|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Российский биотехнологический университет (РОСБИОТЕХ)»** | | | | | | | | | | | | | |
|  |  | |  |  | |  |  |  | |  | |  |
| Кафедра | | | Информатика и вычислительная техника пищевых производств | | | | | | | | | | |
|  |  | |  |  | |  |  |  | |  | |  |
| Направление (Специальность) | | | 09.03.01 - Информатика и вычислительная техника | | | | | | | | | | |
|  |  | |  |  | |  |  |  | |  | |  |
| Профиль | | | Информационные технологии и бизнес-аналитика | | | | | | | | | | |
|  |  | |  |  | |  |  |  | |  | |  |
|  |  | |  |  | |  |  | **К ЗАЩИТЕ** | | | | | |
|  |  | |  |  | |  |  | **(РЕКОМЕНДОВАНО / НЕ РЕКОМЕНДОВАНО)** | | | | | |
|  |  | |  |  | |  |  | зав. кафедрой | | | | | |
|  |  | |  |  | |  |  | к.ф.-м.н., доцент | | | | | |
|  |  | |  |  | |  |  | *(ученая степень, ученое звание)* | | | | | |
|  |  | |  |  | |  |  |  | | |  | Т.А. Санаева | |
|  |  | |  |  | |  |  | *(подпись)* | | | Санаева | *(И.О. Фамилия)* | |
|  |  | |  |  | |  |  | « 21 » 05 2025 г. | | | | | |
|  |  | |  |  | |  |  |  | |  | |  |
| **КУРСОВАЯ РАБОТА** | | | | | | | | | | | | | |
| *по дисциплине* | | | | | | | | | | | | |
| *«Информационные системы и технологии»* | | | | | | | | | | | | |
|  |  | |  |  | |  |  |  | |  | |  |
| на тему: | | Разработка роботизированной системы | | | | | | | | | | | |
|  |  |
|  |  | *(тема курсовой работы)* | | | | | | | | | | | |
|  |  |  |  |  | |  |  |  | |  | |  |
| Обучающийся: | | | Забродин | | « 21 » 05 2025 г. | | | | А.М. Забродин | | | | |
|  |  |  | *(подпись)* | |  |  |  |  | *(инициалы, фамилия)* | | | | |
|  |  | |  |  | |  |  |  | |  | |  |
|  | | | |  |  | | | группа | | |  | 240-090301/БА-1 | |
|  |  | |  |  |  | | |  | | |  | *(шифр группы)* | |
| Руководитель | | | Ящун | | « 21 » 05 2025 г. | | | | доц, к.т.н, Т.В. Ящун | | | | |
|  |  | | *(подпись)* | |  |  |  |  | *(уч. степень, уч. звание, инициалы, фамилия)* | | | | |

# 

[**Введение** 3](#_Toc198832192)

[**История** 4](#_Toc198832194)

[**Актуальность** 5](#_Toc198832195)

[**Аналоги и сравнение** 7](#_Toc198832196)

[**Анализ предметной области** 12](#_Toc198832197)

[**1.1. Выбор платформы** 12](#_Toc198832198)

[**1.2. Выбор периферийных устройств** 14](#_Toc198832199)

[**Программа** 18](#_Toc198832200)

[**2.1. Математические расчёты** 18](#_Toc198832201)

[**2.2. Программная реализация.** 20](#_Toc198832202)

[**Заключение** 22](#_Toc198832203)

[**Список литературы** 23](#_Toc198832204)

# **Введение**

Робототехника - одна из технологий будущего. На данный момент, в робототехнику вкладывают огромное количество ресурсов. Создание роботов лежит в основе множества технологических производств. Технологическая индустрия пытается создать универсального помощника человеку, который будет обладать различными функциями. Уже существуют роботы-пылесосы, роботы-мойщики окон, роботы-питомцы и так далее. Каждый робот необходим в своей сфере и является полезным атрибутом для человеческой жизни. С развитием технологий механические машины становятся все умнее и скоро станут уметь все тоже, что и мы.

На сегодняшний день, роботизированные системы гораздо точнее выполняют работу, особенно с мелкими деталями и, что очень важно, делают это каждый раз одинаково. У робота не дрогнет рука, он никогда не будет расстроен, ему не нужен перерыв на обед или перекур, всегда чёткое выполнение поставленных задач. Робот не боится, находясь в узком, замкнутом пространстве. Такая система никогда не устанет.

Однако такие роботы достаточно дорогостоящи и трудны в производстве. Поэтому, для решения более простых задач, были придуманы более простые и менее технологически трудоёмкие роботы. Чаще всего такие роботы используются в качестве игрушек. Но такие роботы также используются и в научной деятельности. На них тестируют всевозможные модели, создают транспорт будущего на основе алгоритмов движения, добавляют всё более новые устройства и развивают функционал таких роботизированных систем.

В данной работе будет показан пример изготовления простой, многофункциональной роботизированной системы. Будет рассмотрен выбор комплектующих, пример изготовления корпуса и алгоритмы написания программы для управления роботизированной системы.

# **История**

Робот - устройство, которое своими действиями повторяет некую человеческую работу.

Аналоги роботов уходят в далекие времена. Существует множество мифов, в которых упоминаются какие-либо существа, которые были оживлены человеком и в которые человек внес определенные функции. Например, многим известен миф о Големе, существе, созданном из глины. Его цель была защищать еврейский народ. То есть это некий робот - защитник.

Настоящие упоминания механических машин появились еще в эллинистическую эпоху. На острове Фарос стоял маяк, вокруг стояли механические женские фигуры, которые освещались и издавали звуки, чтобы мореплаватели всегда могли их заметить.

В 12 веке изобретатель Аль-Джазари создал лодку с четырьмя механическими музыкантами, которые играли на разных инструментах. В 1495 году Леонардо да Винчи сделал чертеж механического рыцаря, который должен был уметь сидеть, двигать головой, руками и открывать забрало. Однако, неизвестно, пытался ли да Винчи создать данного робота.

В 20 веке уже появилось непосредственно слово «робот», которое впервые упоминается в пьесе Карела Чапека. В 1980 году роботы начали активно производится, появилась торговля данными устройствами, которые производились на основе технологий Японии.

Сейчас роботы начали появляться повсюду и облегчать человеку жизнь. Роботы появились уже во многих профессиях, а Япония создает робота, которого внешне сложно отличить от человека. Данные устройства стали неотъемлемой частью нашей жизни [1].

Однако, если роботы стали таким постоянным явлением в нашей жизни, то как же его собрать?

# **Актуальность**

Актуальность работы заключается в том, что созданная система позволит тестировать дальнейшие разработки, не прилагая больших усилий для переноса модели в реальный мир. Программа написана в отдельных функциях, что позволяет с лёгкостью добавлять или изменять функционал системы для тестирования конкретных элементов модели или всей модели в целом [2].

Также при добавлении к роботу дополнительного функционала и написания функций обработки данных для решения поставленных задач можно использовать систему в более прикладных целях. Например, данного «жука» можно использовать для выполнения ремонтно-строительной деятельности. Если добавить измерители расстояния до объектов, то он сможет измерять размеры пространства и свои вычисления предоставлять человеку, что упрощает и ускоряет процесс работы, то есть робот сможет самостоятельно измерять пространство, создавая 3D-модели по полученным результатам.

При добавлении камеры система сможет получать и обрабатывать данные для поиска объектов. Например, проникать туда, куда человек не может добраться и обследовать шахты, водосточные трубы в поисках различных неисправностей без риска для человека, так как гексапод обладает повышенной проходимостью, чем колесно-гусеничная техника.

Также мы можем добавить «жуку» датчики сканирования почвы, воздуха, тепловизоры и прочее для того, чтобы он изучал местность, различные препятствия, заторы и после служил навигатором. Например, это может выглядеть следующим образом: «Поверните направо, как только увидите первый красный дом, там надо будет доехать до зеленого бака, повернуть налево и возле первого перекрестка остановиться». Такой способ значительно упростит движение и точно разъяснит человеку куда ему ехать.

При усилении конструкции и добавлении захватов, робот может служить неким погрузочно-перевозочным целям. Способен передвигать небольшие предметы, при том, что его проходимость будет выше чем у колёсной техники.

Такая система может быть использована для выполнения огромного спектра задач в различных сферах деятельности, что позволит облегчить и обезопасить человеческий труд. Также она позволяет проводить научные исследования на ее основе, проверяя теорию в реальном мире.

# **Аналоги и сравнение**

Существует множество различных роботов-жуков. Пока большинство их них используется в качестве игрушки, но есть и по-настоящему полезные роботизированные системы. Сравним некоторые из них.

Робот на рисунке 1 имеет корпус в виде настоящего жука, «усики» служат в качестве дальномера и помогают ориентироваться в пространстве. Также, подобно живому насекомому, данный жук боится громких звуков и «дергает лапками» при перевороте на спину. Данный вид роботизированной системы служит исключительно для игровой деятельности [3].



Рисунок 1 - Игрушечный робот-жук

На рисунке 2 представлен робот, который внешне абсолютно прост. Он состоит из платформы Arduino, сервоприводов, батареи, металлической проволоки и смешной рожицы. Однако данный робот умеет ходить на различных скоростях и двигаться под мелодию. Его назначение также является игровым [4].

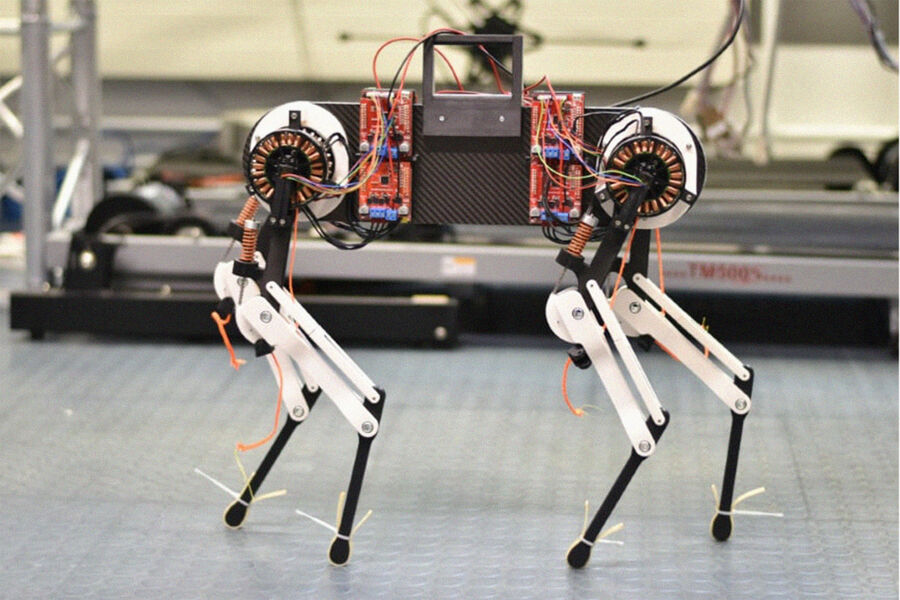


Рисунок 2 - Самодельный робот-жук

На рисунке 3 представлено три робота, которые поражают своим внешнем видом. У каждого тело сделано из различных материалов: железо, дерево и пластик. Данные системы запрограммированы на передвижение, при этом двигаются они в зависимости от их «вида», подражая своим живым родичам.



Рисунок 3- Различные роботы-жуки

На рисунке 4 представлен уже более интересный экземпляр. Данный робот является микроскопическим жуком, разработанным японскими специалистами. Служит от в медицинских целях. Например, в него установлены камеры и сенсоры для помощи врачам, а также робот умеет делать инъекции, которые в будущем способствуют тому, что во многих случает можно будет избежать хирургического вмешательства [5].



Рисунок 4 - Медицинский робот-жук

На рисунке 5 представлен робот-жук, который является ходячим светильником. Такой «питомец» помогает хозяину не ходить в темноте, а для их жизнедеятельности достаточно иметь с собой лампочки в 25 Ватт и периодически их менять.



Рисунок 5 - Робот-жук, выполняющий функцию светильника

На рисунке 6 представлен избавитель от насекомых. Данный робот благодаря запаху привлекает насекомых, например, тараканов, а после заманивает их в ловушку, где вредоносные существа погибают. В хозяйстве может стать незаменимой вещью.



Рисунок 6 - Робот-жук, избавляющий от насекомых

На рисунке 7 представлен робот, который по своему функционалу и внешнему виду схож с роботом, описанным в данной работе. Его создали и запрограммировали для того, чтобы он умел ходить и изучать местность, а в случае чрезвычайных ситуация мог оказать помощь, как в плане сканирования местности, так и оповещения. Также этот робот является своего рода машиной для перевозки 2 человек.



Рисунок 7-Ходящий робот-жук

Можно заметить, что каждый робот служит в различных целях. Большинство существующих «жуков» носит собой развлекательный характер, однако, с каждым годом разработчики придумывают все больше таких помощников с различными целями.

Все роботы имеют разные размеры и различные материалы для своего корпуса. В зависимости от их назначения, выбирается пластик или железо (в редких случаях дерево). Наиболее часто роботизированные системы создаются значительно меньших размеров, чем робот, показанный в данной работе. Большие роботы служат больше для игровых целей или для начинающих сборов. Однако «жуки» средних размеров способны на оказание помощи людям как в доме, так и на улице, в отличие от малогабаритных, которые могут спасать в медицинской деятельности и избавлении от паразитов.

# **Анализ предметной области**

## **1.1. Выбор платформы**

С самого начала мы столкнулись с проблемой выбора. Существует множество платформ для управления периферийными устройствами, реализации передачи и обработки полученных данных.

На сегодняшний день на рынке представлен достаточно большой выбор контроллеров: Raspberry Pi, Arduino, Virt2real, The Shrimp, Maple, Mbed и другие [6].

The Shrimp. Данная платформа является минималистической версией Arduino. Она состоит из микроконтроллера ATmega328, обвязки из нескольких конденсаторов, резистора, диода и кварцевого резонатора. Такую платформу перепрошивают и используют как аналог Arduino.

Maple. Эта платформа является наиболее технически продвинутой. Она базируется на 32-битном микроконтроллере STM32F103RB с архитектурой ARM Cortex M3. Maple имеет 120Кб Flash памяти, 20Кб SRAM памяти, 43 вывода и встроенная система заряда LiPo батарей. Скетчи этой платформы совместимы с Arduino. Несмотря на все плюсы, Maple имеет всего лишь три библиотеки и стоит гораздо дороже Arduino.

Mbed. В основе лежит мощный микроконтроллер NXP LPC1768 с архитектурой ARN Cortex M3. Платформа имеет 512Кб Flash памяти и 32Кб RAM. В отличие от Maple, Mbed имеет множество различных библиотек. Программы составляются при помощи online компилятора. Цена такой платформы превышает цены любой другой.

Raspberry PI - это целый микрокомпьютер. В него входит процессор Broadcom BCM2835, ядро ARM11, графическое ядро, ОЗУ 523Мб, USB порт, выход HDMI, Ethernet 10/200 Мбит/с, выход jack 3.5, 1 выводов. ОС загружается с SD флеш карты. Цена не сильно выше средней цены Arduino.real. Данная платформа имеет маленький размер и мало потребляет энергию, что позволяет обеспечивать видео и аудио контроль, управлять подключенными устройствами по интерфейсам. В него входит Ethernet, Wi-Fi, USB выходы, можно подключить GPRS, 3G и LTE. Также комплект включает модуль камеры, микропроцессорный модуль, разъем для подключения сторонних источников.

Arduino. Эта платформа имеет множество видов, размеров, функциональности. Она самая популярная и наиболее приспособлена для работы. Рассмотрим все платы Arduino [7].

Arduino xxx. Имеет самый стандартный размер. Все платы этой версии имеют одинаковое количество входов-выходов, программируются по USB, в них входит микроконтроллер ATMega. От версии микроконтроллера зависит количество каналов и оперативной памяти. Также имеется USB-UART FT232, который позволяет программировать плату без программатора.

Arduino UNO. При создании UNO разработчики убрали USB-UART и поставили вместо него Atmega8U2. Благодаря этому можно залить свою прошивку и сделать из Arduino все, что душе угодно. Работать с ней труднее, так как придется долго писать свою прошивку, а от FT232 мало чем отличается.

Arduino Leonardo. Данная платформа содержится на одном чипе, а USB ни от чего не зависит. Построена эта плата на ATmega32u4. Также увеличилось количество ШИМ-выходов, объем оперативной памяти, количество аналоговых выходов, а USB и UART разделены. Можно реализовать все тоже, что и на UNO, но гораздо проще.

Arduino Mega xxx. Размер и характеристики этих плат значительно прокачаны, в сравнении с другими. Можно воткнуть почти все шилды, сильно увеличен объем памяти (128/256Кб Flash памяти, 8Кб оперативной, 4Кб еепрома 4 UART). Эти платы построены на ATmega1280/ATmega2560.

Arduino Nano. Компактная плата с mini USB. Шилды к ней не подходят, однако ее саму можно вставлять в макетку. Использует ATmega328 и FT232.

Arduino Mini. Данная плата имеет более маленький размер, чем Nano. Также не совместима с шилдами. Программируется с помощью переходника USB-Serial, так как не имеет выхода USB. Использует ATmega328.

В нашем случае используется платформа Arduino Mega 2560. Выбор был сделан так из-за того, что данная платформа самая доступная, имеет достаточное количество выходов, что позволяет расширять возможности робота, легко программируема и не требует дополнительных знаний для работы с ней. Также она имеет высокую функциональность и большой объем памяти. Материальная сторона также дала преимущество для выбора именно этой платформы.

## **4.2. Выбор периферийных устройств**

### **1.2.1. Выбор сервоприводов**

Для движения робота необходимы сервоприводы. На каждую ногу решено было поставить 3 сервопривода, что в конечном итоге даёт 18 сервоприводов на итоговую модель. Общий вес конструкции не должен превышать 1 килограмма, что при длине бедра в 60 миллиметров и длине голени 100 миллиметров даёт момент в 6\*1 =6 кг\*см. Один сервопривод не способен справиться с таким весом, но в конструкции 6 ног, т.е. в каждый момент времени весь вес распределяется на 3 ноги, что даёт нам предельный момент в 6/3=2 кг\*см. Если учитывать тот фак, что в подъёме ноги участвуют два сервопривода, то предельный момент получается почти в два раза меньше, т.е. около 1кг\*см. Именно на такие характеристики ориентировались при выборе сервоприводов. Идеальным кандидатом стали микросервоприводы SG90. Они имеют вес 13 грамм, крутящий момент до 1,5 кг\*см.

### **1.2.2. Выбор дальномеров.**

При выборе дальномеров использовался принцип простоты и надёжности, по таким характеристикам ультразвуковой дальномер HC-SR04 стоит на голову выше остальных конкурентов. Обладая дальностью от 2 до 400 сантиметров и массой в 9 грамм, стал идеальным решением поставленных задач.

### **1.2.3. Создание корпуса**

Корпус реализован в виде прямоугольной платформы. Размеры платформы были выбраны исходя из компонентов робота, что позволяет с лёгкостью разместить все необходимые устройства для работы робота. На платформе расположен контролер Arduino, батарея, ультразвуковой дальномер и соединительная платформа [8].



Рисунок 8 - Готовый робот-жук

Соединение частей робота происходит посредством стяжек и проклеивания мест соединений. Это связано с тем, что, на данный момент, не проводилась разработка и реализация дополнительных стыковочных конструкций.

Программирование робота

Программирование ведётся через специализированную среду Arduino IDE. Эта среда программирования работает с С++, из-за чего освоение программирования не является проблемой. В среде пишется программа в основе которой стоят две функции:

void setup(){}

void loop(){}

Функция setup отвечает за первоначальную настройку системы и её приведения в стартовое положение.

Функция loop является рабочей функцией, то есть все, что написано в этой функции, повторяется до отключения питания или перезапуска системы. Рассмотрим простой пример, так сказать hello world для среды Arduino IDE:

int ledPin = 13; // LED connected to digital pin 13setup() // run once, when the sketch starts {(ledPin, OUTPUT); // sets the digital pin as output

}loop() // run over and over again

{

digitalWrite(ledPin, HIGH); // sets the LED on

delay(1000); // waits for a second

digitalWrite(ledPin, LOW); // sets the LED off

delay(1000); // waits for a second

}

Во время компиляции код преобразуется в программу стандартного вида:

#include "WProgram.h" setup(); loop(); ledPin = 13; setup()

{

pinMode(ledPin, OUTPUT);

}loop()

{

digitalWrite(ledPin, HIGH);

delay(1000);

digitalWrite(ledPin, LOW);

delay(1000);

} main(void)

{

init();

setup();

for (;;) ();

return 0;

}

Что получилось из нашей программы. Появилась стандартная функция main, в которой запускается функция setup, описанная нами первоначально, и бесконечный цикл for (;;) запускает функцию loop. Все #include перемещаются в верх, как и все созданные нами функции. Именно таким образом происходит компилирование программы в среде Arduino IDE.

# **Программа**

## **2.1 Математические расчёты**

У насекомых есть специальные названия для суставов (Рис 9).

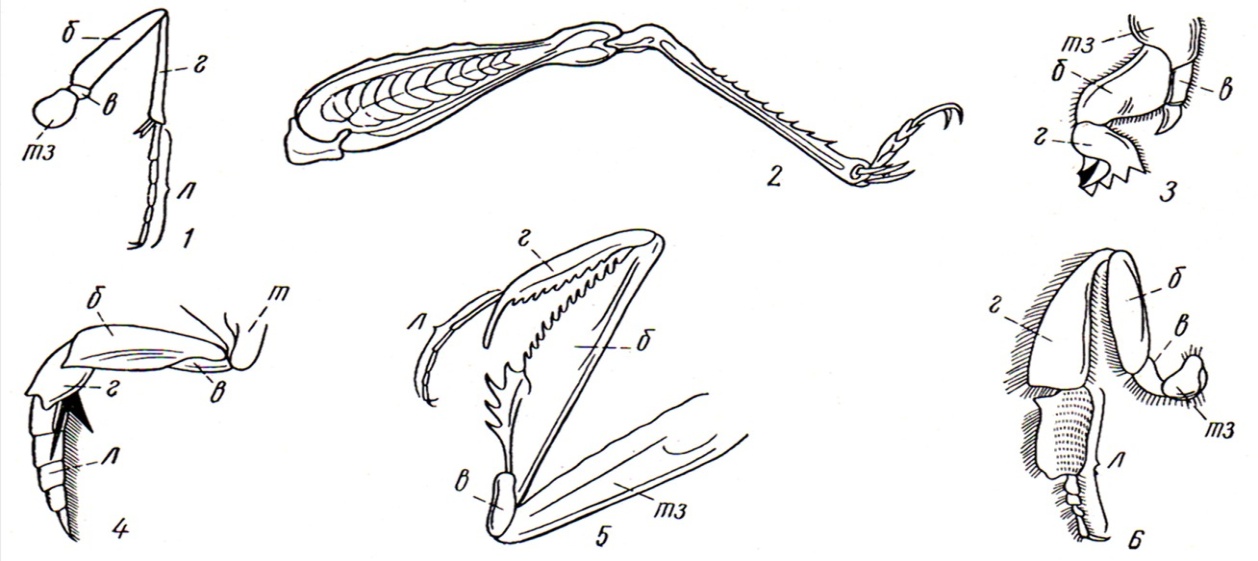
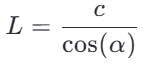


Рисунок 9 - Суставы насекомых

Для простоты понимания и изменения программы, оставим такие же названия для сервоприводов: Coxa, Femur, Tibia соответственно.

Для начала необходимо найти расстояние от проекции точки крепления сервопривода на плоскость земли до точки земли, до которой нужно дотянуться (точка X). Её можно найти по формуле



где L - искомое расстояние, с - максимальное приближение ноги к корпусу, в работе использовалось расстояние равное 10 сантиметрам, α - угол между L. Угол α также является углом поворота ноги относительно корпуса.

Дальше необходимо найти расстояние от точки крепления до точки Х. Его можно найти как гипотенузу треугольника с катетами равными L и катетом, равным желаемой высоте робота, в работе использовалась высота равная 10 сантиметрам. Длинна будет равна, где LL - длинна от точки крепления до точки Х, h - высота робота.

Следующим пунктом будет поиск угла β в треугольнике (Рис. 10).



Рисунок 10 - Поиск угла β

Угол β можно найти по формуле. В последствии угол β пригодится для определения угла поворота сервопривода, отвечающего за Femur.

Далее нужно найти углы в треугольнике со сторонами LL, Femur и Tibia (Рис. 12).

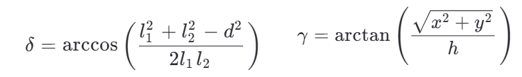


Рисунок 11 - Поиск углов γ и δ.

Таким образом получается, что угол поворота сервопривода, отвечающего за Femur, равен сумме углов β и γ. А угол поворота сервопривода, отвечающего за Tibia, равен δ.

Таблица 1

|  |  |
| --- | --- |
| Параметр | Формула |
| *L* |  |
| *β* |  |
| *γ* |  |
| *d* |  |
| *δ* |  |

## **2. Программная реализация.**

На данном этапе работы, управление пользователя роботом реализовано в виде программы, написанной на языке C#. Передача данных происходит через COM порт, что ограничивает возможности робота по перемещению, максимальная дальность удаления робота от пользователя равна длине кабеля. Клавиши управления роботом:

* W - движение вперёд;
* A - поворот влево;
* S - движение назад;
* D - поворот вправо.

При нажатии клавиш управления, программа посылает флаг выбранной команды на платформу Arduino. Происходит это следующим образом:

1. Для отправки сообщений на платформу, необходимо подключится к ней через COM порт.

void init()

{= new SerialPort();.BaudRate = 9600;.PortName = "COM3";

{.Open();

}

{.Show("Error!");

}

}

Эта функция инициализирует COM порт, задаёт имя и скорость обмена данными, проверяет доступ через порт и если порт закрыт - выдаёт ошибку.

1. Происходит захват нажатия кнопки клавиатуры и отправка соответствующего флага через открытый порт.

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{(e.KeyCode == Keys.W).WriteLine("1");if (e.KeyCode == Keys.A).WriteLine("2");if (e.KeyCode == Keys.S).WriteLine("3");if (e.KeyCode == Keys.D)

Port.WriteLine("4");

}

Функция Form1\_KeyDown отвечает за событие нажатия любой кнопки клавиатуры во время работы с формой, поэтому происходит проверка на

Нажатие конкретных клавиш.

1. Теперь переходим на сторону платформы Arduino. Необходимо считать данные и начать выполнение алгоритмов движения.

void loop() {= Serial.parseInt();(val)

{1: void move ahead();break;2: void move left();break;3: void move back();break;4: void move right();break;

}

}

Функция Serial.parseInt считывает данные из COM порта и предаёт их в переменную val в формате integer. Функции void move выполняют команды пользователя по движению в определённом направлении:

* Ahead - движение прямо.
* Left - поворот влево;
* Back - движение назад;
* Right - поворот направо.

Также существует алгоритм по обходу препятствий, то есть робот посылает через COM порт дальность до препятствия перед ним, и программа принимает решение о том, в какую сторону нужно повернуть для обхода препятствия. Возвращая номер команды управления роботом по СОМ порту.

# **Заключение**

В ходе работы над дипломным проектом были установлены виды конструкций корпуса и варианты соединения деталей робота в единое целое. В работе приведена одна из наиболее удобных конструкций ноги, которая создаёт ощущение реалистичного движения робота, не вызывая лишних действий.

Также, приходилось создавать алгоритмы, вычисляющие углы поворотов сервоприводов, для каждой из создаваемых конструкций, что привело к углубленному изучению библиотек по работе с серво приводами. В связи с тем, что платформа обладает малой вычислительной мощностью и небольшим объёмом памяти, приходилось создавать алгоритмы движения, не требующие больших затрат ресурсов контроллера. (Роман)

Написание программы заставляет работать с одним потоком данных для передачи, что накладывает большие ограничения на создание программы. Таки трудности заставляют изучить и использовать некие уловки для реализации задуманных идей.

Также были получены навыки по работе с СОМ портами и управления периферийными устройствами с его помощью, создания программы для реализации управления через СОМ порты и обработки данных приходящих от контроллера для создания последовательных инструкций и отправки их на контроллер. (Екатерина)

Получены знания в основах тестирования программ и рабочих моделей, проектирование и программирования роботизированных систем, получены знания в осуществлении сборки проектируемого объекта и подготовка рабочей документации.

# **Список литературы**

1. Gorinevsky D. M., Schneider A. Yu. Force control in locomotion of legged vehicles over rigid and soft surfaces // Int. J. Robotics Research. - 1990. - Vol. 9, no. 2. - P. 4-23
2. Охоцимский Д. Е., Голубев Ю. Ф. Механика и управление движением автоматического шагающего аппарата. - М.: Наука, 1984.

* http://hex-bugs.ru/
* http://edurobots.ru/project/arduino-robot-zhuk-ringo/
* http://www.prorobot.ru/09/robot-juk.php
* http://www.poprobot.ru/ideologia/kontroller
* http://robocraft.ru/blog/arduino/1035.html

1. ГурфинкельВ. С., ГурфинкельЕ. В., Девянин Е. А., Ефремов Е. В., Жихарев Д. Н., Ленский А. В., Шнейдер А. Ю., Штильман Л. Г. Макет шестиногого шагающего аппарата с супервизорным управлением // Исследование робототехнических систем. - М.: Наука, 1982. - С. 98-147.