МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

|  |  |  |
| --- | --- | --- |
|  |  |  |

Курс “Аппаратное и программное обеспечение роботизированных систем”

Отчет по лабораторной работе №1

Выполнил: Дормидонтов М.А.

Проверил: Гай В.Е.

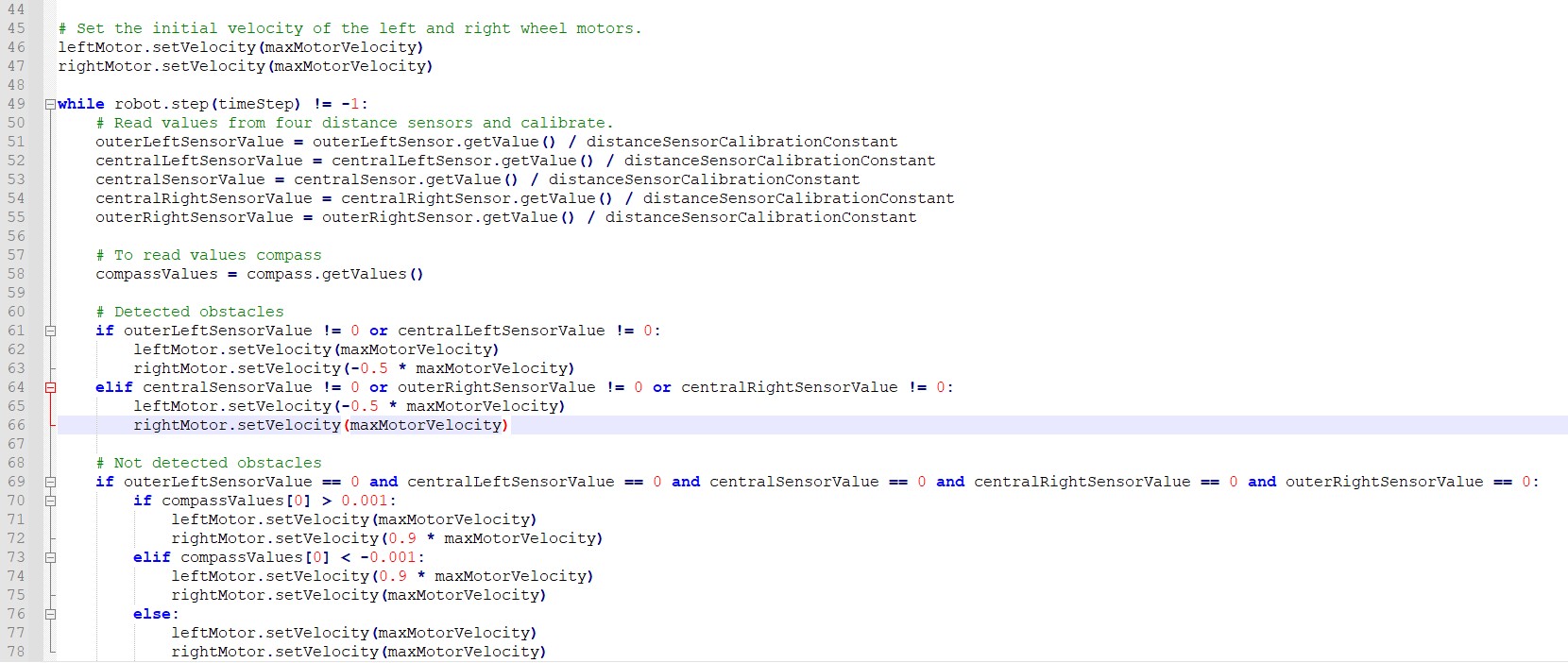
Нижний Новгород 2021

**Задача 1. Обход препятствий**

Этот тест направлен на создание надежного и эффективного алгоритма обхода препятствий для робота Thymio II с использованием языка программирования Python. Цель состоит в том, чтобы робот пересек комнату и достиг противоположной стены как можно быстрее, избегая при этом всех столкновений с препятствиями. Чтобы стимулировать устойчивое поведение избегания, препятствия располагаются случайным образом при каждом пробеге.

**Листинг:**





Для обхода препятствий мы изменяли скорость колес в зависимости от значений датчиков расстояний: если датчики считывают препятствие справа (и по центру) – левое колесо резко снижает скорость, если слева – правое колесо резко снижает скорость.

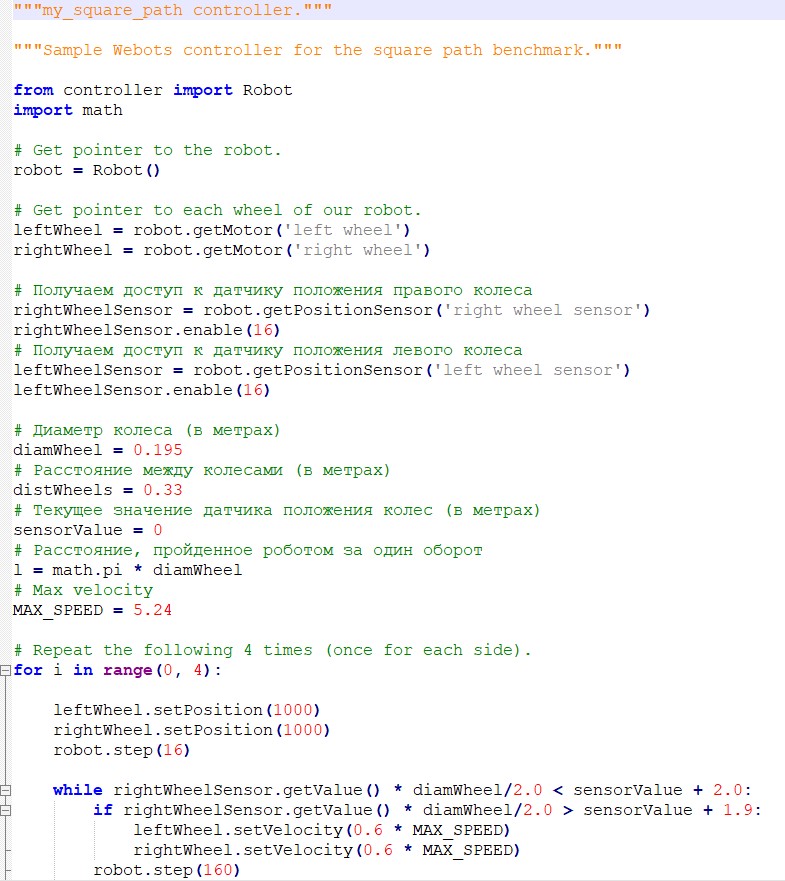
Робот может дезориентироваться, так как не знает точного направления своего пути. Поэтому в нем присутствует компасное устройство, которого нет по умолчанию. Чтобы он заработал, была подключена библиотека *Compass*. Инициализация устройства и считывание значений происходит схожим образом, как и у датчиков расстояний. Для корректировки пути был написан следующий алгоритм: если на пути нет препятствий и робот движется в левую сторону (т.е. первое значение вектора > 0.001) – постепенно снижаем скорость у правого колеса до стабилизации компаса, если движется направо (т.е. первое значение вектора < -0.001) – так же постепенно снижаем скорость у левого колеса.

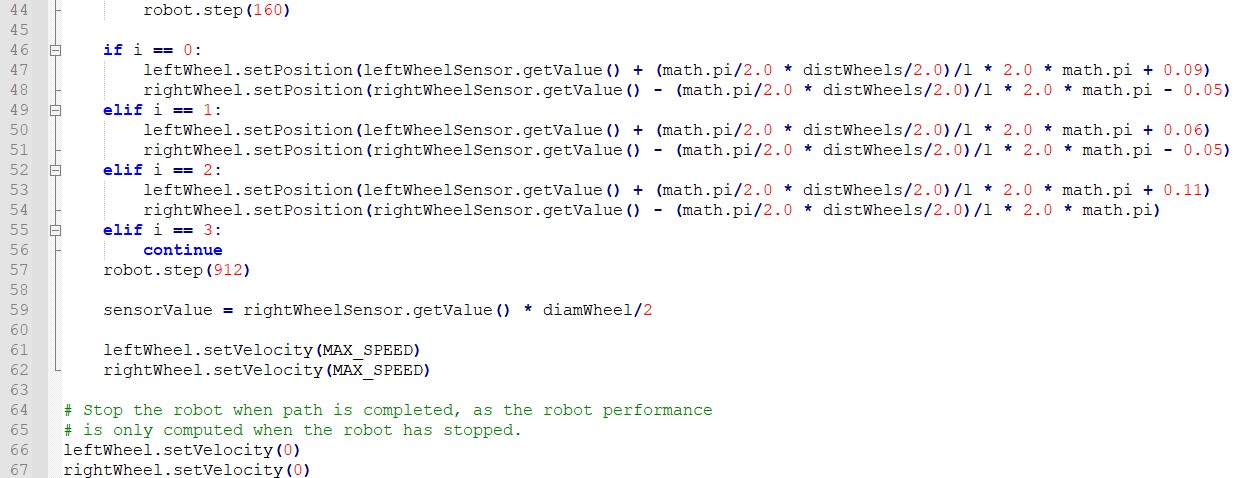
Общую скорость робота мы увеличили до полной, убрав в коде ограничения. Она регулируется только под воздействием датчиков расстояний и компасного устройства.

**Задача 2. Движение по квадрату**

Этот бенчмарк направлен на разработку программы с разомкнутым контуром, которая управляет роботом-пионером, чтобы он следовал квадратной траектории размером 2 на 2 метра.

**Листинг:**





Для прохождения нужного нам расстояния (2 на 2 метра) были подключены датчики положения, считывающие каждые 16 мс текущее положение колес робота (в радианах). В самом алгоритме выполняется цикл «пока робот не проедет 2 метра», где сравнивается значение текущего положения правого колеса, помноженное на его радиус, с положением правого колеса относительно предыдущей вершины + 2 метра (т.е. сравниваются метры, а не радианы). При достижении вершины, робот снижает свою скорость на 40%, после поворота возвращает.

Для того чтобы повернуть робота на 90 градусов, мы так же воспользовались датчиками положения. В зависимости от вершины, устанавливаются различные позиции колес: текущее положение колеса угол поворота погрешность. Угол поворота рассчитывается исходя из расстояния между колесами.

Из-за того, что значения датчиков возвращаются периодически, достигнуть идеального результата довольно трудно, возникают погрешности.