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 №2**

Выполнил: Папанов Р.В.

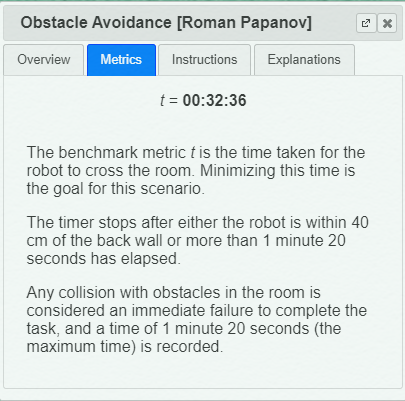
Проверил: Гай В.Е.

Нижний Новгород 2022

Задача №1. Обход препятствий:

Эта работа направлена на создание надежного и эффективного алгоритма обхода препятствий для робота Thymio II с использованием языка программирования Python.

**Результат:**

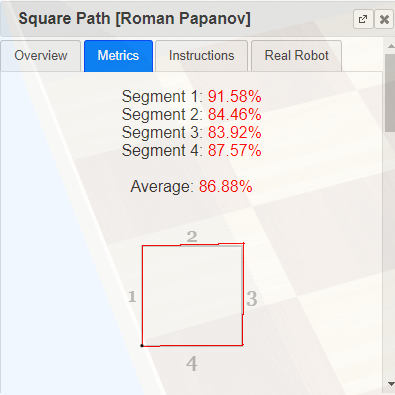
****

В этой работе был использован следующий алгоритм: чтобы заметить препятсвие, мы используем показание сенсоров, установленных на роботе. Сенсоры меняют свое значение относительно нуля, когда они видят, что перед ними препятсвие. Следовательно, если сенсоры, которые установлены на левой части робота меняют свое значение, то нам нужно повернуть направо, для этого мы замедляем правое колесо, тем самым позволяя роботу совершить этот маневр. Соответсвенно, похожим образом мы поступаем и если видим препятствие справа, но замедляем уже левое колесо, чтобы совершить поворот. Наш робот не знает, где финиш, поэтому выполняя множество поворотов, он может потеряться в пространстве. Для этого в коде программы мы используем компас, чтобы после каждого обхода препятсвий, робот выравнивался и ехал непосредственно к финишу. Скорость выранивания робота относительно финиша сделана специально небольшой, что вредит скорости выполнения программы, но дает нам гарантию того, что при повороте робот не врежется в препятсвие, которое только что объезжал. Именно благодаря этим моментам и был получен результат, представленный выше.

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.

**Результат:**

****

Алгоритм этой программы был представлен следующим образом: метод прохода первого сегмента был самый простой, так как все, что нам нужно было сделать – это проехать по прямой. Длину прямой мы узнали при помощи нескольких пробных экспериментов и использования сенсора на правом колесе, который нам и показал длину прямой. Это значения мы приняли за константу, чтобы работать с ним на оставшихся сегментах, так как у нас квадрат, а, как известно, его стороны равны. После каждого поворота происходила корректировка этой «константы», так как на повороте ее значение изменялось в меньшую сторону, ведь правое колесо немного отъезжало назад. Не стопроцентная точность была получена только лишь из-за поворотов, а именно из-за невозможности повернуть робота ровно на 90 градусов. Причиной этого может быть недостаток моих знаний или же то, что робот действительно не может повернуть на 90 градусов, так как ни один из тех, кто писал код для робота не достиг точности равной 100%. В связи с этим углы, на которые робот поворачивал, просто были подобраны так, что бы отклонение от ожидаемого маршрута было минимальным.