в круге» с движением по датчикам оборотов.





Добавим сонар для более успешного поиска кеглей.

«Танец в кругу» с поиском кеглей.

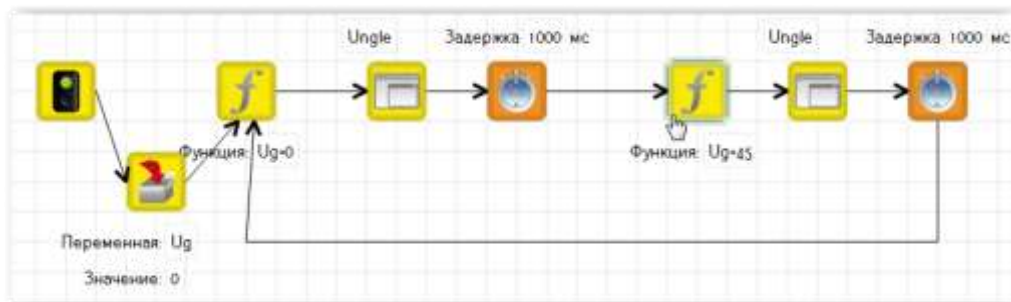


На соревнованиях часто используется алгоритм движения по спирали, перекрывающей всю площадь круга, не использующий датчики.

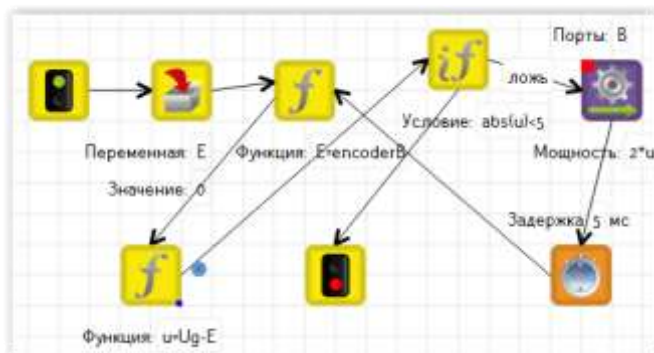
### Занятие 15 Манипуляторы

На сегодняшний день ни одно из промышленных предприятий не обходится без манипуляторов. Для расширения возможностей на работах также устанавливаются манипуляторы. Интересная модель манипулятора с тремя степенями передвижений приведена в книге А.С.Филиппова «Роботехника для детей и родителей». Напишем программу для этого манипулятора.

Первый мотор отвечает за горизонтальное перемещение захвата и фиксируется в двух положениях  $0^{\circ}$  и  $45^{\circ}$ . Используем П-регулятор для стабилизации положений.



Подпрограмма Ungle:



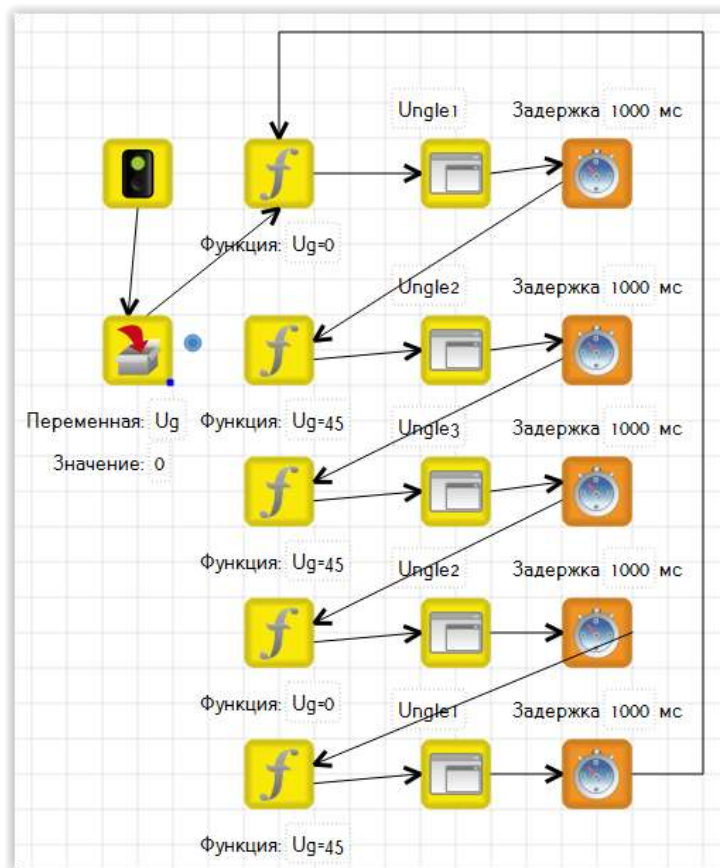
Как мы помним, подпрограмма не может быть цикличной, поэтому условие  $(UgEncoderB) < 5$  и будет означать выход из цикла. На малых мощностях моторов их работа не стабильна, поэтому используем коэффициент 2.

Второй мотор отвечает за вертикальные перемещения и также стабилизируется в двух положениях. Т.е. задача аналогична первой.

Третий мотор отвечает собственно за захват.

В результате имеем.





Подпрограммы работают с разными энкодерами.

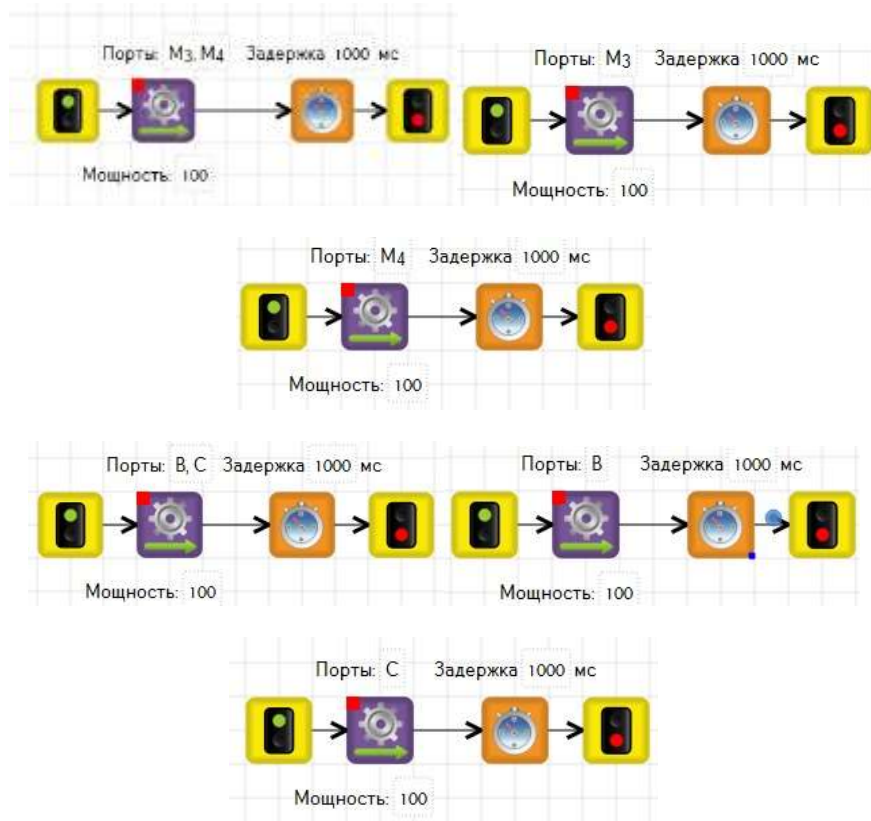
В конструкторе TRIK используются сервомоторы, поэтому программа для него будет гораздо проще:



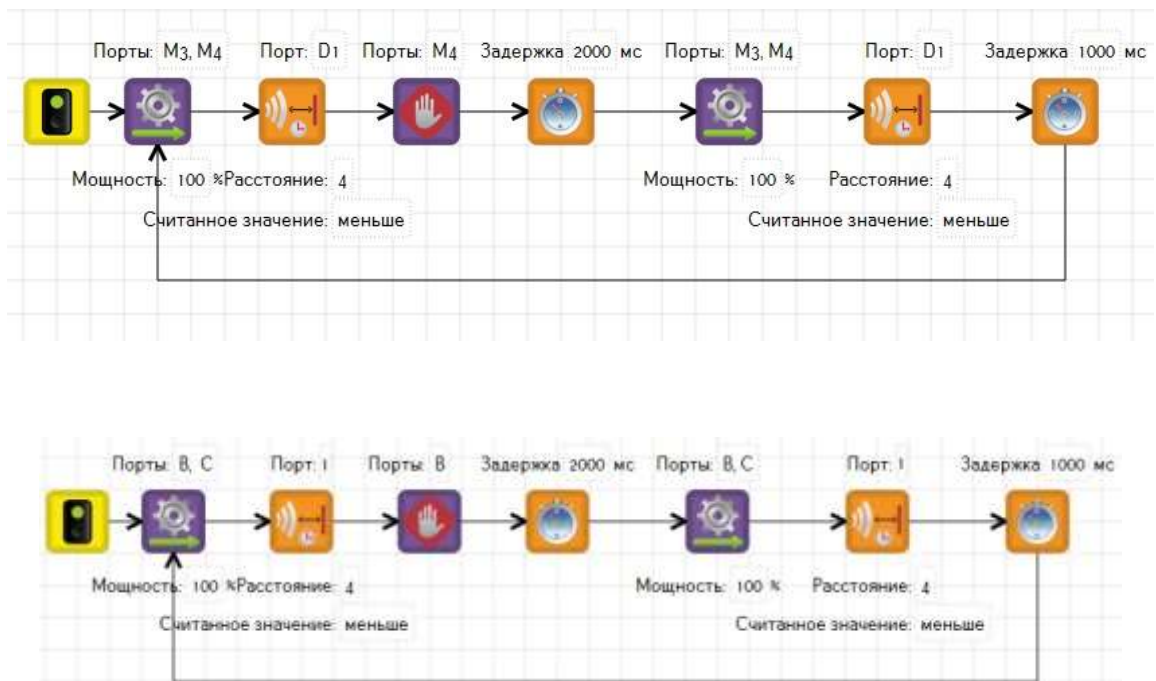
## Занятие 16. Лабиринт

Поиск выхода из лабиринта является классической задачей, которую решают не только роботехники, но и программисты.

Для начала, допустим, что лабиринт нам известен. Напишем алгоритм прохождения по заданному маршруту. Выделим подпрограммы: «шаг вперед», «поворот налево», «поворот на право».



В случае неизвестного лабиринта добавим к нашей модели сонар. В зависимости от показаний датчика будут выполняться наши подпрограммы, к которым добавится разворот на 180°.

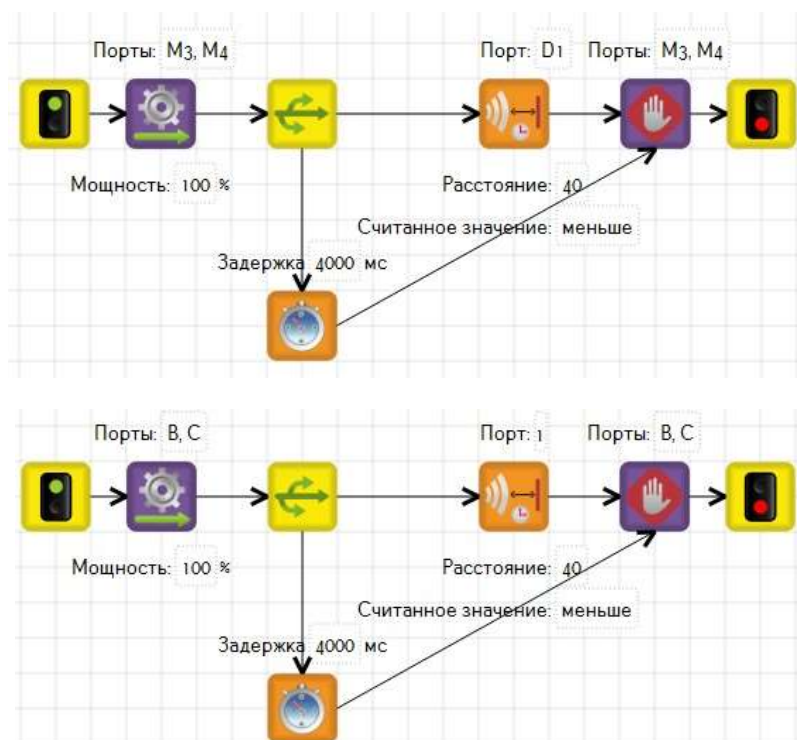


## Занятие 17. Исследователь

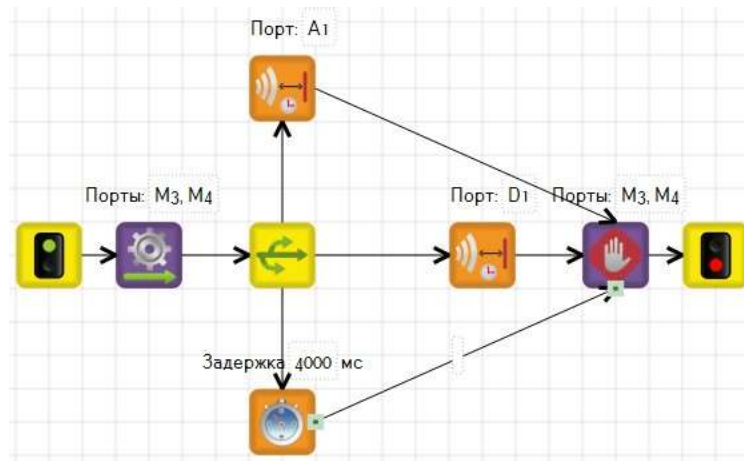
Сконструируем робота для передвижения по комнате так, чтобы он не натыкался на предметы и не застревал. Для начала оснастим его ультразвуковым датчиком. Алгоритм путешествия следующий: вперед до предмета – отъезд – разворот.



Используем таймер для защиты от застревания.



Добавим роботу датчик касания. Программа путешествия примет следующий вид.



Задачи 1. Напишите алгоритм для робота-пылесоса

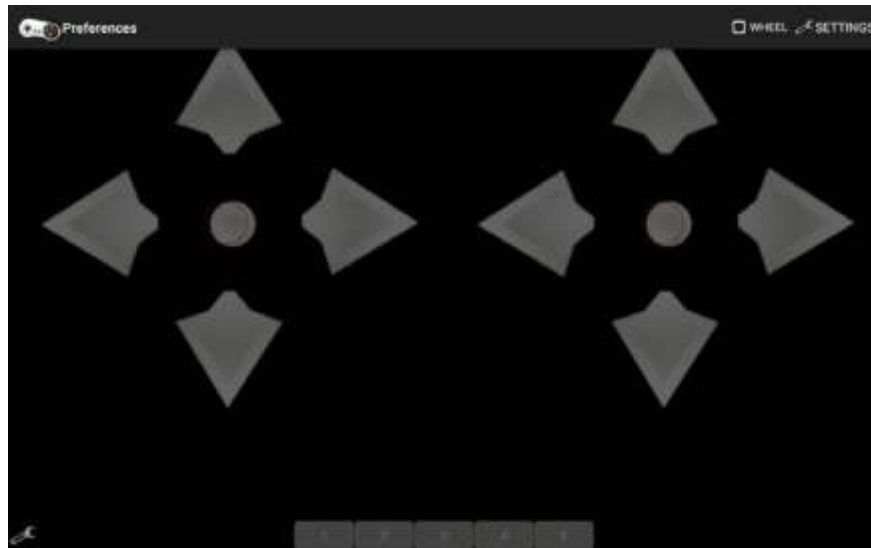
2. Найдите не проходимые места в комнате для робота

### Занятие 18. Дистанционное управление (Только для контроллера TRIK)

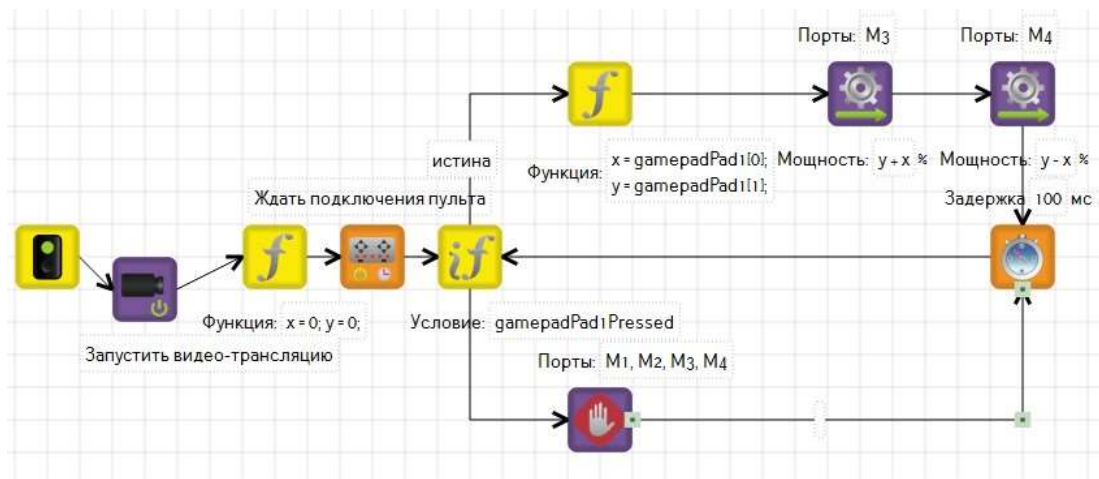
Робот может двигаться не только автономно, им можно управлять при помощи androidприложения TRIKGamepad (ссылка для скачивания <https://play.google.com/store/apps/details?id=com.trik.gamepad&hl=ru>)

В приложении имеется два джойстика, для управления моторами и 5 программируемых кнопок, посылающих сигнал контроллеру. С помощью приложения можно выводить трансляцию видео на экран мобильного устройства.





Рассмотрим программу движения робота, находящуюся в папке примеры (путь по умолчанию C:\TRIKStudio\examples\trik): remoteControl



При помощи запускается трансляция и идет ожидание подключения пульта. В блоке “if” проверяем, нажат ли левый джойстик. Если значение блока “if” истинно, то на моторы подается мощность, в зависящая от того, где находится палец на координатной оси, где осями являются горизонтальные и вертикальные клавиши джойстика.

можно запрограммировать правый джойстик при помощи переменной gamepad2Pad и иконки «ждать “руля” на пульте». Клавиши, расположенные внизу экрана, программируются с помощью иконки «ждать кнопки на пульте».

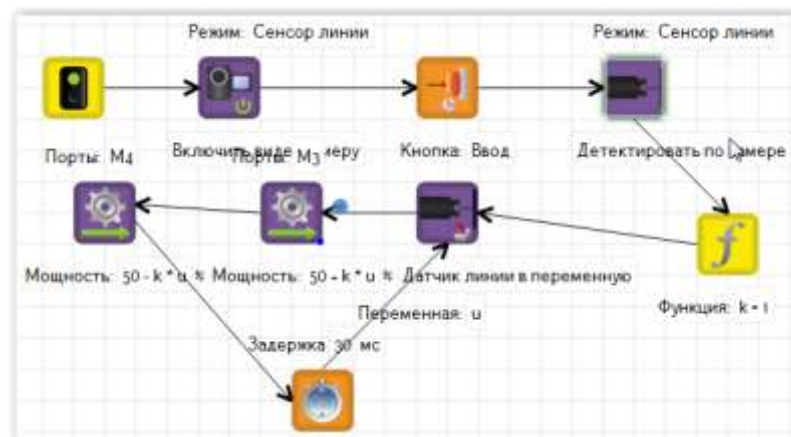
Задачи:

1. После нажатия клавиши «1» на пульте управления тележка должна описать восьмерку, а после нажатия клавиши «2» вернуться задом по той же восьмерке в начально положение
2. К вышеуказанному примеру допишите управление сервомотором разместив на нем видеокамеру.

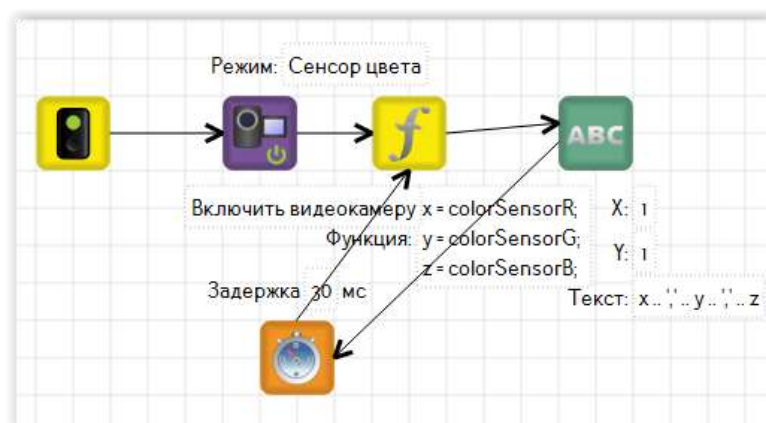
### Занятие 19. Камера контроллера ТРИК.

Видеомодуль или usb-камера, которые можно подключать к контроллеру ТРИК, выполняют несколько различных функций. Их можно использовать как датчик линии, датчик цвета, объектный датчик и непосредственно как камеру.

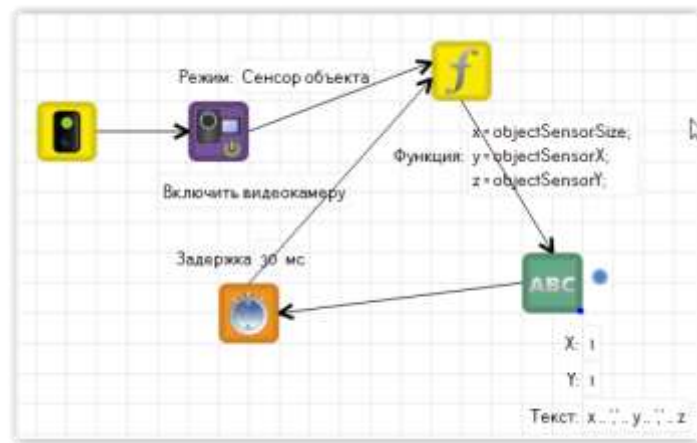
Камера – датчик линии:используется П-регулятор.



Камера – датчик цвета: Цвета передаются тремя числами от 0 до 255 в формате RGB.



Камера – датчик объекта: Передается размер объекта (количество точек на дисплее) и координаты его центра масс.



Видеотрансляция

