

6-осевой робот-манипулятор с компьютерным зрением

Авторы: Котенкова Анастасия, Зарипов Алексей, Гатауллин Мансур

Руководитель: Трофимов Александр Артемович

МБУДО «Городской центр детского технического
творчества им. В.П. Чкалова» г. Казани

О проекте

- Мы выбрали этот проект, так как в последнее время начала активно развиваться сфера промышленной робототехники. Настоящие промышленные роботы, даже обучающие, очень дорогие, по этой причине мы решили создать своего промышленного робота.
- Наверно еще не было команды, которая второй год подряд заявляется с одним и тем же проектом, но тут надо пояснить...
- Хотя основа робота осталась прежней, многое изменилось. Был заново написан код и основной задачей было изучение компьютерного зрения.

Ретроспектива

или как это было

Конструкция робота

- Мы немного поработали над конструкцией робота:
 - Мы программно перенесли зоны расположения элементов.
 - Убрали беспаячную плату, заменив ее разъемами для присоединения модулей.
 - Но самое главное – мы перешли на компьютерное зрение используя Raspberry Pi с камерой вместо датчика GY-31.

Сервоприводы Dynamixel AX-12A

- Они обладают следующими характеристиками:
 - Измерения угла поворота с точностью 0.29 градуса.
 - Сила 1,5Нм
 - Питание от 12 вольт
 - Управление по Half-Duplex UART (UART на одном проводе)
 - Имеется обратная связь, выдающая скорость и угол поворота, нагрузку, температуру, напряжение питания.
- Обратная связь позволяет узнать актуальное положение сервоприводов в любой нужный нам момент, что облегчает разработку анимаций робота.

Интерфейсы робота

- В прошлом году использовались два интерфейса:
 - UART1 – для управления сервоприводами
 - UART2 – для расширений робота
- Сейчас робот использует 4 интерфейса:
 - UART0 – для связи с Raspberry Pi
 - UART1 – для управления сервоприводами
 - UART2 – для расширений робота
 - UART3 – для управления роботом по Bluetooth
 - I2C используется для вывода информации на экран
- Также к роботу присоединена панель управления, хоть она и не работает в данный момент полностью.

Программный код

- Arduino IDE достаточно проста, но в то же время она сильно урезана по функционалу и удобству, по этому мы перешли на VS Code с расширением PlatformIO IDE.
- VS Code – мощный инструмент разработки от Microsoft, это сильно облегченная и кроссплатформенная Visual Studio, которая имеет полный функционал «старшего брата» благодаря расширениям.
- PlatformIO IDE – платформа для создания проектов на микроконтроллерах разных типов, в том числе под фреймворком Arduino. Теперь тут нет «.ino» файлов, весь код рассован в «.h» и «.cpp» файлы, как будто фреймворк Arduino – просто библиотека.
- Мы улучшили код, отвечающий за управление сервоприводами и связью с внешней средой.

Компьютерное зрение

или как мы учили python

GY-31

- В прошлом году для определения цвета объекта мы использовали модуль GY-31, который представляет из себя камеру всего на несколько пикселей и простую схему обработки изображения с нее, выдавая на выходе 3 параметра цвета, а именно: интенсивности красного, зеленого и синего цветов.
- Датчик GY-31 очень сильно зависит от освещения и правильного угла попадания света с подсветки.
- Также этот датчик не способен распознавать форму предмета и уж тем более штрих и QR коды.

Raspberry Pi

- Raspberry Pi – одноплатный микрокомпьютер, который позволяет делать почти все те же вещи, что и большой системный блок, только, естественно, медленнее.
- На Raspberry Pi есть пины GPIO, которые позволяют управлять периферийными устройствами напрямую без дополнительных переходников. Можно как просто помигать светодиодом, так и подключить различные цифровые датчики без дополнительных переходников и микроконтроллеров. Однако мы не использовали эту возможность RPi, от нее мы взяли возможность присоединения камеры и быстрой обработки изображения с нее.

Программа для распознавания

- Программа для распознавания написана на языке python, который является для нас совершенно новым языком программирования. Он не похож на c++, который мы изучали при программировании Arduino. Для нас также была новой операционная система Raspbian и мы изучали как можно взаимодействовать с ней.
- Мы разработали простую систему команд для общения RPi с Arduino по UART.

OpenCV и Pyzbar

- OpenCV – мощная библиотека с открытым исходным кодом для работы с компьютерным зрением. Она позволяет работать с изображением, определять простые формы, если использовать ее поверхностно. С ее помощью можно тренировать нейронные сети для определения сложных образов любого рода.
- Pyzbar тоже является библиотекой для распознавания образов для языка python, однако она строго ориентирована на распознавание штрих и QR кодов.
- Обе эти библиотеки работают в симбиозе в нашем проекте для определения цвета, формы объектов, так же QR и штрих кодов на них.

Обучение

или в чем отличие от роботов KUKA

Стоимость

- Естественно первым же параметром мы рассмотрим стоимость.
- Стоимость обучающих 6-осевых наборов оригинальной KUKA начинается от 2 миллионов рублей.
- Стоимость обучающего набора «Оптима 2» на основе 5-осевого манипулятора от российского производителя «Зарница» менее кусачая – почти пол миллиона рублей.
- Если рассматривать стоимость представленного вам робота, то она будет следующей: 7 сервоприводов по \$50 или по 3000р, резка и гравировка поля из фанеры обойдется еще в 3000р, Arduino Mega стоит 600р, Raspberry Pi обойдется еще в 3000р, если заказывать все из Китая. Если брать российских поставщиков, сервы и RPi по 5000р, Ардуино 1500р. Даже с учетом максимальной стоимости этот робот стоит дешевле 50 тысяч рублей.
- Не надо забывать и тот факт, что сервоприводы у нас уже были, как и в некоторых других учебных центрах, в наборах Robotis Bioloid.

Потенциал обучения готовых наборов

- Роботы KUKA позволяют изучить только язык Kuka Robot Language, который больше похож на G-Code, нежели на обычный язык программирования, это хорошо, если вы учитесь на оператора роботов Kuka, однако недостаточно, если вы хотите изучить полностью работу подобных манипуляторов.
- Робот от Зарницы основан на плате Arduino Mega, что дает уже большой простор для обучения, его можно запрограммировать со стандартной прошивкой на Kuka Robot Language, а можно написать прошивку самим в привычной многим школьникам Arduino IDE. Однако подключение дополнительных внешних модулей корпусом не предусмотрено, как и не предусмотрена сборка этого робота самостоятельно учениками.

Потенциал обучения нашего робота

- Наш робот, в силу отсутствия корпуса как такового, позволяет присоединять к нему всё что угодно, например, RPi с камерой. Помимо этого, конструкцию самого робота определяет сам ученик или его преподаватель, что позволяет собрать не только 6-осевого робота, но и 3 осевого, для начального погружения, или 6 осевого для большей свободы действий и усложнения программы.
- Возможность выбора основы значит, что язык Kuka Robot Language так же может быть затронут в процессе обучения, и не просто на уровне заучивания команд, а на уровне создания транслятора KRL в C++, т.е. разбивки одной KRL команды на внутренние C++ команды, что позволит лучше понимать работу манипулятора и в последствии писать более оптимизированный код на KRL.

Что будет дальше

или обратная кинематика

IKPy

- Raspberry Pi – очень мощное устройство. Программирование на python и использование множества библиотек позволяет делать по настоящему трудные вещи.
- Мы нашли библиотеку для расчёта обратной кинематики под названием IKPy и уже начали ее изучать.

Kuka Robot Language

- Так как за основу робота мы брали промышленный робот-манипулятор, то будет совершенно неправильно программировать нашего робота только написанием прошивки.
- После добавления алгоритма расчета обратной кинематики, следующей задачей станет добавление возможности программирования робота при помощи G-кодов и KRL, так как они работают по принципу «приведи насадку к месту X Y Z под углом A B C», что и является основной задачей обратной кинематики.

Спасибо за внимание