

Raspberry Pi as Wake-On-Lan Proxy

Version 1.0.4

By Martin Rothschink, AxoNet Software GmbH

<https://www.green-it-software.com/download/wolproxy-1.0.4.zip>

Intention

Most SoHo routers do not allow to forward magic packets for wake-on-lan to your internal broadcast address. As a result you cannot use wake-on-lan over the internet (wake-on-wan) to wake your equipment at home. Those routers allow only a forward to a fixed internal ip address which does not work with wake-on-lan. More details can you read here: [wake-on-wan](#).

A Raspberry Pi is an energy efficient, low cost computer which we will use to create a wake-on-lan proxy. Your router will send the magic packet to the Pi and the Pi will then forward the magic packet to the local subnet broadcast address.

Configuring the Pi

We assume that you have an already running Pi with the latest Raspbian Wheezy distro:

(<http://www.raspberrypi.org/downloads>)

During the configuration you need a monitor and a keyboard connected to your Pi. Later on you need only a network connection and a power supply.

Get super user (admin) rights

Start your Pi and log in. You do not need to start X (the graphical UI), we can do these steps in text mode.

Enter `sudo -s`, the prompt should change:

```
pi@raspberrypi ~ $ sudo -s
root@raspberrypi:/home/pi#
```

Create a new project directory

Enter `pwd` (print working directory) to verify that you are in `/home/pi`. Create a new directory `wolproxy` and finally change into that directory:

```
root@raspberrypi:/home/pi# pwd
/home/pi
root@raspberrypi:/home/pi# mkdir wolproxy
root@raspberrypi:/home/pi# cd wolproxy
root@raspberrypi:/home/pi/wolproxy# pwd
/home/pi/wolproxy
root@raspberrypi:/home/pi/wolproxy#
```

Download the project file

Now it's time to download the project package. We use `wget` to download the package from the command line:

```
root@raspberrypi:/home/pi/wolproxy# wget https://www.green-it-
software.com/download/wolproxy-1.0.4.zip
--2015-01-29 13:15:19-- https://www.green-it-software.com/download/wolproxy-1.0.4.zip
Resolving www.green-it-software.com (www.green-it-software.com)... 45.91.101.49
Connecting to www.green-it-software.com (www.green-it-software.com)|45.91.101.49|:443
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 224960 (220K) [application/zip]
Saving to: `wolproxy-1.0.4.zip'

100%[=====>] 5,573      --.-K/s   in 0.02s

2021-09-01 08:44:16 (4,22 MB/s) - >wolproxy-1.0.4.zip.1« saved [224960/224960]
```

Unzip the archive

Enter unzip wolproxy-1.0.4.zip

```
root@raspberrypi4:/home/pi/wolproxy# unzip wolproxy-1.0.4.zip
Archive:  wolproxy-1.0.4.zip
  inflating: makefile
  inflating: wolproxy.service
  inflating: wolproxy.c
  inflating: Raspberry Pi as WolProxy 1.0.4.pdf
```

Compile and do a test run

Enter make to compile the executable:

```
root@raspberrypi:/home/pi/wolproxy# make
gcc -g -c -o wolproxy.o wolproxy.c
gcc -g -o wolproxy wolproxy.o
root@raspberrypi:/home/pi/wolproxy# ls -l
total 520
-rw-r--r-- 1 root root 536 Sep 1 13:09 makefile
-rw-r--r-- 1 root root 237817 Sep 1 09:52 'Raspberry Pi as WolProxy 1.0.4.pdf'
-rwxr-xr-x 1 root root 21016 Sep 1 13:16 wolproxy
-rw-r--r-- 1 root root 225296 Sep 1 13:14 wolproxy-1.0.4.zip
-rw-r--r-- 1 root root 9288 Sep 1 13:07 wolproxy.c
-rw-r--r-- 1 root root 15196 Sep 1 13:16 wolproxy.o
-rw-r--r-- 1 root root 168 Sep 1 12:37 wolproxy.service
```

Let's run the executable and then stop with Ctrl+C:

```
root@raspberrypi:/home/pi/wolproxy# ./wolproxy

WolProxy 1.0.4, Copyright (c) AxoNet Software GmbH, Martin Rothschink 2012-2021
Waiting for interfaces to get up and running
IP 192.168.1.82 netmask 255.255.255.0 broadcast 192.168.1.255

open socket
Ready, waiting for wol packets...
^CSignal 2 received, closing wolproxy
```

You should now write down the IP address of your Pi, here it is 192.168.1.82. We need this IP later for the router configuration.

Install and configure executable and scripts

Run make install to copy the executable and the necessary scripts:

```
root@raspberrypi:/home/pi/wolproxy# make install
cp wolproxy /usr/bin
cp wolproxy.service /etc/systemd/system
rm -f /etc/init.d/wolproxy
systemctl enable wolproxy
```

Start daemon

To start the background service (called a daemon) run

```
root@raspberrypi:/home/pi/wolproxy# systemctl start wolproxy
```

To stop the background service use

```
root@raspberrypi:/home/pi/wolproxy# systemctl stop wolproxy
```

Reboot, log in again and verify that the daemon is again running

Initiate a reboot from the command line with reboot:

```
root@raspberrypi:/home/pi/wolproxy# reboot
```

```
Broadcast message from root@raspberrypi (pts/1) (Tue Jul 3 14:55:05 2012):
The system is going down for reboot NOW!
```

Wait until the Pi has rebooted, then log in again and enter `ps ax | grep wol`. You should see two lines as result.

```
pi@raspberrypi ~ $ ps ax | grep wol
1409 ?        Ss      0:00 /home/pi/wolproxy/wolproxy -d
1527 pts/0    S+      0:00 grep --color=auto wol
```

Finally let's check the system log with `grep wol /var/log/syslog`. You should see some lines like these here:

```
pi@raspberrypi ~ $ grep wol /var/log/syslog
Sep  1 13:18:24 raspberrypi4 systemd[1]: wolproxy.service: Succeeded.
Sep  1 13:28:51 raspberrypi4 wolproxy: Running as daemon: WolProxy 1.0.4,
Copyright (c) AxoNet Software GmbH, Martin Rothschink 2012-2021
Sep  1 13:28:51 raspberrypi4 wolproxy: Waiting for interfaces to get up and
running
Sep  1 13:28:51 raspberrypi4 wolproxy: IP 192.168.1.90    netmask 255.255.255.0
broadcast 192.168.1.255
Sep  1 13:28:51 raspberrypi4 wolproxy: IP 192.168.1.87    netmask 255.255.255.0
broadcast 192.168.1.255
Sep  1 13:28:52 raspberrypi4 wolproxy: open socket
Sep  1 13:28:52 raspberrypi4 wolproxy: Ready, waiting for wol packets...
```

Configuring the router

Most routers allow to assign “fixed” DHCP addresses, i.e. the device will always get the same IP by DHCP. Do this now for the Raspberry Pi.

The last part is the port forwarding rule for

External UDP Port 9, any IP

to Internal UDP Port 9, IP of your Raspberry Pi

That’s it. Now let’s test with all components in place.

How to test your setup

You can use any machine in your network which is configured for wake-on-lan and reliably working. Open a command prompt and run ipconfig/all to get the MAC address. The output will look like that:

```
Windows IP Configuration

Host Name . . . . . : <your machine name>
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : foobar.local

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . : axonet.local
Description . . . . . : Intel(R) PRO/1000 MT Network Connection
Physical Address. . . . . : 00-0C-29-AB-CD-EF
...
```

Look for the Physical Address; here it is 00-0C-29-AB-CD-EF. Click on save energy to put that machine into standby or hibernation.

On a second machine open your web browser and navigate to <http://www.depicus.com/wake-on-lan/woli.aspx> . Enter the MAC , your DynDNS or xxx.homeserver.com domain, 255.255.255.255 as subnet mask and port 9, then click Send. Your sleeping machine should now wake up.

Troubleshooting

If your machine does not wake up, use these steps to locate the issue:

1. Does your DynDNS or xxx.homeserver.com domain resolve into your current external IP? Open a command line and enter `nslookup xxx.homeserver.com`, in your browser go to <http://www.whatismyip.com/> . Both IPs should be identical.
2. Verify that your router is forwarding magic packets. On your Raspberry Pi enter `grep wol /var/log/syslog`
You should see some lines like that

```
Jul  3 16:19:12 raspberrypi wolproxy[1409]: Received a magic packet for MAC  
00-0c-29-ab-cd-ef from 141.44.59.15:52404  
Jul  3 16:19:12 raspberrypi wolproxy[1409]:     forwarding to...  
Jul  3 16:19:12 raspberrypi wolproxy[1409]:         192.168.1.255
```
3. Use a command line `wol.exe` to wake your machine from within your network to verify that wol is working with that machine: <http://gammadyne.com/cmdline.htm#wol>