

РЕАЛИЗАЦИЯ СЕРВЕРА ПРОТОКОЛА SCPI ДЛЯ ИЗМЕРИТЕЛЬНЫХ УСТРОЙСТВ СЕМЕЙСТВА ALMAMETER

И. М. Челевич

*Белорусский государственный университет, пр. Независимости, 4,
220030, г. Минск, Беларусь, i.chelevich@gmail.com
Научный руководитель — М.А. Павлышко, старший преподаватель*

Данная статья посвящена разработке программно-аппаратного комплекса для удаленного управления и мониторинга измерительных устройств семейства AlmaMeter. В проекте описаны процесс создания TCP-сервера, поддерживающего протокол SCPI, и его интеграция с измерительными приборами через сеть Интернет.

Ключевые слова: SCPI; Измерительные устройства; Linux; TCP/IP; C++; USB; Ethernet; WEB.

ВВЕДЕНИЕ

Измерительные устройства семейства Almameter, производимые Белорусским государственным университетом (БГУ), не поддерживают протокол SCPI и не имеют web-интерфейса. Отсутствие поддержки протокола SCPI, который является стандартом для управления и взаимодействия с измерительными приборами, ограничивает функциональность и интеграционные возможности устройств Almameter и затрудняет их интеграцию в современные системы автоматизации и мониторинга, требующие стандартизированных протоколов для обмена данными и управления. Для решения этой проблемы на базе библиотеки ASIO и технологии TCP/IP был разработан сервер, способный принимать и обрабатывать SCPI-команды через локальную сеть[1].

SCPI

SCPI (Standard Commands for Programmable Instruments) – это стандартный протокол для управления и взаимодействия с программируемыми измерительными приборами, такими как осциллографы, мультиметры и источники питания. Он используется для унификации командного интерфейса, что облегчает интеграцию и автоматизацию приборов от разных производителей[2]. SCPI был выбран для обеспечения совместимости устройств семейства AlmaMeter с современными системами автоматизации и мониторинга. Примеры техники, где используется SCPI, включают осциллографы Tektronix, мультиметры Keysight, анализаторы спектра Rohde

& Schwarz и источники питания Rigol. SCPI применяется в телекоммуникациях, авиации, автомобильной промышленности и научных исследованиях, помогая стандартизировать процедуры тестирования и калибровки оборудования.

ASIO

ASIO (Asynchronous I/O) – это библиотека для языка C++, обеспечивающая высокопроизводительную работу с асинхронным вводом-выводом, включая сетевые операции[3]. В данном проекте ASIO использовали из-за её способности выполнять неблокирующие операции, что позволяет эффективно обрабатывать многочисленные подключения и запросы параллельно. Она предоставляет удобный интерфейс для работы с TCP[1], что идеально подходит для разработки сервера, обрабатывающего SCPI-команды. Будучи частью Boost, ASIO гарантирует надежность и совместимость с различными платформами, что делает её оптимальным выбором для интеграции устройств семейства AlmaMeter в современные системы автоматизации и мониторинга.

АРХИТЕКТУРА ПРОЕКТА

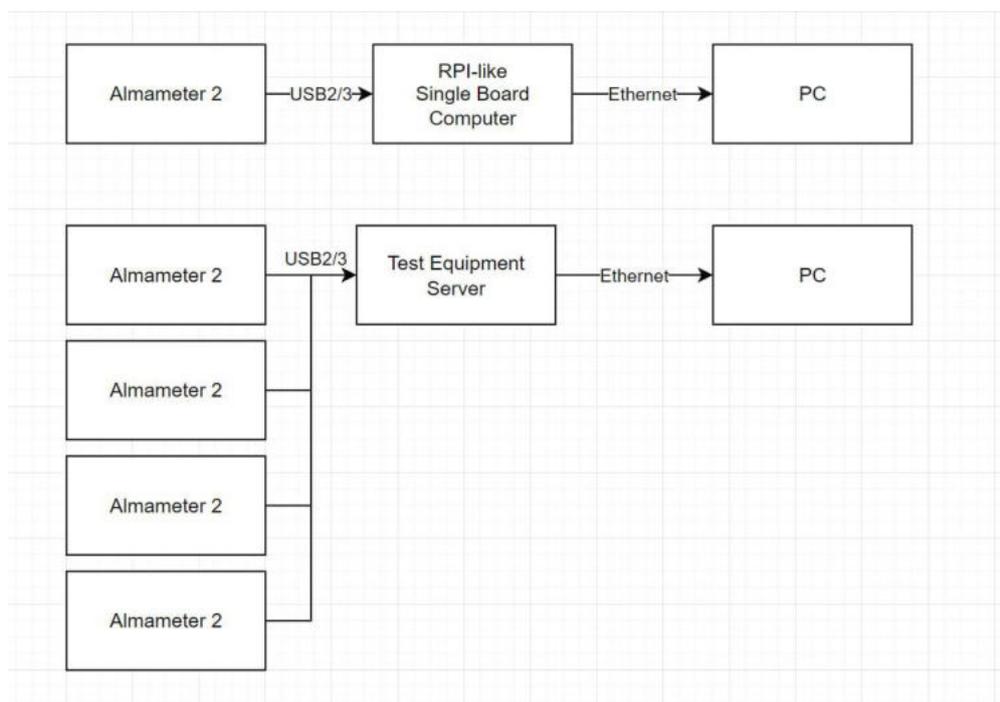


Рис. 1. Блок-схема проекта

Верхний рисунок отображает текущую систему, где одно устройство Almameter 2 подключено к одноплатному компьютеру (например,

Raspberry Pi) через интерфейс USB 2/3. Этот одноплатный компьютер затем подключается к ПК через Ethernet. Такая конфигурация поддерживает только одно устройство Almameter 2, что ограничивает масштабируемость и гибкость системы.

Нижний рисунок показывает планируемое усовершенствование системы. В этой будущей конфигурации несколько устройств Almameter 2 подключаются к серверу измерительного оборудования через интерфейсы USB 2/3. Этот сервер обрабатывает данные от всех подключенных устройств и передает их на ПК через Ethernet. Такой подход значительно расширяет возможности системы, позволяя поддерживать и управлять несколькими устройствами Almameter 2 одновременно. Это улучшение позволит централизовать управление и мониторинг, повысив эффективность и удобство использования системы.

АРХИТЕКТУРА РАЗРАБОТАННОГО ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА



Рис. 2. Блок-схема проекта

«Клиент» был создан для проверки работоспособности программного обеспечения, он подключается по TCP соединению к серверу, отправляет шаблонную команду и слушает ответ от сервера.

«Сервер» принимает и обрабатывает команды от клиентов. На начальном этапе сервер инициализируется и ожидает подключения клиентов. После установления соединения сервер слушает поступающие команды. При получении команды сервер проверяет её наличие в своем списке поддер-

живаемых команд. Если команда существует, сервер обрабатывает её и отправляет ответ клиенту. В случае отсутствия команды сервер возвращает соответствующий SCPI-код ошибки.

«SCPI-транслятор» служит промежуточным звеном между сервером и измерительными приборами. На начальном этапе транслятор инициализируется с использованием стандартных настроек. Транслятор ожидает поступления SCPI-команд от сервера. При получении команды транслятор сопоставляет её с API устройства и выполняет соответствующий вызов API. Получив ответ от API устройства, транслятор отправляет его обратно серверу.

«Библиотека устройства» при наличии подключенного по USB устройства, отправляет API через Vulk-pipe, в противном случае запускается эмулятор устройства со стандартными настройками, где и инициализируются команды[4].

```
int main(int argc, char** argv) {
    (void) argc;
server_t::server_t(
    const std::string ip, const unsigned short port,
    const unsigned short num_threads) noexcept
: m_ios_acceptors{boost::make_shared<io_service>()},
  m_ios_work_acceptors{boost::make_shared<io_service::work>(& *m_ios_acceptors)},
  m_ios_executors{boost::make_shared<io_service>()},
  m_ios_work_executors{boost::make_shared<io_service::work>(& *m_ios_executors)},
  m_endpoint{ addr: address::from_string( str: ip), port_num: port},
  m_acceptor{ & *m_ios_acceptors, endpoint: m_endpoint},
  m_session{boost::make_shared<session_t>(& *m_ios_executors)},
  m_signals{ & *m_ios_acceptors, signal_number_1: SIGINT, signal_number_2: SIGTERM}
{
    // Add signal handling for graceful termination (CTRL + C)
    m_signals.async_wait( handler: boost::bind( f: &server_t::stop, a1: this));

    spdlog::info( fmt: "starting [{}:{}", ip, port);

    for (unsigned int i = 0; i < num_threads; ++i)
    {
        m_executors_thread_group.create_thread( threadfunc: boost::bind( f: &server_t::worker_thread_callback,
                                                                           a1: this,
                                                                           a2: m_ios_executors));
    }

    m_acceptor.async_accept( & m_session->get_socket(),
                            handler: boost::bind( f: &server_t::accept_handler, a1: this, a2: m_session, a3: error));

    spdlog::info( msg: "started, press CTRL+C to quit");
}
}
```

Рис. 3. Пример инициализации обработки SCPI-команды

ЗАКЛЮЧЕНИЕ

Реализация сервера протокола SCPI для измерительных устройств семейства AlmaMeter представляет собой важный шаг в улучшении и расши-

рении функциональности этих приборов. Она включает создание серверного программного обеспечения с использованием современных технологий, таких как библиотека ASIO на языке C++, обеспечивая высокую производительность и надежную передачу данных через TCP-соединения. Реализация также включает разработку интерфейса для взаимодействия с приборами по протоколу SCPI, что упрощает интеграцию новых устройств и повышает их совместимость с другими системами. В будущем можно рассмотреть добавление веб-интерфейса для удаленного управления, увеличение числа поддерживаемых устройств и улучшение средств безопасности, чтобы сделать приборы AlmaMeter еще более универсальными и востребованными.

Библиографические ссылки

1. Boost. Asio C++ Network Programming/ John Torjo, Wisnu Anggoro // Chapter 2. Understanding the Networking Concepts, С. 63 – 105
2. Mastering Binary Data Transfers With Your Instrument [Электронный ресурс]. URL: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1sl381/1SL381_1e_Binary_transfer_using_SCPI.pdf (дата обращения 30.05.2024).
3. (ASIO) ASIO.BOOST Documentation [Электронный ресурс]. URL: https://think-async.com/Asio/boost_asio_1_30_2/doc/html/boost_asio/overview.html (дата обращения 30.05.2024).
4. USB endpoints and their pipes [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/en-us/windows-hardware/drivers/usbcon/usb-endpoints-and-their-pipes_ (дата обращения 30.05.2024).