**Оглавление**

Введение……………………………………………………………………………………………..3

1. Шаговый двигатель, его устройство и принцип работы……………………………………………………………………………………..…4
	1. Устройство шагового двигателя…………………………………………..4
	2. Принцип работы…………………………………….……….………………….....7
	3. Способы управления шаговым двигателем.....................................9
	4. Основные параметры и характеристики шагового двигателя……………………………………………………………………………..10
2. Транспортные роботы……………………………………………………………….12
3. LPT-порт. Его функциональное назначение
	1. Описание LPT-порта. Выражения. Типы данных. Объекты.…15
4. Delphi. Её возможности и применение……………………..………………18
5. Программа управления Шаговым Двигателем………………………....20
6. Тестирование программы…………………………………………………………21

Заключение……………………………………………………………………………….24

Список литературы………………………………………………………………………….....25

Приложение…………………………………………………………………………………..……26

**Введение**

**Шаговые электродвигатели** очень удобны для применения в приводах роботов, поскольку не требуют датчиков обратной связи для определения положений звеньев. Такие двигатели позволяют с высокой точностью преобразовывать цифровые электрические сигналы непосредственно в дискретные угловые перемещения (шаги) ротора.

По принципу действия шаговый двигатель относится к двигателям синхронного типа: в нем существует связь между сигналом питания и положением ротора. Благодаря периодическим переключениям обмоток статора, его магнитодвижущее поле, поворачиваясь на определенный угол (шаг), производит поворот на такой же определенный угол ротора двигателя, представляющего собой постоянный магнит либо переменное магнитное сопротивление.

В данной выпускной работе была поставлена задача изучить способ управления шаговым двигателем через LPT-порт. Была разработана и написана программа для макета транспортного робота . В лаборатории имеется модель робота. Данный макет робота представляет собой два шаговых двигателя, закрепленных на тележке. Программа написана с помощью языка программирования Delphi.

Рис.1 Макет Транспортного робота

1. **Шаговый двигатель, его устройство и принцип работы**

1.1Устройство шагового двигателя

**Шаговый электродвигатель** — это синхронный бесщёточный [электродвигатель](http://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%B4%D0%B2%D0%B8%D0%B3%D0%B0%D1%82%D0%B5%D0%BB%D1%8C) с несколькими обмотками, в котором ток, подаваемый в одну из обмоток [статора](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%BE%D1%80), вызывает фиксацию ротора. Последовательная активация обмоток двигателя вызывает дискретные угловые перемещения (шаги) [ротора](http://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D1%82%D0%BE%D1%80).



Рис.2 Шаговый двигатель

**Статор**— неподвижная часть [электрической машины](http://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0), взаимодействующая с подвижной частью — [ротором](http://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D1%82%D0%BE%D1%80_%28%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0%29). Статор может быть либо постоянным магнитом, либо электромагнитом



Рис.3 Статор

**Ротор**— вращающаяся часть [электрической машины](http://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0). Ротор асинхронной электромашины обычно представляет собой собранное из листовой электротехнической стали цилиндрическое тело с пазами для размещения обмотки. Ротор в электромашинах постоянного тока называется [якорем](http://ru.wikipedia.org/w/index.php?title=%D0%AF%D0%BA%D0%BE%D1%80%D1%8C_%28%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0%29&action=edit&redlink=1).

Конструктивно шаговые электродвигатели состоят из [статора](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%BE%D1%80), на котором расположены обмотки возбуждения, и ротора, выполненного из магнито-мягкого (ферромагнитного) материала или из магнито-твёрдого (магнитного) материала. Шаговые двигатели с магнитным ротором позволяют получать бо́льший крутящий момент и обеспечивают фиксацию ротора при обесточенных обмотках.

Главное преимущество шаговых приводов - точность. При подаче потенциалов на обмотки шаговый двигатель повернется строго на определенный угол. Шаговый привод, как недорогая альтернатива [сервоприводу](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%BE%D0%BF%D1%80%D0%B8%D0%B2%D0%BE%D0%B4), наилучшим образом подходит для автоматизации отдельных узлов и систем, где не требуется высокая динамика.

Чтобы управлять шаговым двигателем необходим контроллер.

 Контроллер – схема, которая подает напряжение к любой из четырех катушек статора. Устройство может быть построено с использованием интегральной микросхемы  типа ULN2003 (отечественный аналог К1109КТ22) состоящая из набора  мощных составных ключей с защитными диодами на выходе. Наличие защитных диодов позволяет подключать индуктивные нагрузки без  дополнительной защиты от выбросов обратного напряжения.



Рис.3 Принцип работы

Контроллер шагового двигателя предназначен для создания автономного управления системой различного рода технологическими процессами, такими как: упаковочные роботы, дозаторов сыпучих продуктов и жидкостей, маркировщиков, в качестве устройства управления открытием-закрытием окон, дверей, для конструкций вращения, управления за камерами слежения за объектами и т.д.

 Рис.4 Контроллер

Контроллер шагового двигателя имеет следующие технические характеристики: ток 0,3-7А, питание 24-80В, защита от КЗ (для ADR401С), 4 оптически развязанных входа управления.

**1.2.Принцип работы**

Шаговые двигатели отличаются от регулируемых двигателей постоянного тока. Вместо того чтобы вращаться как двигатели постоянного тока, шаговый  двигатель совершает дискретное вращение под воздействием серии импульсов. В нашем примере двигателю необходимо 200 шагов  чтобы совершить полный оборот в 360 градусов.

Рис. 5 Устройство шагового двигателя

Другое преимущество шаговых двигателей – то, что их скорость вращения может быть достигнута почти мгновенно при изменении направления вращения на противоположное.

Рис.6 Принцип работы Шагового Двигателя

Двухполюсный ротор из магнитомягкой стали с клювообразными выступами помещен в четырехполюсный статор. Одна пара полюсов выполнена из постоянных магнитов, на другой – находится обмотка управления. Пока тока в обмотках управления нет, ротор ориентируется вдоль постоянных магнитов и удерживается около них с определенным усилием, которое определяется магнитным потоком полюсов. При подаче постоянного напряжения на обмотку управления возникает магнитный поток примерно вдвое больший, чем поток постоянных магнитов.

Под действием электромагнитного усилия, создаваемого этим потоком, ротор поворачивается, преодолевая нагрузочный момент и момент, развиваемый постоянными магнитами, стремясь занять положение соосное с полюсами управляющей обмотки.

Для фиксации ротора при обесточенной обмотке управления не требуется потребление энергии, угол поворота сохраняет свое значение и при перерывах в питании. Двигатели этого типа отрабатывают импульсы с частотой до 200-300 Гц.

Особенностью устройства управления шаговым двигателем является цифровой способ задания скорости вращения, которая позволяет устанавливать частоту вращения мотора с дискретностью в 1 шаг на всем диапазоне.

**1.3.Способы управления**

Есть несколько способов, которые можно использовать, чтобы управлять шаговым двигателем.**Одиночные импульсы** – самый простой способ. Одновременно подключается только одна катушка.  Необходимо  200 шагов чтобы ротор совершил один полный оборот. Каждый шаг перемещает ротор на 7,5 градусов.

Рис. 7 Одиночный импульс

**Двойной импульс** – одновременное подключение двух соседних катушек. В этом случае также необходимо 200 шагов чтобы ротор совершил один полный оборот. Каждый шаг перемещает ротор на 7,5 градусов.

Рис.8 Двойной импульс

**Комбинированные импульсы** – чередование первого и второго способа. Двигатель нуждается в 400 шагах, чтобы совершить один оборот. Каждый шаг перемещает ротор приблизительно на 3,75 градуса.

Рис. 7 **Комбинированные импульсы**

## 1.4.Основные параметры и характеристики шаговых двигателей

Специфика конструкции шаговых двигателей и многообразие режимов их работы вызывают необходимость оценивать эти двигатели по следующим параметрам: частоте собственных круговых колебаний; электромагнитным постоянным времени; коэффициенту внутреннего демпфирования и характеристикам – предельным механическим и предельным динамическим.

Частота собственных круговых колебаний – это угловая частота колебаний ротора около устойчивого положения при отсутствии момента нагрузки. Она является обобщенным параметром, зависящим от момента инерции **J**, амплитуды максимального синхронизирующего момента **Mmax**, числа пар полюсов **p**. Период собственных круговых колебаний, равный **1/w0**, может служить внутренним эталоном времени. Действительно, момент инерции **J** определяет инерционность двигателя и механизма, амплитуда максимального синхронизирующего момента **Mmax** дает характеристику шагового двигателя как преобразователю энергии, число пар полюсов **p** определяет степень электромеханической редукции угла поворота и скорости вращения. Отношение **Mmax/J** дает теоретически предельное ускорение ротора шагового двигателя.

Электромагнитная постоянная времени обмоток управления **Tэм = L/R** характеризует скорость протекания электромагнитных переходных процессов. Часто для уменьшения **Тэм** последовательно с обмоткой управления включают добавочное сопротивление. Уменьшать постоянную времени необходимо потому, что чем она больше, тем до меньшего значения нарастает ток за время импульса напряжения, меньше становится синхронизирующий момент, а, следовательно, и допустимый момент сопротивления.

 Коэффициент внутреннего демпфирования определяется отношением амплитуды потокосцепления ротора с фазой обмотки статора к ее активному сопротивлению. Этот параметр относится только к шаговым двигателям с активным ротором, поскольку его физический смысл заключается в образовании электромагнитного тормозного момента, вызванного взаимодействием поля ротора с током статора, наведенным этим полем и замыкающимся по цепи статор-источник тока. При этом механическая энергия колеблющегося ротора превращается в электрическую энергию с последующим рассеиванием ее в теплоту в активных сопротивлениях обмоток статора.

Предельная механическая характеристика – это зависимость допустимого момента сопротивления от частоты управляющих импульсов в установившемся режиме работы. С ростом частоты сказывается запаздывание в нарастании тока, обусловленное индуктивностью обмоток.

Предельная динамическая характеристика – зависимость частоты приемистости от момента сопротивления и момента инерции нагрузки.

## 2.Транспортные роботы

Транспортные роботы предназначены для автоматизированного транспортирования объектов, а также для управления различными транспортными системами. Исследования и разработки по созданию транспортных роботов интенсивно ведутся во всем мире. При этом выделяются четыре принципиально различных типа - наземные, воздухоплавающие, водоплавающие и подземные. Теория и практика трех последних типов не достигли еще в целом того уровня, чтобы говорить о них сегодня как о всеобщей реальности. Практическое развитие получили ныне наземные транспортные роботы, которые могут быть колесными, шагающими и гусеничными.

 Так же по способу перемещения грузов они разделяются на транспортирующие груз захватом манипулятора, буксирующие транспортные средства с грузом и перемещающие груз без захвата.

По функциональным возможностям — выполняющие только одну операцию, связанную с перемещением груза, и выполняющие несколько операций (перемещение, загрузка, выгрузка и т.п.).

По типу ходового устройства — пневмоколесные, на рельсовом ходу, гусеничные и шагающие.

 По способу управления — с ручной системой, автономные, работающие по внешним сигналам, и с комбинированной системой. В первом случае параметры траектории движения груза выбирают операторы, система робота реализует и поддерживает их значения. Во втором случае указание параметров движения транспортного робота осуществляется полностью его автономной системой автоматического управления. В третьем — все параметры движения (скорость, путь, остановка и др.) формируется под воздействием внешних сигналов, передаваемых бесконтактным способом.

Система управления транспортных роботов, построенная на комбинированных способах задания программы, обеспечивает расшифровку этих сигналов и выработку команд на исполнительные механизмы.

Транспортные роботы по приспособляемости к воздействию внешних условий делятся на роботы с жесткой программой и адаптивные. Большинство применяемых транспортных роботов относится к жесткопрограммируемым, т.е. программа их действия содержит полный набор информации, не изменяющийся в процессе работы и не корректирующийся при изменении внешней среды. Адаптивные транспортные роботы имеют сенсорное обеспечение, позволяющее корректировать управляющую программу в соответствии с информацией, полученной в процессе работы. Система этих роботов позволяет обойти препятствие, возникшее на пути движения, что очень важно как с точки зрения техники безопасности, так и сохранности самого транспортного робота.

Наибольшее развитие и распространение в настоящее время получили колесные транспортные роботы, используемые достаточно широко в промышленных автоматизированных транспортно-складских системах и гибких автоматизированных производствах в виде мобильных автоматических кранов, автоматических управляемых тележек (АУТ), робокаров и др., оснащаемых во многих случаях различными манипуляционными устройствами. В самом простом виде такие роботы следуют по рельсам либо по маршруту над кабелем, проложенным под поверхностью пола. Генератор частоты, подавая ток по кабелю, создает магнитное поле, улавливаемое двумя датчиками приемного устройства тележки, направляющими ее по требуемому маршруту. Даже такие 58 простые системы АУТ позволяет включать маршруты с несколькими ветвями и петлями посредством использования различных частот для каждого пути. В более сложном варианте тележка оборудуется автономной управляющей ЭВМ и средствами очувствления.

Транспортные роботы различаются скоростью и высотой перемещения захвата, способом выполнения операции перемещения, типов устройства управления, составом дополнительных устройств, механизмов, путевых устройств, а также массой и типов приводов перемещения роботов и их захватов.

Различие в технических характеристиках транспортных роботов обусловливает различие в их экономических показателях и, следовательно, в экономической эффективности их применения.

**2.LPT-порт**

2.1 Описание LPT порта

IEEE 1284 (порт принтера, параллельный порт, [англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Line Print Terminal, LPT) — международный стандарт [параллельного интерфейса](http://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29) для подключения периферийных устройств персонального компьютера.

В основном используется для подключения к компьютеру [принтера](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80), [сканера](http://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%B0%D0%BD%D0%B5%D1%80) и других внешних устройств (часто использовался для подключения внешних устройств хранения данных), однако может применяться и для других целей (организация связи между двумя компьютерами, подключение каких-либо механизмов телесигнализации и [телеуправления](http://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BB%D0%B5%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5)).В основе данного стандарта лежит интерфейс Centronics и его расширенные версии (ECP, EPP).

Название «LPT» образовано от наименования стандартного устройства принтера «LPT1» (Line Printer Terminal или Line PrinTer) в операционных системах семейства MS-DOS. Внешний вид устройства представлен на рис.8.



Рис.8 Внешний вид LPT-порта.

25-контактный разъём DB-25, используемый как LPT-порт на персональных компьютерах (IEEE 1284-A)

Порт на стороне управляющего устройства (компьютера) имеет 25-контактный 2-рядный разъём DB-25-female («мама») (IEEE 1284-A).

Не путать с аналогичным male-разъёмом («папа»), который устанавливался на старых компьютерах и представляет собой 25-пиновый [COM-порт](http://ru.wikipedia.org/wiki/COM-%D0%BF%D0%BE%D1%80%D1%82). На периферийных устройствах обычно используется 36-контактный [микроразъемHYPERLINK "http://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D1%80%D0%B0%D0%B7%D1%8A%D0%B5%D0%BC\_%D0%BB%D0%B5%D0%BD%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D0%B3%D0%BE\_%D1%82%D0%B8%D0%BF%D0%B0" ленточного типа](http://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D1%80%D0%B0%D0%B7%D1%8A%D0%B5%D0%BC_%D0%BB%D0%B5%D0%BD%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D0%B3%D0%BE_%D1%82%D0%B8%D0%BF%D0%B0) Centronics (IEEE 1284-B), поэтому [кабели](http://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B1%D0%B5%D0%BB%D1%8C) для подключения периферийных устройств к компьютеру по параллельному порту обычно выполняются с 25-контактным разъёмом DB-25-male на одной стороне и 36-контактным IEEE 1284-B на другой (AB-кабель). Существуют также CC-кабели с разъёмами MiniCentronics на обоих концах, предназначенные для подключения приборов в стандарте IEEE 1284-II, который применяется редко.

Длина соединительного кабеля не должна превышать 3 метров. Конструкция кабеля: витые пары в общем экране, либо витые пары в индивидуальных экранах. Изредка используются ленточные кабели.

Для подключения сканера, и некоторых других устройств используется кабель, у которого вместо разъема (IEEE 1284-B) установлен разъем DB-25-male. Обычно сканер оснащается вторым интерфейсом с разъемом DB-25-female (IEEE 1284-A) для подключения принтера (поскольку обычно компьютер оснащается только одним интерфейсом IEEE 1284). Схемотехника сканера построена таким образом, чтобы при работе с принтером сканер прозрачно передавал данные с одного интерфейса на другой. Базовый интерфейс Centronics является однонаправленным параллельным [интерфейсом](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), содержит характерные для такого интерфейса сигнальные линии (8 для передачи данных, [строб](http://ru.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D1%80%D0%BE%D0%B1&action=edit&redlink=1), линии состояния устройства).

Рис.9 Интерфейс LPT-порта.

На рисунке 10 показано возможное подключение шагового двигателя к LPT-порту.



Рис.10 Схема контроллера шагового двигателя

Это довольно простая схема контроллера шагового двигателя, которая позволит точно управлять однополярным шаговым двигателем через параллельный порт компьютера.

**3.Язык программирования Delphi.**

3.1 Delphi.Его возможности и применение

Delphi — [императивный](http://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [структурированный](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [объектно-ориентированный](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), диалект [Object Pascal](http://ru.wikipedia.org/wiki/Object_Pascal%22%20%5Co%20%22Object%20Pascal). Начиная со среды разработки Delphi 7.0, в официальных документах [Borland](http://ru.wikipedia.org/wiki/Borland%22%20%5Co%20%22Borland) стала использовать название Delphi для обозначения языка [Object Pascal](http://ru.wikipedia.org/wiki/Object_Pascal%22%20%5Co%20%22Object%20Pascal). Начиная с 2007 года уже язык Delphi (производныйот [Object Pascal](http://ru.wikipedia.org/wiki/Object_Pascal)) начал жить своей самостоятельной жизнью и претерпевал различные изменения связанные с современными тенденциями (например, с развитием [платформы .NET](http://ru.wikipedia.org/wiki/.NET_Framework)) развития языков программирования: появились [class helpers](http://ru.wikipedia.org/w/index.php?title=Class_helper&action=edit&redlink=1" \o "Class helper (страница отсутствует)), перегрузки операторов и другое.

[**Выражения**](http://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) в Object Pascal(Delphi) бывают арифметическими, логическими и строковыми.

Основные логические выражения:
Логические выражения в основном употребляются в условных операторах. К логическим выражениям относятся такие операции как:
not — «НЕ», отрицание логического выражения (NOT <логическое\_выражение>)
and — «И», умножение (пересечение) логических выражений (<логическое\_выражение1> AND <логическое\_выражение2>)
or — «ИЛИ», объединение логических выражений (<логическое\_выражение1> OR <логическое\_выражение2>)
xor — «Исключающее ИЛИ», отрицание ИЛИ, все что не попадает под ИЛИ (<логическое\_выражение1> XOR <логическое\_выражение2>)

[**Типы данных**](http://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

Обрабатываемые в программе данные подразделяются на переменные, константы и литералы. Константы представляют собой данные, значения которых установлены в разделе объявления констант и не изменяются в процессе выполнения программы.

**Объекты**

[Объекты](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29) как экземпляры класса объявляются в программе в разделе var как обычные переменные. Например:

var

 CCircle1: TColorCircle;

 CircleA: TCircle;

Как и в случае записей, для обращения к конкретному элементу объекта (полю, свойству или методу) указывается имя объекта и имя элемента, разделенные точкой, то есть имя элемента является составным.

**4.Программа управления шаговым двигателем**

Данная программа должна состоять из 3 подпрограмм.

Траектория движения транспортного робота состоит из элементарных участков движения вперед – назад и поворот на заданный угол. Был поставлена задача разработать эти подпрограммы для каждого отдельного элемента этой траектории.

Один оборот равен 200 шагов.

**2πR/200**

* Зависимость длинны шага от радиуса колеса,

Где **R**- радиус колеса, равный 44 мм.

Из геометрических соображений угол поворота вычисляется по формуле

**α=arctg ∆x/k,**

**k=1\2 K,**

где **K** - длинна оси от одного колеса до другого, равная 150 мм,

**∆x** – путь пройдённый колесом за один шаг.

Из теоретической механики можно вывести формулу скорости

**Ѵ=ωR,**

а угловую скорость можно расписать как

**ω=2πα/T,**

и тогда, подставляя вторую формулу в первую, получим

**Ѵ=2πRα/T.**

**5.Тестирование программы**

По техническим причинам программа представлена в исполнительном виде с помощью светодиодной платы. Зажигающиеся светодиоды имитируют активизацию определенной фазы шагового двигателя.

Интерфейс программы выглядит таким образом



Рис. 11 Интерфейс программы

При нажатии на одну из кнопок импульсы, подаваемые персональным компьютером, преобразовываясь в сигналы, позволяют транспортному роботу совершать движения в зависимости от направления стрелок.

Так же мы можем задавать количество шагов, и период вращения шаговых двигателей.

После того, как двигатели совершат требуемое количество шагов, программа закончит свою работу.

Программа вычисляет пройденное расстояние в зависимости от радиуса колеса и количества шагов.

Ниже приведена одна из нескольких процедур программы.

procedure TForm1.Timer1Timer(Sender: TObject);

var

speed: integer;

speed1:real;

PortValue: byte;

begin

K:=strTofloat(edit3.text); {количество шагов}

if i<round(k) then

begin

trystrtoint(edit1.Text, speed);

Timer1.Interval:=round((speed/200)\*1000);

PortValue:= motor1[step1] or motor2[step2];

Lpt.WritePort (GetCurrentPort,0,PortValue);

 step1:=step1+direction1;

 step2:=step2+direction2;

 if step1=-1 then step1:=3;

 if step1=4 then step1:=0;

 if step2=4 then step2:=0;

 if step2=-1 then step2:=3;

 i:=i+1;

 end;

R:=strTofloat(edit2.text); {радиус колеса}

S:=(2\*3.14\*R\*K)/200; {расстояние}

label2.caption:=FloatToStr(s);

end;

procedure TForm1.SpeedButton1Click(Sender: TObject);

begin

i:=0;

direction1:=1;

direction2:=1;

timer1.Enabled:=true;

end;

**6.Заключение**

В ходе выполнения выпускной работы было проделано:

1)Ознакомилась с методами управления шаговыми двигателями.

2)Был проведен анализ траектории, который позволил выявить элементарные участки

3)Разработана компьютерная программа, в которой каждый участок траектории обрабатывается соответствующей программой.

Были выделены элементарные участки траектории, которые оформленны в виде соответствующих подпрограмм.

Задание траектории сводиться к заданному набору соответствующих участков траектории. Была написана программа на языке Delphi.

**Список литературы**

1. Программируемые роботы. «Создаем робота андроида своими руками»/ Дж. Вильямс; пер. с англ. А.Ю. Карцева.-М.: НТ Пресс, 2006.-240 с.
2. Лебедев М.Б. CodeVisionAVR: пособие для начинающих. – М.:Додэка–XXI,2008.–592 с.: ил.
3. Юревич Е.И. Основы робототехники – 2-ое изд., перераб. и доп. – Спб.: БХВ–Петербург. –2005. –416с.: ил
4. Википедия: Свободная энциклопедия [Электронный ресурс]:Шаговые Электродвигатели. URL: http://ru.wikipedia.org/wiki (дата обращения: 25.05.2011).
5. Электропривод. URL: http://electroprivod.ru/ (дата обращения: 26.05.2011).
6. Автоматизированные технологические комплексы «Оборудование - робот». Методические рекомендации. М.: НИИМАШ, 1981. – 104с.
7. Промышленные роботы агрегатного типа / Ю.Г.Козырев, В.Б.Великович, С.В.Житомирский и др. М.: НИИМАШ, 1978. – 50с.

**Приложение**

В данном приложении приведена программа управления макетом транспортного робота.

unit Unit1;

interface

uses

 Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, LPTIO,

 ExtCtrls, StdCtrls, Buttons;

type

 TForm1 = class(TForm)

 Panel1: TPanel;

 ComboBox1: TComboBox;

 Label1: TLabel;

 SpeedButton1: TSpeedButton;

 SpeedButton2: TSpeedButton;

 SpeedButton3: TSpeedButton;

 SpeedButton4: TSpeedButton;

 GroupBox1: TGroupBox;

 Edit1: TEdit;

 Timer1: TTimer;

 GroupBox2: TGroupBox;

 Edit2: TEdit;

 GroupBox3: TGroupBox;

 Edit3: TEdit;

 Label2: TLabel;

 Label3: TLabel;

 procedure FormCreate(Sender: TObject);

 procedure FormDestroy(Sender: TObject);

 {procedure SpeedButton1MouseDown(Sender: TObject; Button: TMouseButton;

 Shift: TShiftState; X, Y: Integer);

 procedure SpeedButton2MouseDown(Sender: TObject; Button: TMouseButton;

 Shift: TShiftState; X, Y: Integer);

 procedure SpeedButton3MouseDown(Sender: TObject; Button: TMouseButton;

 Shift: TShiftState; X, Y: Integer);

 procedure SpeedButton4MouseDown(Sender: TObject; Button: TMouseButton;

 Shift: TShiftState; X, Y: Integer); }

 procedure Timer1Timer(Sender: TObject);

 procedure SpeedButton1Click(Sender: TObject);

 procedure SpeedButton2Click(Sender: TObject);

 procedure SpeedButton3Click(Sender: TObject);

 procedure SpeedButton4Click(Sender: TObject);

 private

 { Private declarations }

 Lpt : TLptPortConnection;

 public

 { Public declarations }

 function GetCurrentPort : byte; // Читает список ComboBox1 и возвращает соответствующий номер порта

 // function GetCurrentRegister : byte; // Читает список ComboBox2 и возвращает соответствующий номер регистра

procedure InOut; //-- EPP: вход / выход переключатель

 end;

var

 Form1: TForm1;

 direction1:integer;

 direction2:integer;

 step1:integer;

 step2:integer;

 R:real;

 K:real;

 S:real;

 i:integer;

 motor1: array[0..3] of byte;

 motor2: array[0..3] of byte;

implementation

{$R \*.DFM}

procedure TForm1.FormCreate(Sender: TObject);

var msg : AnsiString;

begin

timer1.Enabled:=false;

step1:=0;

step2:=0;

 motor1[0]:=48; // 0011

 motor1[1]:=96; // 0110

 motor1[2]:=192;// 1100

 motor1[3]:=144; // 1001

 motor2[0]:=3; //0011

 motor2[1]:=9; //1001

 motor2[2]:=12; //1100

 motor2[3]:=6; //0110

 Lpt := TLptPortConnection.Create;

 if not Lpt.Ready then

 begin { объект не готов -- покажем код ошибки }

 msg := 'Ошибка при создании объекта Lpt, драйвер ERROR, код = ' + IntToStr(GetLastError());

 Application.MessageBox(PChar(msg),'ERROR',MB\_OK);

 Application.Terminate; // выход

 end;

 // Проверим наличие портов и заполним их список в ComboBox-e

 ComboBox1.Items.Clear;

 ComboBox1.Text := '';

 if Lpt.IsPortPresent(LPT1) then ComboBox1.Items.Add('$3BC');

 if Lpt.IsPortPresent(LPT2) then ComboBox1.Items.Add('$378'); //-ВНИМАНИЕ: $378 здесь назван LPT2!

 if Lpt.IsPortPresent(LPT3) then ComboBox1.Items.Add('$278');

 if 0<>ComboBox1.Items.Count then ComboBox1.ItemIndex := 0;

 Lpt.WritePort (GetCurrentPort,0,0);

end;

procedure TForm1.FormDestroy(Sender: TObject); // Обработчик разрушения

begin

 Lpt.Destroy;

{ Здесь разместите все таймеры, которые должны быть отключены }

end;

function TForm1.GetCurrentPort : byte; // Читает список ComboBox1 и возвращает соответствующий номер порта

begin

 if '$3BC' = ComboBox1.Text then

 GetCurrentPort:=LPT1

 else

 if '$378' = ComboBox1.Text then

 GetCurrentPort:=LPT2 else

 GetCurrentPort:=LPT3;

end;

 procedure TForm1.InOut; //--- переключ. направления в реж. EPP вход / выход

 begin

 if Lpt.IsPortBidirectional(GetCurrentPort)= true then

 begin

 Lpt.WritePort(GetCurrentPort,2, (DIRECTION xor Lpt.ReadPort(GetCurrentPort,2))) ; //-- 32 (бит 5 )

 end else

 MessageBox(Handle,PChar('Данный порт не является активным в EPP режиме. Для активации войдите в Bios, раздел INTEGRATED PERIPHERALS или др., найдите строку PARALLEL PORT MODE: измените режим работы порта на EPP или SPP/EPP.'),

 PChar('Не доступен EPP режим '),MB\_ICONINFORMATION + MB\_OK +MB\_DEFBUTTON2);

 end;

procedure TForm1.Timer1Timer(Sender: TObject);

var

speed: integer;

speed1:real;

PortValue: byte;

begin

K:=strTofloat(edit3.text); {количество шагов}

if i<round(k) then

begin

trystrtoint(edit1.Text, speed);

Timer1.Interval:=round((speed/200)\*1000);

PortValue:= motor1[step1] or motor2[step2];

Lpt.WritePort (GetCurrentPort,0,PortValue);

 step1:=step1+direction1;

 step2:=step2+direction2;

 if step1=-1 then step1:=3;

 if step1=4 then step1:=0;

 if step2=4 then step2:=0;

 if step2=-1 then step2:=3;

 i:=i+1;

 end;

R:=strTofloat(edit2.text); {радиус колеса}

S:=(2\*3.14\*R\*K)/200; {расстояние}

label2.caption:=FloatToStr(s);

end;

procedure TForm1.SpeedButton1Click(Sender: TObject);

begin

i:=0;

direction1:=1;

direction2:=1;

timer1.Enabled:=true;

end;

procedure TForm1.SpeedButton2Click(Sender: TObject);

begin

i:=0;

direction1:=-1;

direction2:=1;

timer1.Enabled:=true;

end;

procedure TForm1.SpeedButton3Click(Sender: TObject);

begin

i:=0;

direction1:=-1;

direction2:=-1;

timer1.Enabled:=true;

end;

procedure TForm1.SpeedButton4Click(Sender: TObject);

begin

i:=0;

direction1:=1;

direction2:=-1;

timer1.Enabled:=true;

end;

end.