

# МОДЕРНИЗАЦИЯ КОМПЛЕКСА ПРОГРАММ АВТОМАТИЗИРОВАННОГО КОНТРОЛЯ И ТЕСТИРОВАНИЯ ДЛЯ РАБОТЫ В СОВРЕМЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМАХ

*Н. В. Цепцова, А. Г. Кузякин, Е. А. Осипова, А. О. Серова, А. А. Чеснов*

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

## Введение

Особенностью последних лет развития оборонной промышленности России является повышение требований к комплексной автоматизации контроля работоспособности объектов специального назначения. К таким объектам могут относиться приборы автоматики [1].

По оценке отечественных и зарубежных специалистов трудоемкость контрольных операций при производстве сложного радиоэлектронного оборудования составляет более 30 % от производственного цикла прибора. При эксплуатации приборов автоматики для поддержания их работоспособности и готовности необходим большой объем работ по контролю технического состояния. Автоматизация процесса контроля позволяет увеличить эффективность работы, повысить надежность измерений, сократить время проверки и избежать возможных ошибок, вызванных действиями оператора.

Для решения задач автоматизации контрольных операций в отделении 11 был разработан комплекс программ автоматизированного контроля и тестирования (ПАКТ). В настоящее время ПАКТ успешно применяется для проверки приборов автоматики и пультовой аппаратуры.

Цель работы – модернизация программного комплекса ПАКТ для использования его в современных операционных системах (ОС), таких как GNU/Linux, Windows 7, 10.

## Задачи, решаемые программным комплексом ПАКТ

ПАКТ – комплекс программных средств, позволяющий проводить проверку объекта контроля (ОК) в автоматическом режиме.

Главной задачей данного комплекса является снижение доли участия человека в процессе контроля измерительной и пультовой аппаратуры. Наряду с этим комплекс решает следующие задачи:

- Удаленное управление измерительной аппаратурой;
- Проведение экспериментов;
- Сбор и обработка данных;
- Принятие решения о годности ОК;
- Простой унифицированный пользовательский интерфейс;

– Единый вид протокола проверки.

Работа ПАКТ основана на принципах кросс-платформенности, модульности и переносимости.

## Существующая архитектура ПАКТ

ПАКТ представляет собой клиент-серверное приложение, использующее веб-технологии. Организация пользовательского интерфейса комплекса ПАКТ в виде веб-приложения позволяет добиться максимальной кросс-платформенности, т. к. на текущий момент веб-браузеры доступны практически на всех операционных системах.

Для работы ПАКТ необходимо, чтобы на персональный компьютер была установлена 32-разрядная версия ОС Windows XP. В качестве базовой программной платформы была выбрана инфраструктура веб-приложения, работающего на связке Apache (веб-сервер) [2] + Django (конструктор сайтов) [3] + SQLite (база данных) [4]. В качестве основного языка программирования выбран язык Python 2.7 [5].

Python – мощный и простой для изучения язык программирования. Он может применяться для написания пользовательского интерфейса, драйверов и программ контроля. Он дает возможность использовать эффективные высокоуровневые структуры данных и предлагает простой, но эффективный подход к объектно-ориентированному программированию. Сочетание изящного синтаксиса, динамической типизации в интерпретируемом языке делает Python идеальным языком для написания сценариев и ускоренной разработки приложений в различных сферах и на большинстве платформ.

Интерпретатор Python и развивающаяся стандартная библиотека находятся в свободном доступе в виде исходных текстов и двоичных файлов для всех основных платформ на официальном сайте Python и могут распространяться без ограничений. Кроме того, на сайте содержатся дистрибутивы и ссылки на многочисленные модули сторонних разработчиков для языка Python, различные программы и инструменты, а также дополнительная документация.

Django – это высокоуровневый Python веб-фреймворк, который реализован на основе архитектуры MVC (Model – View – Controller). MVC – одна из наиболее фундаментальных архитектур для при-

ложений, которая разделяет базовый функционал приложения на ряд отдельных компонентов.

Django имеет лаконичный дизайн и дает возможность для оперативной разработки веб-приложений, позволяет разрабатывать динамические веб-сайты.

Отличительные особенности Django:

- Любой запрос обрабатывается программно и перенаправляется на свой адрес (URL);
- Разделение контента и представления с помощью шаблонов;
- Абстрагирование от низкого уровня баз данных.

Для связи с измерительной аппаратурой выбрана библиотека VISA 1.3, как наиболее законченное решение, поддерживаемое большинством крупных производителей измерительной техники. В дополнение к ней используется модуль PyVISA для получения программного интерфейса к библиотеке VISA из языка Python.

Схема взаимодействия модулей ПАКТ показана на рис. 1. ПАКТ состоит из двух основных модулей: «Пользовательский интерфейс», «Измерительная система». Обмен данными между модулями производится через общую базу данных (БД). Процесс тестирования ОК осуществляется программами автоматизированного контроля, которые состоят из тестов, проверяющих конкретный параметр ОК.

Пользовательский интерфейс представляет собой веб-приложение, построенное на основе веб-сервера Apache. Данные отображаются с помощью Google Chrome. Все данные, вводимые оператором для запуска программы контроля (задачи) поступают в БД. «Измерительная система» состоит из набора программ. «TASKRUNNER» – сервисное приложение для ОС Windows XP, которое постоянно обращается к БД, чтобы найти новые задачи, поставленные

пользователем. Если такая задача найдена, то она передается в программу «CURRENTTEST» для выполнения тестирования ОК. Результаты проверки ОК передаются в БД, а оттуда по запросу поступают в «ПАКТ GUI» для формирования протокола проверки. Таким образом, «пользовательский интерфейс» и «измерительная система» функционируют как два отдельных параллельных потока, которые постоянно обращаются к БД. БД является узким местом производительности системы. С одной стороны, «Пользовательский интерфейс» постоянно запрашивает данные на чтение, с другой стороны, «Измерительная система» постоянно записывает данные. Это может создавать очереди в обработке данных, проведение тестов может затягиваться, а пользовательский интерфейс может зависать.

Зачастую возникают ситуации, когда в ходе проведения проверки ОК оператору необходимо ввести какие-либо дополнительные данные. Возникает проблема, как доставить эти данные в уже выполняющуюся программу контроля. Для решения этой проблемы в программный комплекс ПАКТ была добавлена программа-мессенджер для обмена сообщениями с пользователем.

### Пользовательский интерфейс ПАКТ

Пользовательский интерфейс комплекса ПАКТ прост и унифицирован. Для запуска автоматизированной проверки оператору достаточно ввести информацию о заводском номере ОК, указать свое ФИО, выбрать вид проверки и запустить программу контроля. На рис. 2 изображена страница запуска программы контроля.

После запуска программы контроля открывается страница «Протокол проверки» (рис. 3). Она имеет двойное назначение: во-первых, отображение хода

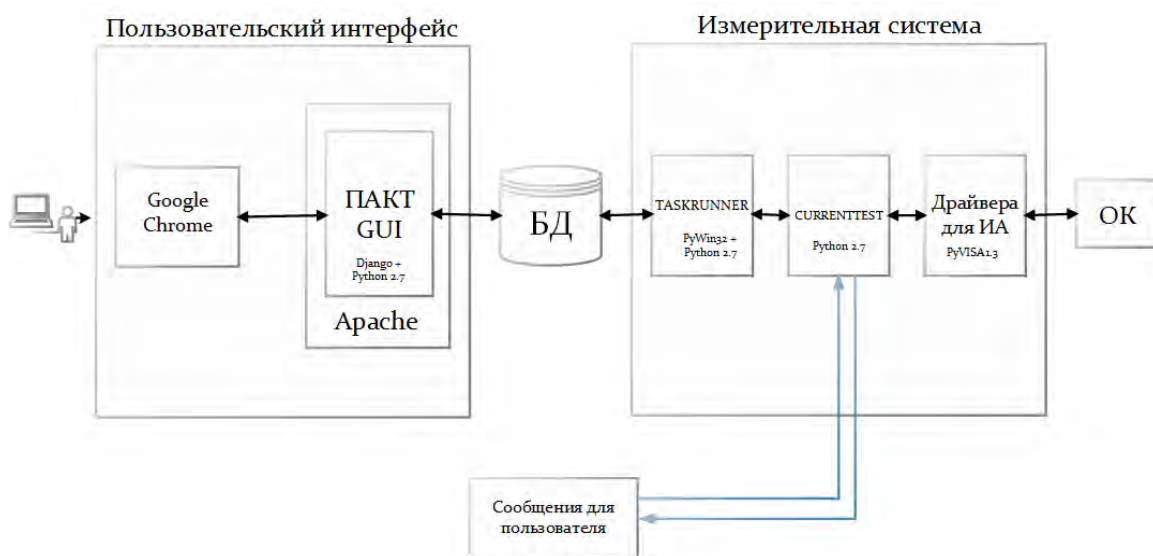


Рис. 1. Схема взаимодействия модулей программного комплекса ПАКТ

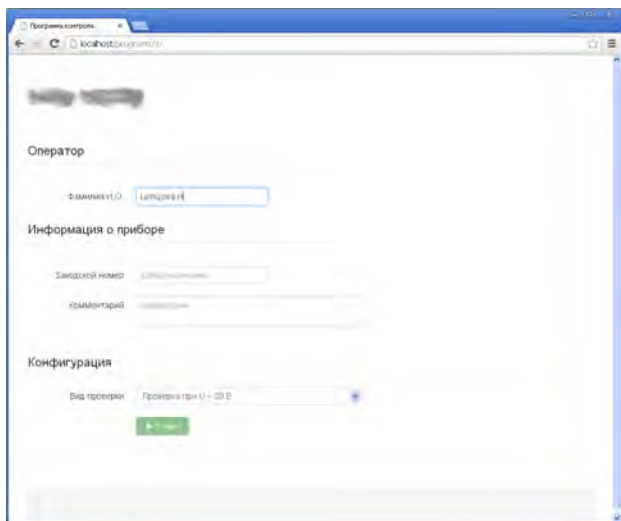


Рис. 2. Страница запуска программы контроля

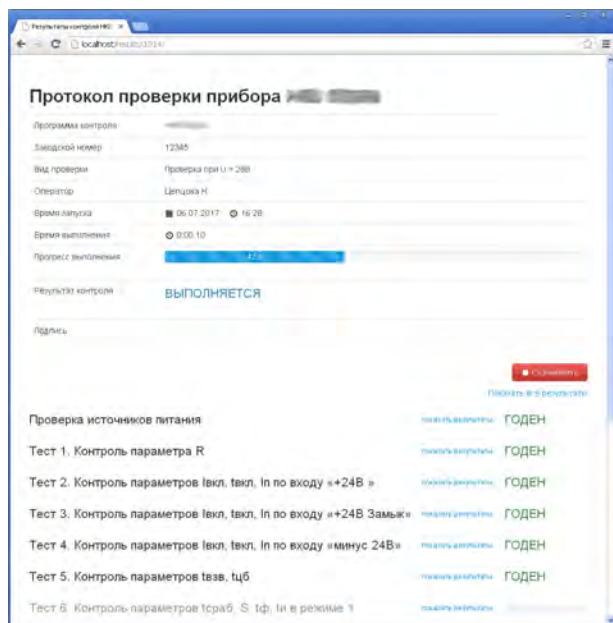


Рис. 3. Страница протокола проверки

текущей проверки ОК, во-вторых – представление результатов контроля.

Результаты проверки могут быть сохранены в виде PDF-файла или HTML-страницы, доступных для последующего вывода на печать.

Унификация пользовательского интерфейса дает следующие преимущества:

- Стандартизация инструкций и методик проверки (алгоритм действий пользователя не меняется от ОК к ОК);
- Привычный интерфейс и алгоритм работы для оператора (не тратится дополнительное время на обучение операторов).

### Новая технологическая платформа для ПАКТ

Для того, чтобы отказаться от использования БД, сократить количество используемых модулей и дополнительных программ и перевести ПАКТ на работу в современных ОС, при этом сохранить функциональность и кроссплатформенность программного комплекса, была предложена новая технологическая платформа для реализации ПАКТ.

Для модернизации ПАКТ были выбраны следующие программные средства:

- Среда программирования Electron;
- Язык программирования Python 3.6;
- Микрофреймворк Flask, модуль flask языка программирования Python 3.6
- Набор библиотек PyVISA 1.8;
- Драйверы и библиотеки из комплекта разработчика контрольно-измерительной аппаратуры.

Electron – среда программирования для создания приложений, написанных на языке программирования JavaScript [7]. Она уже включает в себя Node.js и минимальную версию веб-браузера Chro-

mium. Node.js – интерпретированный язык программирования, построенный на основе JavaScript, добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки, написанные на разных языках. Таким образом, мы получаем управляемое, программируемое окно браузера.

На базе Electron построен не только текстовый редактор для программистов «Atom», но и такие программные продукты для разработчиков, как «Visual Studio Code», «Light Table». На базе Electron также написано клиентское приложение чата «Slack», десктопный клиент «WordPress» и многое другое.

Принцип работы Electron основан на двух типах процессов: основной процесс и рендер-процесс. Основной процесс отвечает за интеграцию и взаимодействие с GUI ОС, например, интеграцию в панель задач для Windows, сворачивание в трей, полноэкранный режим и т. д. Такой процесс может быть запущен один раз на всю жизнь приложения. Рендер-процесс отвечает за отображение окна браузера, в котором может быть открыта страница приложения. Таких процессов может быть произвольное количество.

Flask – фреймворк для создания веб-сайтов на языке Python со встроенным веб-сервером, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2 [6]. Относится к категории так называемых «микрофреймворков» – минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.

В качестве основного языка программирования выбран Python 3.6 [5], т. к. языки программирования Python 2.x уже не поддерживаются разработчиками, а Python 2.7 будет поддерживаться только до

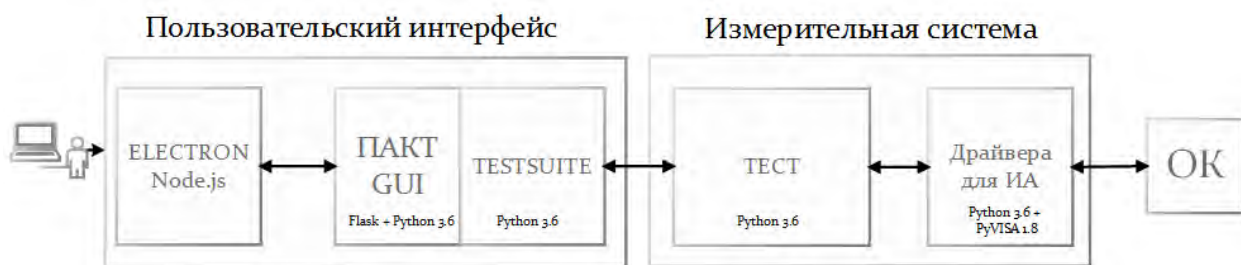


Рис. 4. Новая схема взаимодействия модулей ПАКТ

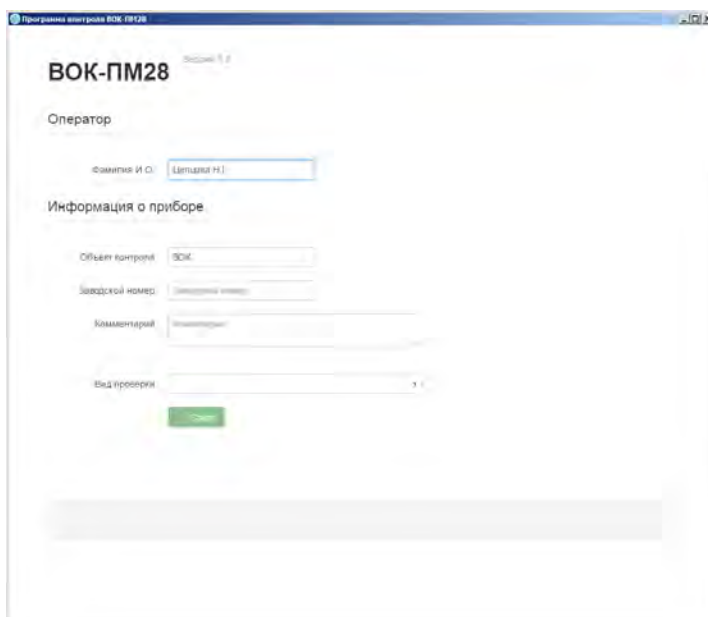


Рис. 5. Форма запуска программы контроля в среде Electron

2020 года. Python 3 был выпущен с целью устранения внутренних конструктивных недостатков предыдущих версий языка, он сосредоточен на поддержке чистой базы кода и устранении избыточности. Хотя между Python 2 и Python 3 существует несколько существенных различий, перейти с одной версии на другую очень просто, это требует лишь небольшой дополнительной настройки.

Набор библиотек ввода/вывода и инструментов управления предоставляет следующие ключевые возможности:

- Быстрое подключение к приборам;
- Автоматическая идентификация измерительных приборов;
- Отражение информации, ориентированное на приборы;
- Удобный пользовательский интерфейс.

Для связи с измерительной аппаратурой выбрана библиотека VISA 1.8. В дополнение к ней используется модуль PyVISA для получения программного интерфейса к библиотеке VISA из языка Python.

С использованием выбранных программных средств структура ПАКТ существенно сократилась. По-прежнему ПАКТ состоит из двух основных модулей: «пользовательского интерфейса» и «измери-

тельной системы», но схема взаимодействия между ними значительно упростилась (рис. 4). «Пользовательский интерфейс» также организуется как веб-приложение, но строится в среде программирования Electron на основе веб-сервера, организованного с помощью модуля Flask.

Теперь модули ПАКТ взаимодействуют друг с другом напрямую, без использования БД и сервиса «TUSKRUNNER», что ведет к увеличению производительности программного комплекса в целом. Сохранение и печать данных протокола проверки будет реализовываться средствами «пользовательского интерфейса». Открытие каждой страницы «ПАКТ GUI» в отдельном процессе с собственным URL-адресом, позволяет отказаться от использования программы-мессенджера для обмена сообщениями с оператором, т. к. пользовательские сообщения могут посылаются по заданному адресу к конкретному тесту.

На рис. 5 показана форма запуска программы контроля «Пользовательского интерфейса», созданная с помощью средств Electron.

Пользовательский интерфейс выглядит в Electron так же, как и в Chrome, а функциональность его значительно увеличивается. Кроме того, было добав-

лено поле «Объект контроля», чтобы привязать электронный адрес страницы к набору тестов для конкретного ОК.

### **Заключение**

Программный комплекс ПАКТ является отечественным кроссплатформенным решением для проведения автоматизированного контроля параметров приборов и блоков, разрабатываемых в подразделении. Модернизация программного комплекса ПАКТ для работы в современных операционных системах дает ряд существенных преимуществ.

Основным преимуществом новой технологической платформы является компактность. Сокращается количество используемых программных модулей, упрощаются связи между ними. Пользователю нет необходимости каждый раз при установке программы устанавливать веб-сервер Apache, веб-браузер Google Chrome.

Свойство Electron открывать каждую веб-страницу в своем процессе позволяет организовать работу нескольких программ проверки различных приборов на одном рабочем месте оператора, т. е. реализовать многозадачность.

Выбор веб-технологий для создания графического интерфейса пользователя, а также поддержка языка программирования Python большинством современных ОС позволяет добиться максимальной переносимости и кроссплатформенности программного комплекса.

### **Литература**

1. Раннев, Г. Г. Измерительные информационные системы М.: Академия, 2010
2. Режим доступа: <http://httpd.apache.org> [Электронный ресурс]
3. Режим доступа: <https://www.djangoproject.com/> [Электронный ресурс]
4. Режим доступа: <http://www.sqlite.org/> [Электронный ресурс]
5. Режим доступа: <http://www.python.org/> [Электронный ресурс]
6. Режим доступа: <http://flask.pocoo.org/> [Электронный ресурс]
7. Режим доступа: <https://electron.atom.io> [Электронный ресурс]