

Чегаев Данил Александрович,
магистрант,
Амурский Государственный Университет,
г. Благовещенск

РАЗРАБОТКА МОДУЛЯ КАМЕРЫ УДАЛЕННОГО ДОСТУПА ДЛЯ ЧАТ-БОТА В МЕССЕНДЖЕРЕ TELEGRAM

Аннотация: Рассматривается разработка и внедрение модуля взаимодействия с камерой удаленного доступа в чат-боте Telegram, представлены скриншоты готового решения.

Ключевые слова: умный дом, чат-бот, Telegram, камера удаленного доступа, видеонаблюдение.

В условиях стремительного роста интереса к домашней автоматизации и безопасности, интеграция видеонаблюдения в интерфейс чат-бота становится первым шагом автоматизации контроля за происходящим в доме в режиме реального времени. Созданный ранее чат-бот в Telegram, благодаря открытому API и широкой поддержке мультимедийных функций, имеет инструменты для реализации подобного рода решений.

Разработанный модуль камеры удаленного доступа позволяет пользователю получать актуальные изображения и видеопотоки (в разработке) с IP-камер, интегрированных в систему умного дома, прямо в интерфейсе мессенджера Telegram. Пользователь может отправить соответствующую команду боту, после чего бот осуществляет подключение к RTSP-потoku камеры, делает снимок текущего кадра или получает короткое видео и отправляет его обратно пользователю.

Основные функции модуля включают:

- просмотр текущего состояния дома через камеры в формате фото;
- интеграцию с системой уведомлений, при которой бот самостоятельно отправляет изображения при срабатывании датчиков движения;
- сохранение снимков и истории обращений для последующего анализа.

На рисунке 1 представлен программный код модуля камеры, он реализован с использованием библиотеки OpenCV для обработки изображений и взаимодействия с видеопотоком, а также библиотеки python-telegram-bot для построения логики взаимодействия с пользователем. Это обеспечило гибкость в настройке функций бота.



```
12
13 # Конфигурация камеры
14 RTSP_PORT = 554
15 RTSP_URL_HIGH = f"rtsp://{TAPO_USERNAME}:{TAPO_PASSWORD}@{TAPO_IP}:{RTSP_PORT}/stream1"
16 RTSP_URL_LOW = f"rtsp://{TAPO_USERNAME}:{TAPO_PASSWORD}@{TAPO_IP}:{RTSP_PORT}/stream2"
17
18 # Глобальная переменная для хранения данных устройства
19 device_data = {}
20
21 # Функция для получения снимка с камеры
22 > def get_snapshot(rtsp_url):...
23
24 # Обработчик команды /get_snapshot
25 async def get_snapshot_command(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
26     await update.message.reply_text("Запрашиваю снимок...")
27     try:
28         image_path = get_snapshot(RTSP_URL_HIGH)
29         with open(image_path, 'rb') as img:
30             await update.message.reply_photo(photo=img, caption="Ваш снимок с камеры")
31     except Exception as e:
32         await update.message.reply_text(f"Ошибка при получении изображения: {e}")
33
34 # Обработчик кнопки для получения снимка
35 async def button_callback(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
36     query = update.callback_query
37     await query.answer() # Подтверждение нажатия кнопки
38     if query.data == "get_snapshot":
39         await query.edit_message_text(text="Запрашиваю снимок...")
40         try:
41             image_path = get_snapshot(RTSP_URL_HIGH)
42             with open(image_path, 'rb') as img:
43                 await query.message.reply_photo(photo=img, caption="Ваш снимок с камеры")
44         except Exception as e:
45             await query.message.reply_text(f"Ошибка при получении изображения: {e}")
46     elif query.data == "add_device":
47         device_data.clear() # Очищаем данные для нового устройства
48         await query.message.reply_text("Введите тип устройства (например, 'камера'):")
49         context.user_data["awaiting_input"] = "device_type"
50
51
52
53
54
55
56
57
58
59
60
```

Рисунок 1 – Программный код модуля камеры

Одним из ключевых преимуществ реализации модуля камеры является отсутствие необходимости устанавливать отдельное приложение для просмотра видеонаблюдения. Всё взаимодействие осуществляется в привычной среде Telegram, что делает систему доступной даже для пользователей без технической подготовки.



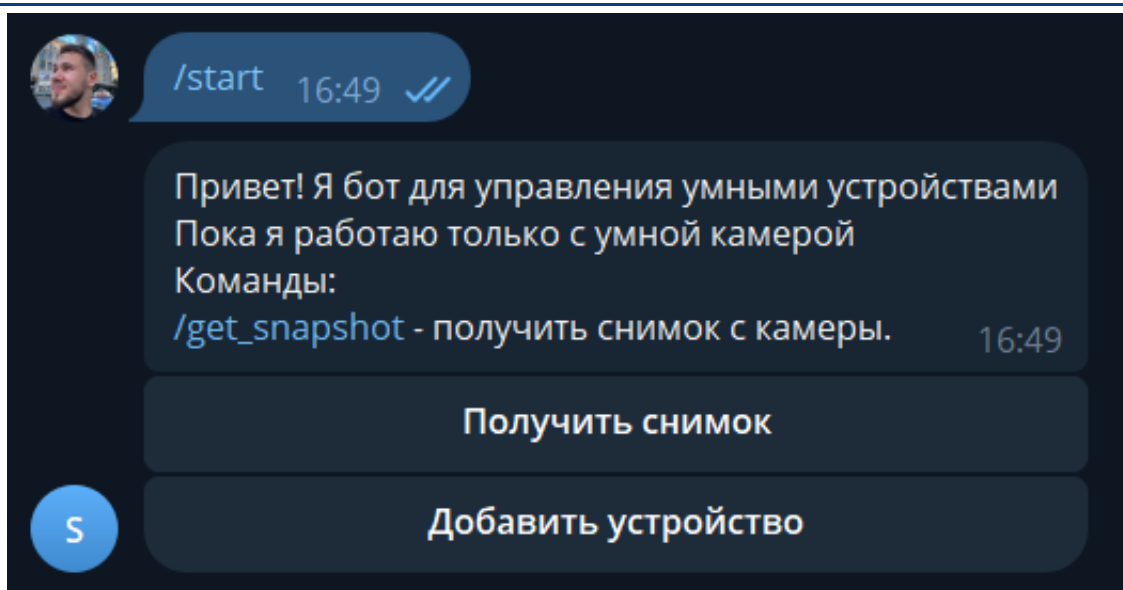


Рисунок 2 – Главное меню

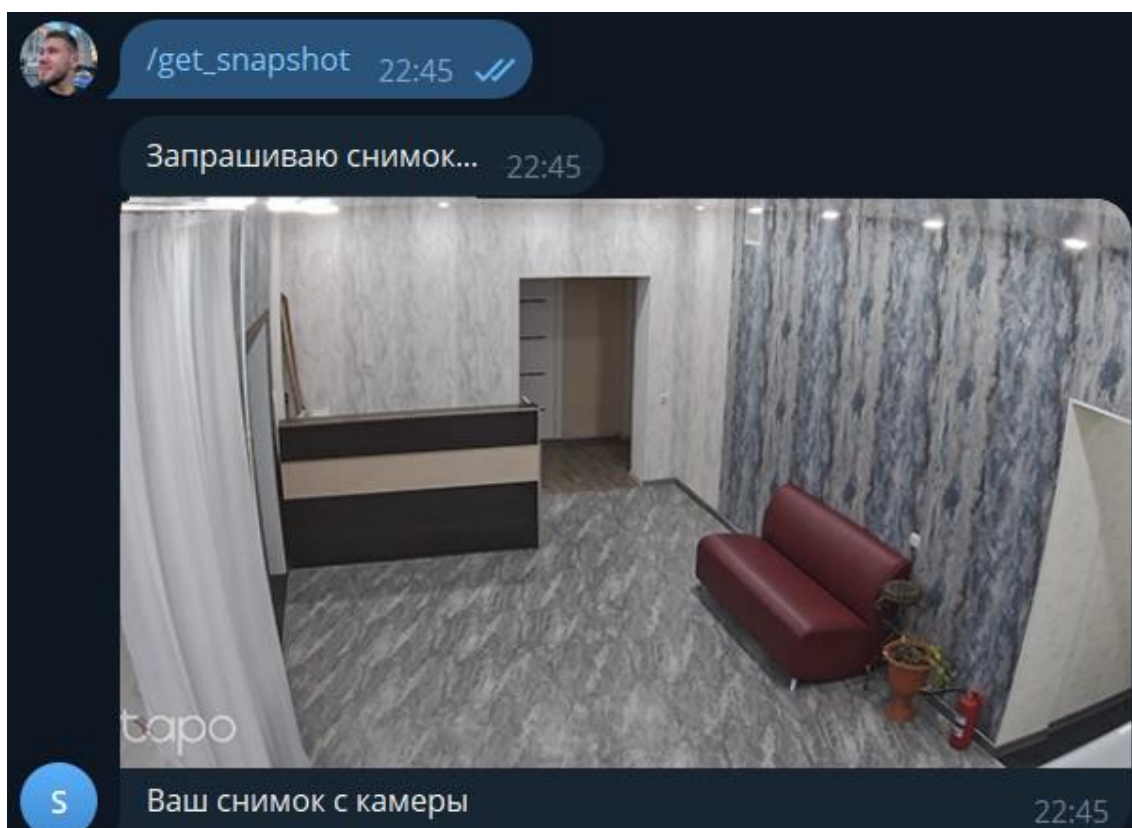


Рисунок 3 – Выполнение запроса

Запуск меню и выполнение запроса на рисунках 2 и 3 происходит по нажатию кнопки или отправки встроенной команды в виде сообщения, после получения команды бот получает снимок и выводит в чате. При невозможности получения снимка или неверной команде, предусмотрены сообщения об ошибке.

С точки зрения безопасности доступа к камере, реализована авторизация только определенным пользователям по Telegram ID, также возможно ограничить доступ к командам



у пользователей разного уровня доступа (администратор, гость и т.д.). Предусмотрено логирование действий пользователя и регистрация всех обращений к камерам.

Разработка и внедрение данного модуля продемонстрировали его высокую практическую значимость – пользователи могут оперативно получать визуальную информацию о происходящем в доме в их отсутствие, что особенно важно в случае тревожных событий или для удаленного контроля за пожилыми членами семьи и детьми.

Таким образом, модуль камеры удаленного доступа, встроенный в Telegram-бот, расширяет возможности системы умного дома, обеспечивая безопасность, удобство и мобильность, оставаясь при этом доступным инструментом даже для пользователей без специальных знаний.

Список литературы:

1. Крюкова, А. А. Особенности развития концепции «умный дом»: российский и зарубежный опыт / А. А. Крюкова, К. О. Шматок. – URL: <https://cyberleninka.ru/article/n/osobennosti-razvitiyakontseptsii-umnyy-dom-rossiyskiy-izarubezhnyy-opyt/viewer> (дата обращения: 03.04. 2025)
2. Обзор систем «Умный дом» // Научная электронная библиотека. – URL: <https://elibrary.ru> (дата обращения: 03.04. 2025).
3. Петров В.Ю. Технология "Интернет вещей", как перспективная современная информационная технология / Петров В.Ю., Рудашевская Е.А. // Фундаментальные исследования [Текст]: научный журнал/ "Академия естествознания", издательский дом. – Москва: Изд. дом Акад. естествознания, 2017. – № 9-2. – С. 371-396.

