

Проектирование и исследование интерфейса защищенной системы передачи аудио-видеоинформации

А.В. Черепанов¹, А.А. Мавричев², А.Г. Долматов²,
А.С. Лучинин², И.В. Мальгин²

¹ ООО «Екатеринбург-2000»
620075, Российская Федерация, г. Екатеринбург
ул. Мамина-Сибиряка, 85

² Уральский федеральный университет им. первого Президента России Б.Н. Ельцина
620002, Российская Федерация, г. Екатеринбург
ул. Мира, 19

В настоящее время существует достаточно технических средств беспроводной передачи аудио-видеоинформации в реальном режиме времени: WiFi, WiMAX, LTE, 3G, PPJ. Несмотря на многочисленные достоинства, присущие этим системам, их эксплуатации сопутствует один общий недостаток – они не предназначены для работы в условиях сильных помех. С целью обеспечения работоспособности в сложной помеховой обстановке авторами разработана система передачи данных, использующая шумоподобные сигналы. Предлагаемая вниманию статья посвящена вопросам проектирования, настройки и исследования интерфейса для связи этой системы со стандартными источниками и приемниками аудио и видеоинформации.

Ключевые слова: шумоподобные сигналы, M-последовательности, spread spectrum, UART, RaspberryPI.

Введение

В любых системах передачи данных (проводных или беспроводных) решается противоречивая задача обеспечения помехоустойчивости и скорости передачи информации. В данной разработке предпочтение отдано помехоустойчивости, поэтому возникла проблема создания максимально медленного интерфейса формирования и передачи цифрового потока, несущего аудио и видеоинформацию. Потребовалось разработать аппаратно-программный интерфейс для передачи аудио и видео потоков от стандартных источников сигнала (микрофона и видеокамеры) к модулятору системы связи и в обратном направлении – от демодулятора до монитора со входом HDMI (или аналогичным) и громкоговорителя. Интерфейс должен работать автономно, то есть не требовать для своего функционирования обратного канала.

1. Анализ возможных вариантов решения поставленной задачи

В процессе анализа поставленной задачи, были рассмотрены следующие варианты реализации системы:

1. Реализация на ПЛИС фирмы Altera. [6]
2. Реализация на платформе Arduino [5].
3. Реализация на микрокомпьютере Raspberry Pi [2].
4. Реализация на камере LS-Y201 [4] с выходом RS-232.

Использование ПЛИС Altera признано хорошим, но сложным вариантом реализации. Решение на платформе Arduino и совместимого модуля камеры представляется довольно простым, но ввиду недостаточной производительности платы было отвергнуто. Последний вариант был принят в качестве запасного.

В качестве основного варианта реализации интерфейсного модуля выбрана схема, при которой приемник и передатчик представляют собой два одинаковых связанных между собой компьютера Raspberry Pi с необходимой периферией (рис. 1). Такая архитектура способствует модульности системы, облегчая замену и модернизацию узлов, благодаря поддержке Raspberry Pi большого количества стороннего программного обеспечения (ПО) и широкого круга периферийных устройств.

Среди преимуществ выбранного варианта следует отметить простоту управления и настройки

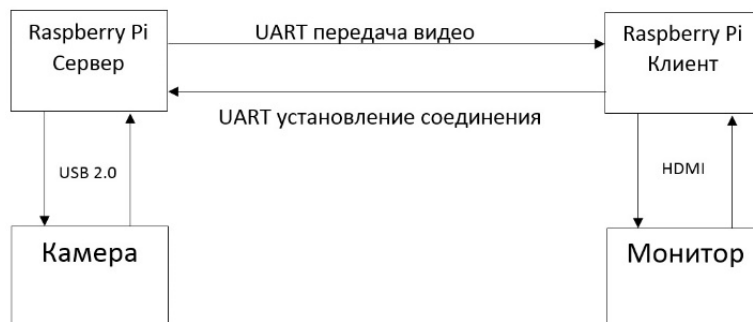


Рис. 1. Блок-схема организации канала связи

Таблица

Стеки используемых протоколов для модели OSI

Уровень модели OSI	Протокол, параметры и описание
Физический	UART 1.5 МБод/с, 1 стоп-бит, без контроля четности, низкий логический уровень: 0 В, высокий +3,3 В
Канальный	PPP без аутентификации, без проверки разрыва соединения
Сетевой	IP
Транспортный	RTP, передача данных с помощью UDP-пакетов без подтверждения о получении

микрокомпьютера, создаваемые операционной системой (ОС) на базе Linux, и хороший запас производительности, важный для обеспечения возможности последующего повышения качества передаваемого видео потока, учитывая современные ресурсоемкие кодеки.

Одним из недостатков выбранного варианта реализации является то, что микрокомпьютер Raspberry Pi не предназначен для автономного режима работы; это потребовало соответствующей настройки операционной системы.

2. Состав системы

Система состоит из следующих основных узлов:

1. Два компьютера Raspberry Pi модели B+ с операционной системой RASPBIAN (Debian Wheezy), выполняющие роли сервера и клиента.
2. USB веб-камера Logitech C170, подключенная к серверу.
3. USB звуковая карта CMedia TRUA71-USB.
4. Монитор, совместимый со стандартом HDMI, подключенный к клиенту.
5. USB ключ для выключения системы.

В окончательном варианте макета радиосистемы приемный модуль интерфейса, включающий микрокомпьютер Raspberry Pi и монитор, реализованы на базе notebook.

Применяются еще два модуля – преобразователя USB в UART и обратно.

При подаче питания на микрокомпьютер, начинается загрузка ОС. По завершению загрузки запускается скрипт, осуществляющий поиск второго компьютера, а также установление и контроль соединения. Когда соединение установлено, сервер запускает USB-камеру и начинает трансляцию, а клиент переходит в режим приема и выводит изображение на подключенный монитор. При обрыве соединения, оба компьютера начинают поиск и автоматически восстанавливают трансляцию. Как только компьютер обнаруживает USB-ключ на одном из портов, он останавливает трансляцию и завершает работу ОС.

3. Организация передачи сигнала

Передача видео от микрокомпьютера-сервера к микрокомпьютеру-клиенту на физическом уровне происходит с помощью протокола UART, поверх которого на PPP канале организована IP сеть, через нее передаются UDP-пакеты в сторону клиента, содержащие видео из камеры, аппаратно закодированное компьютером с помощью кодека h.264. Организация канала выполнена в соответствии с моделью OSI (табл.).

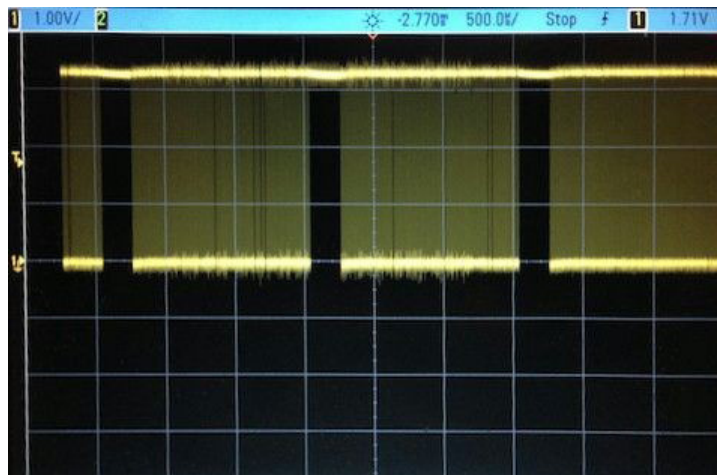


Рис. 2. Показаны периоды времени, когда передача сигнала приостанавливается

4. Результаты испытаний

По результатам разработки проведен ряд экспериментов с целью определения устойчивости работы интерфейса и выявления наилучших параметров кодека для заданной ширины пропускания канала. Заданная ширина пропускания канала ограничивается пропускной способностью системы радиосвязи и не превышает 1 Мбит/с.

В качестве кодека был выбран стандарт h.264, который аппаратно поддерживается видеочипом Raspberry Pi, что позволяет снизить нагрузку на центральный процессор микрокомпьютера при кодировании и декодировании видеопотока. Также данный стандарт предлагает относительно высокую степень сжатия видеопотока при сохранении высокого качества изображения. Трансляция, кодирование и декодирование видео организовано посредством мультимедийного фреймворка с открытым исходным кодом – GStreamer.

В ходе экспериментов по критерию визуального качества изображения подобраны следующие оптимальные параметры кодирования видеопотока:

- Разрешение: 320 × 240.
- Частота кадров: 30 в секунду.
- Битрейт: 500 Кбит/сек.

Данные значения являются оптимальными для заданных условий, так как при уменьшении битрейта ниже указанного на изображении проявляются серьезные артефакты сжатия и значительная потеря цветности. При указанных выше параметрах система вела себя устойчиво, и видеопоток передавался без существенных задержек и искажений. В ходе тестирования проблем в работе комплекса выявлено не было.

5. Оптимизация параметров цифрового потока аудио и видео данных

В ходе тестирования системы выявлено, что при установке битрейта видеопотока на уровне 800 Кбит/с и выше независимо от частоты, на которой работает UART-передатчик, возникают задержки передачи, снижающие пропускную способность интерфейса.

Для выявления причин возникшей проблемы произведен анализ структуры последовательного сигнала на выходе UART-передатчика. Полученные осциллограммы (рис. 2) свидетельствуют о наличии периодов времени, в течение которых передача приостанавливается. При этом частота, на которой работает UART-приемопередатчик в периоды нормальной передачи, соответствует установленной. При повышении частоты передачи продолжительность пауз увеличивается. Таким образом, выявлены изменения структуры сигнала в канале связи, а также их характер при различных режимах работы.

Для исключения причин, связанных с кодированием видеопотока, а также всеми сопутствующими операциями, осуществляемыми на более высоких уровнях, предложено протестировать передачу нулевой информации из системного источника нулевых байтов (/dev/zero) непосредственно в предоставляемый операционной системой интерфейс взаимодействия с асинхронным приемопередатчиком (/dev/ttyUSB). Использование описанного жестко оптимизированного метода значительно понизило потребление ресурсов передатчика, но не изменило соотношения периодов передачи и периодов паузы, что позволило исключить обозначенные выше причины. Чтобы проверить зеркальность



Рис. 2. Фотография лабораторного макета системы защищенной связи

наблюдаемого явления, выполнены те же самые действия, что и в предыдущем пункте, но на программно-аппаратной платформе приемника. Выяснено, что система, использующаяся в качестве приемника, способна передавать данные без задержек и пауз на установленной частоте и с соответствующей максимальной пропускной способностью, то есть, работает в нормальном режиме.

Очевидно, проблема связана с особенностями программно-аппаратной платформы Raspberry Pi. Поскольку для передачи данных используется внешний UART-приемопередатчик, содержащий преобразователь USB-UART, выдвинуто предположение, что проблема связана с возможными особенностями в работе микрокомпьютера Raspberry Pi с высокоскоростными USB-устройствами.

Интерфейс USB предусматривает контроль исходящей передачи данных. Для этого используется пакет типа NAK, сообщающий, что устройство временно не способно выполнять передачу (чаще всего, из-за переполнения буфера). В микрокомпьютере Raspberry Pi для предотвращения чрезмерной загрузки центрального процессора гигантским количеством процессорных прерываний, генерируемых каждый раз при получении пакета NAK, реализован механизм «NAK holdoff», на некоторое время блокирующий центральный процессор от обработки прерываний.

Причиной проблемы явилось такое программное ограничение количества процессорных прерываний в единицу времени, установленное по

умолчанию в операционной системе микрокомпьютера Raspberry Pi. Увеличение этого предела избавило от блокировок. Проверка процессорной загрузки показала, что система работает стабильно даже при пропускной способности в 3 МБод. Таким образом, в конечном счете, механизм «NAK holdoff» был полностью отключен.

В ходе тестирования также было выявлено, что при случайных потерях пакетов (например, при разрыве линии связи) соединение через некоторое время полностью прекращается. Поскольку в системе протокол RTP, используемый для передачи видеоизображения, не осуществляет обратной передачи какой-либо информации, выдвинуто предположение, что управление соединением выполняет протокол более низкого уровня.

PPP – двухточечный протокол канального уровня [7], в своем составе имеющий протокол управления линией связи LCP [8], осуществляющий первоначальное согласование двух точек, а также на всем протяжении соединения обрабатывающий эхо-запросы. Клиенты автоматически выполняют запросы (отправкой специального пакета) и ждут подтверждения эхо-ответом. При этом если эхо-ответ будет потерян, запрашивающая сторона даже после следующего ответа на запрос будет ожидать текущий ответ, и перестанет отвечать на запросы. Если ответ не приходит, через установленное время, происходит разрыв соединения.

Такой метод контроля линии подходит для проводных соединений, но оказывается непри-

менимым в беспроводной передаче, так как со-единение не является стабильным и возможны потери пакетов.

Решение состоит в том, чтобы не использовать контроль канала посредством LCP.

Заключение

В процессе исследовательской работы был разработан, протестирован интерфейс системы защищенной связи для передачи видеоинформации. Устранены проблемы, возникшие в процессе тестирования. Экспериментальная проверка разработанной системы показала устойчивую работоспособность при передаче видеоинформации со скоростью 500 кБит/с при отношении сигнал/шум -10 дБ. Ширина спектра сигнала в эксперименте составляла 56 МГц. Фотография лабораторного макета системы связи показана на рис. 3. Основные характеристики системы связи и принципы ее функционирования изложены в [1; 3].

Список литературы

1. Реализация защищенной системы радиосвязи по технологии SDR на основе трансивера AD9361 / А.Г. Долматов [и др.] // СВЧ-техника и телекоммуникационные технологии – КРЫМИКО 2015. Материалы докладов 25-й Международной Крымской конференции: в двух томах. 6–12 сентября 2015. Севастополь.
2. The Official Raspberry Pi Projects Book. URL: https://www.raspberrypi.org/magpi-issues/Projects_Book_v1.pdf (дата обращения: 11.11.16).
3. Защищенная система радиосвязи на основе кодированного по времени, частоте и фазе сигнала / А.Г. Долматов [и др.] // Компьютерный анализ изображений: Интеллектуальные решения в промышленных сетях: сборник трудов по материалам I Международной конференции 5–6 мая 2016 / под общ. ред. А.Г. Тягунова. Екатеринбург: УМЦ УПИ, 2016. 234 с.
4. Step-by-Step Tutorial for Using LinkSprite JPEG Color Camera Serial UART Interface Model LS-Y201. URL: https://www.sparkfun.com/data-sheets/Sensors/Imaging/JPEG_UART_camera_tutorial.pdf (дата обращения: 05.09.16).
5. Getting Started with Arduino and Genuino products. URL: <https://www.arduino.cc/en/Guide/HomePage> (дата обращения: 05.09.16).
6. Стешенко В.Б. Оценка производительности новых семейств ПЛИС ALTERA для применения в системах связи с OFDM // Цифровая обработка сигналов. 2005. № 3. С. 48–50.
7. Point-to-Point (PPP) Protocol Field Assignments. URL: <https://www.iana.org/assignments/ppp-numbers/ppp-numbers.xhtml> (дата обращения: 11.01.17).
8. PPP Link Control Protocol (LCP). URL: http://www.tcpipguide.com/free/t_PPPLinkControlProtocolLCP.htm (дата обращения: 11.01.17).

Research and design of interface for audio and video secure communication system

*A.V. Cherepanov, A.A. Mavrichev, A.G. Dolmatov,
A.S. Luchinin, I.V. Malygin*

Currently, audio-video information can be transmitted by various communication systems such as WiFi, WiMAX, LTE, or 3G. Despite the fact that these systems fulfill their functions quite well, they are united by a common drawback - they are not designed to work in conditions of strong interference. To transfer audio-video information in the presence of strong interference, the authors have developed a data transfer system using spread spectrum signals. This article is devoted to the design, configuration and interface study of how to link this system with standard sources and receivers, both audio and video.

Keywords: spread spectrum, M-sequence, interface, UART, RaspberryPI.
