**Языки программирования и технологии**

**Python**

Причины использования:

* Высокая скорость разработки, благодаря простоте синтаксиса.
* Наличие большого количества библиотек для анализа данных, которые могут быть полезны для обработки данных о производительности.
* Возможность быстрой интеграции с другими языками и библиотеками, что позволяет комбинировать Python с C/C++ для достижения оптимальной производительности.

**C/C++**

Причины использования:

* Высокая производительность и возможность низкоуровневого программирования, что позволяет эффективно управлять аппаратным обеспечением машин.
* Поддержка работы с реальным временем, что критично при управлении технологическими процессами.
* Широкое использование в промышленности и наличие обширных библиотек для работы с различными протоколами и устройствами.

**Java**

Причины использования:

* Платформенная независимость: можно разрабатывать приложения, которые будут работать на различных операционных системах без необходимости перекомпиляции.
* Широко применяется в крупных корпоративных системах, что позволяет легко интегрироваться в уже существующие бизнес-процессы и ERP-системы.
* Наличие мощных инструментов для разработки интерфейсов (JavaFX, Swing).

**Технологии:**

1. Операционные системы

* Встраиваемые системы: можно рассмотреть использование встраиваемых операционных систем (например, FreeRTOS), если нужна высокая производительность и надежность.
* Linux: Подходит для разработки более сложных систем, обеспечивая гибкость и мощные инструменты для разработки.

2. Технологии сборки данных

* IoT-платформы: использование Internet of Things (IoT) для сбора данных с датчиков в реальном времени. Платформы, такие как AWS IoT или Azure IoT, могут помочь в этом.
* Базы данных: реляционные или NoSQL решения для хранения и обработки данных.

3. Программное обеспечение для управления

* SCADA-системы: можно использовать Supervisory Control and Data Acquisition (SCADA) для мониторинга и управления процессами в реальном времени.
* PLC (Программируемые логические контроллеры): Интеграция с PLC для управления механическими частями машины.

4. Программирование интерфейсов

* Графические интерфейсы пользователя: можно использовать технологии, такие как Qt или WPF (Windows Presentation Foundation), для создания интуитивных интерфейсов для операторов.

5. Обеспечение безопасности

* Шифрование данных: использование SSL/TLS для защиты данных, передаваемых между устройствами.
* Безопасность сети: Firewall и VPN для защиты системы от внешних угроз.