

**КУРСОВОЙ ПРОЕКТ**

**по МДК 01.01 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**специальности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**на тему:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил студент

группы 4ИС3010

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, инициалы)

Проверил преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, инициалы)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

Москва, 2022

Содержание

Содержание 2

Введение 3

1 Теоретическая часть 4

1.1 Описание предметной области 4

1.2 Постановка задачи 6

2 Практическая часть 8

2.1 Описание сценариев использования 8

2.2 Проектирование модели данных 8

2.3 Проектирование интерфейса 8

2.4 Разработка программы 8

2.5 Тестирование программы 8

Заключение 9

Список использованных источников 10

Приложения 11

Введение

В древности производство было традиционным и зависело только от ручного труда. С течением времени каждый вид деятельности человека стал приобретать черты автоматизации процесса производства, что привело к появлению механизмов, которые научились обрабатывать информацию и на основе этой информации делать определенные действия.

Данная тема актуальна тем, что такие механизмы могут обладать малой вычислительной мощностью, поэтому не могут быть обработаны все возможные форматы информации, которые потребуются программисту для реализации того или иного алгоритма действий или предоставлять только необработанную информацию, что сильно замедляет процесс разработки программного продукта.

Практическая значимость заключается в использовании программного продукта другими программистами для разработки собственного программного обеспечения, которое позволит им управлять данными механизмами.

1. Теоретическая часть
   1. Описание предметной области

Робот – это автоматическое устройство, предназначенное для осуществления различного рода механических или математических операций, которое действует по заранее заложенным в него действиям. Каждый из роботов может выполнять несколько задач сразу. Обычно он получает информацию о состоянии окружающего пространства посредством датчиков, установленных отдельно или встроенных в него, а также может получать команды от человека через специальное программное обеспечение.

Датчики могут собирать различную информацию, передающейся в текстовом виде и обрабатывающейся специальной комбинацией инструкций, такая комбинация инструкций, выполняемых последовательно, именуется программой.

Каждый робот требует своей собственной уникальной программой или полноценным программным обеспечением для совершения действий, в которое может входить визуальный интерфейс.

Роботы так же могут соединяться между собой в одну сеть, такой сетью в современном мире может выступать конвейерная лента, площадка для взаимодействия механизмов.

В качестве объекта будет выступать площадка, включающая в себя две четырёхцветные сигнальные лампы, робот-политайзер и робот-манипулятор, а также удаленный терминал для управления площадкой.

Четырёхцветная сигнальная лампа – это устройство для отображения фазы выполнения роботом поставленной задачи. Лампа содержит в себе цвета: зеленый, синий, оранжевый и красный и подключается к площадке различными способами по сети. Возможными способами может быть WIFI сеть или проводное соединение. Данная лампа может только принимать четыре разных значения, смена который определит включен или выключен определенный цвет лампы. Цвета лампы могут отображать не только фазу выполнения задачи роботом, но и могут быть произвольно включены или перепрограммированы.

Удаленный терминал – это устройство для управления и отправки данных на роботов. Он содержит четыре лампы для отображения состояния фазы выполнения роботом поставленной задачи, дисплей для отображения IP адресов, нужных для подключения к нему, и произвольной информации, которую может задать программист или оператор платформы, три кнопки: зеленую, красную и желтую, для использования их в непосредственном удаленном управлении любым элементом на площадке, а так же рычаг, выполняющий функции “зажатой кнопки” и джойстик для непосредственного управления вращением робота при прямом подключении терминала к нему. Данное устройство может рассылать данные в сеть из которых можно получить IP адрес устройства, состояние рычага и количество нажатий на красную, зеленую или желтую кнопку за время работы устройства. Для управления лампами на терминале может быть принята текстовая информация. Дисплей и джойстик не обладают принимаемой или отправляемой информацией.

Робот-политайзер – это устройство для совершения механических действий. Данный тип робота обладает манипулятором с предметами – вакуумной присоской, но может иметь и другие виды инструментов, к примеру карандаш, и может вращаться на 360 градусов. Он может рассылать данные о нагреве своих 6 двигателей, расположенных на подвижных местах робота, состоянии вакуумной присоски, а также может принимать данные для управления по декартовой системе координат и включения или выключения вакуумной присоски.

Робот-манипулятор – это устройство для совершения механических действий. Этот тип роботов обладает манипулятором с предметами, который захватывает предмет и может его перемещать. Обладает пятью двигателями на местах смещения роботов и постоянно рассылает данные о их состоянии в которых так же входит температура, управляется по декартовой системе координат.

Для взаимодействия с роботами выше используется протокол сети - User Datagram Protocol (UDP). Данный протокол уникален тем, что данные приходят быстрее, чем в остальных случаях за счет отсутствия проверки на целостность данных. Такой протокол может быть использован для реализации систем, не требующих получения данных в последовательном порядке, а перезаписывающий их. Каждое из перечисленных устройств использует этот протокол и отправляет данные в сеть раз в несколько миллисекунд.

Чтобы обрабатывать данные приходящие по этому протоколу программисты разрабатывают специальные интерфейсы для межпрограммного обмена, которые называются API.

API – интерфейс для упрощения создания программ, который может позволяет программистам уменьшать время разработки программы за счет предоставления разработчикам программы специальных, подготовленных заранее, способов общения с программой на более низком уровне. Помимо использования для ускорения времени разработки оно так же может быть использовано для доступа к каким-то данных напрямую.

* 1. Постановка задачи

Для упрощения взаимодействия с роботами необходимо реализовать API в формате библиотеки для обработки данных роботами. Каждый пользовательский метод библиотеки должен быть публичным для вызова API из других программ.

Для робота-политайзера нужно реализовать методы:

* Получения информации о данных с моторов массивом.
* Получение информации о данных с моторов по номеру мотора.
* Установки положения робота.
* Установка положения робота изменяя только одну координату.
* Получение информации о текущих координатах робота.
* Получение информации о состоянии захвата робота.

Для робота-манипулятора нужно реализовать методы:

* Получения информации о данных с моторов массивом.
* Получения информации о данных с моторов по номеру мотора.
* Установки положения робота.
* Установки положения робота изменяя только одну координату.
* Получение информации о текущих координатах робота.
* Получение информации о состоянии захвата робота.

Для сигнальной четырехцветной лампы нужно реализовать методы:

* Установки состояния цветов лампы отдельно.
* Установки методов ламп по отдельности.
* Получения текущего состояния цветы лампы.
* Получение текущего состояния цветов лампы массивом.
  + 1. Выбор технологии реализации

Для реализации поставленной задачи необходимо выбрать один или несколько языков программирования. Язык программирования – это формальный язык для записи компьютерных программ. Язык программирования определяет набор конкретный набор инструкций для общения с компьютером.

Чтобы обеспечить связь между API и роботами понадобится так же локальная вычислительная сеть. Локальная вычислительная сеть – это компьютерная сеть, которая обычно покрывает небольшую территорию, например, офис компании или небольшую группу зданий.

Чтобы получать данные с роботов нужно собирать данные из локальной вычислительной сети. Каждое устройство в площадке работает таким образом, что данные с устройства можно получить только после настройки устройств на определенный IP адрес отправки.

IP адрес – это уникальный адрес устройства в сети. IP означает «Интернет-протокол» - набор правил, который регулирует формат данных, отправляемый через интернет или локальную вычислительную сеть.

Соответственно, каждое устройство в площадке может быть идентифицировано по его IP адресу, а также после анализа данных, приходящих с него. Это означает, что для конкретной работы API должен иметь данные с каких IP адресов собирается информация и на какие адреса отправляется информация.

Данный проект должен быть напрямую связан со повышенной скоростью обработки данных, так как роботы отсылают данные по протоколу UDP для наиболее быстрой обработки будет использоваться объектно-ориентированное программирование на языке С++.

С++ является компилируемым, что означает собираемым в машинный код при сборке программы, а не при выполнении, статически типизированным, подразумевая наличие строгих правил при использовании, языком программирования общего назначения, что позволяет ему быть безопасным и универсальным для разработки любого программного обеспечения.

Чтобы качественно выполнить поставленную задачу каждый из методов должен быть прикреплен к устройству, созданному виртуально – специальным объектом, названным классом.

Класс в программировании означает шаблон для создания однотипных объектов в котором могут содержаться методы, свойства объекта, которые позволяют быстро получить к информации.

1. Практическая часть
   1. Описание сценариев использования
   2. Проектирование модели данных
   3. Проектирование интерфейса
   4. Разработка программы
   5. Тестирование программы

Заключение

Список использованных источников

Приложения