



УДК 537.8

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ СИСТЕМОЙ ПИТАНИЯ НА ОСНОВЕ WIENER MPOD**SOFTWARE DEVELOPMENT FOR MULTI-CHANNEL LV/HV SUPPLY SYSTEM BASED ON WIENER MPOD****Я.Т. Схоменко ^{1,2}, И.Е. Внуков ¹, А.Ю. Исупов ², В.П. Ладыгин ², С.Г. Резников ²
Ya.T. Skhomenko ^{1,2}, I.E. Vnukov ¹, A.Yu. Isupov ², V.P. Ladygin ², S.G. Reznikov ²**¹⁾ Белгородский национальный исследовательский университет, Россия, 308015, г. Белгород, ул. Победы, 85²⁾ ЛФВЭ им. В.И. Векслера и А.М. Балдина, ОИЯИ, Россия, г. Дубна, 141980, ул. Жолио-Кюри, 6¹⁾ Belgorod National Research University, 85 Pobedy St, Belgorod, 308015, Russia²⁾ VBLHEP, JINR, 6 Joliot-Curie str., Dubna, 141980, Russia

E-mail: cxom4uk@bk.ru

Аннотация. В статье представлены результаты разработки программы управления системой многоканального низковольтного и высоковольтного питания на основе Wiener Mpod. Данная работа выполнена в рамках проекта DSS (Deuteron Spin Structure) по изучению структуры легких ядер на Нуклотроне ОИЯИ. Приложение разработано под Unix-подобные операционные системы. Программное обеспечение выполнено с использованием кросс-платформенного пакета Qt на языке программирования C++. Связь приложения с контроллером системы питания осуществляется посредством пакета Net-SNMP. При разработке ПО был применен метод разделения на нити с целью улучшения интерактивного отклика приложения. В процессе работы приложение сохраняет информацию с использованием программного пакета syslog(3).

Resume. The article presents the results of the control software development for the multi-channel low-voltage and high-voltage power supply system on the base of Wiener Mpod. This work was performed as a part of the project DSS (Deuteron Spin Structure) for the study of the structure of light nuclei at the JINR Nuclotron. The application is developed under Unix-like operating systems. The application is implemented using the cross-platform software package Qt on the C++ programming language. Communication of the application with the power system controller is carried out by means of Net-SNMP package. The threads have been used to improve the application interactive response. During the application run the LV/HV information is stored using a syslog (3) software package.

Ключевые слова: Qt, MPod, Net-SNMP, QProcess, QThread, запись в файл, чтение из файла.

Key words: Qt, MPod, Net-SNMP, QProcess, QThread, output file, input file.

Введение

Основной целью эксперимента DSS[1,2] (Deuteron Spin Structure) является изучение спиновой структуры двухнуклонных и трехнуклонных короткодействующих корреляций (SRC) с помощью измерений поляризационных наблюдаемых в реакциях с участием дейтронов на Нуклотроне ОИЯИ. Большое значение имеет изучение спиновых эффектов в этих реакциях, поскольку они чувствительны к спиновой структуре короткодействующих корреляций. Нуклотрон позволяет проводить поляризационные исследования в широком диапазоне энергий.

Экспериментальная программа DSS на станции внутренних мишеней Нуклотрона позволяет получить информацию о 2- и 3-нуклонных силах (в том числе их спин-зависимой части) из двух процессов: dp-упругого рассеяния при энергиях 300 - 2000 МэВ и развала дейтрона с регистрацией двух протонов при энергиях 300 - 500 МэВ.

Мотивацией для этой программы служит теоретический анализ экспериментальных результатов, полученных при низких и промежуточных энергиях для реакций с дейтронами в начальном состоянии.

Так как основой детектирующей аппаратуры в эксперименте на внутренней мишени на пучке поляризованных дейтронов являются сцинтилляционные счетчики на базе фотоумножителей Hamamatsu, то требуется обеспечение высоковольтного питания, для чего будет использоваться крейт Wiener MPod[3]. Поскольку крейт установлен вблизи экспериментальной установки, то



исключается ручное управление с использованием локального алфавитно-цифрового дисплея, которое к тому же является достаточно трудоемким.

В рамках модернизации экспериментальной установки была поставлена задача по разработке программного обеспечения управления системой питания Wiener MPod. Приложение должно включать графическую оболочку (GUI), что значительно облегчает работу оператора. Основными задачами приложения являются управление модулями, установка, чтение и визуализация информации из них (имя канала, его статус, напряжение, ток и др.). Необходимо, чтобы программа позволяла работать с файлами для сохранения и загрузки параметров модулей.

Существует вариант настройки крейта через USB-порт контроллера с компьютера под управлением 32-разрядной Windows XP, Vista или Windows 7. Для работы необходимо установить драйверы и приложение для работы. Запуск программы предоставляет краткий обзор MPod. Информация предоставляется только о модулях низкого напряжения. Модули высокого напряжения не отображаются и не дают возможности управления посредством USB. Данный метод не подходит по ряду причин, в том числе, потому что при выполнении эксперимента необходимо использовать высоковольтные модули.

Также, подключив контроллер крейта к персональному компьютеру через Ethernet, можно просмотреть информацию о параметрах LV/HV модулей через веб-браузер, введя в качестве URL IP-адрес устройства. Данный способ позволяет увидеть следующую информацию: имя канала; напряжение, заданное пользователем; порог силы тока на канале; измеряемое напряжение на канале; измеряемый ток; рабочий статус канала.

Недостатком такого способа является, в первую очередь, отсутствие возможности изменения параметров модулей. Также данный метод не предоставляет информацию о скорости возрастания и падения напряжения и тока.

Данные способы являются ограниченными в функциональности и не дают возможности управления высоковольтными модулями, установленными в систему питания. В связи с этим была поставлена задача о разработке программного обеспечения для установки, считывания и визуализации параметров LV/HV модулей под Unix-подобные операционные системы с использованием протокола SNMP.

Используемое оборудование

Особенностью системы MPod является то, что она позволяет использовать высоковольтные (HV) и низковольтные (LV) модули одновременно. Полноразмерный крейт MPod имеет 10 секторов для силовых модулей, которые обеспечивают большое количество выходных каналов (до 80 для низковольтных модулей и до 160 для высоковольтных). Модульная конструкция крейта дает возможность легко заменить лоток вентилятора, контроллер, основной источник питания или дополнительный воздушный фильтр.

При разработке приложения использовался Wiener MPod миникрейт, который позволяет использовать до четырех модулей низкого и высокого напряжения. Он включает в себя первичный источник питания на 600 Вт для модулей, а также систему охлаждения с высокой производительностью вентилятора. Первый сектор крейта зарезервирован для контроллера MPod, который управляет первичным источником питания и обеспечивает Ethernet, USB и CAN-Bus интерфейсы для удаленного мониторинга и управления.

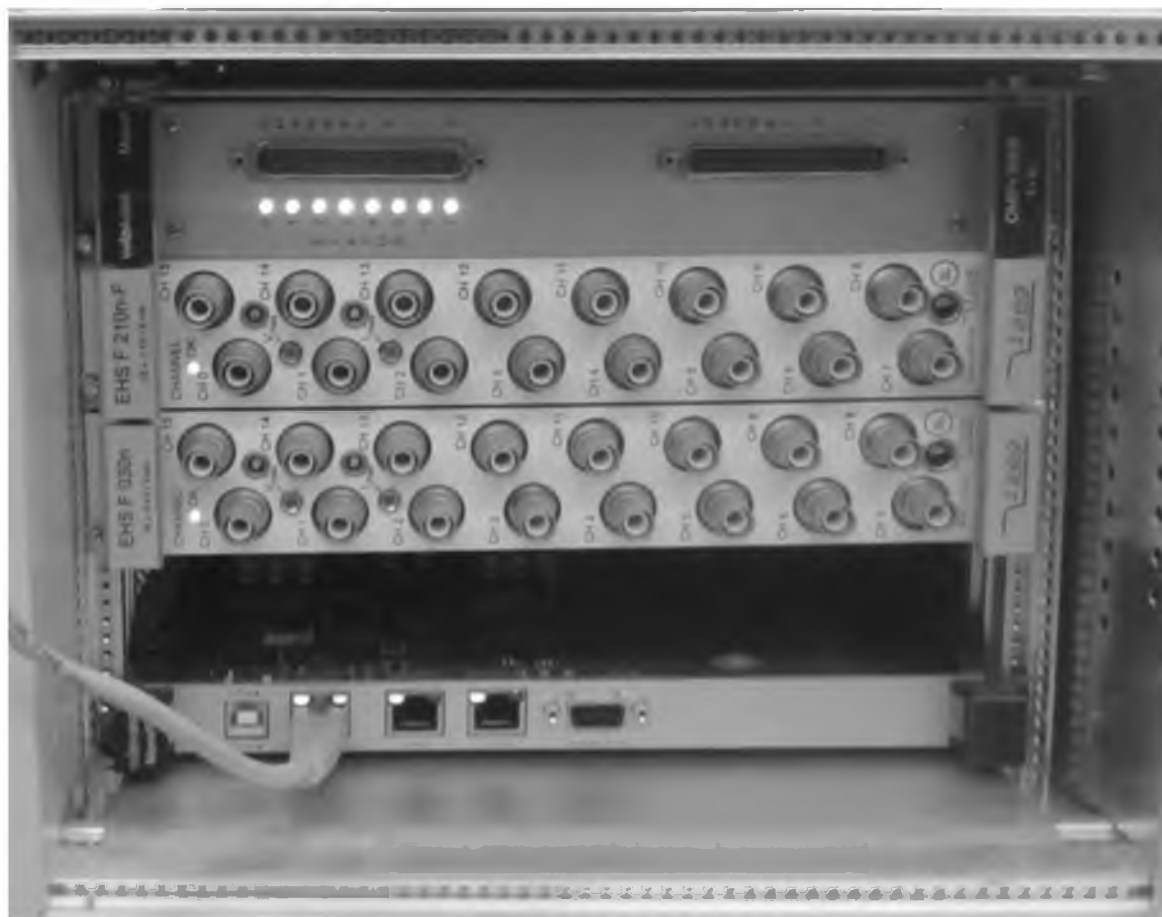


Рис. 1. Фотография MPod миникрейта с крейт-контроллером, низковольтным модулем OMPV.8008 и высоковольтными модулями EHS F 030n и EHS F 210n-F.

Fig. 2. Photo minicrate with MPod crate-controller, low voltage module OMPV.8008 and high voltage modules EHS F 030n and EHS 210n-F.

При разработке приложения в крейте были установлены четыре модуля (рис. 1): контроллер MPod; EHS F 030n – 16-канальный высоковольтный модуль на 3kV/3mA; EHS F 210n-F – 16-канальный высоковольтный модуль на 1kV/8mA; OMPV.8008 – 8-канальный низковольтный модуль на 8V/10A.

Несмотря на то, что использованное в процессе работы устройство имеет всего четыре сектора, разработанное приложение позволяет работать также с большим крейтом, рассчитанным на десять секторов, поскольку они имеют одинаковые контроллеры.

В данной работе контроллер MPod соединен с компьютером посредством Ethernet через сетевой коммутатор.

Используемые программные пакеты

При разработке приложения были использованы следующие программные пакеты: Qt creator 3.5.1, Net-SNMP и syslog(3).

Qt – кросс-платформенный инструментальный разработчик ПО на языке программирования C++, который позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путем простой компиляции программы для каждой ОС без изменения исходного кода. Он включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Отличительная особенность Qt от других библиотек — использование Meta Object Compiler (МОС) — предварительной системы обработки исходного кода. МОС позволяет во много раз увеличить мощь библиотек, вводя такие средства взаимодействия, как слоты и сигналы.

Начиная с версии 4.5.0 в комплект Qt включена среда разработки Qt Creator, которая включает в себя редактор кода, справку, графические средства Qt Designer и возможность отладки приложений.

Программный пакет Net-SNMP – предоставляет реализацию стандартного протокола SNMP (Simple Network Management Protocol) для управления устройствами в IP-сетях. С его помощью осуществляется взаимодействие компьютера с контроллером.

Использование syslog(3) обеспечивает сохранение данных в архивных файлах. Источники данных посредством syslog(3) формируют простые текстовые сообщения о происходящих в них событиях и передают их на обработку серверу Syslog (т.н. syslogd или syslog daemon), используя локальное или удаленное соединение через сокет. Формирование сообщений о событиях и их передача происходит по определенным правилам, называемым протоколом Syslog, который реализуется библиотечными функциями syslog(3). Как правило, сообщение имеет небольшой размер (до 1024 байт) и отсылается в открытом виде.

Общий алгоритм приложения

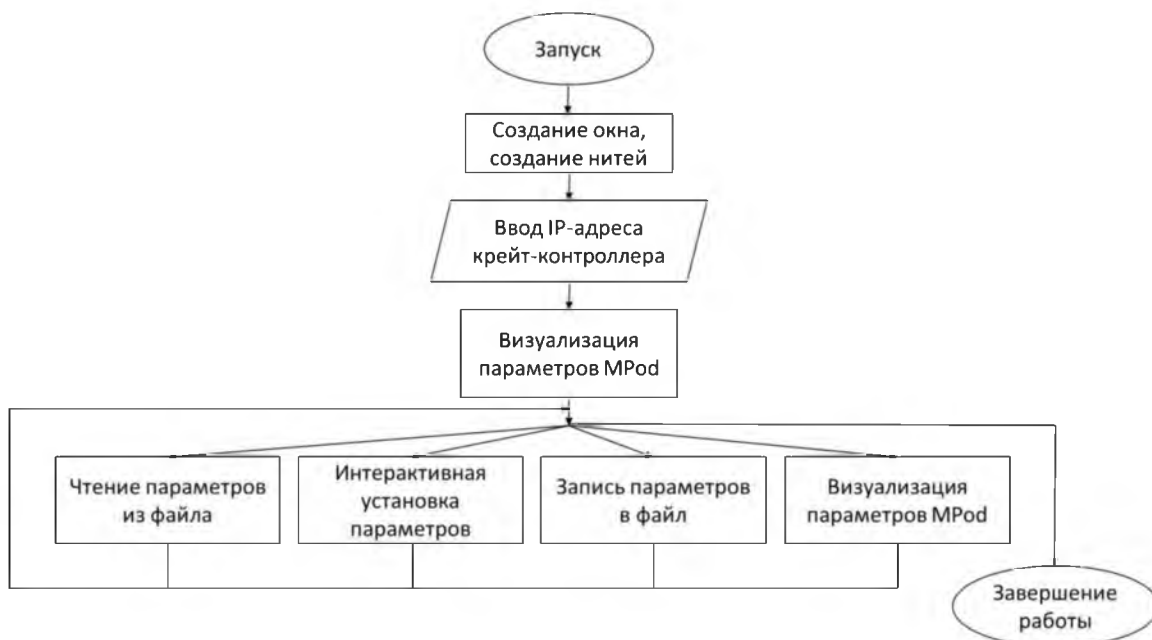


Рис. 2. Общий алгоритм приложения.

Fig. 2. The General algorithm of the application.

На рис. 2 представлен схематический алгоритм работы приложения, который демонстрирует основные его блоки без детального разбора каждого. После запуска приложения происходит построение окна, создание нитей. Далее оператору необходимо ввести в специальную область (однострочное поле ввода) IP-адрес, установленный в контроллере. После нажатия кнопки запуска приложение производит поканальное чтение параметров из кейта. Далее для пользователя доступны 4 функции:

1. Чтение параметров из файла;
2. Интерактивная установка параметров;
3. Запись параметров в файл;
4. Визуализация параметров модулей.

Визуализация параметров доступна также и в автоматическом режиме через определенные промежутки времени (при активации соответствующей кнопки).



Исходный код приложения

Qt-проект содержит 9 файлов, в том числе файл проекта MPod.pro, хранящий в себе настройки компиляции. Заголовочный файл mainwindow.h содержит несколько компонентов. В самом начале объявлены константы. Далее подключаются заголовочные файлы Qt-библиотек. Далее определяется структура, содержащая компоненты, необходимые для построения окна и вывода информации. Затем определяется класс, содержащий заголовки всех функций сигналов и слотов. Данный заголовочный файл является основным.

Заголовочный файл snmp.h определяет класс, который отвечает за выполнение функций считывания параметров из крейта и выполняется в отдельной нити. Класс содержит два сигнала: информирующий об окончании выполнения считывания параметров из крейта и пересылающий информацию о степени выполнения в основное окно. Также класс содержит слот, выполняющий запуск считывания, и объявляет несколько функций считывания информации из крейта.

Заголовочный файл apply.h определяет класс, отвечающий за выполнение функций загрузки параметров в крейт и выполняющийся в отдельной нити. Структура файла схожа с snmp.h. Данный файл отличается лишь тем, что вместо функций считывания объявляются функции записи параметров в крейт.

Файл исходного кода main.cpp содержит в себе функцию main(), создающую основное окно.

В файле mainwindow.cpp реализованы основные функции приложения: конструктор основного окна MainWindow; несколько функций, считывающих информацию из крейта; ряд функций, описывающих поведение компонентов окна; два слота, выполняемых при получении сигналов из других нитей.

Файл исходного кода snmp.cpp реализует функции, которые отвечают за считывания параметров из крейта, а также слот startReload(), который объединяет работу этих функций.

Файл apply.cpp аналогично предыдущему реализует функции, которые отвечают за загрузку параметров в крейт.

Файлы формы mainwindow.ui – XML-файл, описывающий свойства окна и расположенных в нем графических элементов (виджетов).

QProcess

Для взаимодействия компьютера с крейт-контроллером MPod применяется класс QProcess[4]. Он позволяет приложению запускать внешние процессы и взаимодействовать с ними. Для запуска процесса необходимо передать имя соответствующего исполнимого файла и параметры его командной строки в качестве аргументов в QProcess::start(). Аргументы поставляются как набор отдельных строк QStringList. Вслед за запуском QProcess переходит в состояние Starting, а после запуска процесса – в состояние Running и посылает сигнал started(). QProcess позволяет рассматривать процесс как последовательное устройство ввода/вывода и таким образом писать и читать из процесса. Можно писать в стандартный ввод процесса с помощью write() и читать из стандартного вывода с помощью read(), readLine() и getChar(). Когда процесс завершается, QProcess повторно входит в состояние NotRunning (исходное состояние) и посылает сигнал finished(). Сигнал finished() предоставляет код и статус завершения процесса как аргументы (также их можно получить посредством exitCode() и exitStatus()). Если в любой момент времени возникает ошибка, то QProcess посылает сигнал error(). Также можно вызвать error() для определения типа последней возникшей ошибки и state() для определения текущего состояния процесса.

Процессы имеют два предопределенных потока вывода: стандартный вывод (stdout) предоставляет обычный терминальный вывод, а стандартный вывод ошибок (stderr) обычно предоставляет ошибки, которые печатаются процессом. Установить читаемый поток можно посредством setReadChannel(). QProcess посылает сигнал readyRead(), когда в текущем потоке чтения доступны данные, сигнал readyReadStandardOutput() – когда доступны новые данные в стандартном потоке вывода, и сигнал readyReadStandardError() – когда доступны новые данные в стандартном потоке ошибок. Вместо read(), readLine() или getChar() есть возможность явно прочитать все данные из любого из двух потоков, вызывая readAllStandardOutput() или readAllStandardError().



QProcess предоставляет набор функций, которые позволяют использовать его без цикла обработки событий, приостанавливая вызывающий поток исполнения (нить) до момента отправки определенных сигналов:

waitForStarted() блокирует до тех пор, пока процесс не будет запущен;

waitForReadyRead() блокирует до тех пор, пока в текущем потоке чтения не появятся новые данные, доступные для чтения;

waitForBytesWritten() блокирует до тех пор, пока в процесс не будут записаны данные;

waitForFinished() блокирует до тех пор, пока процесс не будет завершен.

Вызов этих функций из основной нити может привести к «замораживанию» пользовательского интерфейса.

В зависимости от указанных параметров командной строки мы получаем определенный формат вывода информации. Ниже приведен пример командной строки, в которой запрашивается имя канала:

```
snmpget -OQev -v2c -m +WIENER-CRATE-MIB -c guru 192.168.0.10 outputName.u100
```

QThread

Класс QThread обеспечивает платформонезависимые нити[5]. QThread представляет собой отдельный поток исполнения в программе; он может обмениваться данными со всеми другими нитями, но выполняется независимо тем способом, каким отдельная программа выполняется в многозадачной системе. QThread производит уведомление сигналами started(), finished() и terminated() о своем начале, завершении и прерывании соответственно, также можно использовать функции isFinished() и isRunning(), чтобы запросить состояние нити. Необходимо использовать функцию wait() для блокировки до тех пор, пока нить не закончит выполнение. Каждая нить получает свой собственный стек от операционной системы. Операционная система также определяет размер стека по умолчанию. Каждый QThread может иметь свой собственный цикл обработки событий. Существует возможность запуска цикла обработки событий вызовом exec(); также цикл можно остановить вызовом exit() или quit(). Его наличие в нити позволяет соединять сигналы из других нитей со слотами в данной нити, используя механизм, называемый соединением через очередь сигналов. Это также позволяет использовать внутри нити классы, которые требуют цикл обработки событий, такие как QTimer и QTcpSocket. Однако, следует отметить, что невозможно использовать внутри нити любые классы виджетов. В крайних случаях есть возможность принудительно прерывать выполнение нити функцией terminate(). Статические функции currentThreadId() и currentThread() возвращают соответственно: платформозависимый идентификатор текущей нити и указатель на нее. QThread также предоставляет различные платформонезависимые функции сна: sleep() (секунды), msleep() (миллисекунды) и usleep() (микросекунды).

Из-за длительного выполнения ввода и вывода информации с крейт-контроллера MPod в данной работе было использовано разделение приложения на несколько нитей, чтобы улучшить его интерактивный отклик. Для работы с нитями первоначально необходимо в отдельном файле создать класс, содержащий в себе функции, которые необходимо выполнить в отдельной нити. Далее создаются объект этого класса и объект нити, и данный объект класса запускается через объект нити. Взаимодействие между нитями происходит посредством сигналов и слотов (обычного метода взаимодействия в библиотеке Qt).

Работа с файлами

Для удобства пользователя введена возможность сохранения параметров LV/HV модулей в обычный текстовый файл и загрузка их параметров из него. Важно учесть, что необходимо сохранять позицию модулей в крейте, поскольку изменение их положения может привести к некорректной загрузке параметров. В файл сохраняется количество каналов в модуле, напряжение, ток и скорости подъема и снижения напряжения и тока.

На рис. 3 представлены блок-схемы алгоритмов чтения из файла и записи в файл.

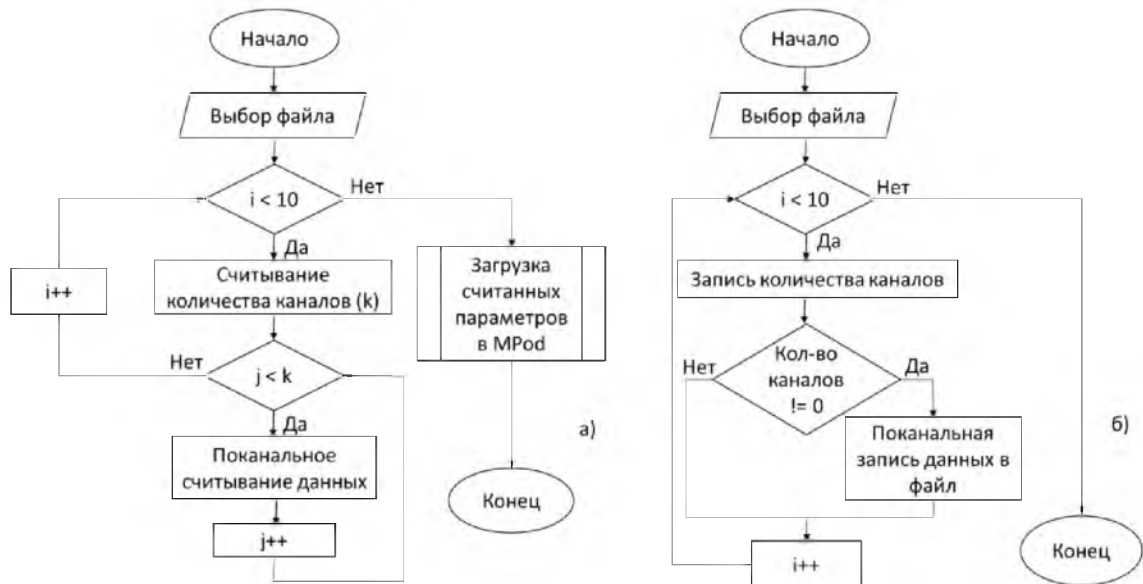


Рис. 3. Алгоритмы работы с файлами: а) считывание из файла; б) запись в файл.

Fig. 3. Algorithms for working with files: a) read from a file; b) writing to a file.

Внешний вид

На рис. 4 представлен вид основного окна приложения. В данном случае видно, что в крейте находятся три модуля, отображается информация с первого модуля, в котором включены только первые три канала. Построение компонентов окна происходит динамически, то есть наличие и положение большинства графических элементов определяется количеством модулей и количеством каналов в каждом модуле.

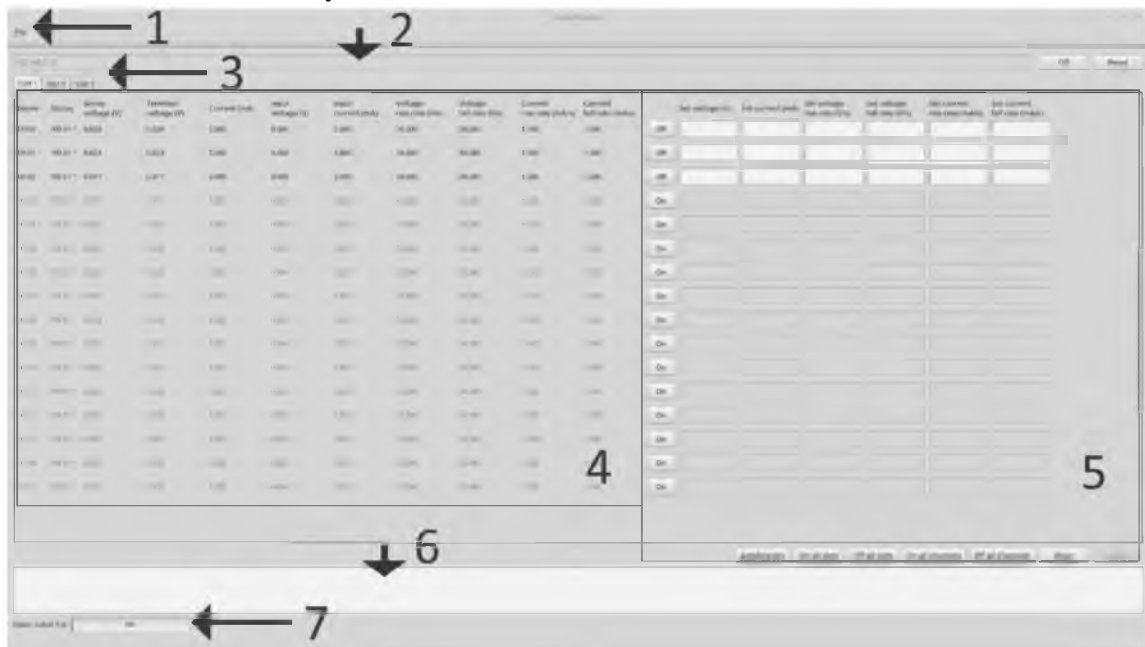


Рис. 4. Основное окно запущенного приложения

Fig. 4. The main window of the running application

Визуально окно можно разделить на несколько частей частей.

1. Меню окна, которое содержит в себе ссылки на функции записи параметров в файл, чтения из файла и выхода из приложения.
2. Область ввода IP-адреса контроллера и последующей активации приложения, которая содержит поле для ввода IP, кнопку включения/выключения и кнопку перезагрузки.



3. Панель переключения между модулями, которая позволяет перейти к отображению информации о том или ином модуле, установленном в крейте.

4. Область информации модуля, которая содержит для каждого канала имя, рабочий статус, измеренные и установленные пользователем напряжение и ток, скорости возрастания и убывания напряжения и тока.

5. Область ввода параметров включает в себя: кнопку включения/выключения канала, поля ввода напряжения, тока, скоростей подъема и снижения напряжения и тока. В нижней части панели имеются несколько общих кнопок: включение/выключение автообновления параметров каналов, активация и деактивация всех каналов и всех каналов текущего модуля, чтение данных из крейта, кнопка применение параметров.

6. Многострочное поле выдачи информации об ошибках во время исполнения.

7. Строка состояния окна, которая показывает степень выполнения той или иной операции.

Заключение

В ходе модернизации высоковольтной системы экспериментальной установки DSS было разработано и успешно протестировано программное обеспечение для управления LV/HV модулями в крейте Wiener MPOd, реализованное с использованием программных пакетов Qt, Net-SNMP, syslog.

Работа частично поддержана грантом РФФИ 16-02-00203а.

Список литературы References

1. Ladygin V.P. 2016. The Spin Studies in Few-Body System at Nuclotron. International Journal of Modern Physics: Conference Series, 40 : 1660074.
2. Ladygin V.P. 2014. Few-body Studies at Nuclotron – JINR. Few Body System, 55 : 709.
3. 2014. MPOD HV& LV Power Supply System Technical Manual. 71.
4. QProcess Class. Qt Documentation - <http://doc.qt.io/qt-5/qprocess.html>.
5. QThread Class. Qt Documentation - <http://doc.qt.io/qt-5/qthread.html>.