

**Список цитированных источников**

1. Wahli U., Leybovich L., Prevost E., Scher R., Venancio A., Wiederkom S., MacKinnon N. Business Process Management: Modeling through Monitoring Using WebSphere V6 Products, 2006
2. Service Component Architecture [Электронный ресурс] - Электронные данные. - Режим доступа: <http://www.ibm.com/developerworks/webservices/library/specification/ws-sca/>
3. WebSphere Process Server [Электронный ресурс] - Электронные данные. - Режим доступа: <http://www.ibm.com/software/integration/wps/>

УДК 004.896

**АЛГОРИТМ ДВИЖЕНИЯ АВТОНОМНОГО МОБИЛЬНОГО РОБОТА  
ДЛЯ УЧАСТИЯ В КОЛЬЦЕВЫХ ГОНКАХ**

**Хомиченко Д.В., Пучик А.А.**

*УО «Брестский государственный технический университет», г. Брест  
Научный руководитель – Дунец А.П., доцент*

В данной работе описан алгоритм ранее созданного автономного робота для участия в соревнованиях, таких как RoboRace. Алгоритм разрабатывался на основе технических характеристик робота. Алгоритм обеспечивает роботу максимально точное движение по линии и минимальную вероятность ее потери.

**Правила гонок**

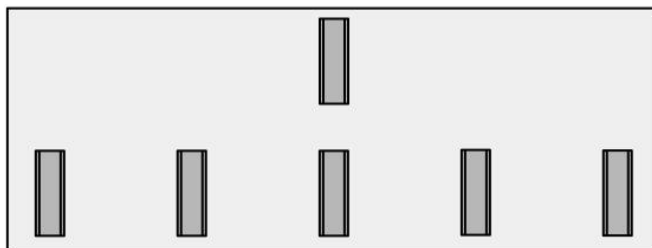
RoboRace - это соревнование автономных мобильных роботов, организованное по принципу Формулы 1. Движение роботов осуществляется по трассе с заранее нанесенными на неё черными линиями и установленными бортами. Допускается движение робота по любой из линий или любым иным способом, который подразумевает автономную ориентацию.

Алгоритм движения робота разрабатывается каждой командой для своего робота.

**Технические данные робота**

Характеристики робота: четырёхколёсная платформа, 6 инфракрасных датчиков, определяющих линию, 2 механических датчика, воспринимающих столкновения, микроконтроллер и драйвер двигателя, позволяющий управлять скоростью и направлением движения робота, сервопривод, отвечающий за поворот передних колес.

Инфракрасные датчики находятся на передней части робота и расположены в два ряда: в первом стоит один средний (далее будем называть его “передним центральным”), а во втором (датчики второго ряда, далее будем называть “левый”, “левый центральный”, “центральный”, “правый центральный” и “правый” соответственно) расположено пять датчиков так, чтобы третий был на одном уровне с передним центральным датчиком (Рис. 1).



**Рисунок 1 – Расположение инфракрасных датчиков**

### Логика движения робота

Движение робота осуществляется посредством поворота его передних колёс, угол поворота которых определяется положением черной линии относительно датчиков. Робот движется прямо на полной скорости, если линию обнаруживают оба центральных датчика. При пропадании линии с одного из датчиков - робот снижает скорость на 40% для подготовки к повороту (после нескольких экспериментов привело к выводу, что при меньшей скорости радиус поворота у робота меньше) и ожидает сигнал от «левого центрального» или «правого центрального» датчика, получив который, робот поворачивает влево или вправо соответственно. Если поворот очень резкий и робот не может отследить линию центральными датчиками, то линию фиксируют «левый» или «правый» датчики. Следует заметить то, что при появлении линии на одном «левом» или «правом» датчике робот ожидает сигнала на «левом центральном» или «правом центральном» соответственно. В случае потери линии и этими датчиками, то робот продолжает поворот в ту же сторону до появления сигнала на одном из центральных датчиков, после чего выравнивает движение робота по линии.

Так же был разработан метод отслеживания линии. Этот метод позволяет ехать роботу подобной конструкции вдоль одной линии, не теряя её. Метод основывается на том, что определенные датчики, получив сигнал о черной линии, влияют на поведение робота только при некоторых условиях. «Центральные боковые» датчики активируются в любых ситуациях, т.к. это означает, что линия отклонилась в ту или иную сторону от центра; или же робот пытается центрировать линию относительно себя. Крайние боковые датчики учитываются только после того как линия появится на соответствующем «боковом центральном» датчике - это позволяет нам при появлении другой линии ее не учитывать. Центральные датчики активируются после любого варианта событий. Благодаря этому робот может выбрать более плавную линию для следующего круга.

Алгоритм работы робота описан функциями, и тело основной исполняемой функции выглядит как приведено на рисунке 2.

```
void loop()    // Основной цикл управления
{
  if ( !ButtonRead() ) // Проверяем механические кнопки
  {
    Control();    // Сбор информации с датчиков и анализ ситуации
    Move();      // Движение робота
  }
}
```

**Рисунок 3 – Алгоритм работы основной исполняемой функции**

При срабатывании кнопок на препятствие робот поворачивает передние колеса в одну или другую сторону и отъезжает назад, немного едет прямо (это позволяет роботу иметь некоторую вероятность прохождения трассы в случае поломки датчиков). Скорость обработки запросов очень велика, поэтому нажатие сразу двух кнопок «словить» практически нереально, для этого после нажатия на одну кнопку было вставлено небольшое ожидание (0.1 сек) и снова делался запрос на нажатие кнопок. Если сработали сразу обе кнопки, то робот просто отъезжает назад и дальше пытается ориентироваться по линии. Вероятность того, что кнопки зациклят робота на одном промежутке трассы очень мала, и за все время работы робота таких ситуаций не было. Для считывания кнопок была разработана отдельная функция (рисунок 3).

```
boolean ButtonRead()
{
  if (digitalRead(Butt_L) == 0) // проверка нажатия левой кнопки
  { delay(100); // ожидание нажатия 2-й кнопки
    if (digitalRead(Butt_R) == 0) // проверка нажатия правой кнопки
      CenterBack(); // движение при нажатии 2-х кнопок
    else LeftBack(); // движение при нажатии левой кнопки
  }
  if (digitalRead(Butt_R) == 0)
  { delay(100);
    if (digitalRead(Butt_L) == 0)
      CenterBack();
    else RightBack(); // движение при нажатии левой кнопки
  }
  if (digitalRead(Butt_L) == 0 || digitalRead(Butt_R) == 0)
    return 1; // было срабатывание кнопок
  else return 0; // не было срабатывания кнопок
}
```

*Рисунок 3 - Алгоритм выполнения считывания кнопок*

### **Тестирование**

После прошивки микроконтроллера были проведены первые испытания алгоритма на реальном роботе. С первого раза не достигли лучшего результата, т.к. для всякого алгоритма существуют коэффициенты, которые должны быть откорректированы на опытах.

Тестирование робота на трассе было успешным. Разработанная логика оправдала себя. Робот уверенно ехал по черной линии и не терял её, столкновений практически не было. Робот на соревнованиях во Львове на 3D-трассе занял второе место.

### **Заключение**

В дальнейшем планируется:

- использовать подобную логику для гусеничной платформы,
- разработать логику для робота, воспринимающего информацию с камеры.

Сейчас идет активная работа над разработкой логики, основанной на показаниях дальномеров.