

## ГЛАВА 1

### Разработка ПО

#### 1.1 Формирование алгоритма управления

Сетевая архитектура представлена на Рисунке 1.

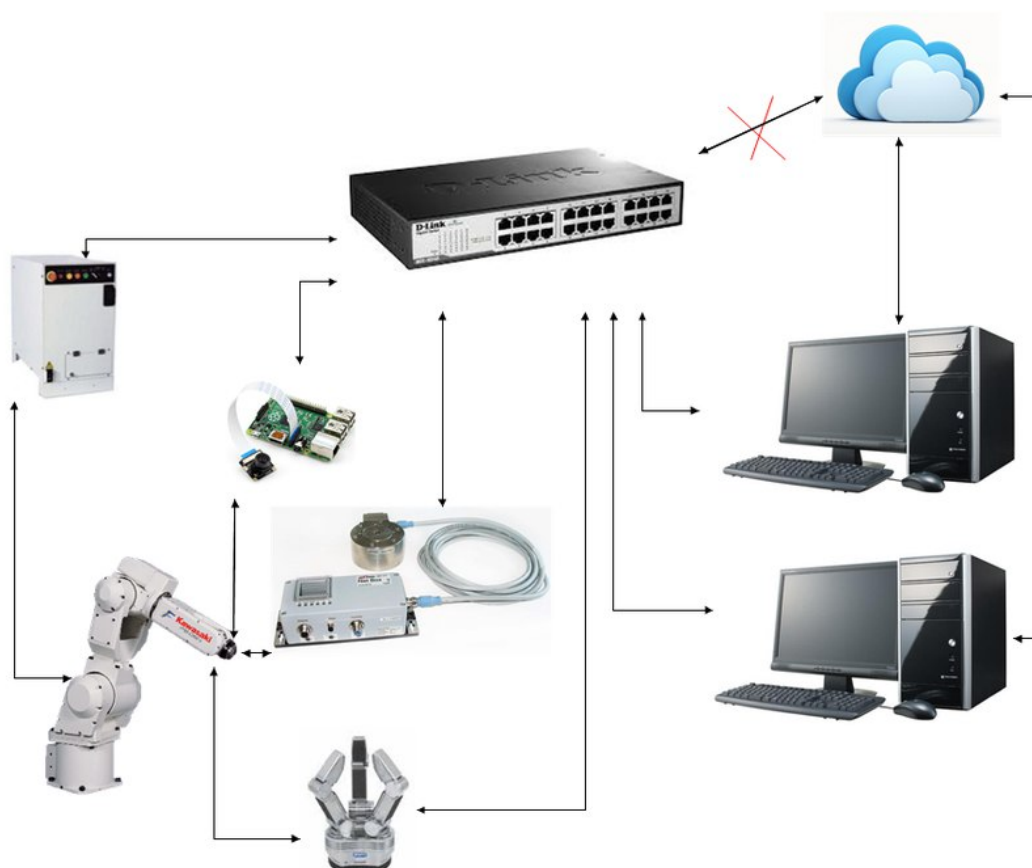


Рисунок 1.1 — Сетевая архитектура

У робота есть всего две группы параметров позиционирования: 1) углы поворотов шести джоинтов 2) Положение фланца, заданное в  $x, y, z$  координатах, и его ориентация, заданная в  $o, a, t$  углах Эйлера в  $zyz$  конвенции.

Т.к. робот получает управляющие команды по протоколу `tcp ip`, и контроль безошибочности передачи заложен в самом протоколе, то нам остаётся разработать собственный формат посылки.

## 1.2 Структура посылки

Т.к. при позиционировании робот использует 6 величин, то будем строить посылку следующим образом.

В силу функциональных особенностей внутреннего языка программирования контроллеров RoboticAS, самым оптимальным способом передачи данных по протоколу TCP IP является отправка символьных строк, содержащих в себе данные, разделённых пробелом (разделительным символом). Единственное требование к такому представлению - это чтобы каждое число было представлено шестью символами. В случае, если число меньше 105, то недостающие разряды должны быть заполнены цифрой 0. Например, число 123 должно быть представлено в выходной строке, как 000123. Минимальная единица посылки, называемая кадром, - это целое число.

Т.к. при использовании протокола tcp ip все символы строки гарантированно будут доставлены от передатчика к приёмнику, то механизм обработки входящих сообщений можно строить следующим образом: все входящие символы добавляются в буфер с конца, как только длина буфера превышает число  $n=k*w$ , где  $k$  - кол-во кадров в посылке,  $w$  - размер кадра в символах, то из буфера берутся первые  $n$  символов и переводятся в  $k$  целых чисел.

Т.к. параметров позиционирования у робота-манипулятора шесть, то в состав посылки заложим 9 кадров.

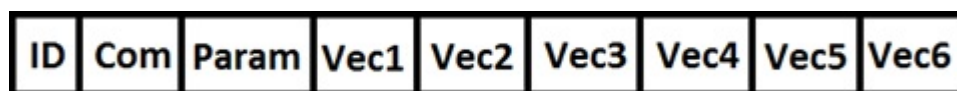


Рисунок 1.2 — **Структура посылки:** ID- уникальный номер управляющей команды(id); Com - номер управляющей команды; Param- параметр управляющей команды(запасной); Vec1...Vec6 - вектор параметров управляющей команды

## 1.3 Управление Кавасаки

Всего есть три основных группы управления:

- а) По положению. Вектор параметров имеет следующую структуру:
  - Vec1 – X координата фланца
  - Vec2 – Y координата фланца
  - Vec3 – Z координата фланца
  - Vec4 – O угол Эйлера
  - Vec5 – A угол Эйлера
  - Vec6 – T угол Эйлера
- б) По абсолютным углам поворота джоинтов:  $i$ -ая координата отвечает за поворот  $i$ -го джоинта.
- в) По относительным углам поворота джоинтов:  $i$ -ая координата отвечает за поворот  $i$ -го джоинта.

Внутренняя программа контролера построена следующим образом: запущено 3 параллельных процесса: первый в бесконечном цикле отправляет данные о положении фланца, его ориентации и углах поворота шести джоинтов. Второй процесс в бесконечном цикле проверяет, не пришло ли новых символов и по каналу связи и добавляет их к буферу. Третий процесс в бесконечном цикле обрабатывает входящие управляющие команды и запускает вспомогательные программы внутреннего управления, если такое управление было вызвано.

#### **1.4 Внешние управляющие программы.**

- а) Вывод робота в заданную точку по координатам (X,Y,Z,O,A,T). В вектор параметров передаются соответствующие координаты.
- б) Вывод робота в заданную конфигурацию поворотов джоинтов. В вектор параметров передаются соответствующие углы поворота джоинтов.
- в) Вывод робота в заданную конфигурацию относительных поворотов джоинтов. В вектор параметров передаются соответствующие углы поворотов джоинтов. При получении любой из первых трёх команд робот сразу же запускает внутреннюю команду перемещения. Поле Param остаётся нулевым.

- г) Передача значений показаний силомоментного датчика в базовой системе координат (преобразование реальных показаний датчиков производится на стороне РС). Поле Param остаётся нулевым.
- д) Задание скоростей и ускорений. Поле Param остаётся нулевым, значения  $vec1$ - $vec4$  следующие: ускорение в процентах, замедление в процентах, модульная скорость в процентах, механическая скорость в мм/с. Реальная скорость робота является произведением механической на модульную скорости.
- е) Задание коэффициентов регулирования. При поле Param, равном нулю задаются пропорциональные коэффициенты  $kx, ky, kz, kmx, kmy, kmz$ , при поле Param, равном 1 задаются интегральные коэффициенты  $ix, iy, iz, imx, imy, Imz$ .
- ж) Запуск программы force feedback. При Param, равном 0 – запускает режим стабилизации показаний  $x, y, z$  компонент силомоментного датчика в нулевых значениях за счёт смещений по  $x, y, z$  осям. ForceFeedBackPosition. При Param равном 1 – запускается режим стабилизации  $mx, my, mz$  показаний силомоментного датчика в нулевых значениях. ForceFeedBackOrientation. При Param равном 2 останавливается выполнение force feedback.

## 1.5 Работа Force Feedback

Из-за низкой производительности контроллера Kawasaki все математические вычисления производятся на ПК. Контроллер получает уже преобразованные значения  $x, y, z$  компонент силы в базисе базовой системы координат.  $Mx, My$  и  $Mz$  – моменты, возникающие вдоль осей  $x, y$  и  $z$  базовой системы координат. Force feedback Position реализует при помощи ПИ регулятора, где за ошибку берётся значение  $x, y, z$  компонент силомоментного датчика. Выходом регулятора является новое положение робота на следующей итерации главного цикла программы.

Force FeedBack Orientation планируется организовать более сложным способом. Также строится ПИ регулятор, за ошибку берутся показания мо-

ментов  $M_x$ ,  $M_y$ ,  $M_z$ , На выходе мы получаем углы поворота относительно осей  $x, y, z$  базовой системы координат.

Итого, мы имеем 3 задачи:

- а) Преобразование значения силы из системы координат датчика в базовую систему координат и компенсация силы тяжести инструмента.
- б) Преобразование значений моментов из системы координат датчика в базовую систему координат и компенсация момента силы тяжести инструмента.
- в) Поворот инструмента вдоль ортов базовой системы координат.