

интерфейс JTAG, совместимый со стандартом IEEE 1149.1, который также используется для доступа к встроенной системе отладки и для программирования. Загрузка кодов команд в память микроконтроллера осуществляется из персонального компьютера через внутрисхемный программатор JTAG.

Отладка схемы выполнялась с использованием осциллографического адаптера PCS64i, подсоединенного к персональному компьютеру стандартной конфигурации. На рисунке 4 изображен вид панели управления осциллографа.

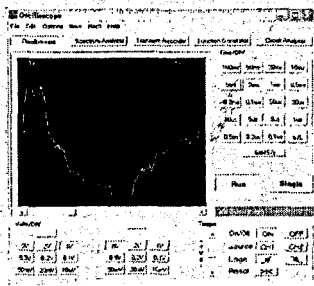


Рисунок 4 – Вид панели управления осциллографа

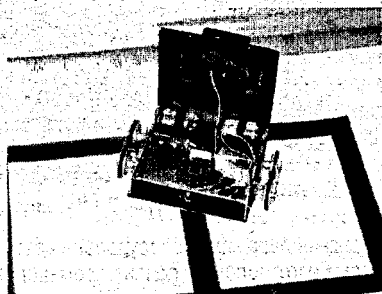


Рисунок 5 – Фотография робота в сборе

В конструкции робота также предусмотрена возможность дальнейшего функционального расширения возможностей робота путем установки ультразвуковых датчиков расстояния. Такая модификация позволит дополнить реализованную в настоящий момент функцию остановки при столкновениях и ограниченные возможности объезда препятствия бесконтактным методом их распознавания. Общий вид робота с открытым корпусом изображён на рисунке 5.

#### ЛИТЕРАТУРА

1. Юревич Е. Основы робототехники. – СПб.: ВНУ, 2005. – 416 с.
2. Вильямс Д. Программируемые роботы. – М.: ИТ Пресс, 2006. – 240 с.
3. Склипус Д.Б. Микроконтроллерная лаборатория // Информационные технологии управления в экономике 2006: материалы респ. науч.-практ. конф. – Брест, 2006. – С. 35.

УДК 004.896:621.865

*Склипус Д.Б.*

*Научный руководитель: к.т.н., доцент Костюк Д.А.*

#### АЛГОРИТМ ДВИЖЕНИЯ ПО ЧЁРНОЙ ЛИНИИ ДЛЯ АВТОНОМНОГО МОБИЛЬНОГО РОБОТА

В работе рассматриваются алгоритм управления и его программная реализация для автономного мобильного робота [1, 2]. Алгоритм обеспечивает движение робота по чёрной линии с корректной обработкой разветвлений и тупиковых отрезков. Применение такой системы в промышленных предприятиях позволит повысить интеллектуальность автоматизированной доставки грузов, увеличив тем самым эффективность.

Алгоритм обеспечивает движение робота по маршруту, заданному нанесенной на поверхность пола чёрной линией. Для устойчивого распознавания маршрута сформулированы следующие требования к чёрной линии.

- Линии должны пересекаться только под прямым углом.
- Повороты должны быть под прямым углом.

- Расстояние между перекрёстками не должно быть меньше длины робота.
- Длинные кривые должны быть отмечены метками на карте (Метка представляет собой чёрный прямоугольник, длина которого не менее 180 мм, а ширина 18-25 мм).

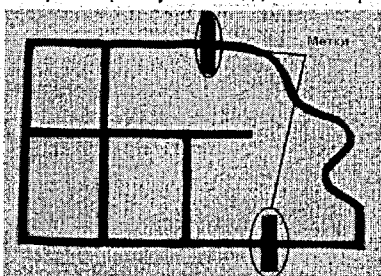


Рисунок 1 – Фотография линии – траектории движения робота

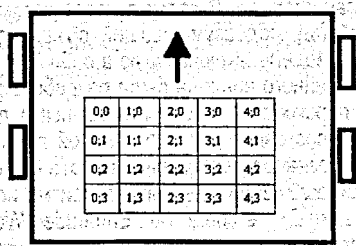


Рисунок 2 – Нумерация матрицы датчиков робота

Алгоритм основывается на последовательном преодолении перекрёстков, соответствующих описанным выше требованиям. Если робот движется вдоль левой границы чёрной линии, он повернёт налево на ближайшем перекрёстке, но если он двигался вдоль правой границы чёрной линии, то повернёт направо. Для исключения каких-либо поворотов и гарантированного прямолинейного движения робот движется без отклонений вправо и влево. Данный алгоритм может быть применён для уже известных карт чёрной линии. Иначе при попытке двигаться в сторону, где нет чёрной линии, например, вперёд на Т-образном перекрёстке, направление движения робота будет не определено.

Блок датчиков, считывающих изображение маршрута с находящейся под роботом поверхности [3, 4], представляет из себя матрицу размерностью 5x4 элементов. Нумерация датчиков изображена на рисунке 2. При движении робота вдоль левой границы чёрной линии используются два датчика с индексами (2,2) и (1,2). Таблица истинности, на основе которой обрабатываются поступающие от датчиков входные данные, представлена в табл. 1.

Таблица 1 – Таблица истинности для движения робота вдоль левой границы чёрной линии

Направление	Чёрная поверхность	Белая поверхность
Вперёд	(2,2)	(1,2)
Поворот налево 1	(1,2) и (2,2)	-----
Поворот налево 2	(1,2)	(2,2)
Поворот направо	-----	(1,2) и (2,2)

При движении робота вдоль правой границы чёрной линии, используется пара датчиков (2,2) и (3,2) в соответствии с табл. 2.

Таблица 2 – Таблица истинности для движения робота вдоль правой границы чёрной линии

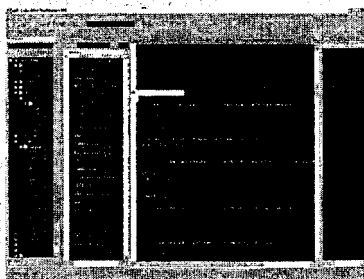
Направление	Чёрная поверхность	Белая поверхность
Вперёд	(2,2)	(3,2)
Поворот направо 1	(3,2) и (2,2)	-----
Поворот направо 2	(3,2)	(2,2)
Поворот налево	-----	(3,2) и (2,2)

Прямолинейное движение осуществляется за счёт работы центральных датчиков чёрной линии, имеющих индексы (2,i), где  $i = \overline{1-4}$ . Если они сигнализируют о наличии чёрной линии под роботом, выполняется перемещение прямо. Иначе показания датчиков правой и левой половины матрицы сравниваются, и направление движения выбирается в

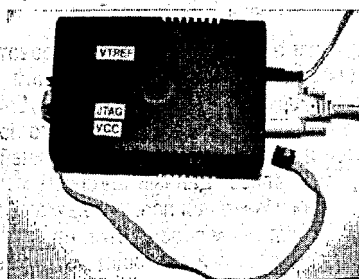
зависимости от результатов сравнения: если на правой половине больше число датчиков сигнализирует о наличии чёрной линии, то робот повернёт направо, иначе – налево.

Определение перекрестков возможно одним из двух методов. Первый подразумевает сравнение со всеми возможными типами перекрестков, число которых в нашем случае равняется  $2^{20}$ . В связи с недопустимой трудоёмкостью данного процесса был выбран метод, подразумевающий объединение возможных типов перекрестков в обобщённую модель, включающую в себя Х- и Т-образные перекрестки, а также повороты. В рамках данного подхода была разработана функция для определения положения робота на перекрёстке. Входными данными для неё являются двумерный массив, описывающий состояние датчиков чёрной линии. Возвращаемое значение имеет логический тип: 1 обозначает нахождение робота на перекрёстке, а 0 - на участке чёрной линии.

Для реализации данного алгоритма использовался высокоуровневый язык программирования C++ в среде JAR Embedded Workbench 4.1. На рисунке 2 (а) изображен процесс отладки разработанного программного обеспечения в среде программирования, а также JTAG контроллер (б), использованный для удаленной отладки.



а)



б)

Рисунок 2 – а) Среда программирования в режиме отладки; б) JTAG контроллер

Алгоритм был протестирован на действующем макете робота. Эксперименты продемонстрировали устойчивое движение по заданному маршруту при отсутствии сбоев в аппаратном обеспечении робота. Исследование зафиксированных ошибок в распознавании маршрута выявило следующие аппаратные причины:

1. Отсутствие питания на блоке электроники;
2. Обрыв проводов;
3. Контакты датчиков.

Разработанное программное обеспечение требует настройки только одного коэффициента, характеризующего скорость робота. Данный коэффициент настраивается экспериментальным способом и не изменяется на протяжении всей эксплуатации робота.

Для проверки устойчивости алгоритма к механическим повреждениям, было искусственно ослаблено крепление левого ведущего колеса робота. Данная неисправность может возникать в условиях реальной эксплуатации робота с превышением максимально-допустимой нагрузки на заднюю ось. Ее последствиями являются, увеличение времени поворота робота и движение с постоянным уклоном в сторону ослабленного крепления. Эксперименты продемонстрировали устойчивое движение робота по чёрной линии при наличии данной неисправности, показав эффективность разработанного алгоритма также при разной скорости вращения ведущих колёс.

Оценка эффективности алгоритма проводилась по двум параметрам: безопасности и скорости.

На рисунке 3 представлено условное изображение траектории движения робота, задающей маршрут чёрной линии, а также площади, образованной пересечениями этих

кривых. Безопасность оценивается по формуле  $B=S/l$ , где  $l$  – длина отрезка, а  $S$  – площадь. Чем меньше значение  $B$ , тем более безопасным является движение робота с точки зрения возможных столкновений и необратимой потери маршрута. Безопасность движения может быть также охарактеризована величиной граничного расстояния – максимального расстояния, на которое робот может удалиться от черной линии в рамках выполнения поставленной задачи. В большинстве случаев это расстояние не превышает ширины корпуса робота. Для определения площади  $S$  применен теоретический подход на основе использования интегрального исчисления.

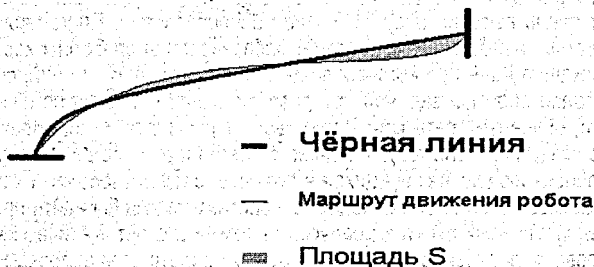


Рисунок 3 – Схематическое изображение маршрута робота, черной линии и площади образованной этими кривыми

Критерием оценки эффективности робота также является отношение скорости его движения с учетом задержек на перекрестках к скорости движения по прямой черной линии. В проведенных экспериментах это отношение составило  $4 \text{ см/с} / 5.5 \text{ см/с}$  или  $0.28$ .

Дальнейшее развитие алгоритма направлено на более эффективную обработку данных, поступающих от оптических датчиков, для получения более качественного изображения линии.

#### ЛИТЕРАТУРА

1. Склипус Д.Б. Робот SRS. [http://www.prorobot.ru/nauka/Sklipus\\_Robot\\_System.doc](http://www.prorobot.ru/nauka/Sklipus_Robot_System.doc) 23/12/2008.
2. Юревич Е. Основы робототехники. – СПб.: BHV, 2005. – 416 с.
3. Вильямс Д. Программируемые роботы. – М.: ИТ Пресс, 2006. – 240 с.
4. Склипус Д.Б. Микроконтроллерная лаборатория: сб-к тезисов по секции «Информатика» Междунар. конф. «VI Колмогоровские чтения». – Москва, 2006. – С. 26.

УДК 004.8.032.26

Войцехович Г. Ю.

Научный руководитель: д.т.н., профессор Головкин В.А.

#### НЕЙРОСЕТЕВЫЕ МЕТОДЫ РАСПОЗНАВАНИЯ ЧЕЛОВЕКА ПО ИЗОБРАЖЕНИЮ ЛИЦА

**Введение.** Каждый человек обладает рядом уникальных биометрических характеристик, таких как изображение сетчатки глаза, узор радужной оболочки, отпечатки пальцев, голос, лицо и прочие. Часть из этих характеристик успешно используется для идентификации человека в криминалистике или при решении задач контроля доступа [1]. Методы, основанные на применении биометрических параметров, имеют множество преимуществ по сравнению с традиционными, такими как ключ, пароль и др., которые можно подделать, забыть или потерять. Однако имеется и ряд недостатков. Один из них заключается в том, что сравнение полученных биометрических параметров с имеющимися в базе данных является весьма сложной и трудоемкой задачей. Вторым недостатком – сложность собственно получения характеристик. До сих пор задача идентификации че-