**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Доцент департамента  программной инженерии  факультета компьютерных наук  канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. Л. Макаров  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.02.07-01 81 01-1 | | **СИСТЕМА СЛЕЖЕНИЯ ЗА УДАЛЁННЫМ ПОМЕЩЕНИЕМ**  **НА ОСНОВЕ ИНТЕРНЕТА ВЕЩЕЙ**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.02.07-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ184  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /А. Д. Курылев /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | |
|  | | |
|  | |  |

**Москва 2019**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.02.07-01 81 01-1 | | УТВЕРЖДЁН  RU.17701729.02.07-01 81 01-1-ЛУ  **СИСТЕМА СЛЕЖЕНИЯ ЗА УДАЛЁННЫМ ПОМЕЩЕНИЕМ**  **НА ОСНОВЕ ИНТЕРНЕТА ВЕЩЕЙ**  **Пояснительная записка**  **RU.17701729.02.07-01 81 01-1**  **Листов 20** | | | | |
|  |  | | | |
|  | | | |
|  | | | | |
|  | | |  | |

**Москва 2019**

**СОДЕРЖАНИЕ**

[1. ВВЕДЕНИЕ 3](#_Toc9435919)

[1.1. Наименование программы 3](#_Toc9435920)

[1.2. Условное обозначение 3](#_Toc9435921)

[1.3. Документ, на основании которого ведется разработка 3](#_Toc9435922)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 4](#_Toc9435923)

[2.1. Функциональное и эксплуатационное назначение 4](#_Toc9435924)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 5](#_Toc9435925)

[3.1. Постановка задачи на разработку программы 5](#_Toc9435926)

[3.2. Описание алгоритма и функционирования программы 6](#_Toc9435927)

[3.2.1. Главная форма 6](#_Toc9435928)

[3.2.1.1. Общий вид 6](#_Toc9435929)

[3.2.1.2. Показ транслируемого видео 6](#_Toc9435930)

[3.2.1.3. Сохранение изображения с камеры 7](#_Toc9435931)

[3.2.1.4. Поворот камеры 8](#_Toc9435932)

[3.2.1.5. Режим обнаружения движений 9](#_Toc9435933)

[3.2.1.6. Сообщения оператору 10](#_Toc9435934)

[3.2.2. Форма настроек 11](#_Toc9435935)

[3.2.2.1. Общий вид 11](#_Toc9435936)

[3.2.2.2. Экспорт файлов настройки 11](#_Toc9435937)

[3.2.2.3. Импорт файлов настройки 12](#_Toc9435938)

[3.2.2.4. Сообщения оператору 13](#_Toc9435939)

[3.2.3. Форма с файлами 15](#_Toc9435940)

[3.2.3.1. Общий вид 15](#_Toc9435941)

[3.2.3.2. Загрузка файлов 16](#_Toc9435942)

[3.2.3.3. Удаление файлов 16](#_Toc9435943)

[3.2.3.4. Сообщения оператору 17](#_Toc9435944)

[3.2.4. Форма информации 18](#_Toc9435945)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных 18](#_Toc9435946)

[3.4. Описание и обоснование выбора состава технических и программных средств 19](#_Toc9435947)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 20](#_Toc9435948)

[4.1. Экономические преимущества разработки 20](#_Toc9435949)

[5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 21](#_Toc9435950)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 22](#_Toc9435951)

1. ВВЕДЕНИЕ
   1. Наименование программы

«Система слежения за удаленным помещением на основе интернета вещей».

* 1. Наименование на английском

«IoT System for Spying on Remote Apartment».

* 1. Условное обозначение

«IoT Room Monitoring».

* 1. Документ, на основании которого ведется разработка

Приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» И. В. Аржанцева № 2.3-02/1012-0 2 от 10.12.2018«Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук».

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ
   1. Функциональное и эксплуатационное назначение

Программа несет в себе функцию трансляции видео в интернет с возможностью записи и последующего просмотра. Может эксплуатироваться при наличии устройства на базе ОС Windows и с доступом в интернет. Ее можно использовать как в качестве охранной системы, так и в целях организации наблюдения за собственностью. Конечными потребителями являются пользователи без специальной подготовки.

* 1. **Характеристика области применения**

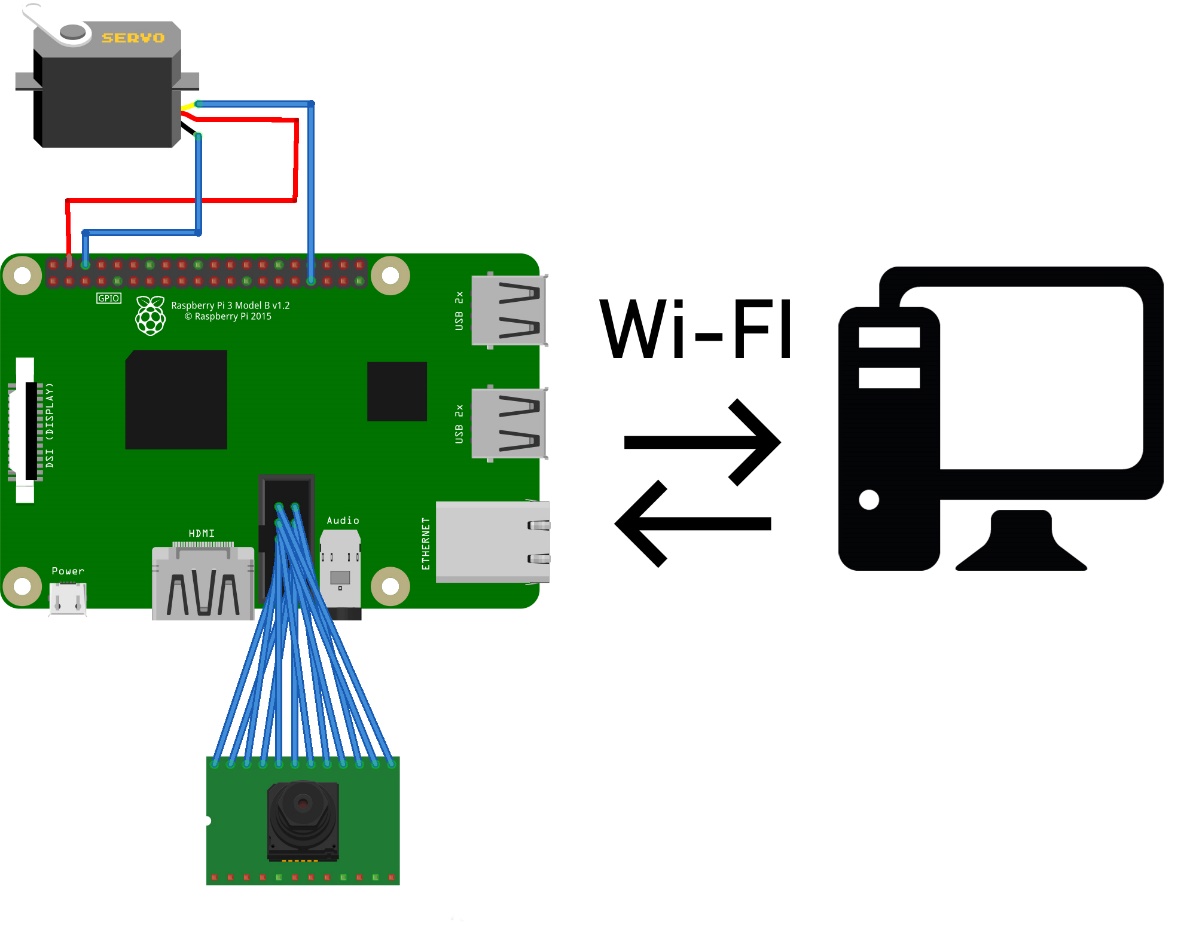
Программа может применяться как в охранной области, так и для персонального использования в наблюдении за личным помещением.

1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ
   1. Постановка задачи на разработку программы

Программа реализует следующие функции:

1. Показ транслируемого в интернет видео;
2. Сохранение кадров с камеры в формате .jpg и .mp4;
3. Поворот камеры по команде пользователя;
4. Режим обнаружения движений;
5. Импорт и экспорт файлов настройки.
   1. Описание архитектуры программы

На *рис. 1* показана архитектура системы программы, где в левой части находится Raspberry Pi 3 с различными модулями, а справа персональный компьютер.

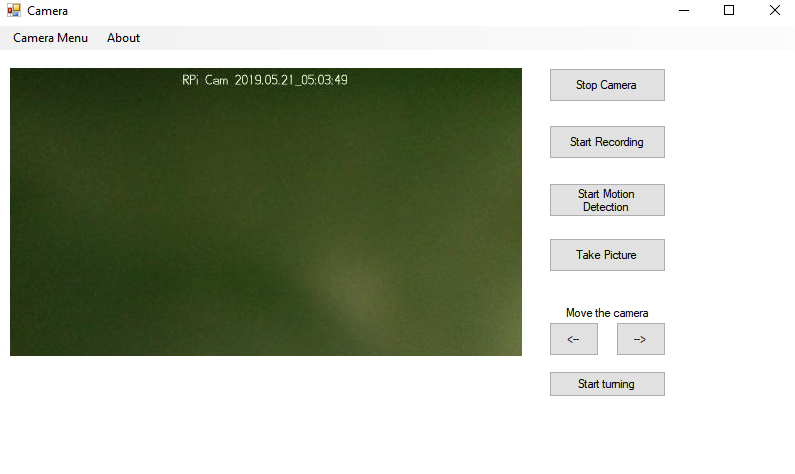


**Рисунок 1 –** «Архитектура системы»

* 1. Алгоритм работы программы
     1. Запись видео

1. Начало алгоритма
2. Запуск программы
3. Нажать кнопку “Connect”
4. Если подключение невозможно, конец алгоритма
5. Иначе, нажать кнопку “Start Recording”
6. Пока нужно записывать видео, не нажимать кнопку “Stop Recording”
7. Конец алгоритма
   * 1. Загрузка видео
8. Начало алгоритма
9. Запуск программы
10. Нажать кнопку “Connect”
11. Если подключение невозможно, конец алгоритма
12. Иначе, открыть форму с файлами, нажав Ctrl + F
13. Выбрать необходимые файлы для загрузки
14. Нажать кнопку “Download Selected”
15. Выбрать путь загрузки
16. Нажать ОК
17. Конец алгоритма
    1. Описание функционирования программы
       1. Показ транслируемого видео

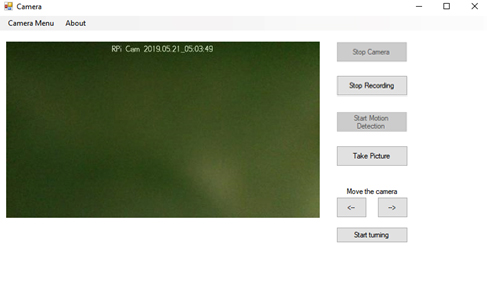
На *рис. 2* демонстрируется трансляция видео из интернета на Windows Form.



**Рисунок 2 –** «Трансляция видео»

* + 1. Сохранение изображения с камеры

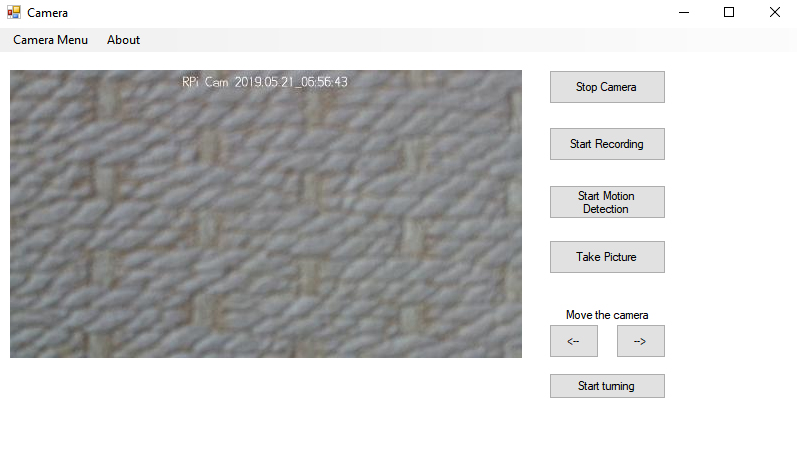
По нажатии на кнопки “Start Recording” или “Take Picture” происходит запись видео или фото соответственно, что и продемонстрировано на *рис. 3.*



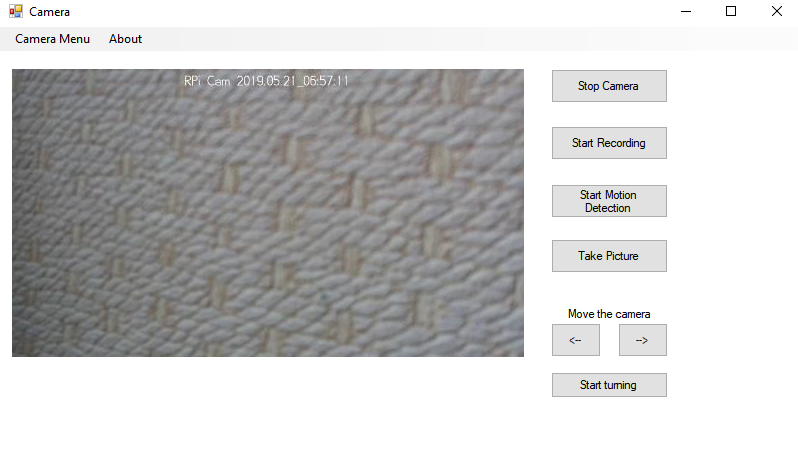
**Рисунок 3 –** «Запись видео»

* + 1. Поворот камеры

На *рис. 4* и *рис. 5* показано состояние камеры до и после нажатия кнопки поворота влево.



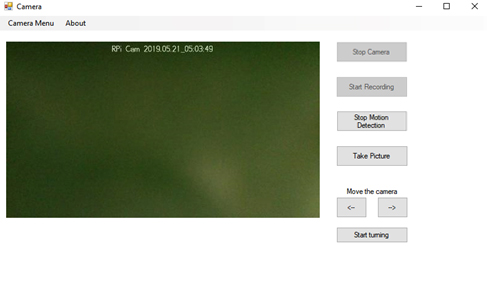
**Рисунок 4 –** «Изображение до поворота»



**Рисунок 5 –** «Изображение после поворота»

* + 1. Режим обнаружения движений

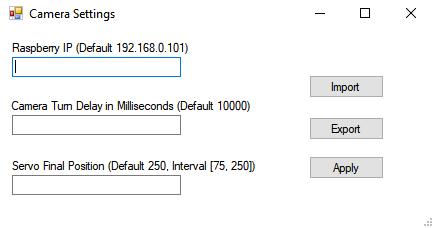
На *рис. 6* демонстрируется работа режима “Motion Detection” при нажатии на соответственную кнопку.



**Рисунок 6 –** «Режим обнаружения движений»

* + 1. Экспорт и импорт файлов настройки

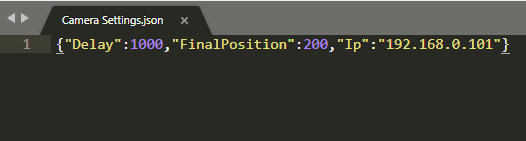
Общий вид формы настроек изображен на *рис. 7.*



**Рисунок 7 –** «Форма настроек»

* + - 1. Экспорт файлов настройки

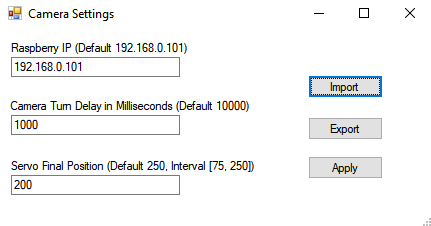
На *рис. 8* показан файл, получившийся в результате сериализации полей элементов TextBox с помощью JsonSerializer



**Рисунок 8 –** «Файл настроек»

* + - 1. Импорт файлов настройки

В результате импорта из файла, показанного на *рис. 8*, получается заполнение полей, изображенное на *рис. 9.*



**Рисунок 9 –** «Импорт из файла»

* 1. Описание и обоснование выбора метода организации входных и выходных данных

Программа может загружать файлы формата .json, .jpg, .mp4 для хранения настроек, изображений и видео соответственно. Пользователь может самостоятельно указать путь к этим файлам на своем компьютере. В качестве входных данных программа принимает файлы формата .json для импорта сохраненных настроек. Описанные методы работы с данными выбраны исходя из простоты использования для пользователей.

* 1. Описание и обоснование выбора состава технических и программных средств

- процессор не ниже Intel Pentium N3520 или аналогичный с тактовой частотой не ниже 1 ГГц;

- 64 Мб ОЗУ или более;

- жесткий диск с объемом свободной памяти не менее 512 Мб;

- монитор, клавиатура;

- тачпад или компьютерная мышь;

- разъем USB 2.0, Ethernet;

- Raspberry Pi 3, Raspberry Pi Camera V2.1, сервомотор Tower Pro SG90;

- ОС Raspbian [12];

- операционная система Windows 7 или выше;

- библиотека Microsoft .NET Framework 3.5 или выше;

- модуль камеры RPi-Cam-Web-Interface 6.4 [10], библиотека Servoblaster [11], библиотека AngleSharp [13].

При использовании как процессора с более низкой частотой, так и компьютер с меньшим количеством ОЗУ, возможна некорректная работа программы. Жесткий диск с объемом свободной памяти 512 Мб нужен для установки программы и возможности сохранения медиа файлов с сервера. Монитор, клавиатура и компьютерная мышь (тачпад) необходимы для взаимодействия с программой. USB-разъем необходим для передачи программы, а Ethernet для доступа в интернет. Raspberry Pi 3, Raspbian, Raspberry Pi Camera V2.1, сервомотор Tower Pro SG90 необходимы для корректной работы сервера, так как программа разрабатывалась именно под эти инструменты. Модуль RPi-Cam-Web-Interface служит для упрощенной работы с камерой Raspberry Pi, а библиотека Servoblaster для работы с сервомотором. Библиотека AngleSharp нужна для работы с html-тегами.

1. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ
   1. Предполагаемая потребность

Данная программа будет полезна для людей, которые хотят установить наблюдение за своим помещением с использованием Raspberry Pi 3 и приложения на Windows Forms.

* 1. Экономические преимущества разработки

После быстрого анализа рынка можно понять, что стоимость данного проекта сильно ниже аналогов за счет недорогих технических решений (Raspberry Pi 3 и Raspberry Camera V2.1), а также из-за низкой стоимости разработки. Например, аналог [14] имеет цену, намного превышающую представленный комплект программы.

1. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способ

10. Библиотека RPi-Cam-Web-Interface - <https://github.com/silvanmelchior/RPi_Cam_Web_Interface> (Дата обращения: 02.03.2019, режим доступа: свободный)

11. Библиотека Servoblaster - <https://github.com/richardghirst/PiBits/tree/master/ServoBlaster> (Дата обращения: 02.03.2019, режим доступа: свободный)

12. ОС Raspbian - <https://www.raspberrypi.org/downloads/raspbian/> (Дата обращения: 02.03.2019, режим доступа: свободный)

13. Библиотека AngleSharp - <https://github.com/AngleSharp> (Дата обращения: 02.02.2019, режим доступа: свободный)

14. Комплект видеонаблюдения – <https://carcam.ru/product/carcam-videokit-5m-1.html> (Дата обращения: 20.05.2019, режим доступа: свободный)

15. Программа Fritzing - <http://fritzing.org> (Дата обращения: 20.05.2019, режим доступа: свободный)

1. ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

|  |  |
| --- | --- |
| Класс | Назначение |
| Fieldset | Представляет собой удобный для обработки класс, отражающий важные для программы свойства html-тега fieldset |
| SettingsClass | Представляет собой класс, содержащий все необходимые поля и свойства для реализации настроек приложения |
| MainFom | Представляет класс главной формы, в которой совершаются все основные действия, такие, как нажатие кнопок, открытие других формы, запись видео и фото |
| SettingsForm | Представляет собой класс формы настроек, в которой происходит выбор IP и других важных параметров |
| FilesForm | Представляет собой класс формы файлов, в которой можно просмотреть, скачать или удалить все доступные файлы. |
| AboutForm | Представляет собой класс формы информации, в которой представлены все необходимые сведения о проекте |

1. ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, СВОЙСТВ И МЕТОДОВ

MainForm.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| buttons | Button[] | private | Массив кнопок | Объявлен для более удобного доступа к свойствам кнопок |
| cameraButtons | Button[] | private | Массив кнопок | Объявлен для более удобного доступа к свойствам кнопок |
| uri | string | public static | Адрес Raspberry Pi | Задает IP Raspberry |
| fw | string | private | Вспомогательный адрес Raspberry Pi | Нужен для изменения и передачи в FilesForm |
| client | HttpClient | private | Клиент | Связывается с сервером и отправляет GET запросы |
| flag | int | private | Счетчик | Служит для корректной работы поворота сервомотора |
| finalPosition | int | public static | Конечное положение сервомотора | Объявлен для расчета количества поворотов |
| turnValue | int | private | Количество поворотов | Нужен для корректной работы поворота сервомотора |
| turningTimerDelay | int | public static | Задержка | Служит для изменения задержки между поворотами сервомотора |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| MainCameraForm\_Load | void | private | Обработчик события загрузки формы | Отвечает за начальное состояние кнопок и установку интервала таймера |
| connectButton\_Click | void | async private | Обработчик события нажатия на кнопку "Connect" | Делает кнопки активными и запускает таймер обновления изображения |
| stopCameraButton\_Click | void | async private | Обработчик события нажатия на кнопку "Stop Camera" | Меняет состояния кнопок и посылает запрос об остановке камеры на сервер |
| startCameraButton\_Click | void | async private | Обработчик события нажатия на кнопку "Start Camera" | Меняет состояния кнопок и посылает запрос о запуске камеры на сервер |
| stopRecordingButton\_Click | void | async private | Обработчик события нажатия на кнопку "Stop Recording" | Меняет состояния кнопок и посылает запрос об остановке записи на сервер |
| startRecordingButton\_Click | void | async private | Обработчик события нажатия на кнопку "Start Recording" | Меняет состояния кнопок и посылает запрос о начале записи на сервер |
| stopMDButton\_Click | void | async private | Обработчик события нажатия на кнопку "Stop Motion Detection" | Меняет состояния кнопок и посылает запрос об остановке режима Motion Detection на сервер |
| startMDButton\_Click | void | async private | Обработчик события нажатия на кнопку "Start Motion Detection" | Меняет состояния кнопок и посылает запрос о запуске режима Motion Detection на сервер |
| takePictureButton\_Click | void | async private | Обработчик события нажатия на кнопку "Take Picture" | Посылает запрос о съемки фотографии на сервер |
| moveLeftButton\_Click | void | async private | Обработчик события нажатия на кнопку "Move Left Button" | Посылает запрос о смене состояния сервомотора на сервер |
| moveRightButton\_Click | void | async private | Обработчик события нажатия на кнопку "Move Right Button" | Посылает запрос о смене состояния сервомотора на сервер |
| StartTurningButton\_Click | void | async private | Обработчик события нажатия на кнопку "Start Turning" | Меняет интервал таймера и состояние кнопок, отправляет запрос на сервер об изменении положения сервомотора, запускает таймер |
| stopTurningButton\_Click | void | async private | Обработчик события нажатия на кнопку "Stop Turning" | Останавливает таймер и меняет состояние кнопок |
| settingsToolStripMenuItem\_Click | void | private | Обработчик события нажатия на элемент "Settings" меню | Открывает форму SettingsForm и меняет некоторые настройки, если была нажата кнопка "Apply" |
| filesToolStripMenuItem\_Click | void | private | Обработчик события нажатия на элемент "Files" меню | Открывает форму FilesForm |
| aboutToolStripMenuItem\_Click | void | private | Обработчик события нажатия на элемент "About" меню | Открывает форму AboutForm |
| informationToolStripMenuItem\_Click | void | private | Обработчик события нажатия на элемент "Information" меню | Открывает форму AboutForm |
| cameraPictureTimer\_Tick | void | private | Обработчик события тика таймера | Пробует загрузить изображение с камеры, если не удается, деактивирует кнопки и уведомляет пользователя |
| turningTimer\_Tick | void | async private | Обработчик события тика таймера | Отслеживает количество поворотов, отправляется запросы на сервер о повороте сервомотора, меняет направление поворота |

**FilesForm.cs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| client | HttpClient | private | Клиент | Служит для связи с сервером |
| uri | string | public | IP Raspberry Pi | Адрес Raspberry |
| url | string | public | Хранилище файлов | Ссылка, по которой можно получить доступ к файлам |
| gbArray | GroupBox[] | private | Массив GroupBox | Объявлен для упрощенной работы с элементами GroupBox |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| FillThePanel | void | async private | Заполняет панель файлов | Нужен для поучения, обработки и отображения медиа файлов с сервера |
| GetFieldsets | Fieldset[] | private static | Возвращает массив Fieldset | Объявлен для упрощенного получения элементов Fieldset со страницы |
| FilesForm\_Load | void | private | Обработчик события загрузки формы | Вызывает метод FillThePanel, заполняющий форму |
| selectButton\_Click | void | private | Обработчик события нажатия на кнопку "Select All" | Помечает свойство Checked всех элементов CheckBox как true |
| deselectButton\_Click | void | private | Обработчик события нажатия на кнопку "Deselect All" | Помечает свойство Checked всех элементов CheckBox как false |
| downloadButton\_Click | void | private | Обработчик события нажатия на кнопку Download Selected" | Выполняет загрузку выбранных файлов в указанный каталог |
| deleteAllButton\_Click | void | private | Обработчик события нажатия на кнопку "Delete All" | Отправляет POST запрос на сервер об удалении всех файлов и закрывает форму |

**SettingsForm.cs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| ImportButton\_Click | void | private | Обработчик события нажатия на кнопку "Import" | Импортирует настройки из выбранного файла |
| ExportButton\_Click | void | private | Обработчик события нажатия на кнопку "Export" | Экспортирует настройки в выбранный файл |
| applyButton\_Click | void | private | Обработчик события нажатия на кнопку "Apply" | Инициализирует объект класса SettingsClass, возвращает OK в качестве DialogResult и закрывает форму |

**Fieldset.cs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Свойства | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| Date | string | public | Дата | Служит для хранения даты записи файла |
| Time | string | public | Время | Служит для хранения времени записи файла |
| Type | string | public | Тип | Объявлен для того, что пользователь понимал, видео файл он может скачать или фото |
| Image | string | public | Превью | Ссылка на превью |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| Fieldset | - | public | Конструктор | Инициализация свойств класса |
| GetType | string | private | Метод типа | Возвращает тип файла |
| GetDate | string | private | Метод даты | Служит для получения даты записи файла |
| GetTime | string | private | Метод времени | Объявлен для получения времени записи файла |
| GetImageSource | string | private | Метод получения ссылки на превью | Служит для получения ссылки на превью медиа файла |

**SettingsClass.cs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Свойства | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| Ip | string | public | Адрес Raspberry Pi | Служит для хранения IP |
| Delay | int | public | Задержка | Хранит задержку между поворотами сервомотора |
| FinalPosition | int | public | Конечная позиция | Объявлено для хранения конечного положения сервомотора |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Название | Тип | Модификаторы | Описание | Назначение |
| SettingsClass | - | public | Конструктор | Служит для объявления свойств и проверки введенных данных на корректность |
| ToString | string | public override | Строковое представление | Возвращает строковое представление объекта класса |

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированых |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |