

# Разработка специализированного терминала для системы ЧПУ на базе мобильных компьютеров<sup>1</sup>

М.М. Сорокоумова,  
магистрант, [mspromargo@gmail.com](mailto:mspromargo@gmail.com),  
Н.В. Козак,  
к.т.н., доц., [kozak@ncsystems.ru](mailto:kozak@ncsystems.ru),  
МГТУ «СТАНКИН», г. Москва

Работа посвящена актуальной практической задаче разработке конфигурируемого пользовательского интерфейса для системы числового программного управления (ЧПУ) на основе мобильного приложения с помощью кросс-платформенной среды разработки приложений Qt-QML. Решение этой задачи позволит повысить мобильность операторов станочного оборудования. Разработан механизм конфигурации пользовательского интерфейса на основе XML-данных. Представлено практическое использование мобильного терминала для управления роботом-манипулятором.

The work is devoted to the urgent practical task of designing a configurable user interface for the computer numerical control (CNC) system- based mobile apps using cross-platform application development environment in Qt-QML. The solution of this problem will allow to increase the mobility of operators of machine tools. The mechanism of the configuration user interface (UI) based on the XML data. The practical use of the mobile terminal to control a robot manipulator.

При выполнении работ по запуску и наладке крупногабаритных станков инженеры-операторы сталкиваются с множеством проблем. Одной из таких проблем является мобильность и доступ к функциям управления оборудованием станка. Терминал управления системы числового программного управления (ЧПУ), как правило, встраивается в конструкцию станка (жёстко закрепляется на станине), и не имеет возможности перемещения по периметру станка, при этом оператору требуется свобода передвижения в зоне настройки. Мобильность наладчика повышает выносной пульт, но пульт имеет рабочую зону, которая ограничивается длиной кабеля, при помощи которого он подключается к станку. Так же выносной пульт имеет стандартный интерфейс, предназначенный для управления, а не для настройки станка. Для решения задачи мобильности инженеров-операторов станочного оборудования предлагается новый способ управления станком - мобильный терминал с конфигурируемым пользовательским интерфейсом, построенный на базе мобильного приложения (Рис.1).

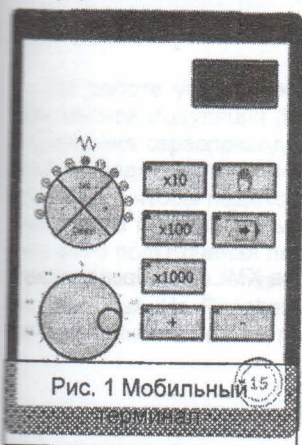


Рис. 1 Мобильный терминал

Целью работы является создание специализированного терминала системы ЧПУ на основе мобильного приложения для основных операционных систем (iOS, Android) для повышения мобильности операторов станочного оборудования.

В список решаемых задач входят:

1. Анализ инструментальных средств для обеспечения кроссплатформенности приложения для устройств на основе ОС Android и ОС iOS.
2. Разработка механизма конфигурации мобильного терминала
3. Разработка специализированных элементов управления для интерфейса терминала: многопозиционный переключатель, маховик, аварийная кнопка, кнопка с индикацией.

Теоретические исследования в работе базируются на методах системного анализа, объектно-ориентированного проектирования (декомпозиции, абстракции, иерархии), концепции объектно-ориентированного программирования. В качестве инструментальных средств построения приложений применяются интегрированная среда разработки Framework Qt, язык программирования C++, язык разметки графических интерфейсов

QML и язык разметки данных XML.

В настоящее время портативные устройства получили широкое распространение в мире вследствие своей доступности и большому списку решаемых задач. Это стало возможным благодаря увеличению вычислительной мощности электроники и вследствие этого применению операционных систем, которые в данный момент представлены в основном операционными системами Android (53%) и iOS (40%). В связи с этим эти платформы выбраны в качестве основных.

Поскольку эти операционные системы имеют разные «экосистемы» для разработки приложений, а также имеют разные уровни открытости, пользовательские интерфейсы, доступ к файловой системе, необходимо учитывать все различия между ними. Существенные различия операционных систем требуют различного инструментария создания приложений. Для iOS родными средствами разработки являются языки Objective-C и Swift, а для Android – Java. Отдельно разрабатывать приложения для каждой операционной системы было бы накладно, поэтому необходимо выбрать кроссплатформенное средство для создания приложения.

В настоящее время существует множество решений для построения мобильных приложений, среди них можно выделить PhoneGap, Xamarin, Qt.

Преимущества Qt заключаются в использовании языка программирования C++, который является более производительным за счёт компиляции в машинный код под конкретную платформу, а не под виртуальную машину как в случае с C#, или интерпретации в браузере (JavaScript). Возможность компиляции для встраиваемых систем так же является преимуществом, т.к. это позволяет использовать приложение не только на мобильных устройствах, но и во

<sup>1</sup> Работа выполнена при финансовой поддержке Министерства образования и науки РФ в рамках государственного задания в сфере научной деятельности.

встраиваемых системах, которые часто применяются в станкостроении. Так же неоспоримым преимуществом является наличие лицензии open-source.

Поэтому для разработки элементов управления, была выбрана кроссплатформенная инструментальная среда Qt – QML.

Преимущества Qt являются:

- хорошая графика: Qt Quick (QML) базируется на OpenGL ES для встраиваемых систем;
- производительность разработки: приложение разделяется на front-end (QML + JavaScript) и back-end (Qt C++) части, что способствует front-end разработчикам сфокусироваться на разработке красивого UI, а back-end разработчики уделяют больше внимания стабильности и производительности;
- кроссплатформенность: можно разработать один код сразу под все поддерживаемые операционные системы (Windows, Linux, MacOS, Android, iOS и др.);
- открытость: Qt является активно развивающимся open-source проектом, разработка которого ведётся открытым сообществом. QML это декларативный язык программирования основанный на JavaScript. QML легок в освоении и прост в отрисовке графики. Написанный код не нужно компилировать, т.к. этот язык является интерпретируемым.

Реализация мобильного терминала

В архитектуре приложения мобильного терминала можно выделить две части (Рис. 2), разработанные соответственно на C++, с использованием библиотек инструментария Qt, и на языке разметки QML. В соответствии с общим подходом в разработке мобильных приложений в структуре выделяются следующие части:

- Front-end. Используется язык программирования QML. Разработка дизайна приложения. Создание эргономичных пользовательских интерфейсов с поддержкой анимации.
- Back-end. Используется кроссплатформенный фреймворк Qt (C++). Реализуется логика приложения, работа с сетью, файловой системой и т.п.

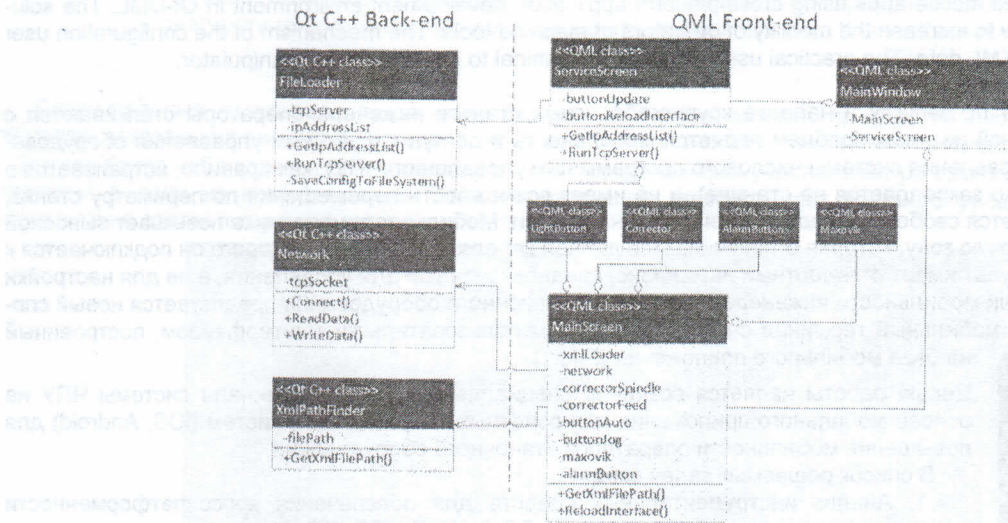


Рис. 2 Архитектура приложения мобильного терминала

Чтобы сделать элементы конфигурируемыми был использован язык разметки документов XML. Он обеспечивает возможность записывать и сохранять параметры конфигурации элементов управления.

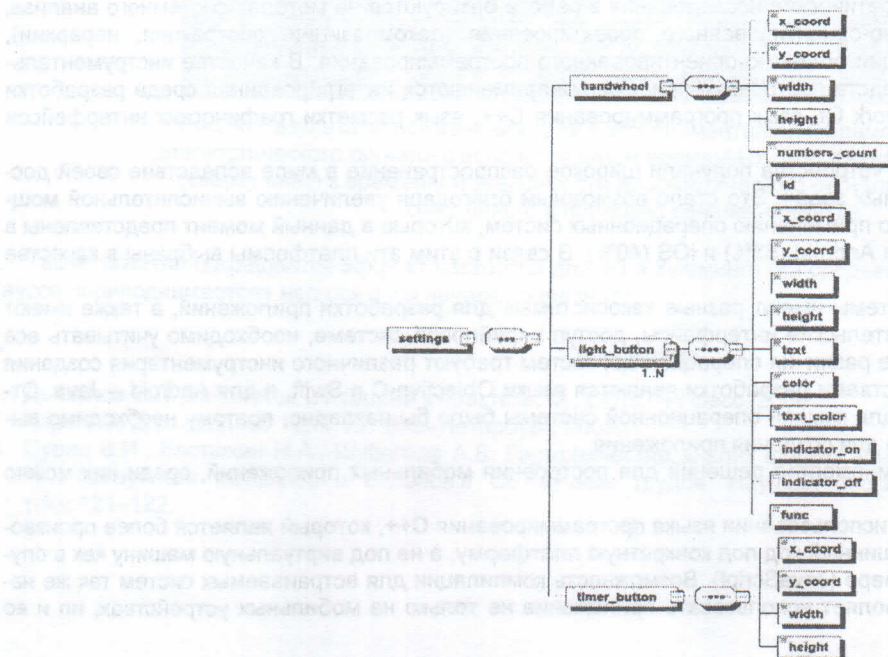


Рис. 3 XSD схема XML-документа для конфигурации пользовательского интерфейса

Разработчик создает стандартный пользовательский интерфейс, который включает в себя создание элементов управления, обеспечение передачи XML-документов на устройство, так же изменять функционал элементов управления. Наладчик в свою очередь же обладает правом настройки пользовательского интерфейса по необходимости: создавать и удалять элементы управления, менять их расположения, менять параметры (размер, цвет, текст, иконки). Наладчик может сам создать xml-документ и редактировать его при соответствующих правах доступа.

На XSD схеме XML-документа (Рис.3) представлена структура конфигурации пользовательского интерфейса. Корневой элемент настройки (settings) содержит в себе список типов элементов управления: маховик (handwheel), кнопка с индикацией (light button), кнопка безопасности (timer button). Элементов каждого типа может быть несколько, у каждого типа свои параметры, которые конечный пользователь может изменить в зависимости от технического задания или своих пожеланий.

При загрузке XML-документа на мобильное устройство возникает проблема сохранения документа на устройстве с ОС iOS. Пользователи не имеют прав прямого доступа к файловой системе iOS. К файловой системе имеет доступ только само приложение, его и нужно использовать для передачи настроек в формате XML. Для этого можно создать универсальный механизм загрузки файлов на основе клиент-серверной модели и протокола TCP/IP.

На Рис.4 представлен процесс загрузки настроек интерфейса пользователя на терминал. Для этого разработан клиент, находящийся на компьютере. Принцип работы заключается в создании сокета и подключения к серверу, затем загрузки XML-документа и закрытия сокета соответственно. Сервер находится на мобильном терминале, для связи с клиентом необходимо установить локальную точку для прослушивания подключений, когда сервер получит подтверждение подключения к клиенту, на мобильном устройстве будет загружен XML-документ. Для обновления пользовательского интерфейса нужно перезапустить приложение.

В качестве примера использования мобильного терминала представлено управление приводами робота-манипулятора.

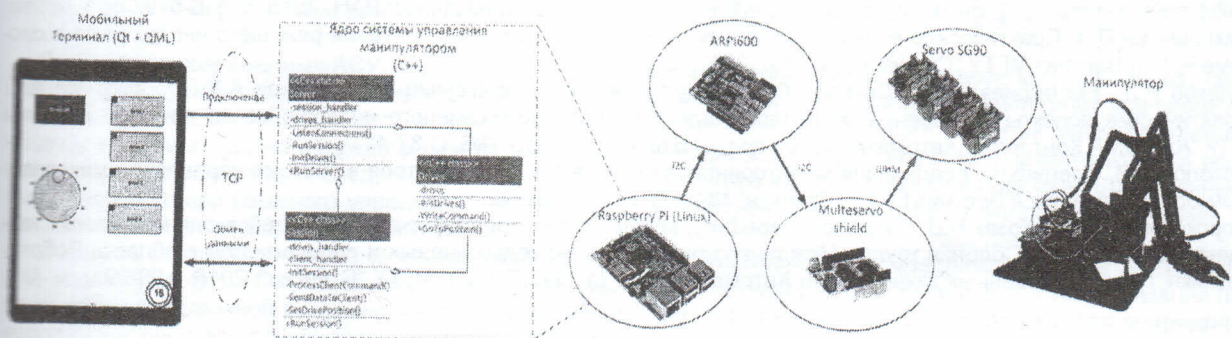


Рис. 4 Архитектура взаимодействия мобильного терминала с роботом-манипулятором

На работе установлены четыре сервопривода. Управление приводами осуществляется с помощью широтно-импульсной модуляции (ШИМ). Сервоприводы подключены к модулю Arduino Multiservo Shield (Рис.4). На плате для управления сервоприводами имеется специальный контроллер, который полностью избавляет от проблемы дрожащих приводов, которое всегда присутствует при управлении напрямую от Arduino. Так как плата Multiservo Shield для Arduino, то чтобы подключить ее к Raspberry Pi используется переходная плата ARPi600. Это плата расширения совместимая с одноплатным миникомпьютером Raspberry Pi, которая добавляет устройству возможность удобного и быстрого подключения платы расширения Arduino Multiservo Shield. Raspberry Pi обладает возможностями практически такими же, что и персональный компьютер. Компьютер работает под управлением дистрибутива Raspbian на базе ОС Linux. На Rasperry запущено серверное приложение (ядро системы управления манипулятором), которое связывается с клиентом по протоколу TCP/IP.

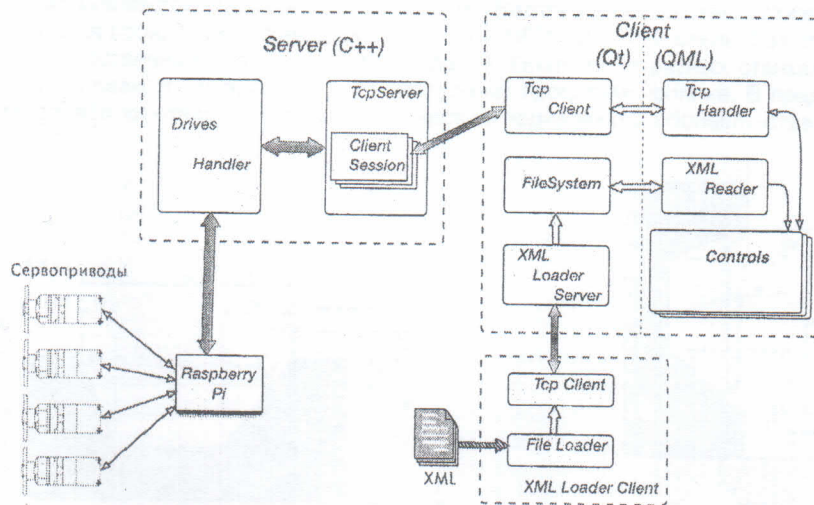


Рис. 5 Структурная схема системы управления роботом-манипулятором

Структурная схема взаимодействия компонентов мехатронной системы показана на Рис.5. От элементов управления (Controls), написанных на языке QML передаются команды в Tcp Handler. Полученная команда направляется в клиент (Tcp Client), написанный на Qt Framework, который по протоколу TCP/IP подключается к серверу (Tcp Server). При подключении клиента к серверу, создается объект Client Session, который принимает команды от клиента, обра-

бывает их и передает объекту Drives Handler. Drives Handler по шине I2C отправляет управляющий пакет приводам через плату ARPI600 и Arduino Multiservo Shield.

Применение мобильных терминалов для обеспечения мобильности операторов станочного оборудования имеет ряд положительных характеристик: возможность перемещения оператора без ограничения; использования устройств разных размеров, от маленьких мобильных устройств до больших планшетов; настройка пользовательского интерфейса в зависимости от решаемых технологических задач; эргономичный и современный пользовательский интерфейс.

Разработаны классы элементов управления, которые наследуют и расширяют функции классов библиотеки Qt Framework. Специализация этих классов направлена на предоставление оператору станка и другого технологического оборудования удобных, эргономичных и безопасных инструментов управления и контроля.

В ходе работы были разработаны варианты компоновки пользовательского интерфейса, в том числе интерфейса терминала для управления роботом-манипулятором. Таким образом, возможна реализация полноценного представления интерфейса терминальной задачи для системы ЧПУ (например, выносного пульта) на основе разработанных элементов управления, механизмов конфигурирования и настройки представления.

#### Литература

1. Козак Н.В., Абдуллаев Р.А. Динамическое конфигурирование прикладных программных компонентов в приложениях электроавтоматики // Вестник МГТУ "Станкин". 2009. №3(7). С. 96–100.
2. Н.В. Козак, М.М. Сорокоумова Разработка специализированных элементов управления для терминальной части системы ЧПУ в инструментальной среде Qt. Труды XXVII Международной инновационно-ориентированной конференции молодых ученых и студентов (МИКМУС-2015). М: Изд-во ИМАШ РАН, 2015. с. 516-519.
3. Никищечкин П.А. Практические аспекты разработки модуля диагностики и контроля режущего инструмента в системе ЧПУ // Вестник МГТУ "Станкин", №4(23), 2012, с. 93-97.
4. Пушков Р.Л., Евстафиева С.В., Соколов С.В., Абдуллаев Р.А., Никищечкин П.А., Кулиев А.У., Сорокоумов А.Е. Практические аспекты построения многотерминального человеко-машинного интерфейса на примере системы ЧПУ "АксиОМА Контрол" // Автоматизация в промышленности. 2013. №5. С.37-41.
5. Соколов С.В. Специфика реализации мониторинга технологических параметров в системе управления мехатронным оборудованием // Вестник МГТУ "Станкин" №4(17), 2011, с. 89-92.
6. Сорокоумова М.М., Козак Н.В. Разработка конфигурируемого пользовательского интерфейса для мобильных терминалов оператора. Сборник трудов Международной школы молодых ученых и специалистов в области Робототехники, Производственных технологий и Автоматизации. М.: ФГБОУ ВО "МГТУ "Станкин", 2016. с.98-99.