

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА СИСТЕМЫ ИНСТРУМЕНТООБЕСПЕЧЕНИЯ ОБРАБАТЫВАЮЩЕГО ЦЕНТРА НА ОСНОВЕ МОБИЛЬНОГО РОБОТА

К. В. Козаченко

*Учреждение образования «Гомельский государственный технический
университет имени П. О. Сухого», Республика Беларусь*

Научный руководитель М. И. Михайлов

Разработан алгоритм прохождения микросигналов в системе управления мобильным роботом. На основании разработанного алгоритма составлена программа с использованием языков программирования C++ и Python, позволяющих работать роботу в автономном режиме.

Ключевые слова: система инструментообеспечения, мобильные роботы, программирование.

Целью данной работы является разработка программного комплекса для робототехнического комплекса (РТК) системы инструментообеспечения. В работе решаются следующие задачи: разработка многоуровневой архитектуры программного обеспечения на базе ROS 2, обоснование выбора языков программирования для каждого уровня, описание взаимодействия узлов и алгоритма смены инструмента.

При разработке программного комплекса для РТК ключевым решением является выбор языков программирования, так как система включает задачи, требующие высокой производительности и работы в реальном времени, а также логические задачи с жесткими временными ограничениями. В проекте используются два языка: C++ и Python. C++ применяется для компонентов, где критичны скорость выполнения и детерминизм: драйверы приводов колес, вычисляющие управляющие сигналы для сервомоторов; узлы обработки данных с лидара (построение карты, локализация, обнаружение препятствий); захват и обработка видеопотока с камеры через OpenCV; драйверы UWB-приемника, где важна точная временная привязка измерений. Дополнительным преимуществом C++ является возможность использования реального времени в ROS 2 с соответствующими настройками ядра Linux.

Python выбирается для задач, где приоритетны гибкость разработки и скорость написания кода: верхний уровень архитектуры, взаимодействующий с системой управления предприятием, ведущий базу данных инструментов и принимающий заявки от станков; алгоритмы управления процессом смены инструмента с реализацией конечного автомата; драйверы простых дискретных датчиков (индуктивных, оптических, ультразвуковых), частота опроса которых невелика; вспомогательные узлы мониторинга, логирования и визуализации. Взаимодействие между узлами на разных языках осуществляется стандартными механизмами ROS 2.

Программный комплекс разрабатывается в среде ROS 2, обеспечивающей модульность, поддержку распределенной работы, наличие готовых пакетов для навигации, SLAM и управления манипуляторами. Предлагается трехуровневая архитектура.

Верхний уровень (стратегическое планирование) отвечает за глобальную координацию РТК в производственной системе. Реализованный на Python, он взаимодействует с системой управления предприятием, принимает заявки на смену инструмента от обрабатывающего центра, планирует маршрут с учетом заряда батареи и наличия инструментов в накопителе, ведет базу данных инструментов.

Средний уровень (управление задачами и навигация) координирует исполнительные узлы. Навигационная подсистема, получающая от верхнего уровня целевые точки, имеет критичные по времени части (обработка облака точек с лидара, локальное планирование траектории), реализованные на C++. Подсистема управления задачами на Python реализует алгоритм смены инструмента через конечный автомат; каждое действие оформлено как ROS-действие для отслеживания хода и прерывания.

Нижний уровень (драйверы и исполнительные узлы) обеспечивает непосредственное управление аппаратурой. На C++ реализованы драйверы приводов колес (команды скорости в сигналы для сервомоторов с обратной связью по энкодерам), драйвер манипулятора (перемещение схвата в заданную точку), драйверы лидара, камеры и UWB-приемника. На Python реализованы драйверы дискретных датчиков (контроль закрытия схвата, наличие инструмента в ячейках, ультразвуковые).

Для точного позиционирования РТК используется комбинированный подход по зонам. На складе инструментов, где нужно приблизительное определение координат, применяется система абсолютного позиционирования на основе UWB-маяков. В углах помещения устанавливаются стационарные маяки, на борту РТК – приемник. Измеряя расстояния методом Time of Flight, система вычисляет координаты с погрешностью 10–15 см, что допустимо для грубой навигации. Обработка UWB-данных реализована на C++.

При движении от склада к станку UWB не используется – применяется метод SLAM на основе данных лидара (частота – 20 Гц). Алгоритм SLAM на C++ с библиотекой PCL одновременно строит карту и оценивает положение РТК. Камера на борту РТК работает постоянно для распознавания препятствий, а ее видеопоток обрабатывается на C++ с использованием OpenCV.

В работе предложена трехуровневая архитектура программного комплекса РТК для автоматизации инструментообеспечения обрабатывающего центра. Обосновано распределение языков: C++ – для производительных и real-time задач (драйверы приводов, обработка лидара, локальное планирование, захват видеопотока); Python – для логических задач, работы с базами данных, алгоритмов управления и драйверов простых датчиков. Уровни взаимодействуют через топики, сервисы и действия ROS 2. Разработанный комплекс может быть реализован на базе ROS 2.

Литература

1. Ху, Чуньсюй. ROS 2. Программирование интеллектуальных роботов : пер. с англ. / Ху Чуньсюй, Ли Цяолун. – М. : ДМК-Пресс, 2025. – 544 с.
2. Бжихатлов, И. А. Операционная система для роботов ROS : учеб. пособие / И. А. Бжихатлов. – СПб. : Университет ИТМО, 2025. – Ч. 1. – 109 с.
3. Лавренов, Р. О. Основы ROS1 и ROS2 : учеб.-метод. пособие / Р. О. Лавренов, Е. А. Магид. – Казань : Изд-во Казан. ун-та, 2024. – 43 с.