

Роботизированный манипулятор с удаленным управлением с контроллера

Работа ученика школы №1532:

Воронова Никиты Рустамовича

Руководитель проекта:

Кузьмин Михаил Игоревич

Введение

В основе разработки роботизированных устройств всегда лежит один принцип – облегчить какую-то рутинную задачу. Ярким примером этого принципа являются умные пылесосы. Они созданы с целью автоматизации и упрощения процесса уборки в доме. Эти роботы способны самостоятельно перемещаться по помещению, обнаруживать загрязнения и эффективно очищать поверхности. Такие устройства значительно снижают необходимость ручной уборки, освобождая время для более важных и интересных занятий.

Но мой проект не о роботе пылесосе. Есть вещи, которые людям сложно выполнять в каких-то местах. Обычно для таких целей создают роботов, которые могли бы в этом помочь. Роботизированный манипулятор – это один из таких роботов.

Суть манипулятора в том, чтобы в зависимости от общего положения всех своих двигателей выдавать определенную форму или позу, из которой в последствии можно выбрать необходимые действия.

О проекте

Проект разработан с целью обеспечить безопасное взаимодействие человека с разными объектами.

Манипулятор по своему строению похож на человеческую руку, с некоторыми модификациями. Подобно тому, как человек управляет своей рукой можно добиться того, как человек управляет роботом.



Рис.1 Пример робота-манипулятора

Вся система состоит из двух частей: роботизированный манипулятор и удалённый контроллер управления. Обмен данными происходит по радиоканалу с использованием модулей *NRF24L01*. Канал достаточно

настроить один раз на обоих устройствах. Конечно, куда проще было бы настроить автоматический поиск свободных радиоканалов, но в таком случае есть риск попасть в чужой радиоканал, который находился в режиме ожидания. Таким образом возрастают риски того, что управление манипулятором может быть опасным или бессмысленным.

В основе идеи лежит сделать подобие роботизированного помощника, подобно тому, которого нам показали в фильме «Железный человек»



Рис.2 Пример манипулятора из фильма

Разумеется, что повторить все функции робота из фильма не получится, но сама идея вполне реализуема.

Подробнее с файлами работы, готовым кодом и примерами использования можно ознакомиться по ссылке на репозиторий GitHub:

<https://github.com/voronov-nikita/arm>

Принцип работы

Как уже было сказано ранее, вся система состоит из ручного контроллера – перчатки и самого роботизированного манипулятора. Обе части работают на базе микроконтроллеров Arduino с различными усилителями. Рассмотрим принцип работы каждого из них по отдельности.

Перчатка – это специальный контроллер, который собирает данные о положении руки человека и отправляет соответствующие данные по радиоканалу. Данные, которыми обмениваются контроллер и манипулятор – это 3 угла отклонения (*roll*, *pitch*, *throttle*), 4 показателя от датчиков холла, положение перчатки по вертикальной оси и положение перчатки по оси

вращения yaw . NRF модули свободно способны обмениваться большими данными в формате массивов. Таким образом для массива достаточно использовать только целые числа, не выходящие за пределы 16 битного представления числа. Следовательно, по формуле поиска общего объема информации:

$$I = k * i$$

Можно получить такие параметры, где k – это количество информации, а i – размер каждого элемента этой информации. В нашем случае k – это размер массива, а i – это размер целых чисел в языке программирования C++, соответственно от 16 до 128 бит информации. Как раз размер массива уместит в себя 9 значений, необходимых для изменения движения манипулятора. А размер каждого должен уместиться в 16-битную разрядность двоичной системы счисления. Соответственно общий объем информации это $16_{\text{бит}} * 9 = 144$ бита. Скорость передачи данных у NRF24L01 в районе 0,25Мбайт/секунду. Переводя в обратные единицы, получим, что наш пакет данных дойдет и вернется за ~0,41 секунд. Разумеется, что это скорость при идеальных условиях, а в нашем мире легко получить помехи в соединении. Поэтому, с условиями погрешности, была взята скорость в диапазоне от 0 до 1 секунды. Этого всего уже достаточно, чтобы обеспечить практически бесперебойную связь на расстоянии до 2-х километров.

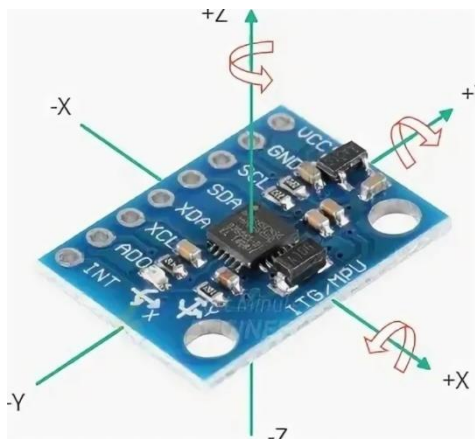


Рис.3 mpu6050

Для получения положения руки управляющего используется модуль гироскоп-акселерометра MPU6050 с Arduino совместимым стабилизатором.

Для симметричных тел, вращающихся вокруг главной оси, момент импульса (L) равен произведению момента инерции тела (I_0) на его угловую скорость вращения (ω_0). Это выражается уравнением:

$$L = I_0 * \omega_0$$

Момент инерции (I) зависит от массы и распределения массы относительно оси вращения. Вектор момента импульса (L) определяет точность работы гироскопических приборов: чем он больше, тем устойчивее и точнее функционируют эти приборы. Но при таком раскладе мало просто использовать некоторые значения, вычисляемые из этой формулы, нам

необходимо сделать что-то вроде преобразователя. Используя данные гироскопа для оценки угла, легко можно найти угол отклонения (θ), интегрируя угловую скорость (ω):

$$\theta(t) = \theta_0 + \int_{t_0}^t \omega(\tau) \Delta \tau$$

Однако, данная формула подвержена ошибкам из-за *дрейфа гироскопа*, и поэтому для более точных результатов часто применяют фильтры, такие как фильтр Калмана, для коррекции данных.

Фильтр Кальмана начинает с предсказания текущего состояния системы, основываясь на предыдущих измерениях и знании о том, как система обычно изменяется. Например, если мы измеряем угловую скорость (как быстро что-то вращается), мы можем предсказать, на сколько изменится угол поворота за следующий короткий период времени. Затем фильтр Кальмана принимает новое измерение текущего состояния системы. Но измерения часто содержат шумы или ошибки. Фильтр сравнивает предсказание с фактическим измерением и определяет, насколько они различаются. Затем фильтр учитывает, насколько ему можно доверять предсказанию и насколько измерение подвержено шумам. На основе этих сравнений фильтр корректирует предсказание, чтобы получить более точное состояние системы. Важно отметить, что фильтр Кальмана учитывает как измерения, так и динамические свойства системы. Он стремится минимизировать ошибки оценки состояния, учитывая неопределенность в данных и модели системы. Это позволяет ему быть эффективным инструментом для оценки состояния в реальном времени в различных областях, таких как навигация, управление роботами и другие приложения.

Для отслеживания режимов работы и изменении каких-то прочих параметров используются цифровые датчики холла, расположенные в районе 4-х пальцев. Всего из 4-х датчиков формируются специальные команды для перчатки. Для активации датчика необходимо поднести выбранный датчик к большому пальцу руки, именно там располагается магнит для активации датчика. К примеру, если замкнуть перчатку на указательный и большой палец, то данные для манипулятора перестанут передаваться, манипулятор застынет на одном месте, а человек сможет передохнуть. Если замкнуть средний и большой пальцы, то можно вызвать команду закрытия или открытия руки манипулятора. Это должно облегчить понимание структуры и процедуры открытия и закрытия манипулятора. Если же замкнуть безымянный и большой палец, то можно подать команду манипулятору

«Вернуться в начальное положение». Это облегчит процесс выключения робота и подготовку к транспортировке. Если замкнуть большой палец и мизинец, то можно вызвать работу, заранее записанного алгоритма действий. Это делает манипулятор более автономным и независимым от пилота. При повторном замыкании автоматический алгоритм просто приостановит работу.

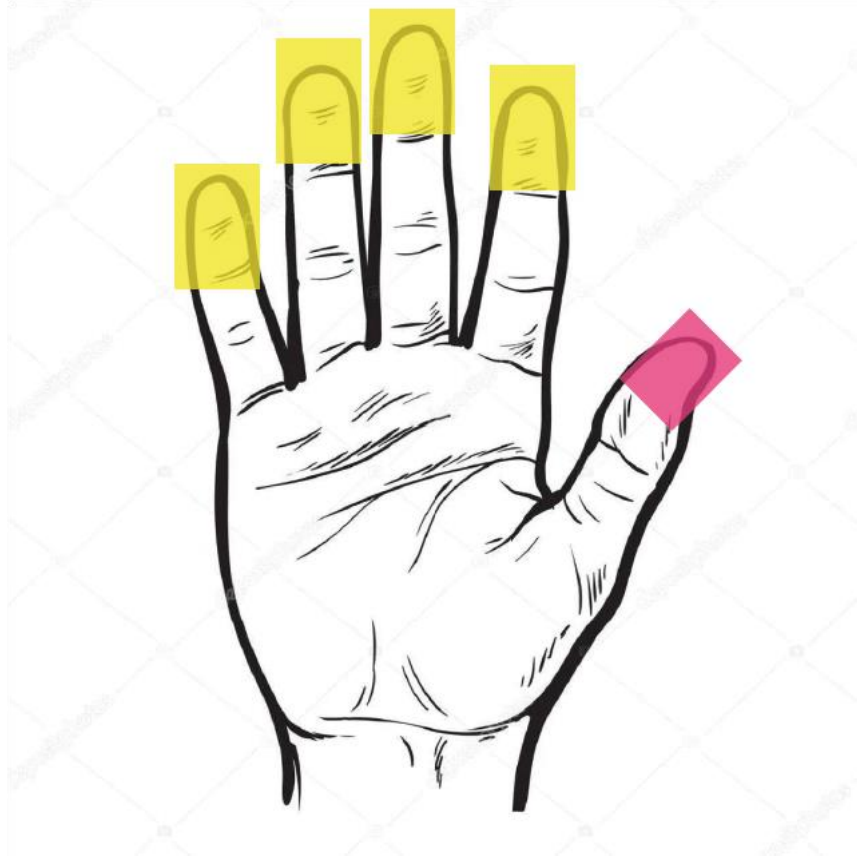


Рис.4 Абстрактное представление расположения датчиков и магнита в перчатке

Схема перчатки довольно простая, но со своими особенностями. Так как устройство должно быть автономным, необходимо предусмотреть возможность питать микроконтроллер от аккумулятора. Но важно помнить, что и аккумулятор нужно периодически заряжать. Для этого момента был взят специальный разъем для одновременной зарядки аккумулятора и питания микроконтроллера Arduino. Таким образом зарядить перчатку можно от обычной зарядки для телефона. Перчатка способна работать от аккумулятора без перерыва на протяжении около 6 часов. Все дело в том, что сама Arduino требует не так много питания и несмотря на то, что у нее присутствуют тяжелые 5V датчики, требующие постоянного питания, это не



Рис. 5 Разъем питания microusb

мешает перчатке работать в экономном режиме. Это делает конструкцию меньше и проще в разработке, однако это усложняет логические цепочки при программировании контроллера.