

Региональная открытая конференция по научно-техническому творчеству
школьников «Лабиринты науки»

Секция «Техника»

Робот-манипулятор для решения складских задач

Выполнил: **Колобов Данила
Сергеевич,**
МОУ «Средняя школа № 18» г.
Ярославля,
ученик 11 б класса

Научный руководитель: **Шамахов
Леонид Михайлович,**
ЯГТУ, Ассистент кафедры
«Строительные и дорожные машины»

Дороднова Елена Александровна,
учитель истории и обществознания,
МОУ «Средняя школа № 18»
г. Ярославля

Ярославль, 2024 год

Оглавление

Введение.....	3
Глава 1. Обзоры существующий конструкций.....	4
1.1 Классификация роботов-манипуляторов.....	4
1.2 Существующие модели.....	6
1.3 Область применения.....	7
Глава 2. Разработанный робот-манипулятор.....	8
2.1 Описание разработанной конструкции.....	8
2.2 Описание использованных компонентов.....	9
Глава 3. Разработка управляющей команды.....	10
Заключение.....	11
Список литературы.....	12
Приложение.....	13

Введение

В настоящее время в производства интегрируются большое количество технологий, новые технологии позволяют заменить человека на робота-манипулятор в опасных условиях работы.

Целью работы является разработка робота-манипулятора на сервоприводах при управлении по средствам микроконтроллера Arduino Uno. Для достижения поставленной цели, требуется решить следующие **задачи**:

1. Изучить существующие технологии
2. Разработать устройства и собрать его
3. Написать управляющую программу для Arduino Uno
4. Собрать электронную схему
5. Объединение всех компонентов и тестирование продукта

В ходе выполнения проекта, требуется разработать продукт, состоящий из металлической конструкции, сервоприводов и платы управления, рассчитанный для использования на предприятиях, с целью заменить человеческий труд в опасных условиях.

Методы: анализ, систематизация, моделирование. Эксперимент.

Глава 1. Обзоры существующих конструкций

1.1 Классификация роботов манипуляторов

Робот-манипулятор — это промышленное оборудование, которое выполняет функции человеческой руки. Это может быть как отдельное устройство, так и составная часть сложного роботизированного комплекса. Манипулятор может осуществлять вращательные или поступательные движения. Роботы-манипуляторы применяются в различных сферах для автоматизации производственных процессов. Например, в автомобильной промышленности они используются для автоматизации логистических систем, в производстве электроники — для сборки и монтажа печатных плат.

Отличительной чертой подобных устройств является их способность выполнять задачи, которые являются физически невозможными для человека, например транспортировка тяжелых деталей. При этом непосредственное присутствие человека при выполнении операций не требуется. Работники предприятия могут удалённо управлять роботом.

Роботы-манипуляторы можно классифицировать по различным признакам:

- По количеству осей: 4-осевые применяются для фасовки и укладки продукции, 6-осевые — для прочих операций.
- По степени подвижности: от 2 до 6 уровня. Чем выше параметр, тем более точно робот-манипулятор производит наклоны, сжатие, вращение.
- По типу операций: Универсальные выполняют основные и вспомогательные задачи, подъёмно-транспортные перемещают объекты в пространстве, производственные выполняют конкретные технологические задачи: покраску, сварку, резку, сортировку, сверление, листогибочные операции.

- По грузоподъёмности: лёгкие: до 10 кг, средней грузоподъёмности: до 200 кг, тяжёлые: до 1 тонны, сверхтяжёлые: более 1 тонны
- По типу монтажа: Стационарные (наиболее распространённый тип) и мобильные (могут выполнять задачи в любой точке производства или на значительном удалении от него).
- По типу применения: Автономные работают по заданной программе без участия человека, коллаборативные — вместе с человеком, управляемые непосредственно оператором.
- По системе координат: Цилиндрическая, сферическая (полярная), прямоугольная и угловая системы координат определяют рабочую площадку робота и те операции, в которых он способен заменить человека.
- По типу привода: **Электроприводы.** В их состав входят двигатели, усилители мощности, механизмы передачи, датчики, отслеживающие скорость перемещения и положение звеньев. Электроприводы отличаются высоким КПД, отсутствием шумов и простотой регулировки. **Пневматические приводы** состоят из пневмоцилиндров одностороннего и двустороннего действия, специализированных моторов, мембранных камер, распределителей, систем подготовки и подачи воздуха. Пневматические приводы легко настраиваются и управляются, защищены от перегрузок, взрывобезопасны и виброустойчивы. **Гидравлические приводы** включают в себя цилиндры с поршнем, который выдвигается под воздействием на шток рабочей среды, подаваемой гидронасосом под определённым давлением. Гидравлические приводы отличаются высокой грузоподъёмностью. **Сервоприводы** включают в себя бесколлекторный электродвигатель, волновой редуктор, блок управления с контроллером и датчик обратной связи. Они обеспечивают

положение робота в соответствии с управляющим сигналом, полученным от контроллера.

1.2 Существующие модели

Декартовый робот. Рука имеет три призматических сустава, оси которых совпадают с декартовым координатором. Используется для подбора и размещения оборудования, нанесения герметика, сборочных операций, обращения со станками и дуговой сварки. Приложение [1]

Цилиндрический робот. Оси робота образуют цилиндрическую систему координат. Используется для сборочных операций, манипуляций на станках, точечной сварки и манипуляций на машинах литья под давлением. Приложение [2]

Сферический робот. Оси робота образуют полярную систему координат. Используется для обработки станков, точечной сварки, литья под давлением, штамповочных машин, газовой и дуговой сварки. Приложение [3]

Робот SCARA. Оснащён двумя параллельными поворотными шарнирами для обеспечения податливости в плоскости. Используется для выполнения монтажных работ, нанесения герметика, сборочных операций и манипулирования станками. Приложение [4]

Шарнирно-сочленённый робот. Рука имеет как минимум три поворотных шарнира. Используется для сборочных операций, литья под давлением, штамповки машин, газовой сварки, дуговой сварки и покраски распылением. Приложение [5]

Параллельный робот. Руки имеют параллельные призматические или поворотные шарниры. Одно из применений — мобильная платформа для управления тренажёрами полёта в кабине пилота. Приложение [6]

Антропоморфный робот. Имеет форму, напоминающую человеческую руку, то есть с независимыми пальцами. Предназначен для исследовательских и медицинских целей, например — как часть протезов. Приложение [7]. [4]

1.3 Область применения

Роботы-манипуляторы активно используются в промышленности и применяются в таких областях как: Автомобилестроение, машиностроение, металлообработка, химическая промышленность, медицина, фармацевтика, продовольствие и сельское хозяйство.

В автомобилестроение роботы-манипуляторы выполняют перенос грузов, установку деталей, сборку кузова и контроль качества этой сборки.

В электронике и электротехнике манипуляторы выполняют сборку жёстких дисков и плат для компьютеров.

В машиностроение и металлообработке манипуляторы выполняют финишную обработку и шлифовку различных деталей из металла и композитных материалов.

В химической промышленности роботов применяют для автоматизации работы с опасными веществами. К манипуляторам, используемым в химической промышленности, предъявляются особые требования: они должны быть устойчивыми к агрессивным средам и иметь высокую точность позиционирования.

В медицине и фармацевтике манипуляторы используют в качестве помощника на операциях. Также такие устройства позволяют совершать хирургическое вмешательство удалённо. Также могут быть использованы как протезы конечностей.

В продовольствие и сельском хозяйстве применяют мобильные манипуляторы для сбора урожая. [2]

Глава 2. Разработанный робот-манипулятор

2.1 Описание разработанной конструкции

Нами был разработан робот-манипулятор с использованием таких компоненты как: плата Arduino Uno, сервоприводы, макетная плата, соединительные провода, металлическая конструкция, блок питания 9V и Power Bank. В приложение [8] под буквой (a) находится Power Bank отвечающий за питание платы Arduino Uno (c), в которой находится управляющая программа для робота-манипулятора, под буквой (b) находится блок питания, отвечающий за работу сервоприводов (d-i), которые выполняют действия программы, под буквой (j) находится макетная плата, которая требуется для подачи питания на все сервоприводы, а также разместить все прочее компоненты.

2.2 Описание использованных компонентов

1. Плата Arduino Uno используется для создания электронных проектов в разных сферах. Также Arduino Uno позволяет обмениваться информацией с

другими устройствами через различные коммуникационные интерфейсы, что даёт возможность создания умных систем и их интеграции в существующую инфраструктуру.

2. Сервопривод — это электродвигатель, редуктор, датчик обратной связи и блок управления в одном корпусе. Сервопривод используется для точного контроля положения или движения механических систем.

3. Макетная плата используется для сборки и моделирования прототипов электронных устройств. Она позволяет создать рабочий макет той или иной схемы, чтобы впоследствии собирать прибор на специально подготовленной плате с уверенностью, что всё заработает как надо.

4. Блок питания выполняют роль электронных регуляторов, преобразуя поступающее переменное напряжение из сети в стабильный постоянный ток, соответствующий требованиям оборудования.

5. Power Bank — портативный аккумулятор представляет собой небольшой корпус со встроенными батареями, которые можно зарядить от сети или другого источника питания.

Глава 3. Разработка управляющей программы

Для написания управляющей программы была выбрана среда программирования — Arduino IDE.

Arduino IDE — интегрированная среда разработки для Windows, macOS и Linux, разработанная на C и C++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей.

В Arduino IDE можно писать код, оптимизировать работу платы, получать данные об эффективности кода, устанавливать сторонние библиотеки и разрабатывать свои. [3]

Приложение [9] — Код программы

`#include <Servo.h>` отвечает за добавление библиотеки, которая позволяет управлять сервоприводами, из данной библиотеки были использованы такие команды как `Servo` и `.write()`. `.write()` отвечает за поворот сервопривода на определенный градус который указывается внутри скобок. `Servo` отвечает за присвоение сервоприводу имени, к которому впоследствии обращаются, чтобы конкретный сервопривод выполнил ту или иную команду. [1]

Также в коде присутствуют встроенные в язык программирования команды, такие как `delay()`, `.attach()`, `switch()`. `delay()` отвечает за задержку перед выполнением следующего действия. `.attach()` позволяет задать управляющий канал для сервопривода, в скобках указывается канал. `switch()` отвечает за выбор алгоритма, по которому будет действовать программа, в скобках указывается номер алгоритма. Например, если у робота-манипулятора присутствует датчик движения, он может переключить на другой алгоритм, который позволит предотвратить происшествие на производстве.

Заключение

В результате выполнения работы удалось разработать робот-манипулятор. Данный продукт можно реализовывать на небольших

предприятиях, которые производят не тяжелые и не хрупкие товары. Также данная конструкция очень дешёвая по сравнению с аналогами.

В будущем данную конструкцию можно усовершенствовать, добавив камеру и джойстик для управления в ручном режиме. А также замена сервоприводов на другой вид приводов, которые смогу увеличить плавность действий.

Продукт можно монетизировать за счет продажи роботов-манипуляторов на различные предприятия. Так как сейчас автоматизация набирает в России, то на данную также повышается спрос. [5]

Список литературы

[1] Статья: Управление сервоприводами. Сервоприводы – устройство и способы управления.

<https://habr.com/ru/articles/750222/>

Дата посещения (15.08.2024)

[2] Статья: Зачем нужны роботы манипуляторы.

<https://osnastik-robots.ru/informaciya/stati/zachem-nuzhny-roboty-manipulyatory/>

Дата посещения (16.08.2024)

[3] Статья: Arduino IDE.

https://ru.wikipedia.org/wiki/Arduino_IDE

Дата посещения (17.08.2024)

[4] Статья: Роботизированная рука: захватные устройства.

<https://top3dshop.ru/blog/robotic-arm-grippers.html>

Дата посещения (01.11.2024)

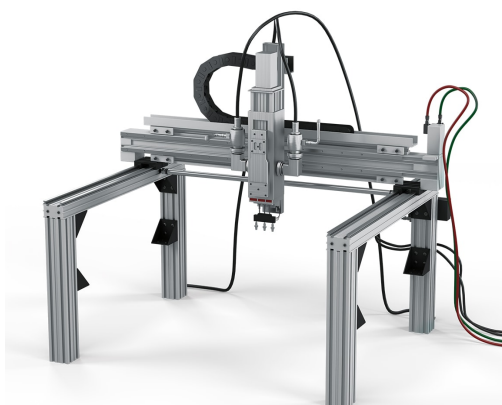
[5] Статья: Перспективы использования промышленных роботов манипуляторов.

<https://vektorus.ru/blog/promyshlennye-roboty-manipulyatory.html>

Дата посещения (14.10.2024)

Приложение

Приложение 1



Приложение 2



Приложение 3



Приложение 4



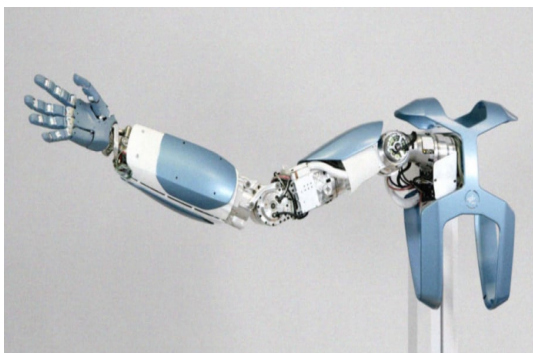
Приложение 5



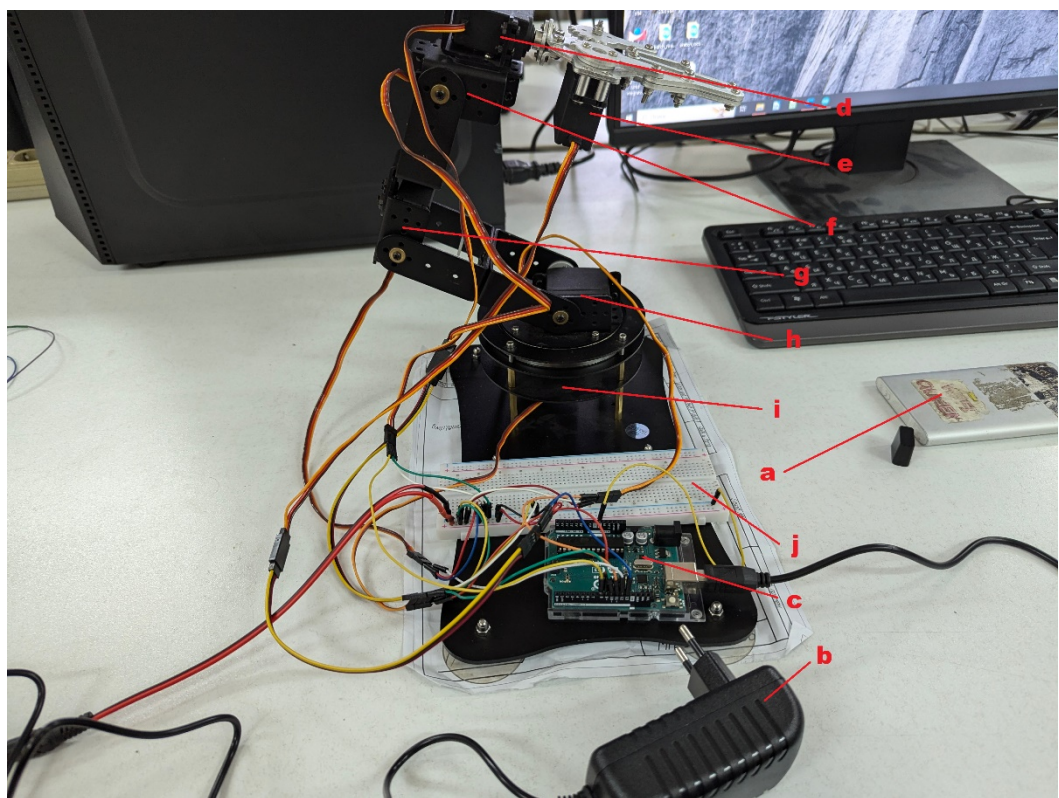
Приложение 6



Приложение 7



Приложение 8



Приложение 9

```

1  #include <Servo.h> //Подключение библиотеки для управления сервоприводами
2  Servo Constanto; // Дает название сервоприводу
3  Servo Rotationx; // Дает название сервоприводу
4  Servo Level_1; // Дает название сервоприводу
5  Servo Level_2; // Дает название сервоприводу
6  Servo Level_3; // Дает название сервоприводу
7  Servo Claw; // Дает название сервоприводу
8  int Plot = 1; // Вводится переменная чтобы менять алгоритм робота-манипулятора
9
10 void setup() {
11     Constanto.attach(8); // Управляющий канал для этого сервопривода 8
12     Rotationx.attach(9); // Управляющий канал для этого сервопривода 9
13     Level_1.attach(10); // Управляющий канал для этого сервопривода 10
14     Level_2.attach(11); // Управляющий канал для этого сервопривода 11
15     Level_3.attach(12); // Управляющий канал для этого сервопривода 12
16     Claw.attach(13); // Управляющий канал для этого сервопривода 13
17 }
18
19 void loop() {
20     switch (Plot) { //Выбирается алгоритм по которому будет действовать робот-манипулятор
21     | case 1: // Алгоритм 1
22         Constanto.write(0); //Начало алгоритма
23         Rotationx.write(90);
24         Level_1.write(0);
25         Level_2.write(0);
26         Level_3.write(0);
27         delay(2000); //Задержка 2 секунды
28         Rotationx.write(0);
29         delay(2000); //Задержка 2 секунды
30         Level_1.write(90);
31         Level_2.write(45);
32         delay(2000); //Задержка 2 секунды
33         Level_1.write(120);
34         Level_2.write(45);
35         Level_3.write(0);
36         Claw.write(0);
37         delay(2000); //Задержка 2 секунды
38         Claw.write(90); //Подбор кубика
39         delay(2000); //Задержка 2 секунды
40         Level_1.write(90);
41         Level_2.write(45);
42         delay(2000); //Задержка 2 секунды
43         Rotationx.write(90);
44         Level_1.write(0);
45         Level_2.write(0);

```

```

45 Level_2.write(0);
46 Level_3.write(0);
47 delay(2000); //Задержка 2 секунды
48 Level_1.write(90);
49 Level_2.write(45);
50 delay(2000); //Задержка 2 секунды
51 Rotationx.write(150);
52 delay(2000); //Задержка 2 секунды
53 Level_1.write(120);
54 Level_2.write(45);
55 Level_3.write(0);
56 delay(2000); //Задержка 2 секунды
57 Claw.write(0); //Опускание
58 delay(2000); //Задержка 2 секунды
59 Level_1.write(90);
60 Level_2.write(45); //Конец алгоритма
61 delay(2000); //Задержка 2 секунды
62 Plot += 1; //Меняет на 2 алгоритм
63
64
65 case 2:
66 Constanto.write(0); //Начало алгоритма
67 Rotationx.write(90); //Start
68 Level_1.write(0);
69 Level_2.write(0);
70 Level_3.write(0);
71 delay(2000); //Задержка 2 секунды
72 Rotationx.write(0);
73 delay(2000); //Задержка 2 секунды
74 Level_1.write(90);
75 Level_2.write(45);
76 delay(2000); //Задержка 2 секунды
77 Level_1.write(120);
78 Level_2.write(45);
79 Level_3.write(0);
80 Claw.write(0);
81 delay(2000); //Задержка 2 секунды
82 Claw.write(90); //Подбор кубика
83 delay(2000); //Задержка 2 секунды
84 Level_1.write(90);
85 Level_2.write(45);
86 delay(2000); //Задержка 2 секунды
87 Rotationx.write(90);
88 Level_1.write(0);
89 Level_2.write(0);

```

```

90 Level_3.write(0);
91 delay(2000); //Задержка 2 секунды
92 Level_1.write(90);
93 Level_2.write(45);
94 delay(2000); //Задержка 2 секунды
95 Rotationx.write(120);
96 delay(2000); //Задержка 2 секунды
97 Level_1.write(120);
98 Level_2.write(45);
99 Level_3.write(0);
100 delay(2000); //Задержка 2 секунды
101 Claw.write(0); //Опускание
102 delay(2000); //Задержка 2 секунды
103 Level_1.write(90);
104 Level_2.write(45); //Конец алгоритма
105 delay(2000); //Задержка 2 секунды
106 Plot -= 1; //Меняет алгоритм на 1
107 }
108 }
109

```