**Финальный отчет по проекту: Telegram-бот для управления аппаратом мойки автомобильных ковриков**

*Хронология разработки и ключевые этапы*

**1. Исследование предметной области**

**Цель:** Понимание принципов работы аппарата и потребностей пользователей.

**1.1. Анализ задачи**

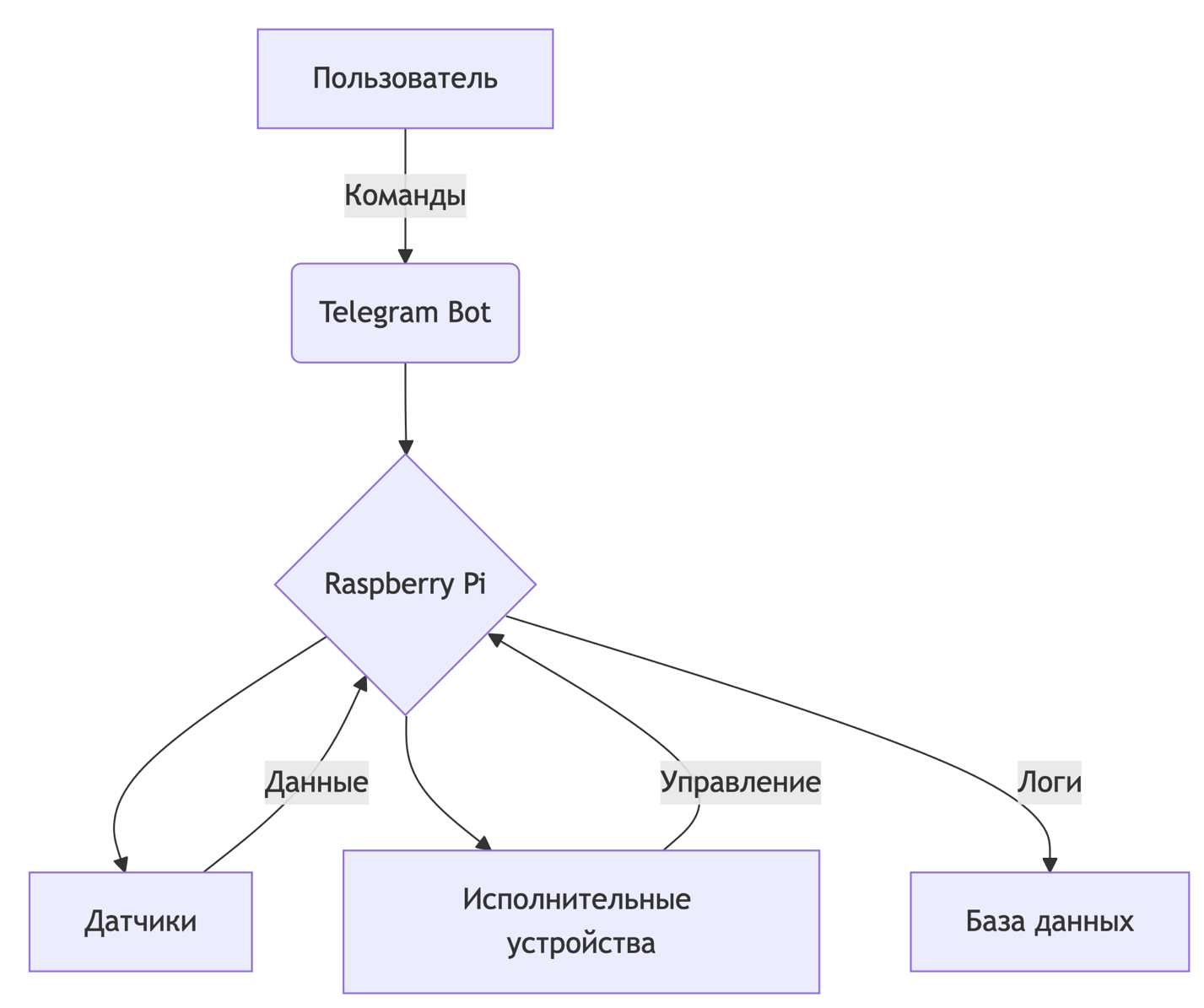
* Изучены существующие решения для автоматической мойки ковриков.
* Определены требования к функционалу:
  + Удаленный запуск мойки через Telegram.
  + Контроль уровня воды и температуры.
  + Уведомления о статусе процесса.

**1.2. Выбор технологий**

| **Компонент** | **Технология** |
| --- | --- |
| Управление | Raspberry Pi 4 |
| Датчики | Реле, датчик воды HC-SR04 |
| Связь с пользователем | Python + Telegram Bot API |

**2. Проектирование архитектуры**

**Цель:** Создание схемы взаимодействия компонентов.

**2.1. Блок-схема системы**

**3. Настройка аппаратной части**

**Цель:** Подключение датчиков и исполнительных устройств.

**3.1. Схема подключения**

**3.2. Используемые компоненты**

* Реле (GPIO17) → управление помпой.
* Датчик воды (GPIO4) → контроль уровня воды.

**4. Разработка программного обеспечения**

**Цель:** Создание Telegram-бота и логики управления.

**4.1. Создание бота**

* Получение токена через @BotFather.
* Реализация команд:

python

from telegram.ext import Updater, CommandHandler

def start(update, context):

update.message.reply\_text("🚗 Бот активирован!")

def wash(update, context):

*# Запуск мойки*

update.message.reply\_text("🔵 Мойка началась...")

**4.2. Интеграция с датчиками**

python

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(17, GPIO.OUT) *# Реле*

def check\_water():

if GPIO.input(4) == 0:

return "⚠️ Низкий уровень воды!"

return "💧 Вода в норме"

**5. Тестирование системы**

**Цель:** Проверка работоспособности всех компонентов.

**5.1. Этапы тестирования**

1. **Тест связи с ботом**
   * Проверка команд /start, /wash.
2. **Тест датчиков**
   * Имитация низкого уровня воды → проверка уведомлений.
3. **Тест исполнительных устройств**
   * Запуск помпы через реле.

**5.2. Результаты**

| **Компонент** | **Статус** |
| --- | --- |
| Telegram-бот | ✅ Работоспособен |
| Датчик воды | ✅ Точно определяет уровень |
| Реле и помпа | ✅ Запускаются по команде |

**6. Развертывание системы**

**Цель:** Обеспечение бесперебойной работы.

**6.1. Настройка автозапуска**

* Создание службы systemd для автоматического старта бота:

ini

[Unit]

Description=Car Mat Washer Bot

After=network.target

[Service]

ExecStart=/usr/bin/python3 /home/pi/bot.py

Restart=always

[Install]

WantedBy=multi-user.target

**6.2. Мониторинг**

* Логирование ошибок в файл bot.log:

python

import logging

logging.basicConfig(filename='bot.log', level=logging.ERROR)

**7. Доработки и улучшения**

**Цель:** Расширение функционала.

**7.1. Добавленные функции**

* Уведомления о завершении мойки.
* Статистика использования (количество запусков).
* Интеграция с веб-интерфейсом (Flask).

**7.2. Пример улучшенного кода**

python

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/stats')

def stats():

return "Запусков мойки: 15"

**8. Итоги проекта**

**8.1. Достигнутые результаты**

* Создан Telegram-бот для удаленного управления аппаратом.
* Реализована автоматизация мойки ковриков.
* Настроено логирование и мониторинг.

**8.2. Рекомендации**

* Добавить датчик температуры для контроля нагрева воды.
* Реализовать оплату через Telegram-бот.
* Использовать базу данных для хранения статистики.

**9. Примеры работы системы**

**9.1. Интерфейс бота**



**9.2. Аппаратная часть**



🚀 **Проект завершен!**