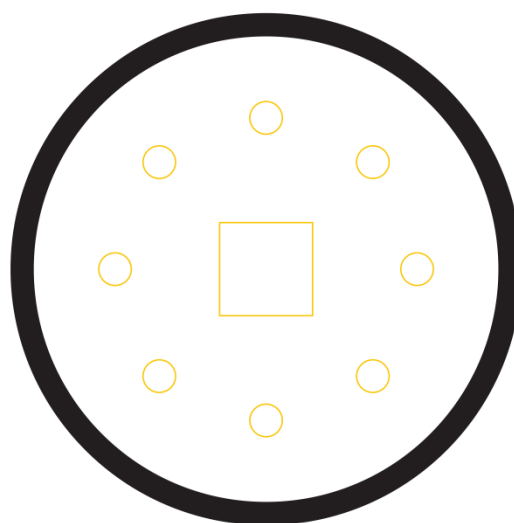


## Кегельринг

Кегельринг – соревнование, в котором за наиболее короткое время робот, не выходя более чем на 5 секунд за пределы круга, очерчивающего ринг, должен вытолкнуть расположенные в нем кегли.

Основные требования к состязанию:

1. На очистку ринга от кеглей дается максимум 2 минуты.
2. Если робот полностью выйдет за линию круга более чем на 5 секунд, попытка не засчитывается.
3. Во время проведения состязания участники команд не должны касаться роботов, кеглей или ринга.



Существует множество способов прохождения кегельринга.

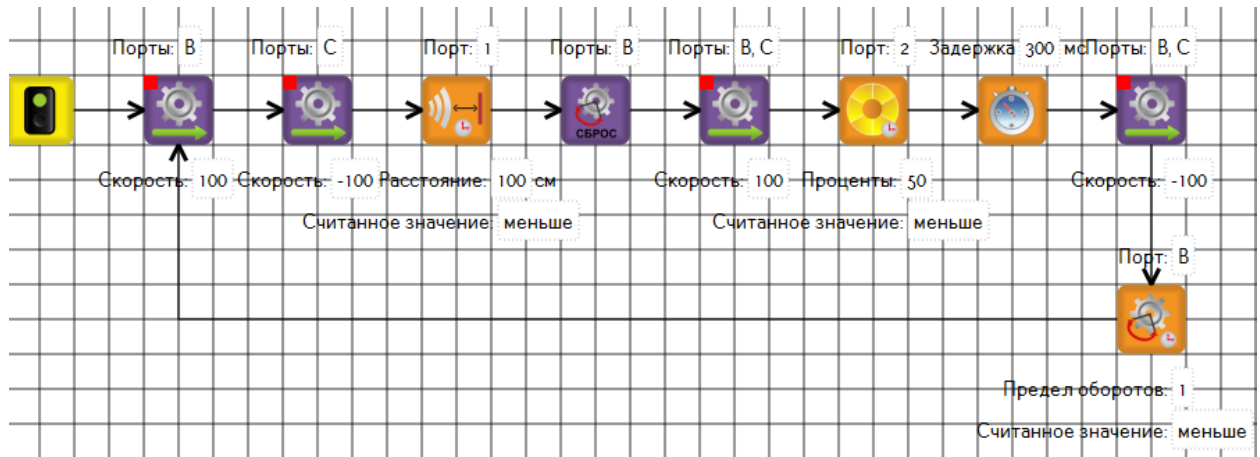
Абсолютный рекорд в нем составляет 2.5 секунды на выбивание всех 8 кегель. В данном пособии мы изучим два из способов: с поиском датчиком расстояния и расчетом по энкодерам.

Первое, о чем стоит упомянуть – это конструкция робота. Он должен входить в регламент (20x20 см) и на нем должны стоять все необходимые датчики, такие как: датчик расстояния, смотрящий вперед, и датчик света. Так же рекомендуется сделать бампер, который будет аккуратно выталкивать кегли и не давать им падать под колеса.

Второе – программа. В первом варианте она будет с поиском кегель датчиком расстояния.

Порядок действий при кегельринге с поиском кегли:

1. Крутимся на месте, пока не увидим кеглю;
2. Обнуляем энкодер;
3. Едем вперед, пока не увидим черную линию;
4. Проезжаем еще немного вперед;
5. Едем назад, пока на энкодере не будет 0.



Кегельринг с поиском кегель в TRIK Studio

```

1  #pragma config(Sensor, S1, , sensorEV3_Ultrasonic)
2  #pragma config(Sensor, S2, , sensorEV3_Color)
3  #pragma config(Motor, motorB, B, tmotorEV3_Large, openLoop, encoder)
4  #pragma config(Motor, motorC, C, tmotorEV3_Large, openLoop, encoder)
5  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!*/
6
7  task main()
8  {
9      while(true)
10     {
11         motor[B]=100;
12         motor[C]=-100;
13         while(SensorValue(S1)>50) sleep(1);
14
15         nMotorEncoder[B]=0;
16         motor[B]=motor[C]=100;
17         while(SensorValue(S2)>50) sleep(1);
18         sleep(300);
19
20         motor[B]=motor[C]=-100;
21         while(nMotorEncoder[B]>0) sleep(1);
22     }
23 }

```

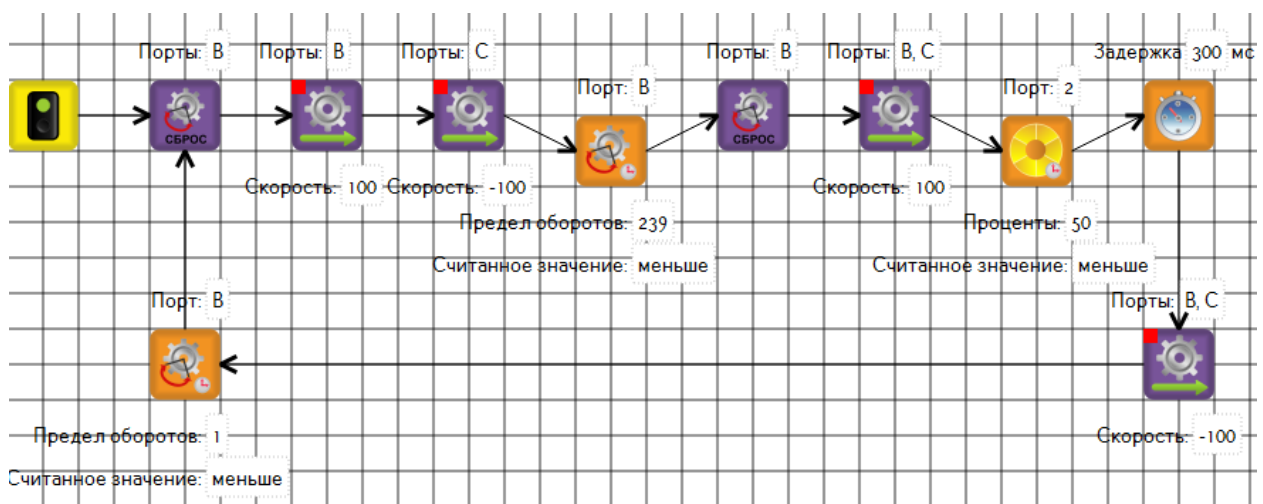
Кегельринг с поиском кегель в RobotC

В данном варианте очень часто возникает одна проблема: датчик расстояния видит землю и бьет в одну точку. Чтобы избежать этой ошибки применяется второй вариант программы. В нем по энкодерам мы будем не

только возвращаться назад, но и поворачиваться в центре круга.

Последовательность действий немного меняется:

1. Обнуляем энкодер;
2. Крутимся на месте, пока на энкодере не будет выбранного числа (у меня вышло 239);
3. Обнуляем энкодер;
4. Едем вперед, пока не увидим черную линию;
5. Проезжаем еще немного вперед;
6. Едем назад, пока на энкодере не будет 0.



Кегельринг с поворотом по энкодерам в TRIK Studio

```

1  #pragma config(Sensor, S4, , sensorEV3_Color)
2  #pragma config(Motor, motorB, B, tmotorEV3_Large, openLoop, encoder)
3  #pragma config(Motor, motorC, C, tmotorEV3_Large, openLoop, encoder)
4  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!**/
5
6  task main()
7  {
8      while(true)
9      {
10         nMotorEncoder[B]=0;
11         motor[B]=100;
12         motor[C]=-100;
13         while(nMotorEncoder[B]<239) sleep(1);
14
15         nMotorEncoder[B]=0;
16         motor[B]=motor[C]=100;
17         while(SensorValue(S2)>50) sleep(1);
18         sleep(300);
19
20         motor[B]=motor[C]=-100;
21         while(nMotorEncoder[B]>0) sleep(1);
22     }
23 }

```

Кегельринг с поворотом по энкодерам в RobotC

В этом случае мы избавляемся от «шумовых» показаний датчика. Но при этом нужно очень точно подобрать градус поворота, чтобы робот не промахивался.

При отладке кегельринга часто возникает несколько типовых ошибок:

1. Робот находит кеглю, начинает ехать вперед и тут же «отпрыгивает назад». Это следствие того, что у вас грязное поле, и он считает, что уже доехал до черной линии. Если это происходит, нужно понизить чувствительность (померяйте, сколько у вас показывает датчик света на черном и прибавьте к этому числу 10).
2. Робот вместо того, чтобы вращаться едет вперед. Проверьте, правильно ли у вас стоит скорость на моторах. Чаще всего в этом случае на обоих стоит 100.
3. Робот после выталкивания банки едет назад вечно (или вечно крутится во втором варианте программы). Это следствие того, что у вас неправильно настроен энкодер. Обратите внимание, что при движении мотора ВПЕРЕД у энкодера значения УВЕЛИЧИВАЮТСЯ, а при движении НАЗАД УМЕНЬШАЮТСЯ. Поэтому можно долго ждать, если мотор крутится назад, а написано жди, пока будет БОЛЬШЕ, чем 100.
4. Обратная проблема – робот только начал движение назад и тут же остановился. Причина та же – если написано, жди, пока на энкодере будет больше 300, а там уже 1000, то он сразу переходит на следующее действие.