Порохова Ирина Алексеевна

РОБОТЫ LEGO WEDO. ЗАНЯТИЕ 5. СЛЕДОВАНИЕ ПО ЛИНИИ И БАЛАНСИРУЮЩИЙ РОБОТ

С каждым годом роботы становятся все более популярными, открываются новые кружки робототехники, осуществляются поставки в школы, проводятся соревнования и выставки. Наверняка вы видели, как роботы ездят по линии или играют в футбол, возможно, вы даже знакомы с роботами Lego NXT и имеете представление о возможностях продвинутых роботов. Оказывается, что и наш детский конструктор Lego WeDo не такой уж и детский, на нем можно решать настоящие робототехнические задачи.

В этой статье мы рассмотрим две задачи для роботов: следование по линии и управление балансирующим на двух колесиках роботом-сигвейем. Обе эти задачи возможно реализовать в среде Скретч и только благодаря тому, что, как обсуждалось в предыдущей статье, датчик расстояния на самом деле является датчиком освещенности, он определяет не расстояние до объекта, а интенсивность отраженного света.

Датчик расстояния по-разному реагирует на светлую и темную поверхности, в чем легко можно убедиться, например, в Скретче. Для этого надо расположить датчик на расстоянии 10 см от стола, положить на стол сначала белый лист, заметить значение, которое выдаст установка 

, а затем, не меняя положения датчика, положить черный лист и зафиксировать новое значение. Оно будет другим.

СЛЕДОВАНИЕ ПО ЛИНИИ

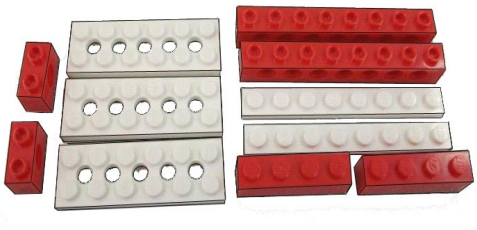
Соберем машинку и оснастим ее датчиком расстояния, который будет смотреть в пол. Конструкция машинки может быть самой разной, единственное условие – у нее должен быть один мотор. Ограничения модели связаны с тем, что среда Скретч версии 1.4 поддерживает только один мотор, поэтому наша машинка не сможет самостоятельно поворачивать. В принципе можно пытаться строить машинку с храповыми механизмами, которая будет поворачивать и на одном моторе, например, по аналогии с машинкой из довольно старого Лего-конструктора «Город и транспортные средства» 9723.

На рис. 1 показаны блочные этапы сборки одного из возможных вариантов машинки. Модель использует коронную передачу для приведения во вращение передних колес. Так как преследовалась цель обойтись деталями одного конструктора, задних колес машинка не имеет, в качестве колеса используется круглый кирпич.

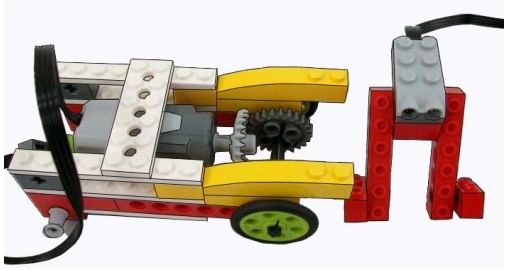
Далее надо подготовить поле. Можно взять белые листы бумаги и наклеить на них черную изоленту, можно взять готовое поле для роботов NXT или найти в интернете поля для соревнований и распечатать их фрагменты.

Черную линию на белом фоне робот видит совсем иначе, чем человек (см. рис. 2). Для робота отсутствует резкая граница между цветами, он видит не белое и черное, а оттенки серого. Граница линии предстает для робота серой областью, и наша первая задача объяснить роботу, что именно является для робота границей. Границей черного и белого будет являться среднее значение между черным и белым, это значение будем обозначать как серое.

а)

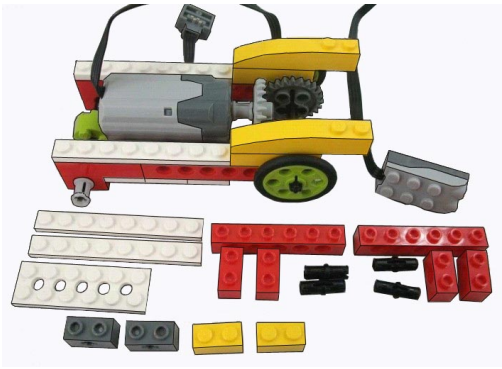


г)



б)

д)



в)

е)

Рис. 1

Создадим в среде Скретч четыре спрайта: вертолетик из стандартных картинок, он будет символизировать нашу машинку, и три кнопки – «установить черное», «установить

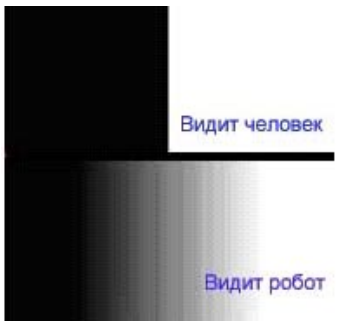
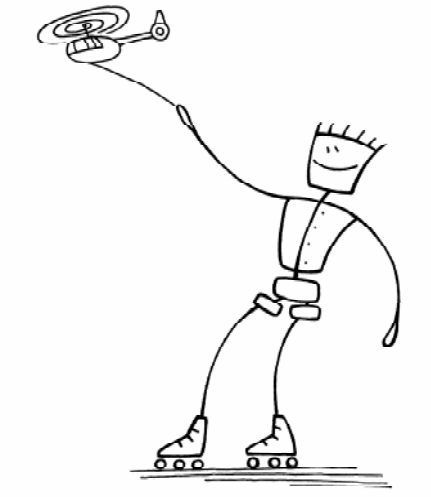


Рис. 2



...вертолетик из стандартных картинок, он будет символизировать нашу машинку...

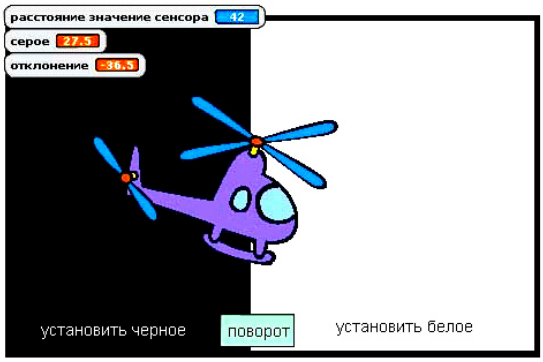


Рис. 3

белое» и «поворот». Сцену можно раскрасить так, как показано на рис. 3.

Для спрайтов-кнопок понадобятся следующие скрипты:

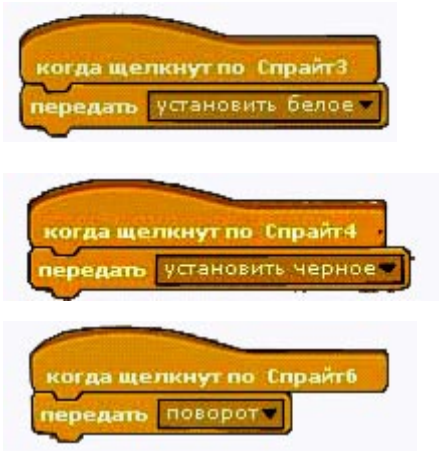


Рис. 4

Основная программа, написанная для спрайта «вертолетик» будет ждать щелчков по этим кнопкам.

Программа использует пять переменных:

белое – значение датчика расстояния на белом фоне,

черное – значение датчика расстояния на черном фоне,

серое – среднее арифметическое между черным и белым,

отклонение – разность текущего значения от датчика расстояния и среднего,

поворот – переменная, принимающая значения 0 и 1 в зависимости от нажатия кнопки «поворот».

Сначала переменные *белое* и *черное* устанавливаются в значение ноль. Когда появится сообщение «Установите белый цвет», поставьте робота на белый фон и нажмите кнопку «установить белое». В переменную *белое* будет записано значение от датчика расстояния на белом фоне. Когда программа скажет: «Установите черный цвет», переставьте робота на черную линию и щелкните мышкой по надписи «установить черное». В переменную *черное* будет записано значение от датчика расстояния на черном фоне. Далее будет вычислено значение переменной *серое*, соответствующее границе черного и белого.

Поскольку наша машинка не умеет поворачивать, в основном цикле программы определяется необходимость поворота, и подаются команды. На каждом шаге вычисляется *отклонение* от линии, то есть разность между *серым* и текущим значением от датчика расстояния. Если *отклонение* мало, то поворачивать не надо, и машинка едет вперед в течение 0,15 секунд. Если *отклонение* велико и отрицательно, т.е. машинка находится над черной линией, программа говорит «поверните меня направо». Надо немного повернуть машинку вправо и нажать кнопку «поворот», после чего выполнение цикла продолжится. Если же *отклонение* велико и положительно, то машинка находится на белом фоне, программа командует «поверните меня налево». Надо повернуть ее чуть-чуть налево и щелкнуть по надписи «поворот». Программа в цикле перейдет к анализу текущего отклонения от серого и примет решение о движении вперед или необходимости еще одного поворота (рис. 5).

Если бы благодаря своей конструкции машинка умела поворачивать, тогда вместо словесных команд на поворот можно было бы использовать команды для мощности мотора и направления вращения. Тогда машинка смогла бы двигаться «самостоятельно» вдоль линии.

Темы для развития, обсуждения, исследований и модернизации модели:

1. Значения от датчика расстояния меняются в диапазоне от 0 до 100. Какие значения будет показывать датчик на белом, черном, красном, зеленом и синем фонах?

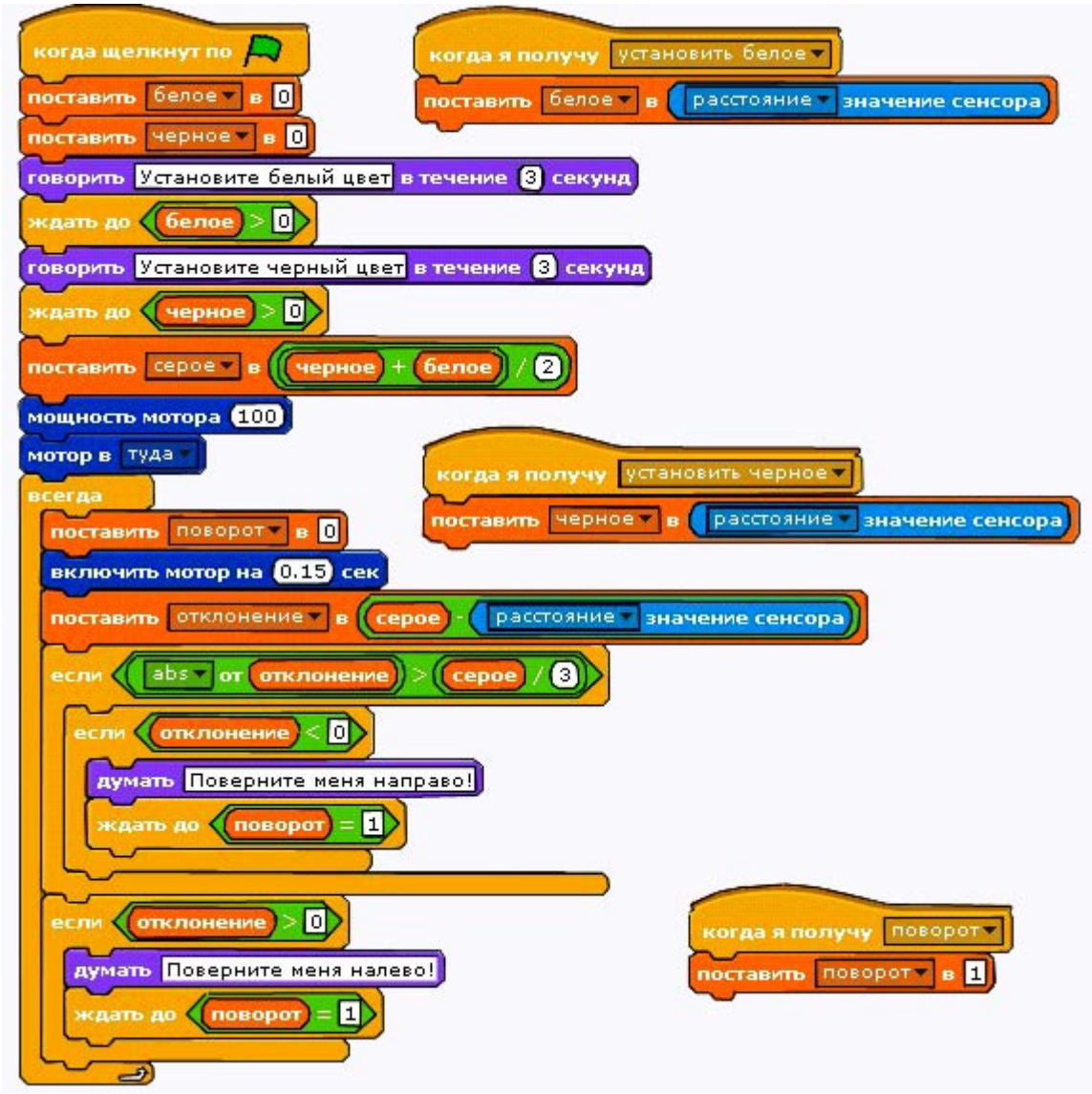
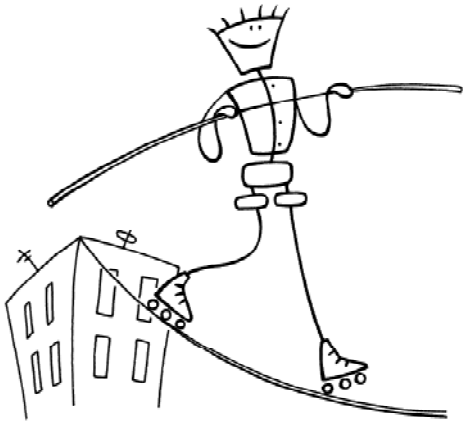


Рис. 5

2. На каком минимальном расстоянии от поверхности значения от датчика расстояния будут отличаться от нуля?

3. Измените программу, добавьте смещение вертолетика от границы черного и белого пропорционально значениям переменной *отклонение*, попробуйте добавить звуки. Пусть команды на поворот сопровождаются звуками.



Модель балансирующего робота является одной из самых впечатляющих, сложных и интересных…

БАЛАНСИРУЮЩИЙ РОБОТ

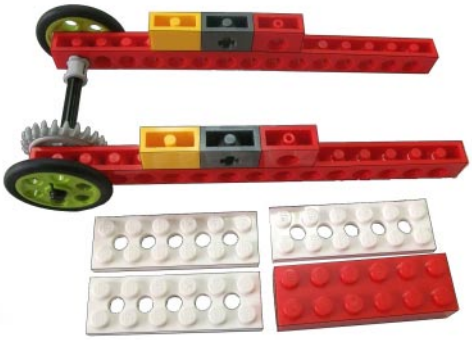
Модель балансирующего робота является одной из самых впечатляющих, сложных

и интересных (см. http://www.youtube.com/watch?v=q9ZONn3p1LI). Существует много модификаций таких роботов, основанных на датчике-гироскопе, датчике цвета и датчике освещенности. Наша модель будет использовать датчик освещенности (то есть датчик расстояния) и будет выполнена из деталей одного конструктора Lego WeDo.

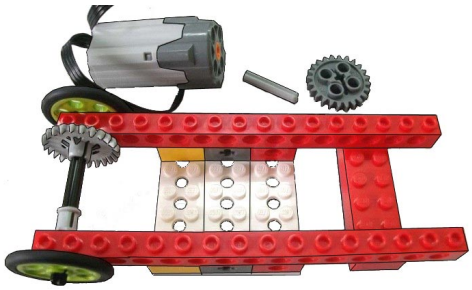
В модели (рис. 6) мотор вращает большое зубчатое колесо. Большое зубчатое колесо вращает коронное колесо, насаженное на ось с колесами робота. Сверху закреплен датчик расстояния. Конструкция может сохранять равновесие, только находясь в постоянном движении.

а)

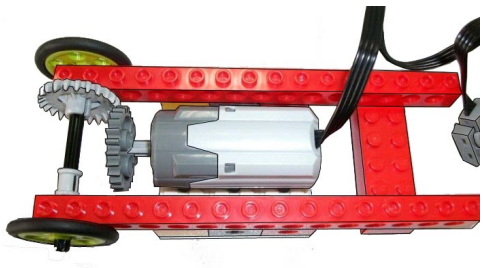
б)



в)

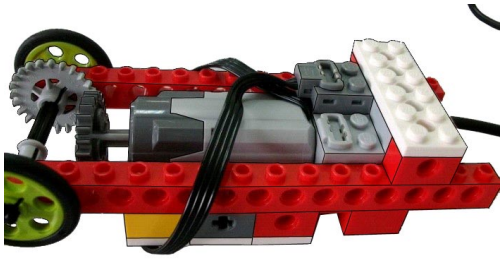


г)



д)

е)



ж)

з)

и)

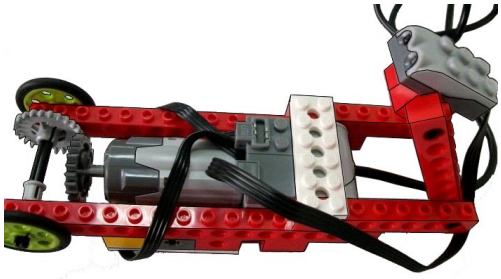


Рис. 6

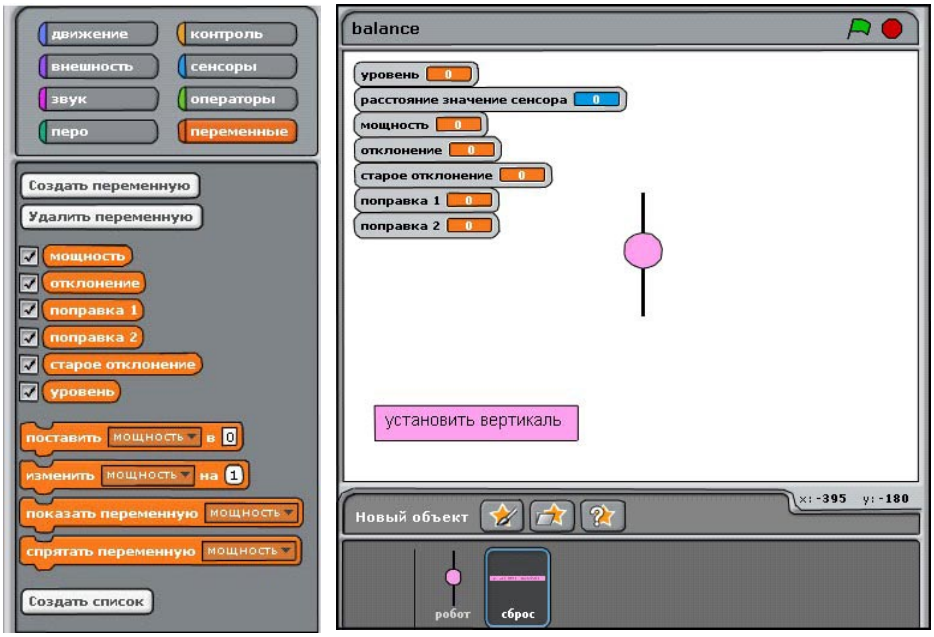


Рис. 7

Робота надо установить вертикально на колесики, придерживая за провод, датчик расстояния должен смотреть четко вниз, провода следует убрать, чтобы они не попадали в область видения датчика.

На рис. 7 показаны экранное поле и переменные для управления роботом.

Для управления балансирующим роботом потребуются следующие переменные:

– *уровень* – значение от датчика расстояния при вертикальном положении робота,

– *отклонение* – ошибка, или разность текущего значения от датчика расстояния и его значения в вертикальном положении (*уровень*),

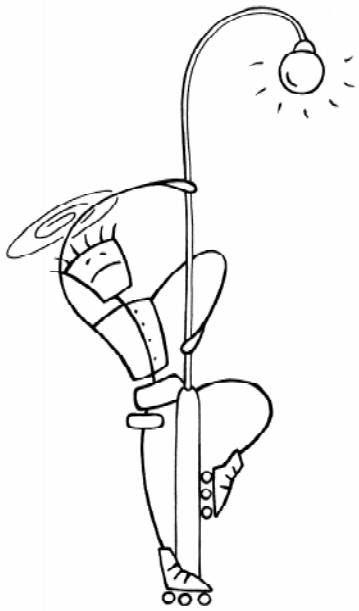
– *старое отклонение* – ошибка в предыдущий момент времени,

поправка 1 – управляющее воздействие, пропорциональный регулятор,

– *поправка 2* – управляющее воздействие, более сложный дифференциальный регулятор,

– *мощность* – значение для мощности мотора и направления вращения мотора.

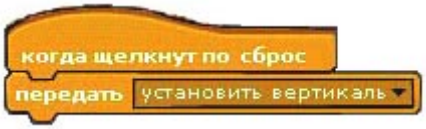
В программе использовано два спрайта. Первый – кнопка «установить вертикаль». Следует установить робота как можно более четко вертикально и нажать эту кнопку. Значение от датчика расстояния будет записано в переменную «уровень». Значения отклонений обнулятся. Второй спрайт – это шарик на линии, демонстрирующий отклонение робота от вертикали. Если линия оказывается в положении, близком к горизонтальному, следует остановить скрипт, снова установить робота вертикально, нажать



Следует установить робота как можно более четко вертикально...

кнопку «установить вертикаль» и перезапустить скрипт.

Скрипт для спрайта «сброс» весьма прост



Скрипт для спрайта «робот» показан на рис. 8.

Для запуска робота следует установить робота вертикально и щелкнуть мышкой по надписи «установить вертикаль». При этом будет передано сообщение «установить вертикаль», переменная *уровень* будет установлена в значение, которое передаст датчик расстояния, а значения отклонений *отклонение* и *старое отклонение* обнулятся. Эти параметры соответствуют вертикальному положению робота.

Далее в основном цикле программы на каждом шаге вычисляется отклонение робота от вертикали, то есть разность текущего значения от датчика расстояния и значения в вертикальном положении *уровень*. Затем вычисляются две поправки и переменная *мощность*. Мощность, подаваемая на мотор, вычисляется как модуль *мощности*, а знак *мощности* определяет направление вращения мотора. Таким образом, если робот начинает падать вперед, мотор начинает вращаться так, чтобы робот ехал вперед, а если робот падает назад, то робот должен ехать назад, причем тем быстрее, чем сильнее он падает.

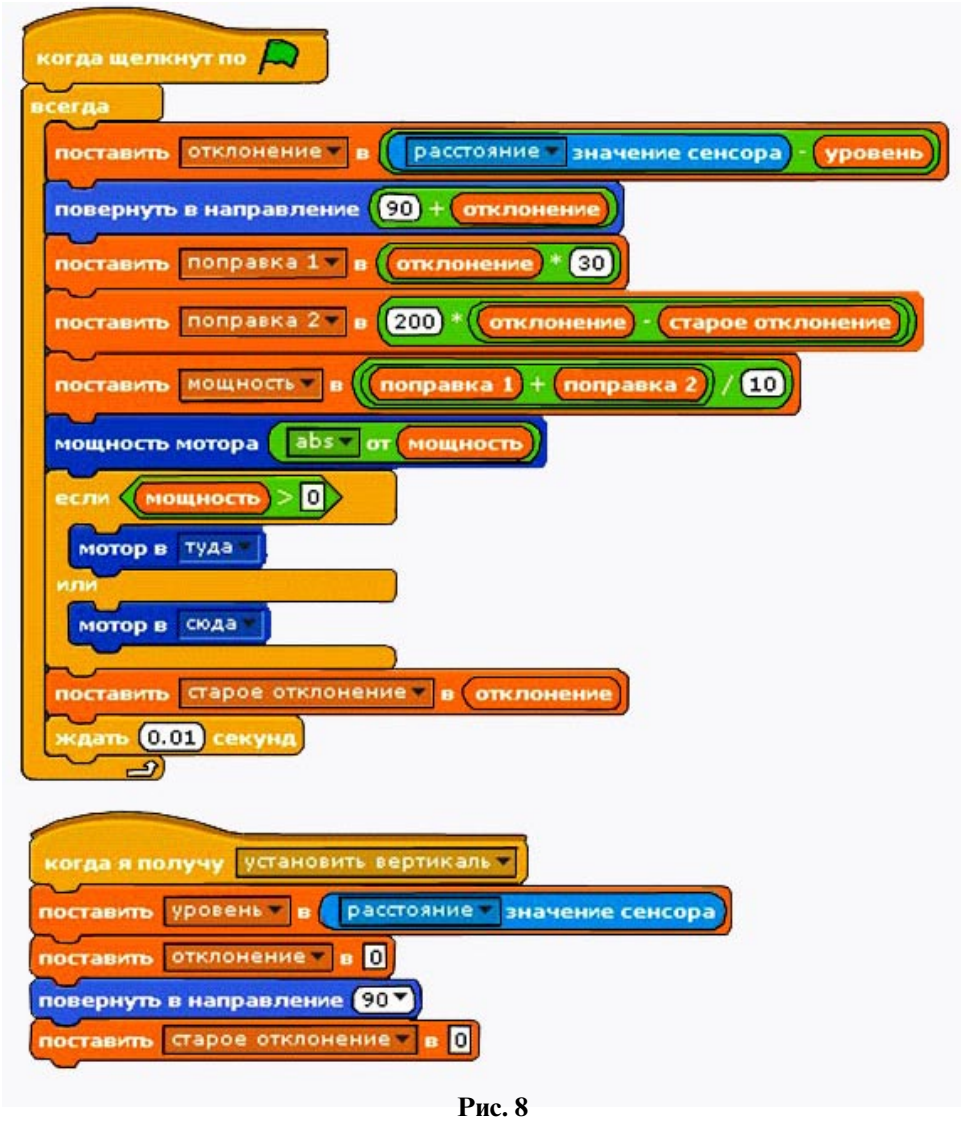


Рис. 8

Темы для обсуждения, исследований и модернизации модели:

1. Как вы думаете, можно ли построить балансирующего робота на датчике наклона вместо датчика расстояния?

2. Попробуйте изменить коэффициенты 30 и 200 в выражениях для управляющих воздействий *поправка 1* и *поправка 2*. Возможно, вам удастся подобрать другие коэффициенты усиления, при которых робот будет двигаться более устойчиво.

3. Добавьте музыкальное сопровождение.

4. В книжке С.А. Филиппова «Робототехника для детей и родителей» на стр. 187, написанной для роботов NXT, приведен немного более сложный алгоритм балансировщика, основанный на трех регуляторах. Этот алгоритм дает еще более устойчивое поведение балансирующего робота. Если у вас есть эта книжка, попробуйте разобраться и реализовать в Скретче этот алгоритм.

В идеале балансирующий робот должен быть построен на ПИД-регуляторе. Пропорционально-интегрально-дифференциальный (ПИД) регулятор используется в системах автоматического управления в управляющих контурах с обратной связью. ПИД-регулятор формирует управляющий сигнал, являющийся суммой трёх слагаемых, первое из которых пропорционально отклонению регулируемой величины от заданного значения, второе – интеграл от этого отклонения, третье – производная от отклонения. Если какие-то из составляющих не используются, то регулятор называют пропорционально-интегральным (ПИ), пропорционально-дифференциальным (ПД), пропорциональным (П) и т. п.

Пропорциональная составляющая вырабатывает выходной сигнал, противодействующий отклонению регулируемой величины от заданного значения. Он тем больше, чем больше это отклонение. Если входной сигнал равен заданному значению, то выходной равен нулю.

Дифференциальная составляющая пропорциональна скорости изменения отклонения регулируемой величины и предназначена для противодействия отклонениям от целевого значения, которые прогнозируются в будущем. Отклонения могут быть вызваны внешними возмущениями или запаздыванием воздействия регулятора на систему.

Интегральная составляющая пропорциональна интегралу от отклонения регулируемой величины. Её используют для устранения статической ошибки. Она позволяет регулятору со временем учесть статическую ошибку.

В рассмотренных задачах мы использовали П-регулятор при движенни по линии и ПД-регулятор при балансировке. При подборе коэффициентов усиления регуляторов сначала надо добиться достаточно устойчивого поведения модели на П-регуляторе, потом включить дифференциальную составляющую, а уже потом интегральную.

Порохова Ирина Алексеевна, кандидат физико-математических наук, методист Образовательного центра «ИНТОКС».