

# ОСОБЕННОСТИ РАБОТЫ С ВИРТУАЛЬНЫМ СОМ-ПОРТОМ В СРЕДЕ РАЗРАБОТКИ QT

Долгов А. Н.<sup>1</sup>, Рахимзода А. К.<sup>2</sup>

<sup>1</sup>Долгов Антон Николаевич / Dolgov Anton Nikolaevich – студент;

<sup>2</sup>Рахимзода Азимжони Кодиржон / Rahimzoda Azimjoni Kodirzhon – студент,  
кафедра систем автоматического управления и контроля, факультет интеллектуальных технических систем,  
Национальный исследовательский университет  
Московский институт электронной техники, г. Зеленоград

**Аннотация:** в данной статье рассматриваются основные теоретические сведения об инструменте программирования на различных языках программирования, интерфейсе для подключения периферийных устройств к вычислительной технике. Здесь приведены определения, принципы и механизмы работы, а также описаны сравнения различных версий USB, некоторые преимущества инструмента и недостатки. Кратко подчеркнуты способы и методы работы в среде разработки QT, приведен один из примеров подключения устройства с помощью QT. Далее в написанной небольшой программе объясняем работу каждой строчки кода, начиная с объявления объекта, заканчивая описанием работы различных функций в составе приложения. Кроме того, описывается важность внедрения данной технологии в современную IT-индустрию.

**Ключевые слова:** виртуальный СОМ-порт, среда разработки QT, микроконтроллер.

USB (ю-эс-би, англ. Universal Serial Bus — «универсальная последовательная шина») — последовательный интерфейс для подключения периферийных устройств к вычислительной технике. Получил широчайшее распространение и фактически стал основным интерфейсом подключения периферии к бытовой цифровой технике. Интерфейс позволяет не только обмениваться данными, но и обеспечивать электропитание периферийного устройства. Сетевая архитектура позволяет подключать большое количество периферии даже к устройству с одним разъемом USB. [1].

USB интерфейс получил широкое применение в современной вычислительной технике. Связано это с двумя основными параметрами: скорость передачи данных и простота реализации интерфейса, если сравнивать с такими интерфейсами как CAN, UART и др. На данный момент времени есть 3 различных версии USB: USB 1.0, USB 2.0, USB 3.0. А так же существуют их различные модификации. Все они совместимы друг с другом. Максимальная скорость передачи данных составляет 480 Мбит/с (USB 3.0) и минимальная скорость 1,5 Мбит/с (USB 1.0 режим с низкой пропускной способностью).

А теперь непосредственно приступим к рассмотрению работы с виртуальным СОМ-портом в кроссплатформенной среде разработки Qt.

Для работы в среде Qt с виртуальным СОМ портом следует использовать библиотеку QSerialPort. Модуль QtSerialPort - это дополнение к библиотеке Qt5, которое предоставляет единый интерфейс для работы, как аппаратными, так и с виртуальными последовательными портами. Последовательный интерфейс благодаря своей простоте и надежности до сих пор популярен в промышленности, при разработке встраиваемых устройств, робототехнике и т.п. Использование модуля QtSerialPort позволяет разработчикам значительно сократить время на реализацию Qt-приложений, требующих доступ к последовательным интерфейсам[2].

Рассмотрим пример подключения устройства.

1. QSerialPort vcom
2. QString com\_name = "COM3";
3. vcom.setPortName(com\_name);
4. vcom.setBaudRate(QSerialPort::Baud115200);
5. vcom.setDataBits(QSerialPort::Data8);
6. vcom.setParity(QSerialPort::NoParity);
7. vcom.setStopBits(QSerialPort::OneStop);
8. vcom.setFlowControl(QSerialPort::NoFlowControl);
9. vcom.open(QSerialPort::ReadWrite);

Рассмотрим каждую строчку код поподробнее.

1. Объявление объекта vcom.  
2. Объявляем переменную типа com\_name, которая отвечает за com-порт к которому будет происходить подключение.

3. Функцию setPortName() отвечает за подключение к заданному порту.

4. Функция setBaudRate() отвечает за скорость работы com- порта, на самом деле, при работе с виртуальным com- портом, значение установленной скорости не играет роли. Master и slave сами определяют с какой скоростью они будут работать. Т. е. заданная скорость может быть проигнорирована и установлена другая.

5. Функцию setDataBits() отвечает за количество передаваемых бит. В данном случае это 8 бит. Но в связи с тем, что обмен данными происходит через виртуальный com-порт, USB не будет использовать

протокол передачи СОМ-порта, вместо этого будет использоваться тип посылки bulk. Это значит, что если вы решите передать массив данных некоему устройству, то каждый элемент массива будет передан не отдельно, а будет находиться в посылке со всеми остальными элементами. Прием же передачи данных выглядит еще необычнее. Предположим что у нас есть микроконтроллер и мы так же используя виртуальный com-порт, пытаемся передать некоторые данные, в случае передачи того же самого массива, допустим размером в 64 элементов. Микроконтроллер не сразу выдаст их принимающему устройству. Данная ситуация справедлива лишь в том случае, если вы непрерывно пытаетесь отправлять данные.

6. Функцию setDataBits() отвечает за режим проверки бита четности. В данном примере он отключен.

7. Функция setStopBits() отвечает за установку количества стоповых битов. В данном случае он один.

8. Функция setFlowControl() за необходимый режим управления потоком.

9. Здесь происходит непосредственное подключение устройства. Задается так же режим ReadWrite означающий, что данные будут не только передаваться, но и считываться.

Вывод. Очевидно, что работать с СОМ-портом не сложно, с СОМ- портом в кроссплатформенной среде Qt работать можно.

### *Литература*

1. [Электронный ресурс]: Определения и принципы работы QtSerialPort. Режим доступа: [https://wiki.qt.io/Qt\\_Serial\\_Port/ru](https://wiki.qt.io/Qt_Serial_Port/ru) (дата обращения: 18.12.2016).
2. [Электронный ресурс]: Материал из Википедии — свободной энциклопедии. Режим доступа: <https://ru.wikipedia.org/wiki/USB> (дата обращения: 28.11.2016).