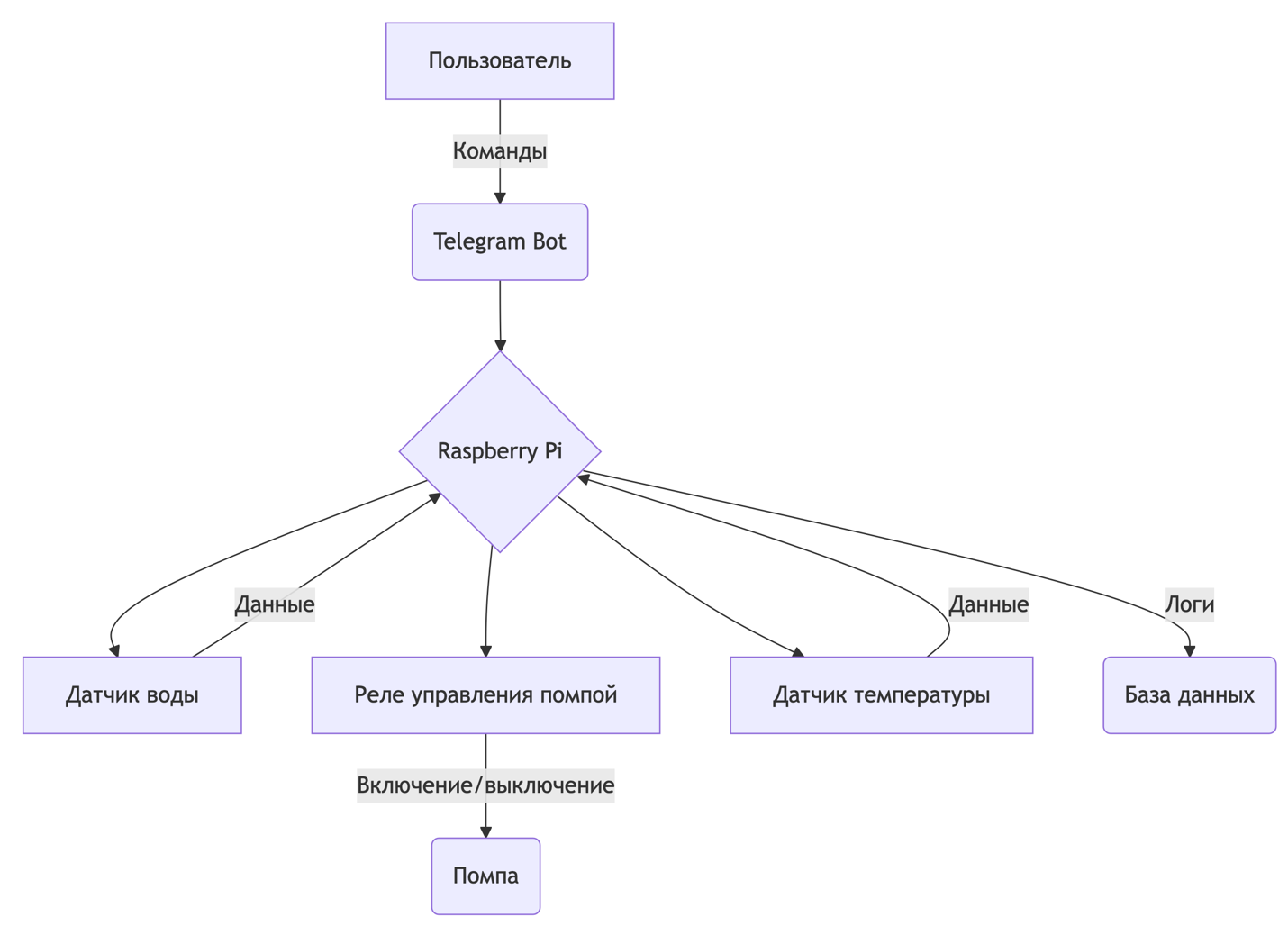
**Техническое руководство: Telegram-бот для управления аппаратом мойки автомобильных ковриков**

**1. Архитектура системы**

**1.1. Блок-схема системы**

****

**1.2. Компоненты системы**

| **Компонент** | **Назначение** | **Пример** |
| --- | --- | --- |
| **Telegram Bot** | Интерфейс пользователя | Bot Icon |
| **Raspberry Pi** | Центр управления |  |
| **Датчик воды** | Контроль уровня воды |  |
| **Реле** | Управление помпой |  |

**2. Пошаговая реализация**

**2.1. Настройка аппаратной части**

**Схема подключения**

Diagram

Code

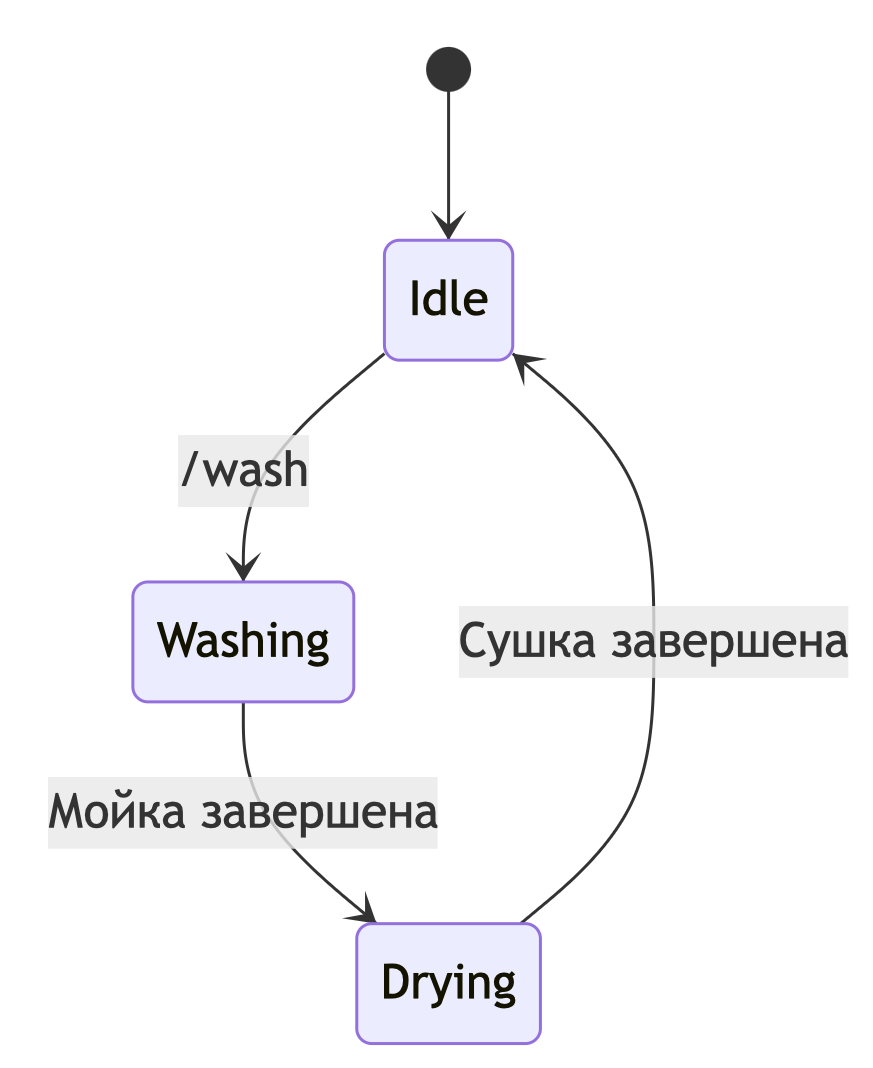
Mermaid rendering failed.

**Необходимые детали:**

* Реле → GPIO17
* Датчик воды → GPIO4 + 5V
* Помпа → 12V (через реле)

**2.2. Создание Telegram-бота**

**Диаграмма состояний бота**

****

**Код бота (Python)**

python

from telegram.ext import Updater, CommandHandler

import RPi.GPIO as GPIO

import time

TOKEN = "ВАШ\_ТОКЕН"

PUMP\_PIN = 17

WATER\_SENSOR = 4

def start(update, context):

update.message.reply\_text("🚗 Бот для мойки ковриков готов к работе!")

def wash(update, context):

GPIO.output(PUMP\_PIN, GPIO.HIGH)

update.message.reply\_text("🔵 Начата мойка...")

time.sleep(300) *# 5 минут мойки*

GPIO.output(PUMP\_PIN, GPIO.LOW)

update.message.reply\_text("✅ Мойка завершена. Запущена сушка.")

GPIO.setmode(GPIO.BCM)

GPIO.setup(PUMP\_PIN, GPIO.OUT)

GPIO.setup(WATER\_SENSOR, GPIO.IN)

updater = Updater(TOKEN)

updater.dispatcher.add\_handler(CommandHandler('start', start))

updater.dispatcher.add\_handler(CommandHandler('wash', wash))

updater.start\_polling()

**3. Интеграция с датчиками**

**3.1. Логика работы датчика воды**

python

def check\_water(update, context):

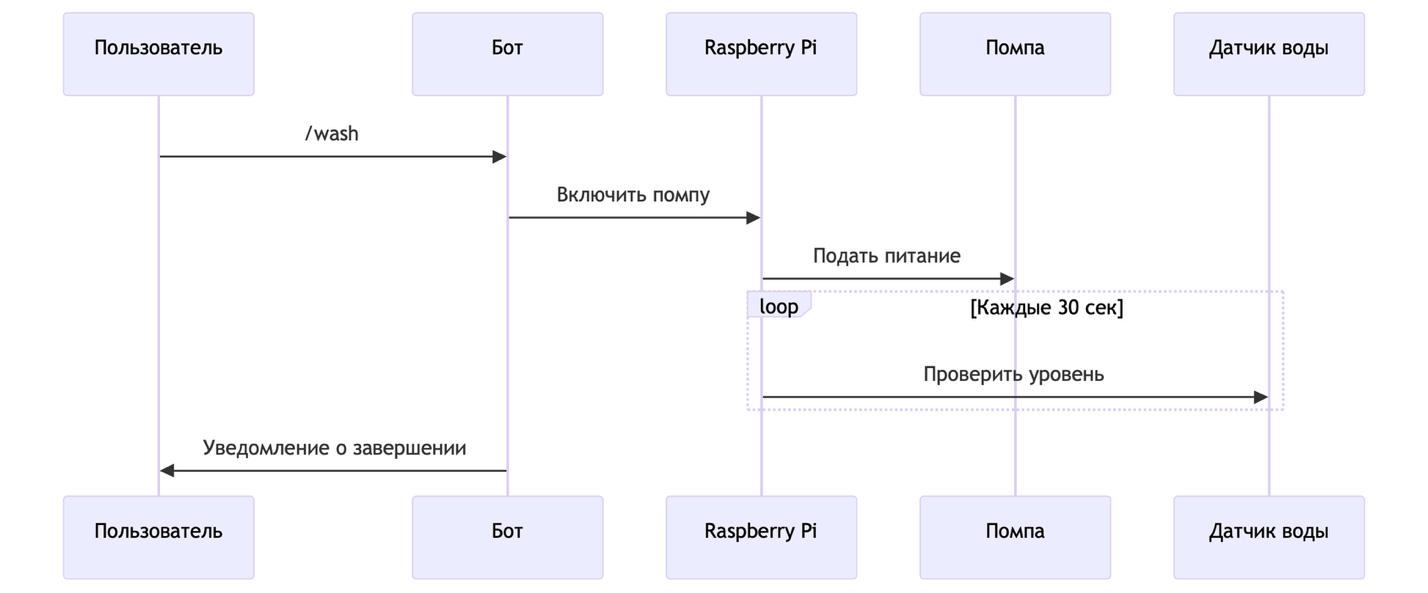
if GPIO.input(WATER\_SENSOR) == 0:

update.message.reply\_text("⚠️ Низкий уровень воды!")

else:

update.message.reply\_text("💧 Вода в норме")

**3.2. Диаграмма последовательностей**



**4. Развертывание системы**

**4.1. Таблица настройки GPIO**

| **Пин** | **Компонент** | **Режим** |
| --- | --- | --- |
| GPIO17 | Реле | OUTPUT |
| GPIO4 | Датчик воды | INPUT |
| 5V | Питание датчика | - |
| GND | Земля | - |

**4.2. Автозапуск бота (systemd)**

Создайте файл /etc/systemd/system/mat\_bot.service:

ini

[Unit]

Description=Telegram Bot for Car Mat Washer

After=network.target

[Service]

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

WorkingDirectory=/home/pi

Restart=always

[Install]

WantedBy=multi-user.target

Запустите:

bash

sudo systemctl enable mat\_bot

sudo systemctl start mat\_bot

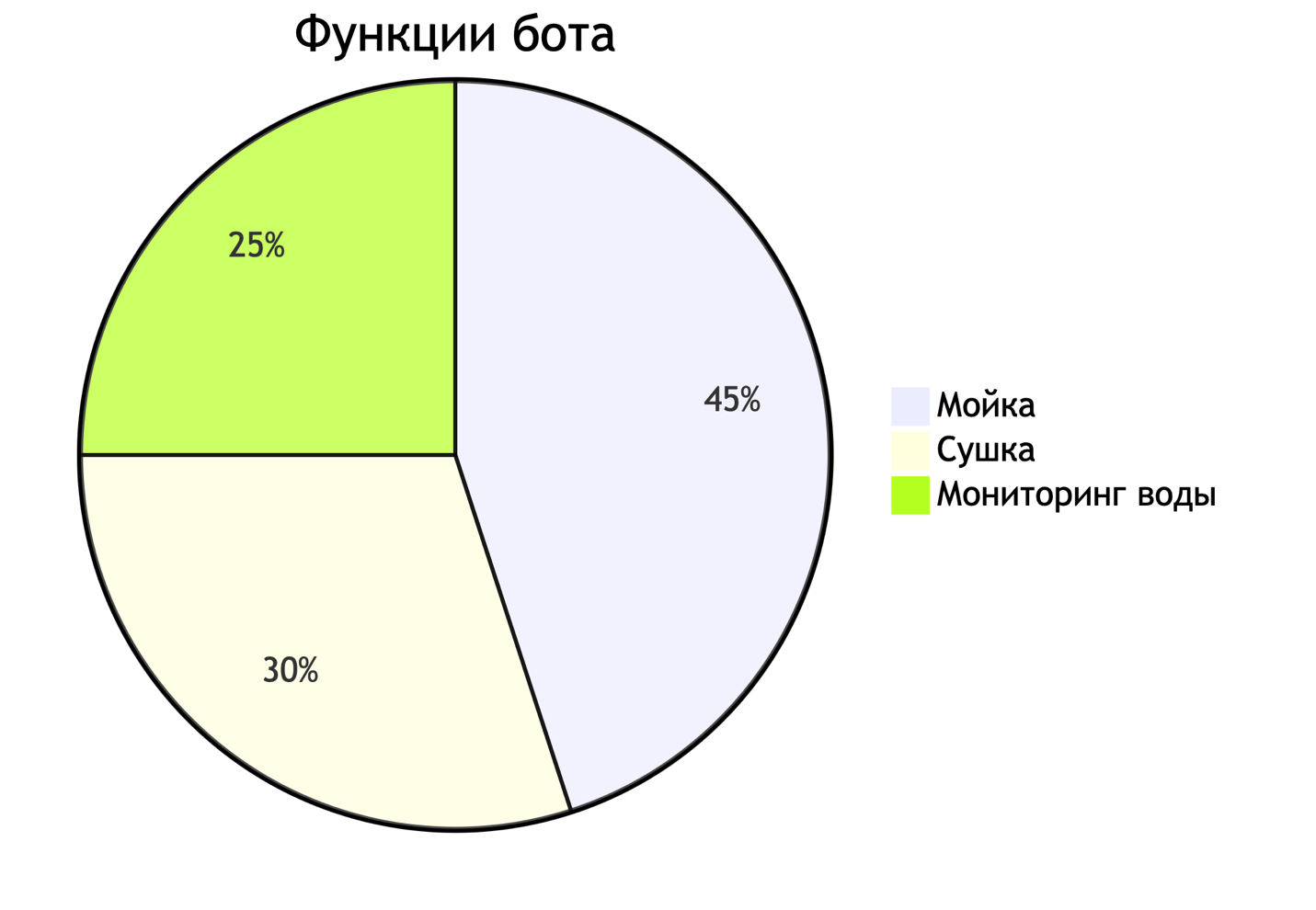
**5. Возможные проблемы и решения**

| **Проблема** | **Решение** |
| --- | --- |
| Бот не отвечает | Проверьте токен и интернет |
| Помпа не включается | Проверьте реле и питание 12V |
| Датчик воды показывает ошибку | Калибровка датчика в воде |

**6. Дополнительные улучшения**

**6.1. Диаграмма использования**

**6.2. Веб-интерфейс (дополнительно)**

****

Используйте **Flask** для создания панели управления:

python

from flask import Flask

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

@app.route('/')

def dashboard():

return "<h1>Статус мойки: Активен</h1>"

**7. Итоговая схема проекта**



🚀 **Готово!** Ваш автоматизированный комплекс:

* Принимает команды через Telegram
* Контролирует аппарат через Raspberry Pi
* Логирует все процессы