



Raspberry Pi Projects

A Practical Guide to Seven Real-World Pi Projects

Web Server · Pi-hole · Jellyfin · WireGuard VPN Home Assistant · GPIO · Resources

osztromok.com

Contents

Project 1 — Setting Up a Web Server

[Apache](#) · [PHP](#) · [MariaDB](#) · [SSL](#) · [Virtual Hosts](#)

Project 2 — Pi-hole DNS Ad Blocker

[Installation](#) · [Configuration](#) · [Upstream DNS](#) · [Regex](#)

Project 3 — Jellyfin Media Server

[Installation](#) · [Libraries](#) · [Transcoding](#) · [Remote Access](#)

Project 4 — WireGuard VPN

[Key Pairs](#) · [Server Config](#) · [Client Config](#) · [QR Codes](#)

Project 5 — Home Assistant

[Installation](#) · [Integrations](#) · [Automations](#) · [Dashboard](#)

Project 6 — GPIO & Physical Computing

[LED](#) · [Button](#) · [DHT22](#) · [Relay](#) · [gpiozero](#)

Project 7 — Further Resources

[Documentation](#) · [YouTube](#) · [Community](#) · [What's Next](#)

Project 1 — Setting Up a Web Server

Apache · PHP · MariaDB · SSL · Virtual Hosts

What you build	A full LAMP web server serving your own websites with HTTPS
Difficulty	Beginner-friendly — each step builds on the last
Time	2 – 3 hours for core setup; longer with SSL and virtual hosts
Pi requirement	Any Pi with network access — Pi 3B+ or Pi 4 recommended

Installation

Install the full LAMP stack in two commands:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y apache2 php php-mysql mariadb-server
sudo mysql_secure_installation
```

Key Locations

Web root	/var/www/html/
Apache config	/etc/apache2/sites-available/
PHP config	/etc/php/8.x/apache2/php.ini
MariaDB data	/var/lib/mysql/

SSL with Let's Encrypt

```
sudo apt install -y certbot python3-certbot-apache
sudo certbot --apache -d yourdomain.com
sudo certbot renew --dry-run # test auto-renewal
```

Virtual Hosts

Host multiple websites on one Pi. Create a config file in /etc/apache2/sites-available/ for each site, then enable it:

```
sudo a2ensite mysite.conf
sudo systemctl reload apache2
```

Essential Commands

```
sudo systemctl start|stop|restart|status apache2
sudo apache2ctl configtest # check config syntax
sudo tail -f /var/log/apache2/error.log
```

```
mysql -u root -p          # connect to MariaDB
```

Project 2 — Pi-hole DNS Ad Blocker

Network-wide ad and tracker blocking via DNS

What you build	A DNS sinkhole that blocks ads for every device on your network
Difficulty	Beginner — one install command; most setup is in the web UI
Time	30 – 60 minutes to a working setup
Pi requirement	Any Pi — even a Pi Zero 2W works well

Installation

```
curl -sSL https://install.pi-hole.net | bash
# Note the admin password shown at the end of installation
pihole -a -p newpassword    # change admin password
```

■ Give your Pi a static local IP before installing — Pi-hole must always be reachable at the same address. Set this via your router's DHCP reservation table.

Router Configuration

In your router's DNS settings (usually under LAN or DHCP), set the Primary DNS to your Pi's local IP address. All devices will then use Pi-hole automatically.

Key Commands

```
pihole status      # show running status
pihole -up         # update Pi-hole
pihole -g          # update gravity (blocklists)
pihole enable / disable # toggle blocking
pihole -w domain.com # whitelist a domain
pihole -b domain.com # blacklist a domain
```

■ Access the admin dashboard at <http://pi.hole/admin> or <http://admin>. The Query Log shows every DNS request on your network in real time.

Project 3 — Jellyfin Media Server

Stream your movies, TV shows, and music to any device

What you build	A self-hosted Netflix-style media server accessible on your LAN and remotely
Difficulty	Beginner — web UI handles most configuration
Time	1 – 2 hours to install and set up libraries
Pi requirement	Pi 4 (4 GB recommended) for transcoding; Pi 3B+ works for direct play

Installation

```
curl -s https://repo.jellyfin.org/install-debuntu.sh | sudo bash
sudo systemctl enable --now jellyfin
# Access setup wizard at http://raspberrypi.local:8096
```

Mount an External Drive

```
sudo mkdir /media/mediadrive
sudo blkid /dev/sda1          # find UUID
# Add to /etc/fstab:
UUID=xxxx /media/mediadrive exfat defaults,auto,nofail 0 0
sudo mount -a
```

Transcoding on Pi 4

Enable hardware transcoding in Dashboard → Playback → Transcoding. Set Hardware Acceleration to Video4Linux2 (V4L2). Add the jellyfin user to the video group:

```
sudo usermod -aG video jellyfin
sudo systemctl restart jellyfin
```

Project 4 — WireGuard VPN

Secure remote access to your home network from anywhere

What you build	A fast, modern VPN server — connect phones and laptops back to your Pi securely
Difficulty	Intermediate — config files require care; PiVPN makes it easier
Time	1 – 2 hours for server + first client
Pi requirement	Any Pi with a static IP or DDNS hostname

Install with PiVPN (recommended)

```
curl -L https://install.pivpn.io | bash
# Choose WireGuard during setup
# Note the port number chosen (default: 51820)
```

Add a Client

```
pivpn add      # create a client config
pivpn qrcode   # display QR code for mobile apps
pivpn list     # show all clients
pivpn remove name # revoke a client
```

Router Configuration

Forward UDP port 51820 from your router to the Pi's local IP. If your ISP gives you a dynamic IP, set up a DDNS hostname (DuckDNS, No-IP) and use that in client configs.

■ The WireGuard app is available free on iOS, Android, Windows, Mac, and Linux. Scan the QR code to import a client config in under 30 seconds.

Project 5 — Home Assistant

Automate your home with a powerful open-source platform

What you build	A home automation hub controlling lights, heating, sensors, and more
Difficulty	Intermediate — installation is easy; automation YAML takes practice
Time	1 – 2 hours to install; days to weeks to fully automate your home
Pi requirement	Pi 4 (2 GB+ recommended) — HAOS runs best on dedicated hardware

Install Home Assistant OS (HAOS)

Flash the HAOS image to an SD card or SSD using Raspberry Pi Imager. HAOS is a complete operating system — it replaces Raspberry Pi OS:

```
# In Raspberry Pi Imager:  
# Choose OS → Other specific-purpose OS  
# → Home assistants and home automation → Home Assistant  
# Access the UI at http://homeassistant.local:8123
```

Example Automation (YAML)

```
automation:  
  alias: 'Turn on lights at sunset'  
  trigger:  
    platform: sun  
    event: sunset  
    offset: '-00:30:00'  
  action:  
    service: light.turn_on  
    target:  
      entity_id: light.living_room  
    data:  
      brightness_pct: 80
```

■ Home Assistant has 3,000+ integrations. Popular ones: Philips Hue, Google/Alexa, IKEA Tradfri, Sonos, Nest, Ring, and many smart home protocols (Zigbee, Z-Wave, Matter).

Project 6 — GPIO & Physical Computing

Connect real-world electronics to your Raspberry Pi

What you build	Four mini-projects: LED control, button input, DHT22 sensor, relay switch
Difficulty	Beginner — safe low-voltage circuits using the gpiozero library
Time	1 – 3 hours depending on how many projects you build
Hardware	LEDs, resistors, push buttons, DHT22 sensor, relay module — all under £10 total

The GPIO Pin Layout

The Pi has 40 GPIO pins. Always use 3.3 V logic (not 5 V directly to GPIO pins). Add a 330 Ω resistor in series with any LED. Ground any unused input pins to avoid floating values.

Mini Project 1 — Blinking LED

```
from gpiozero import LED
from time import sleep

led = LED(17) # GPIO 17 (pin 11)
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Mini Project 2 — Button Input

```
from gpiozero import Button, LED

btn = Button(2) # GPIO 2
led = LED(17)
btn.when_pressed = led.on
btn.when_released = led.off
pause() # from signal import pause
```

Mini Project 3 — DHT22 Temperature & Humidity

```
import adafruit_dht, board

sensor = adafruit_dht.DHT22(board.D4)
print(f'Temp: {sensor.temperature:.1f}°C')
print(f'Humidity: {sensor.humidity:.1f}%')
```

■ Never connect 5V directly to a GPIO pin. The Pi's GPIO pins operate at 3.3V — exceeding this permanently damages the processor.

Project 7 — Further Resources & What's Next

Documentation, communities, and ideas for going further

Official Documentation

Raspberry Pi docs	raspberrypi.com/documentation
RetroPie	retropie.org.uk/docs
Home Assistant	home-assistant.io/docs
Pi-hole	docs.pi-hole.net
WireGuard	wireguard.com
Jellyfin	jellyfin.org/docs

Recommended YouTube Channels

- Jeff Geerling (Pi hardware deep dives)
- NetworkChuck (beginner-friendly networking projects)
- Crosstalk Solutions (Home Assistant and networking)
- Wolfgang's Channel (self-hosting and homelab)
- LearnLinuxTV (Linux fundamentals)

Community

The Raspberry Pi community is one of the friendliest in tech. Great places to ask questions and share projects:

- [r/raspberrypi](https://reddit.com/r/raspberrypi) — the main Pi subreddit
- forums.raspberrypi.com — official forums
- [r/homelab](https://reddit.com/r/homelab) — self-hosting and home infrastructure
- [r/selfhosted](https://reddit.com/r/selfhosted) — running your own services

[*] Congratulations on completing the Raspberry Pi Projects course! You've built a web server, a DNS ad blocker, a media server, a VPN, a home automation hub, and a GPIO electronics workbench. Each of these is a real, production-quality system — and you built it yourself.