Содержание

[Введение 2](#_Toc487801985)

[1. Аппаратные средства 3](#_Toc487801986)

[1.1. Raspberry Pi 3 model B 3](#_Toc487801987)

[1.2. Genius веб-камера 5](#_Toc487801988)

[2. Программные средства 7](#_Toc487801989)

[2.1. Операционная система Rasbian 7](#_Toc487801990)

[2.2. Необходимое программное обеспечение для мультимедиа 8](#_Toc487801991)

[3. Настройка сервера и трансляции мультимедиа 9](#_Toc487801992)

[4. Создание графического интерфейса для клиентского приложения и тестирование 11](#_Toc487801993)

[Заключение 13](#_Toc487801994)

[Список используемых источников 15](#_Toc487801995)

[Приложение А. 15](#_Toc487801996)

# 

# Введение

В наши дни потоковое воспроизведение видео уже вошло в привычку большинства пользователей. Сервер ожидает соединения на определенном порту, а затем всем подключившемся клиентам передает определенный участок видео. Существует возможность настроить вещание в локальную сеть для устройств, передачу на телевизор, плеер или смартфон.

Дистанционные мобильные роботы полезны в военных целях, космических или местах, условия которых затрудняют работу для человека. Удаленное управление дает возможность выполнить необходимую работу на расстоянии, при этом контролировать процесс, наблюдая за движениями робота при помощи бортовой камеры наблюдения.

Дистанционное управление мобильным роботом заключается в передаче управляющего сигнала от оператора, находящемуся на расстоянии от робота, непосредственно к самому роботу. Существует несколько каналов для организации управления на расстоянии. Однако в рамках данной работы будет использовано управление по сети, по текстовому протоколу UDP (протокол пользовательских датаграмм) для управления роботом и протокол RTSP (Потоковый протокол реального времени) для трансляции потокового мультимедиа и позволяющий удалённо управлять потоком данных с сервера.

Такое управление предпочтительнее, так как оператор и робот не всегда могут находиться в зоне прямой видимости, что позволяет использовать робота на весьма больших расстояниях. Однако, в определенных случаях, управление роботом осуществляется глобальную сеть Интернет, что накладывает ряд требований к программной архитектуре системы управления.

Для создания данной системы была использована архитектура клиент-сервер, суть которой состоит во взаимодействии двух программ – на стороне клиента (посылает запросы) и на стороне сервера (выполняет сервисные функции по запросу). Клиентское программное обеспечение (ПО) дает возможность человеку получить доступ к серверному ПО, которое находится в режиме ожидания. После получения сигнала, сервер начинает выполнять полученную команду и будет действовать, согласно прописанной программе.

Целью данной работы является создание системы удаленного управления, которая обеспечивает дистанционное управление мобильной роботизированной платформой, получая потоковый мультимедиа от веб-камеры, которая установлена на роботе и работает как удаленный сервер для передачи информации по сети Интернет.

# Аппаратные средства

## Raspberry Pi 3 model B

Одноплатный компьютер Raspberry Pi дешевый, мощный и очень функциональный, благодаря этим особенностям быстро набрал популярность среди пользователей и за эти годы появилось очень много проектов с использованием этого устройства. Rаspberry Pi уже давно вышел за рамки своего первого предназначения - обучения детей и студентов информатике.

Raspberry PI 3 Model B 1Gb улучшенная и доработанная модель, которая отличается обновленным чипом Broadcom 2837 quad-cоre АRMv8 Cortex-A53 64bit (1,2GHz), наличием встроенного модуль Wi-Fi по стандарту (802.11n) и Bluetooth 4.1. Таким образом, новая модель стала на 50% производительнее своего предшественника PI2. На сегодняшний день для платформы Raspberry PI существует большой выбор операционных систем, однако в данной работе использовалась Rasbian.

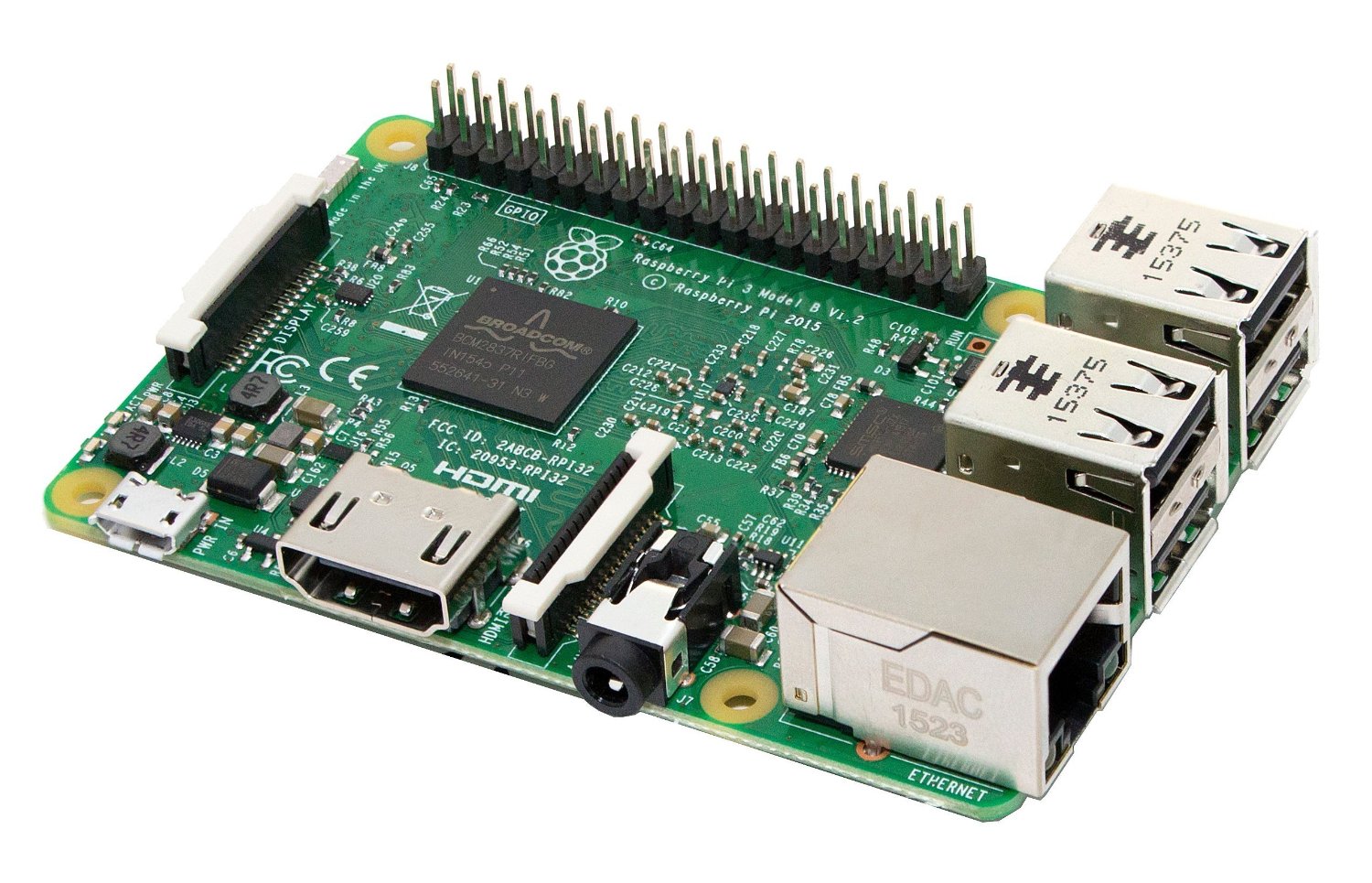


Рисунок 1 – Raspberry Pi 3.

Главное преимущество Raspberry Pi — 40 контактов ввода/вывода общего назначения (GPIO), к которым есть возможность подключать периферию для взаимодействия: [исполнительные устройства](http://amperka.ru/collection/motor-drivers), любые [сенсоры](http://amperka.ru/collection/sensors) и всё, что работает от электричества [1].

Техническое описание:

* Операционная система: Загружается с сайта производителя (NOOBS, Raspbian и др.)
* Процессор компьютера: Broadcom BCM2837 ARMv8 Cortex-A53 (1.2 ГГц)
* Тип видео: Встроенное
* Оперативная память: 1 ГБ LPDDR2 900 МГц
* Стандарты Wi-Fi IEEE 802.11n (BCM43438 2.4 ГГц)
* Интегрированный Bluetooth: Есть, Bluetooth 4.1
* Сеть: 10/100 Мбит/сек
* Слоты для карт памяти microSDHC, microSD
* Разъемы компьютера: 4 x USB 2.0, HDMI, RJ-45, 1 x USB microB, AV-out, GPIO, разъем для камеры
* Охлаждение: Безвентиляторная система охлаждения
* Потребление энергии: 0.5 - 1 Вт
* Протоколы: RDP
* Прочее: Поддержка работы с USB-хабами

## 1.2. Genius веб-камера

Малоразмерная видеокамера, способная в реальном времени фиксировать изображения, предназначенные для дальнейшей передачи по сети Интернет. Веб-камеры, доставляющие изображения через [интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82), закачивают изображения на [веб-сервер](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), либо по запросу, либо непрерывно, либо через регулярные промежутки времени. Это достигается путём подключения камеры к [компьютеру](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) или благодаря возможностям самой камеры.



Рисунок 2 – веб-камера Genius.

Технические характеристики:

* веб-камера с матрицей 2 МП
* разрешение видео 1600x1200
* подключение через USB 2.0
* ручная фокусировка

Камера подключает по USB-кабелю в разъемы платы, как показано на схеме подключения на Рисунке 3. Веб-камера установлена непосредственно на роботе и ведет вещание, а плата в свою очередь транслирует этот видеопоток в сеть, к которой можно обратиться по специальному URL адресу, прописанному в конфигурации и по которому клиент будет делать запрос.

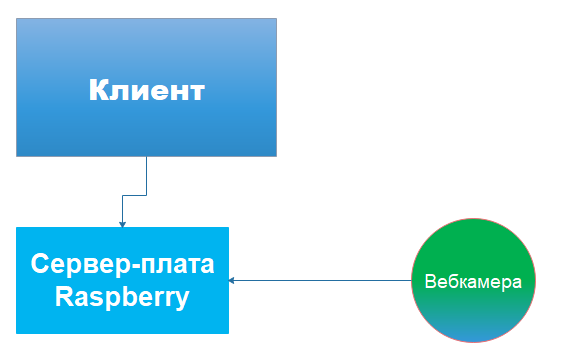


Рисунок 3 – схема подключения аппаратной части.

## 

## 2. Программные средства

## 2.1. Операционная система Rasbian

Raspbian - это официальная операционная система (ОС) для Raspberry Pi, она разработана специально для этого устройства и имеет все необходимое программное обеспечение. Raspbian основана на ARM версии Debian 8 Jessie [2]. Для установки ОС необходимо произвести монтирование ее образа на MicroSD-карту, которая используется одноплатным компьютером как ПЗУ (постоянное запоминающее устройство).

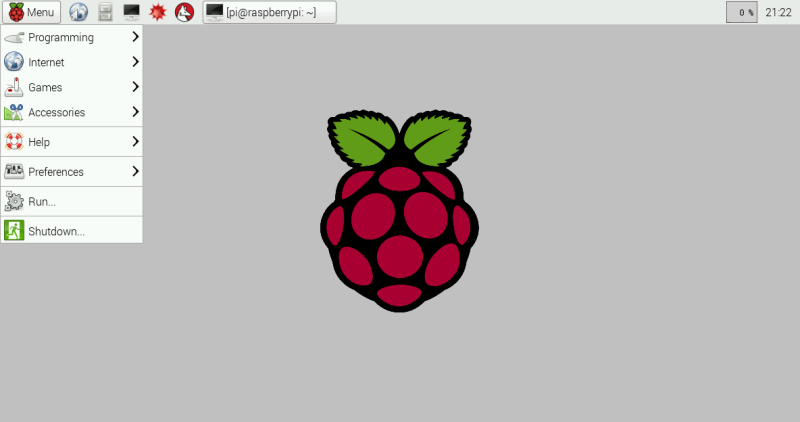


Рисунок 4 – внешний вид интерфейса ОС Rasbian.

Raspbian – свободная ОС основанная на Debian и специально оптимизирована под аппаратное обеспечение одноплатного компьютера Raspberry Pi 3. Тем не менее, она содержит набор базовых программ и утилит, которое делает интерфейс ОС удобнее и понятнее.

## 2.2. Программное обеспечение для потокового мультимедиа

Наиболее подходящая программа видеонаблюдения, позволяющая транслировать поток видео в сеть – Motion. Легковесная программа для семейства UNIX-систем, написанная специально для работы с веб- и телевизионными камерами через video4linux (интерфейс прикладного программирования видео захвата для ОС linux) [3]. Данная программа будет снимать в онлайн режиме и пускать трансляцию в локальную сеть, доступ к которой будет доступен по URL ссылке.

В Linux-системах присутствует командная строка - Терминал, которая служит для выполнения нужных команд, помогая очень эффективно управлять всей системой. Через терминал будут прописаны основные конфигурации для корректной работы программы Motion с веб-камерой.

Для воспроизведения потокового видео будет использована мультимедийная программа VLC (от VideoLAN Client) —[свободный](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [кроссплатформенный](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [медиаплеер](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D0%B0%D0%BF%D0%BB%D0%B5%D0%B5%D1%80), который позволяет принимать поток по сети, а также данная программа может быть имплементирована в клиентское приложение [4]. Данная программа будет упрощать управление роботом на больших дистанциях, так как она будет встроена сразу в клиентское графическое приложение.

## 3. Настройка сервера и трансляции мультимедиа

В системе используется сервера для обеспечение потоковой трансляции мультимедиа. Для этого необходимо настроить программу трансляции так, чтобы она корректно транслировала мультимедиа и работала как сервер, т.е. при включении платы в сеть сразу же начинала трансляцию в сеть, нужно настроить файл конфигурации программы, в котором будет прописаны IP- адрес потока нашей камеры, ее разрешение и частоту кадров, а также видео устройство для захвата и используемый порт. Файл конфигурации полностью регулирует работу видеопотока, поэтому от его настройки будет зависеть качество воспроизведения.

На рис. 5 показано подключение Raspberry Pi и веб-камеры Genius. Так как компьютер уже настроена как рабочий сервер, то нет необходимости подключать дополнительно устройства периферии.

Рисунок 5 – подключение камеры к плате.

Передача мультимедиа использует протокол RTSP - потоковый протокол реального времени, специально разработанный для использования в системах, работающих с мультимедийными данными, и позволяющий удаленно управлять поток данных с сервера [5]. В случае с Raspberry – это оптимальный вариант для использования.

Для подключения к потоку будет достаточно ввести в браузере IP-адрес и порт подключения, после чего будет возможность просмотра потока. На рисунке 6 показано такое подключение, что значит работа на сервере и трансляция видеопотока в сеть настроены и функционируют.

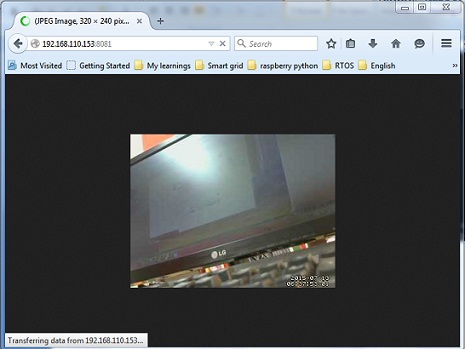


Рисунок 6 – изображение с веб-камеры, подключенной к Raspberry Pi.

## 

## 4. Создание графического интерфейса для клиентского приложения и тестирование

Наиболее удобная среда разработки для создания удобного приложения с понятным интерфейсом – Visual Studio. Она позволяет создать программы для любых задач. В рамках работы было создано графическая программа, позволяющая управлять роботизированной платформой, а также принимает данные из локальной сети, в нашем случае это видеопоток, передаваемый с сервера Raspberry.

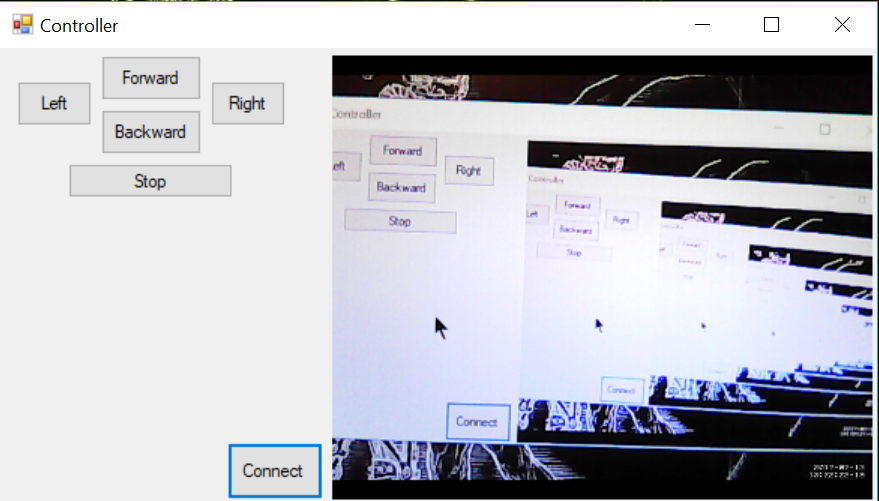


Рисунок 7 – Внешний вид графического приложения.

На рисунке 7 продемонстрирован внешний вид рабочего приложения. Интерфейс программы устроен так, что при нажатии на кнопку “Connect” идет подключение к серверу и транслируемый видеопоток открывается в клиентской программе. Таким образом, идет трансляция с камеры в настоящее время с небольшой задержкой. Остальные кнопки предназначены для управления роботом, который также использует архитектуру клиент-сервер.

Видео транслируется с задержкой около 3 секунд. При правильно настроенной конфигурации задержка уменьшается, но все же присутствует. Данная проблема, как правило, часто появляются в виду проблем с получением информационного потока. В основном, [мультимедиа](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0) информация занимает большие объёмы, так что затраты на хранение и передачу подобной информации всегда велики. Поэтому, в большинстве случаев, передаваемая в поток информация [сжимается](https://ru.wikipedia.org/wiki/%D0%A1%D0%B6%D0%B0%D1%82%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) при передаче в сеть вещания, что может вести за собой задержку.

Листинг кода клиентского приложения представлен в приложении А.

# 

# Заключение

В результате выполнения задания была разработана система трансляции потокового мультимедиа для дистанционно управляемого робота. Такая система может быть применена в системах удаленного управления для различных систем. Стоит отметить, что использование системы довольно упрощено, потому может быть использовано даже человеком, не знакомым с различными аспектами разработки программного обеспечения. В рамках выполненной работы были получены следующие результаты:

* реализована архитектура клиент – сервер для взаимодействия двух компьютеров
* разработано программное обеспечение сервера, транслирующая видеопоток в настоящем времени по сети Интернет
* расширен функционал клиентского программного обеспечения: добавлена возможность получения видеопотока по сети и его воспроизведения в клиентском приложении с графическим интерфейсом пользователя
* определены оптимальные параметры настройки используемых средств

Также, были проведены тестирование и отладка, в результате которых была подтверждена работоспособность системы, выявлены и устранены неполадки при работе. При дальнейшей работе над системой, планируется ее

оптимизация и улучшение функциональности, что позволит ее использовать в дальнейшем в других системах удаленного управления.

# Список используемых источников

1. [Амперка](http://www.wikipedia.ru) [Электронный ресурс]: Raspberry Pi 3 Model B. URL: http://amperka.ru/product/raspberry-pi-3-model-b (дата обращения: 07.07.2017)
2. Losst [Электронный ресурс]: Лучшие операционные системы Raspberry Pi 3. URL: <http://losst.ru/luchshie-operatsionnye-sistemy-raspberry-pi-3> (дата обращения: 07.07.2017)
3. Википедия [Электронный ресурс] : Motion (программа видеонаблюдения). URL: <http://ru.wikipedia.org/wiki/Motion> (дата обращения: 07.07.2017)
4. Википедия [Электронный ресурс] : VLC (медиаплеер). URL: <http://ru.wikipedia.org/wiki/VLC_(медиаплеер)> (дата обращения: 07.07.2017)
5. Википедия [Электронный ресурс] : RTSP. URL: <http://ru.wikipedia.org/wiki/>RTSP (дата обращения: 07.07.2017)

# Приложение А.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

using System.Net;

using System.Net.Sockets;

using AxAXVLC;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

static string remoteAddress; // хост для отправки данных

static int remotePort; // порт для отправки данных

static void client()

{

remoteAddress = "192.168.137.157"; // адрес, к которому мы подключаемся

remotePort = 4210; // порт, к которому мы подключаемся

}

static void SendMessage(string message)

{

UdpClient sender = new UdpClient(); // создаем UdpClient для отправки сообщений

try

{

while (true)

{

byte[] data = Encoding.Unicode.GetBytes(message);

sender.Send(data, data.Length, remoteAddress, remotePort); // отправка

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

finally

{

sender.Close();

}

}

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

SendMessage("w");

}

private void button2\_Click(object sender, EventArgs e)

{

SendMessage("a");

}

private void button3\_Click(object sender, EventArgs e)

{

SendMessage("d");

}

private void button1\_Click\_1(object sender, EventArgs e)

{

SendMessage("s");

}

private void button1\_Click\_2(object sender, EventArgs e)

{

vlc\_window.playlist.add(@"http://169.254.6.231:8081/");//URL адрес

vlc\_window.playlist.playItem(0); // запуск видеопотока

}

private void Stop\_Click(object sender, EventArgs e)

{

SendMessage("v");

}

}

}

}