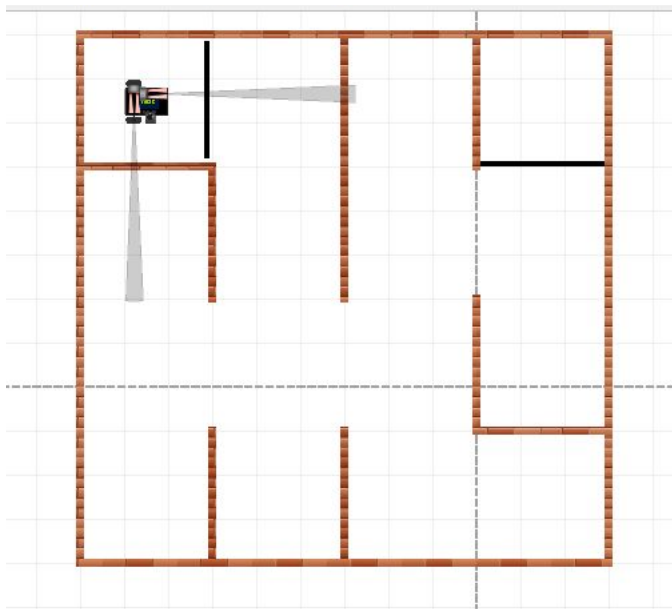


Презентация

Лабиринт (для дистанционных соревнований)

Логинов Андрей Анатольевич

Задача робота: Двигаясь внутри лабиринта, за время не более 3 минут, добраться из зоны старта в зону финиша, или как можно ближе к зоне финиша



Полигон для соревнования «Лабиринт» представляет собой прямоугольную поверхность с размером сторон три клетки. Вся поверхность полигона мысленно можно разделить на квадратные ячейки со сторонами в три клетки, расположенный смежно друг с другом. Между ячейками могут быть установлены не преодолимые для робота перегородки. По периметру полигона имеется ограждение той же высоты. Зоны старта и финиша выделены линиями черного цвета

Требования к роботу

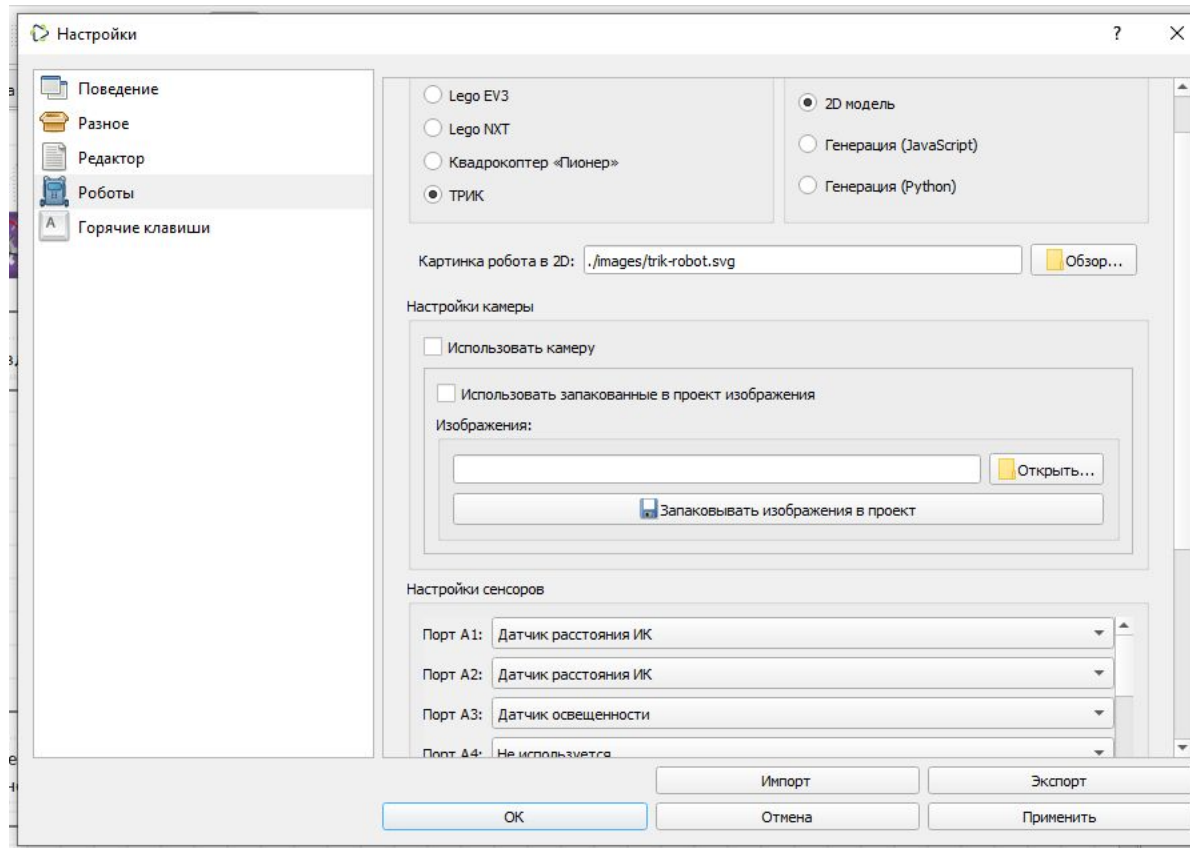


Для дистанционных соревнований «Лабиринт» робот строится на основе двухмоторной тележки из контроллера Lego EV3, Lego NXT или TRIK. Датчики расстояния ориентируются один – вперёд, другой – вправо или влево и располагаются так, чтобы они не выступали за габариты тележки для исключения зацепления ими за стенку при поворотах. Датчик света располагается аналогичным образом

Время заезда останавливается по истечении 3 минут, когда робот прибыл в финишную ячейку, либо когда робот не может покинуть ячейку в течении 15 секунд. Местонахождения робота фиксируется и производится подсчёт ячеек, которые роботу требуется пройти до зоны финиша по кратчайшему маршруту. Чем меньше количество таких ячеек, тем лучше результат. При равном количестве, побеждает тот, кто затратил меньше время.

Для нахождения выхода из лабиринта применяют правило правой (левой) руки: Если двигаться по лабиринту, держась рукой за правую (левую) стену, то рано или поздно выйдешь выходу из него, либо в точку начала пути, если лабиринт замкнут.

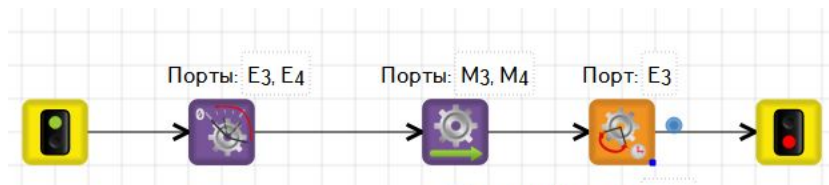
Для начала настроим работа в Trik Studio



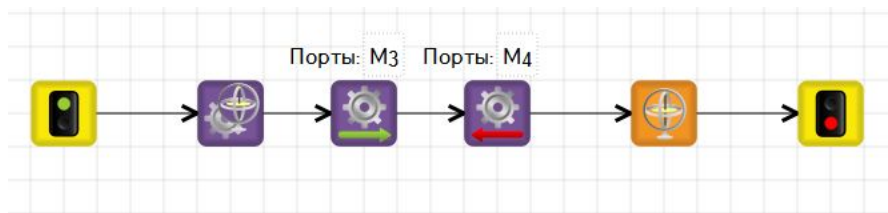
Робот ТРИК, датчики расстояния подключены к портам 1 и 2. Модель робота – 2D модель. Физика – реалистичная физика

Для оптимизации программного кода, сначала не будем составлять полную диаграмму, а создадим три подпрограммы: «Вперед», «Направо» и «Налево».

«Вперед» :



«Налево» :



«Направо» :



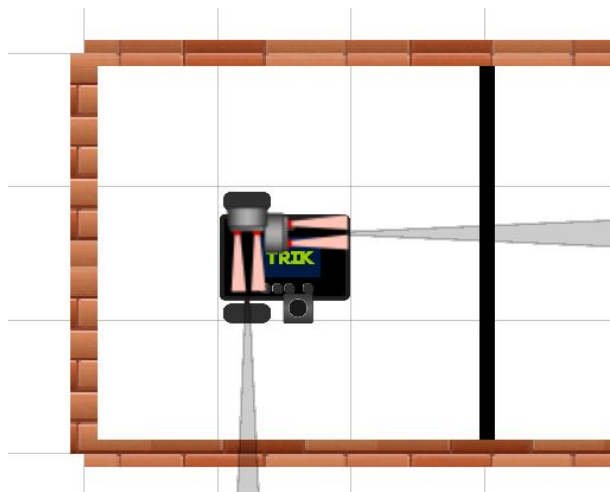
Перед написанием главной диаграммы, мы должны настроить каждую из них на виртуальном роботе. Здесь намеренно нет готовых параметров. Их вы должны подобрать при настройке подпрограмм.

В Trik Studio имеется возможность сохранить созданную подпрограмму в коллекцию подпрограмм. Тогда Вы можете её использовать при написании другой программы и сэкономите своё время. Делается это так:

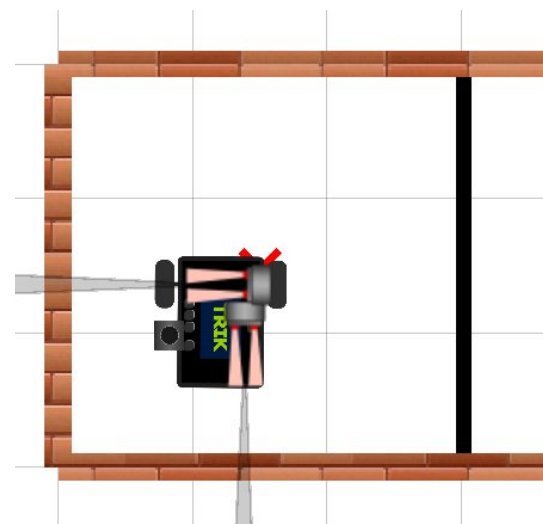
Инструменты -> Подпрограммы -> Сохранить в коллекцию

Поставить галочку напротив подпрограммы "Направо Gyro", кликнуть ОК

Однако, как Вы заметили, если подпрограмма «Вперёд» настраивается достаточно просто, то после поворота налево и направо, робот смещается относительно центра ячейки. А это негативно будет сказываться на всём дальнейшем его движении. При каждом повороте смещение всё больше и когда-то робот просто упрётся в стену.



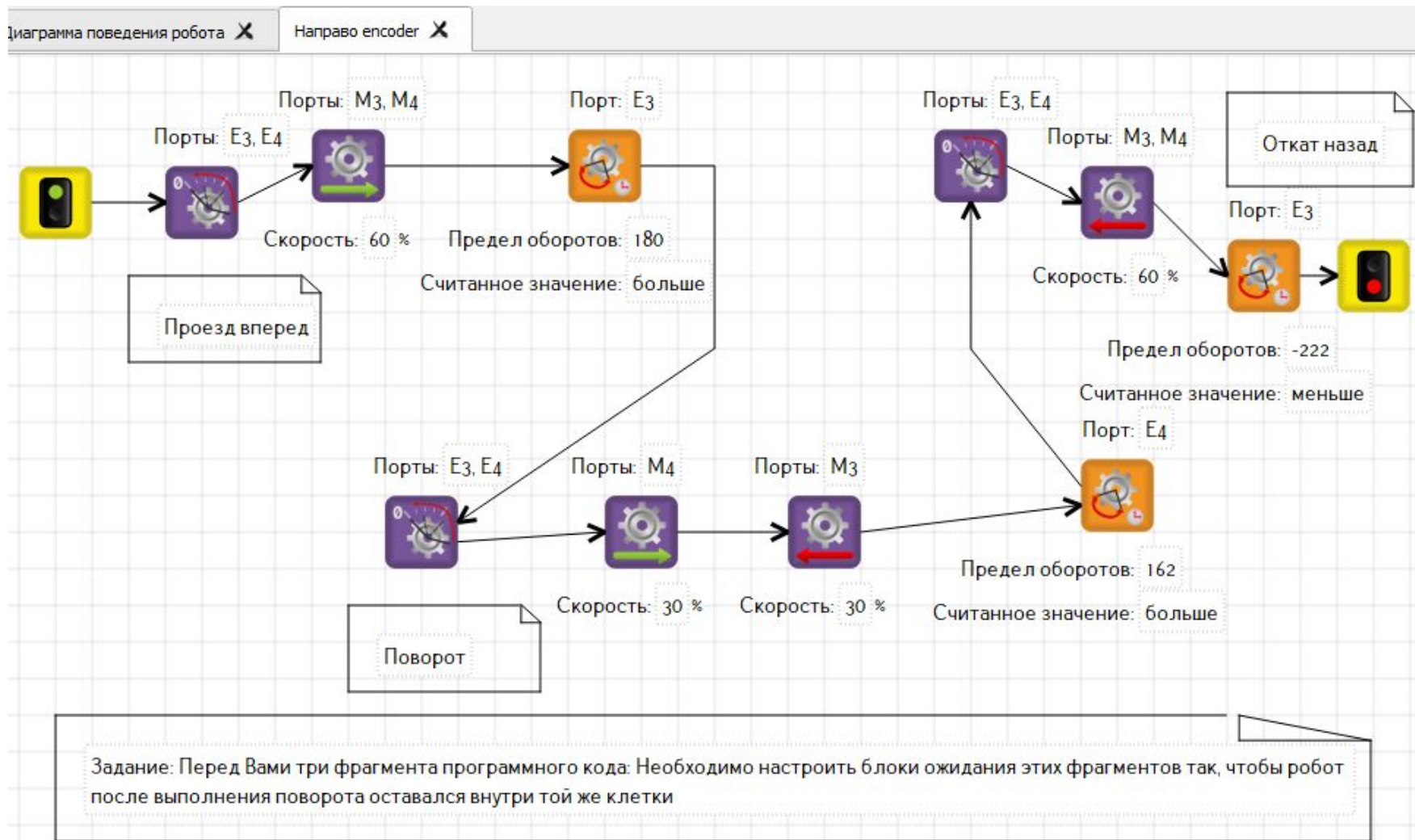
Перед поворотом



После поворота

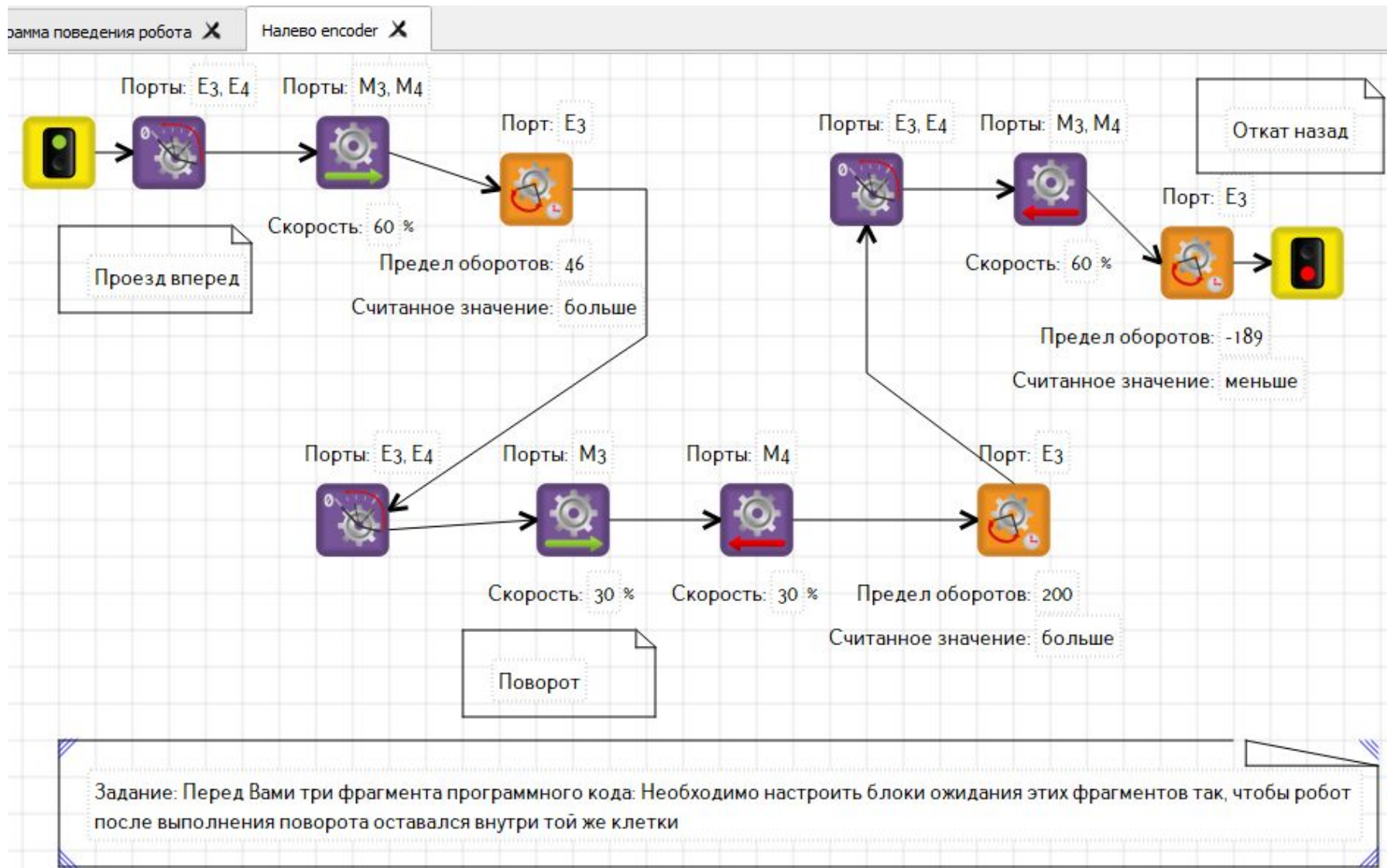
Для точного поворота на месте, кроме собственно поворота, необходимо сделать ещё два корректирующих движения: перед поворотом – проезд немного вперед, и откат назад после поворота. Тогда робот встанет точно в центр клетки. Таким образом, начиная каждый раз движение из центра клетки, робот постоянно будет находится посередине коридора.

Так выглядит подпрограмма для поворота направо по энкодеру:



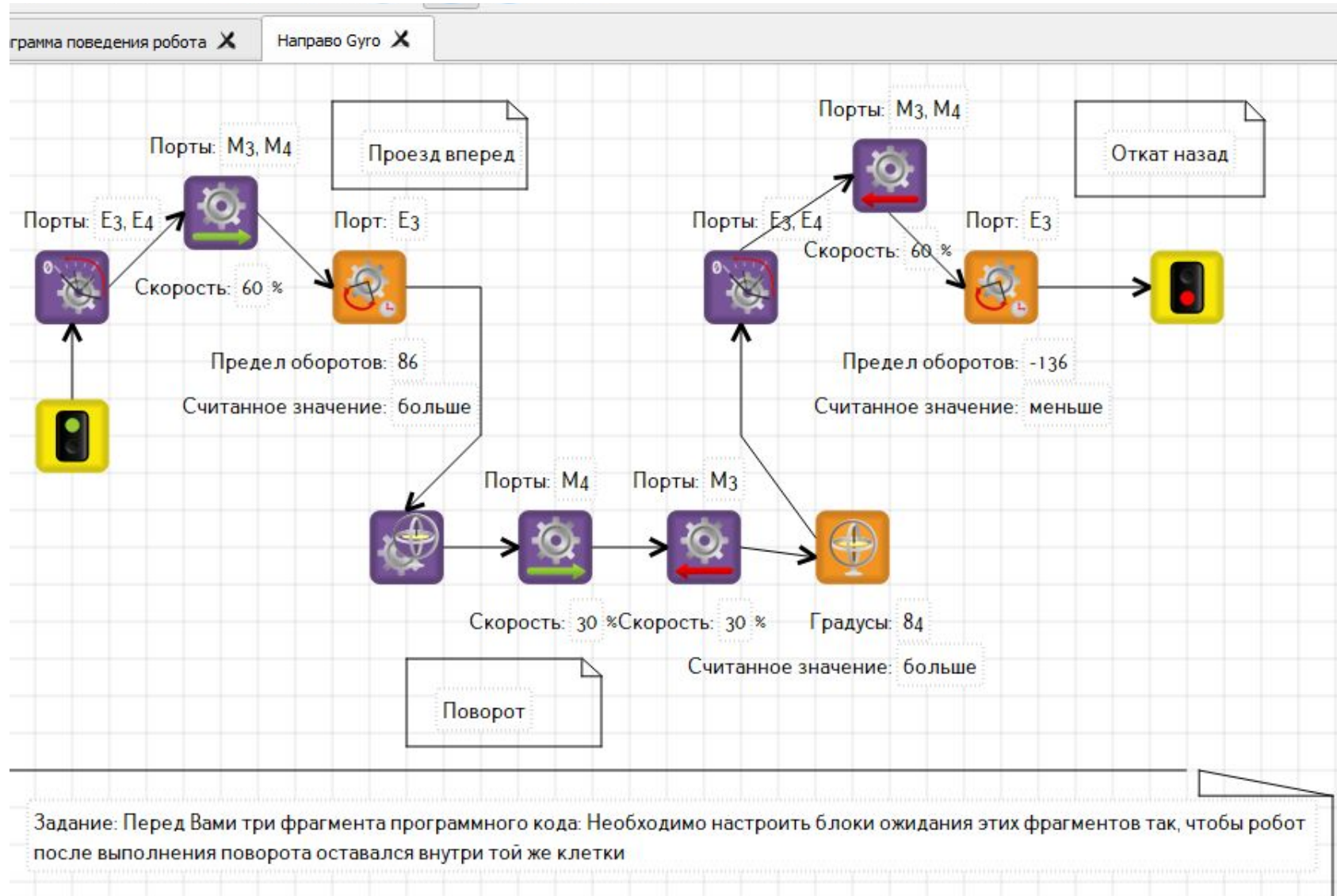
Здесь намеренно искажены настройки блоков ожидания. Их вы должны подобрать при настройке подпрограмм сами

Так выглядит подпрограмма для поворота налево по энкодеру:

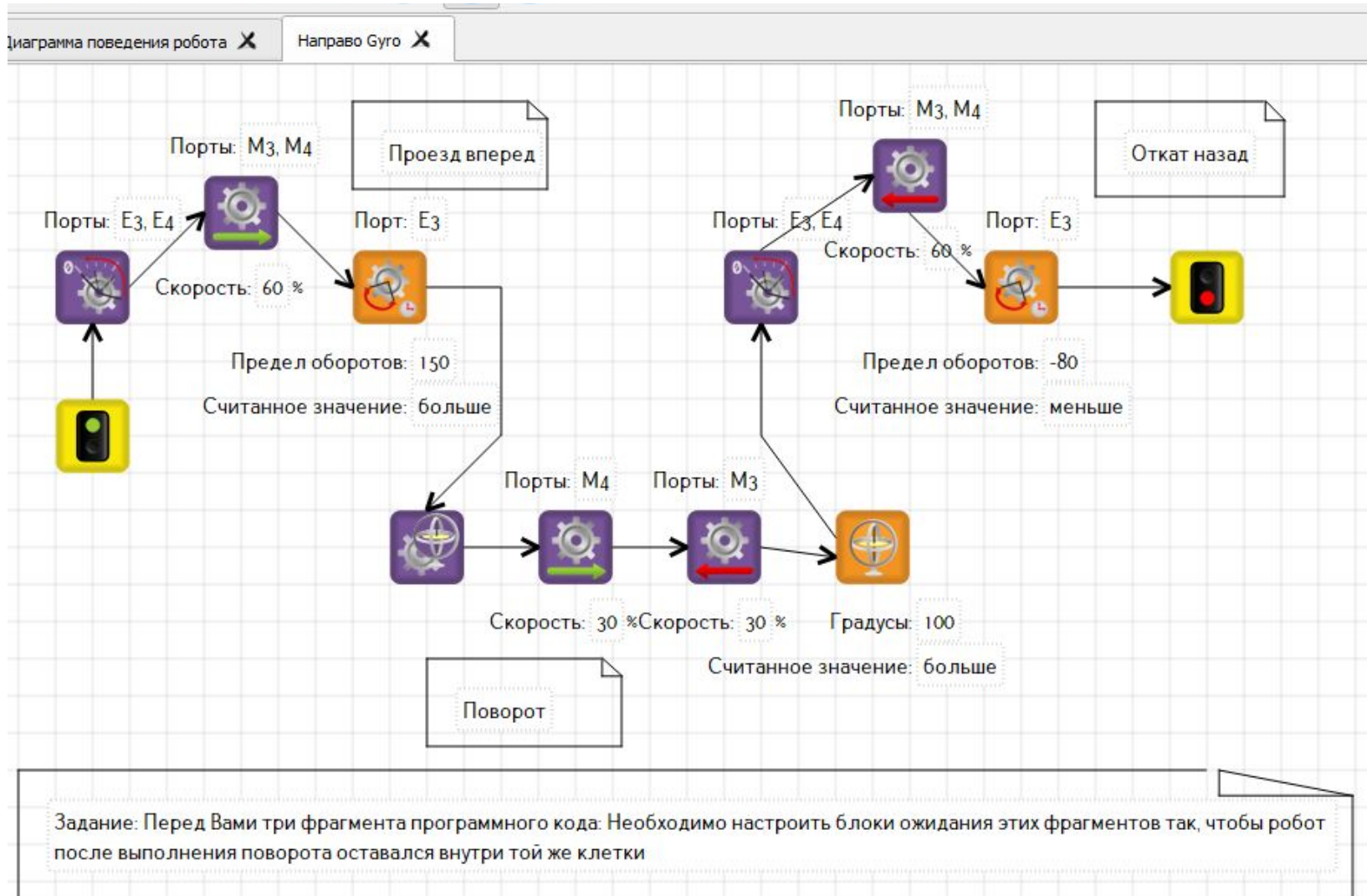


Здесь намеренно искажены настройки блоков ожидания. Их вы должны подобрать при настройке подпрограмм сами

Но кроме энкодера, выполнить поворот можно и по гироскопическому датчику. Поворот направо по гиродатчику:

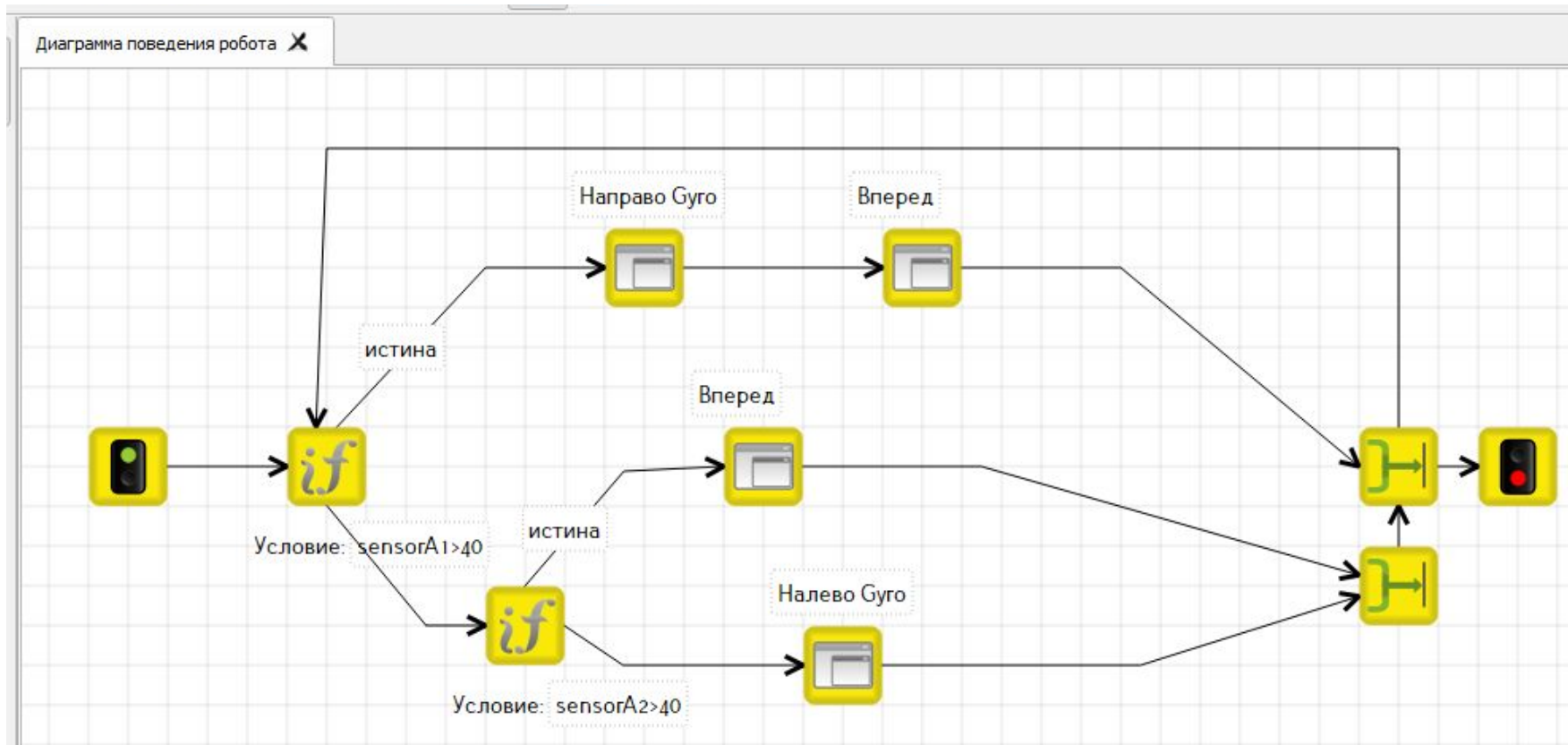


Поворот налево по гиродатчику:



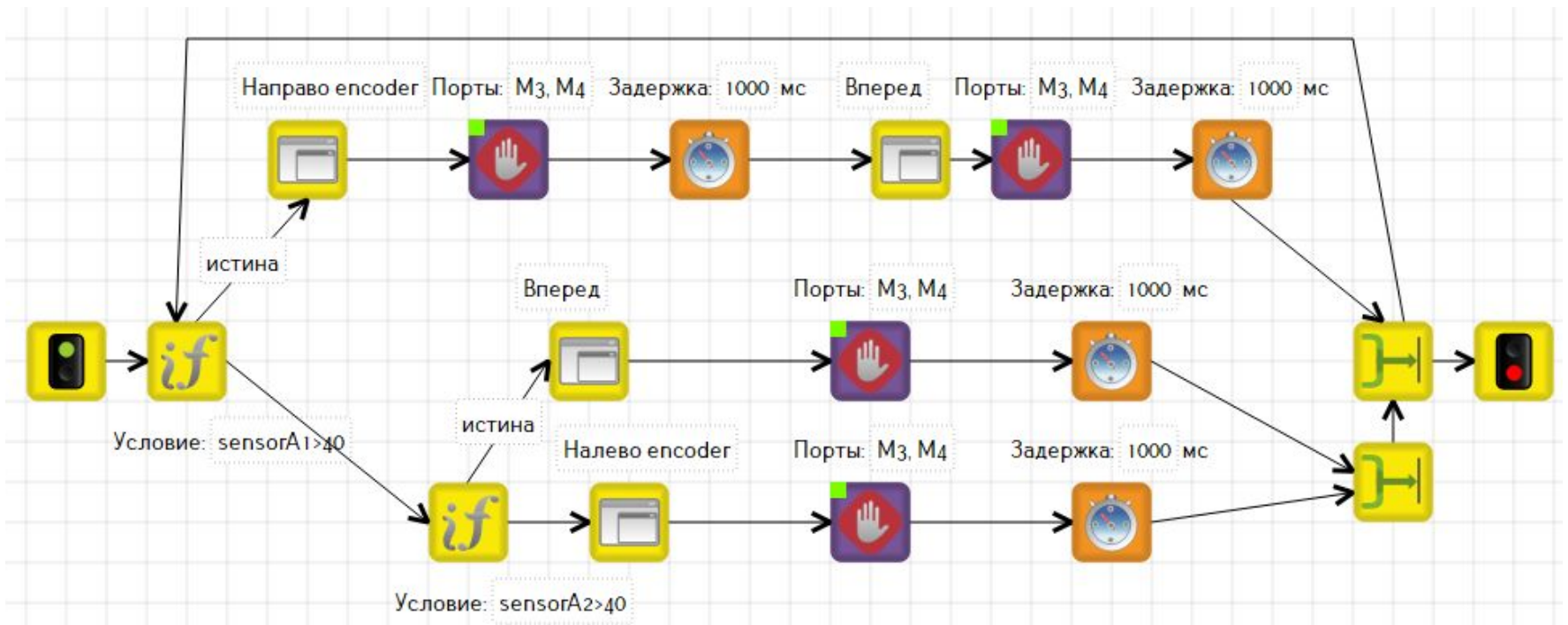
Напоминаю: Во всех примерах намеренно искажены настройки блоков ожидания. Их вы должны подобрать при настройке подпрограмм сами

Теперь, когда подпрограммы движений робота написаны, настроены и сохранены в коллекцию, можно приступить к сборке главного алгоритма. Главная программа (Диаграмма поведения робота) будет выглядеть так:



Здесь применены повороты по гиродатчику. Но можно написать программу с поворотами по энкодеру, или комбинировать: например, правый – по энкодеру, левый – по гиродатчику.

Для точной настройки, можно встроить в ветки Диаграммы поведения робота остановки и задержки времени. Тогда вы увидите, где робот действует не точно и какую подпрограмму необходимо поправить:



Отладка робота в Trik Studio

The screenshot displays the Trik Studio interface. On the left is a toolbar with icons for 'Редактор' (Editor) and 'Отладка' (Debug). The main workspace shows a 'Двумерная модель' (2D model) of a maze with a robot at the top. A grid is visible with coordinates ranging from -157.5 to 52.5. On the right, there are two panels: 'Дисплей' (Display) showing a virtual screen with a 'TRIK STUDIO' logo and 'Порты' (Ports) section with three dropdown menus (A1: Датчик расстояния, A2: Датчик расстояния, A3: Датчик освещеннос). Below these is a 'Переменные' (Variables) table.

Имя	Значение
10 colorSensorG	0
11 colorSensorR	0
12 encoder1	0
13 encoder2	0
14 encoder3	0
15 encoder4	0
16 gyroscope	{ 0 }
17 lineSensor	{ 0 }
18 objectSensorSize	0
19 objectSensorX	0
20 objectSensorY	0
21 pi	3...
22 sensorA1	0
23 sensorA2	0

Подпрограммы настраиваются отдельно. Особое внимание следует обратить на точность движений, так как, от этого зависит насколько робот сместится от желаемой траектории в конце лабиринта.

После настройки приложений переходим к настройке всей программы. Случается, что при настройке приложения не так видно неточность, как при отладке всей программы. И тогда необходимо вернуться к настройке приложения и произвести корректировку.