WATCHDOG v4.3 Pro

Network Monitoring System with Automatic Power Reset

Professional documentation for Raspberry Pi-based network monitoring
with GPIO-controlled power outlets (SSR relays)

Version	4.3.0
Platform	Raspberry Pi 1-5
License	MIT
Author	
GitHub	github.com/MaraXa-CZ/watchdog

Table of Contents

1. Features Overview

Core Functionality

- ■ **Network Monitoring** - Continuous ping/HTTP/TCP monitoring
- ■ **Automatic Power Reset** - GPIO-controlled SSR relay switching
- ■ **Statistics & Charts** - Ping history, uptime percentage, latency
- ■ **Scheduled Restarts** - Per-group maintenance schedules
- ■■ **Live Status** - Real-time server status indicators

Multi-User System

- ■ **Role-Based Access Control** - Admin, Operator, Viewer roles
- ■ **SSL/HTTPS Support** - Auto-generated self-signed certificates
- ■ **Audit Logging** - Track all user actions
- ■ **Auto-Backup** - Configuration backup on every change

v4.3.0 New Features

- ■■ **GitHub Auto-Update** - One-click updates from web interface
- ■ **Multi-ping Mode** - Fail only when ALL servers are down
- ■ **HTTP Health Check** - Check HTTP endpoints (status 200)
- ■ **TCP Port Check** - Verify if TCP port is open
- ■■ **Latency Alerts** - Warning/critical thresholds with logging
- ↕■ **Drag & Drop Groups** - Reorder groups on dashboard
- ■■■ **Dark/Light Theme** - User selectable theme
- ■ **Export CSV** - Download statistics history

2. Requirements

Hardware

- Raspberry Pi (any model: 1, 2, 3, 4, 5, Zero, Zero 2 W)
- SSR (Solid State Relay) modules (e.g., Fotek SSR-25DA)
- MicroSD card (8GB+ recommended)
- Network connection (Ethernet or WiFi)

Software

- Raspberry Pi OS (Bullseye or Bookworm)
- Python 3.9+ (installed automatically)
- Git (for installation and updates)

3. Installation

Fresh Install via Git

```
cd /opt
sudo git clone https://github.com/MaraXa-CZ/watchdog.git
cd watchdog
sudo bash install.sh
```

Update Existing Installation

```
cd /opt/watchdog
sudo git pull
sudo systemctl restart watchdog watchdog-web
```

Or use the Updates page in web interface for one-click updates!

First Login

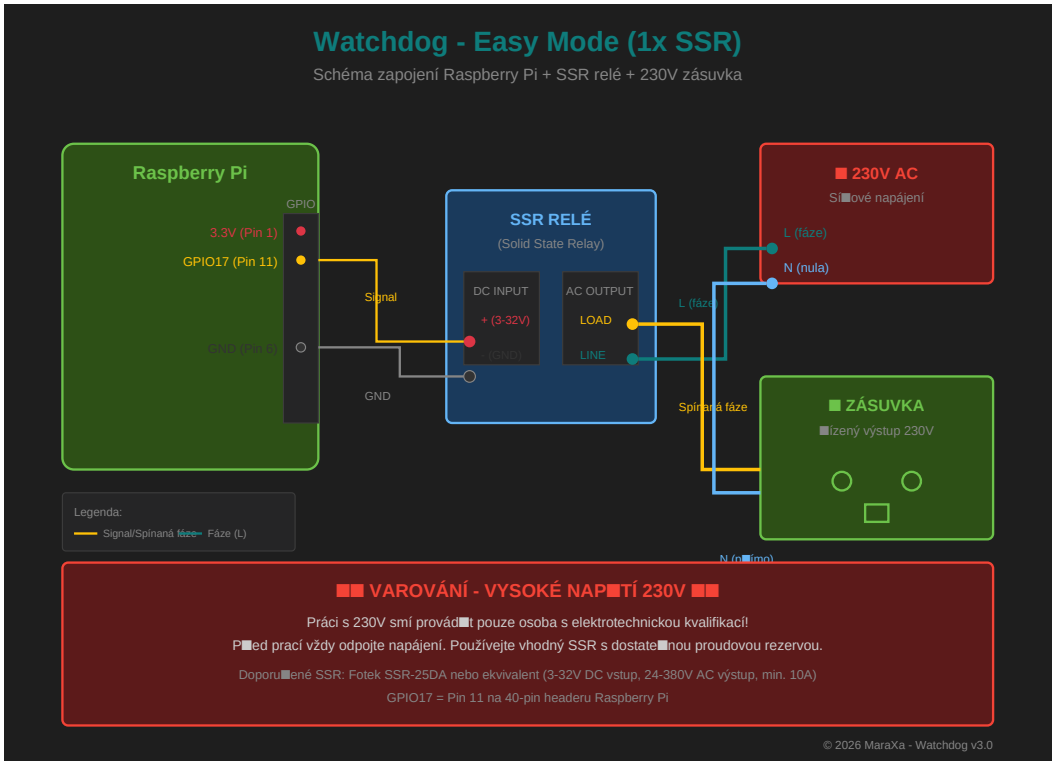
URL	http://<raspberry-ip>/
Username	admin
Password	admin

■■ Change the default password immediately after first login!

4. Wiring Diagrams

Basic Setup (1 SSR)

Connect Raspberry Pi GPIO to SSR relay for single device control.

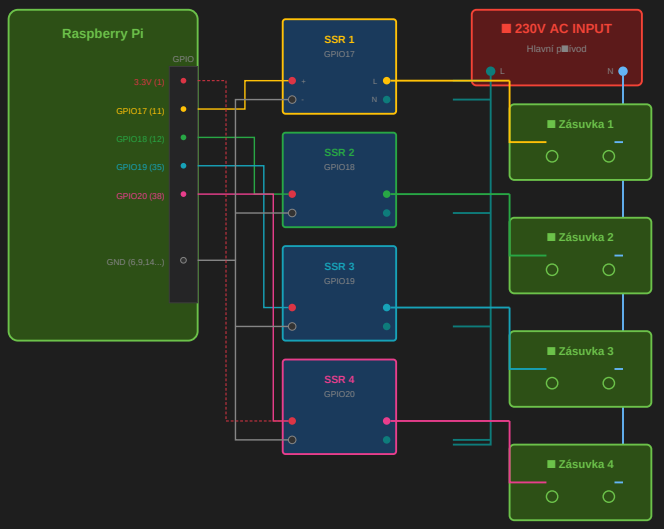


Full Setup (4 SSR)

Multiple SSR relays for controlling several devices independently.

Watchdog - Full Mode (4x SSR)

Schéma zapojení Raspberry Pi + 4x SSR relé + 4x 230V zásuvka



P■i■azení GPIO pin■

Outlet	GPIO	Pin #	Funkce
Zásuvka 1	GPIO17	11	External Network
Zásuvka 2	GPIO18	12	Internal Services
Zásuvka 3	GPIO19	35	Custom Group 1
Zásuvka 4	GPIO20	38	Custom Group 2
GND	-	6,9,14,20,25...Spole■ná zem	

Bill of Materials (BOM)

- 1x Raspberry Pi (any model with GPIO)
- 4x SSR relé (Fotek SSR-25DA nebo ekvivalent)
- 4x Zásuvka 230V (IP20 nebo vyšší)
- 1x Rozvodná krabice (min. 200x150x100mm)
- Vodi■e: 1.5mm² pro 230V, dupont pro GPIO
- Svorkovnice, DIN lišta (volite■ln■)
- microSD karta min. 8GB pro Raspberry Pi OS

■■ BEZPE■NOSTNÍ UPOZORN■NÍ

- Práci s 230V smí provád■t pouze kvalifikovaná osoba
- P■ed prací VŽDY odpojte hlavní p■ívod
- SSR musí mít rezervu min. 2x max. proud zát■že
- Zajist■te dostate■né chlazení SSR p■i vyšších zát■žích

5. Configuration

GPIO Pin Assignment

GPIO Pins	Physical Pin	Status
4, 5, 6, 12, 13, 16	Various	✓ Safe
17, 18, 19, 20, 21, 22	Various	✓ Safe
23, 24, 25, 26, 27	Various	✓ Safe
2, 3	3, 5	I2C (if unused)
7-11	Various	SPI (if unused)
14, 15	8, 10	UART (if unused)

Example Configuration

```
{
  "outlets": {
    "outlet_1": {"name": "Server Rack 1", "gpio_pin": 17},
    "outlet_2": {"name": "Network Switch", "gpio_pin": 18},
    "outlet_3": {"name": "Router", "gpio_pin": 27}
  },
  "groups": [
    {
      "name": "Web Servers",
      "servers": ["192.168.1.10", "192.168.1.11"],
      "outlet": "outlet_1",
      "fail_count": 3,
      "off_time": 10,
      "check_type": "ping",
      "enabled": true
    }
  ]
}
```

6. User Roles & Permissions

Permission	Admin	Operator	Viewer
View Dashboard	✓	✓	✓
View Logs	✓	✓	✓
View Statistics	✓	✓	✓
Control Relays	✓	✓	✗
Configure Groups	✓	✗	✗
Manage Users	✓	✗	✗
System Settings	✓	✗	✗
View Audit Log	✓	✗	✗
GitHub Updates	✓	✗	✗

7. Web Interface

Dashboard - Main overview with group status, live indicators, relay controls
Groups - Configure monitoring groups, servers, check types, thresholds
Outlets - Manage GPIO pin assignments for power outlets
Statistics - View ping history, uptime charts, export CSV
Scheduler - Set up scheduled restarts for maintenance
Logs - View system logs with filtering, delete old logs
SMTP - Configure email notifications, test connection
Users - Manage user accounts and roles
System - General settings, SSL, timezone, hostname
Maintenance - Restart services, system info, updates
Backups - View and restore configuration backups

8. Mobile App (PWA)

Watchdog includes a Progressive Web App that works on any smartphone:

1. Open http://<raspberry-ip>/mobile/ in your browser
2. Chrome (Android): Menu → 'Add to Home screen'
3. Safari (iOS): Share → 'Add to Home Screen'

Features:

- Real-time group status
- Relay control (for operators and admins)
- Offline indicator
- Auto-refresh

9. API Reference

Authentication

```
POST /api/auth
```

```
{"username": "admin", "password": "admin"}
```

Response: {success, token, user}

Get Groups

```
GET /api/groups
```

```
Header: Authorization: Bearer <token>
```

Control Relay

```
POST /api/control
```

```
{"group": "Web Servers", "action": "restart"}
```

Actions: on, off, restart

Live Status

```
GET /api/live-status
```

Returns real-time status of all servers

10. Service Management

Check Status

```
sudo systemctl status watchdog watchdog-web
```

Restart Services

```
sudo systemctl restart watchdog watchdog-web
```

Stop Services

```
sudo systemctl stop watchdog watchdog-web
```

View Daemon Logs

```
journalctl -u watchdog -f
```

View Web Logs

```
journalctl -u watchdog-web -f
```

File Structure

```
/opt/watchdog/app.py - Web application
/opt/watchdog/watchdog.py - Monitoring daemon
/opt/watchdog/config.json - Configuration
/opt/watchdog/users.json - User database
/opt/watchdog/log/ - System logs
/opt/watchdog/stats/ - Ping statistics
/opt/watchdog/backups/ - Config backups
```

11. Troubleshooting

Services won't start

```
journalctl -u watchdog -n 100
```

```
journalctl -u watchdog-web -n 100
```

GPIO not working

1. Verify wiring connections
2. Check GPIO permissions: sudo usermod -aG gpio $USER
3. For Pi 5, ensure lgpio is installed

Web interface not accessible

1. Check if service is running: systemctl status watchdog-web
2. Check firewall: sudo ufw allow 80
3. Verify IP address: hostname -I

12. Changelog

v4.3.0

- GitHub Auto-Update - Check and update directly from GitHub
- Live Server Status - Real-time indicators on dashboard
- Multi-ping Mode - Group fails only when ALL servers down
- HTTP/TCP Health Check - Multiple check types
- Latency Alerts - Warning/critical thresholds
- Drag & Drop Groups - Reorder on dashboard
- Dark/Light/Auto Theme - User selectable
- Export Statistics CSV - Download history
- Delete Logs - Clear from UI

v4.2.x

- Maintenance page with system controls
- Automatic SSL certificate generation
- System time display on dashboard
- Auto service restart on config changes

v4.0.0 - v4.1.x

- Multi-user support with roles
- Multi-language (EN/CZ)
- Scheduled restarts
- Ping statistics and charts
- Mobile PWA
- Up to 16 GPIO outlets