

# Home Appliance Hack-and-IoT Guidebook

**Affordable solutions with  
the ESP8266 and 3D printing**  
*Command listing*

## Content:

<b>2. Components.....</b>	<b>3</b>
<b>4. emoncms installation .....</b>	<b>3</b>
<b>4.1. Update and upgrade the Raspberry Pi .....</b>	<b>3</b>
4.2.1. MariaDB & apache. ....	3
4.2.2. phpMyAdmin.....	3
4.2.3. PECL.....	4
<b>4.3. emoncms code .....</b>	<b>4</b>
4.3.1. Configure emoncms .....	4
4.4.1. Dashboard .....	5
4.4.2. Graph.....	6
4.4.3. Apps.....	6
<b>5. MQTT (mosquitto) installation on a Raspberry Pi.....</b>	<b>6</b>
<b>5.1. Installation of the Mosquitto broker. ....</b>	<b>6</b>
<b>5.2. Installation of the Mosquitto client (RPI).....</b>	<b>7</b>
<b>5.3. Verification of the Mosquitto installation.....</b>	<b>7</b>
<b>9. ESP8266 Over The Air – update/programming (OTA).....</b>	<b>7</b>
9.1.3.1. Firmware build. ....	7
9.1.3.2. Firmware update .....	7
9.1.4. CLI OTA .....	7
<b>12. VPN gateway. ....</b>	<b>7</b>
12.2.1. Basic configuration.....	7
12.2.2. OpenVPN installation .....	8
12.2.3.1. OpenVPN Authorization file .....	8
12.2.3.2. Start at boot time. ....	9
12.2.3.3. Configuration of the VPN gateway (fun with IP tables).....	9
<b>13. Software/firmware design. ....</b>	<b>10</b>
13.2.1. Installation of the utility functions. ....	10
<b>25. Software listing. ....</b>	<b>10</b>
25.1.1. PHP .....	10
25.1.2. c-code .....	10

## 2. Components

```
pi@raspberrypi:~ $ passwd
```

## 4. emoncms installation

### 4.1. Update and upgrade the Raspberry Pi

```
pi@raspberrypi:~ $ sudo apt-get update && sudo apt-get upgrade
```

```
pi@raspberrypi:~ $ sudo reboot
```

```
pi@raspberrypi:~ $ sudo apt-get dist-upgrade && sudo rpi-update
```

#### 4.2.1. MariaDB & apache.

```
pi@raspberrypi:~ $ sudo apt-get install -y apache2 mariadb-server mysql-client php7.0 libapache2-mod-php7.0 php7.0-mysql php7.0-gd php7.0-opcache php7.0-curl php-pear php7.0-dev php7.0-mcrypt php7.0-common redis-server php-redis git build-essential php7.0-mbstring
```

```
pi@raspberrypi:~ $ sudo mysql
```

```
MariaDB [(none)]> SELECT user, host, plugin FROM mysql.user;
```

```
MariaDB [(none)]> UPDATE mysql.user SET plugin = '' WHERE plugin = 'unix_socket'; FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> SELECT user, host, plugin FROM mysql.user;
```

#### 4.2.2. phpMyAdmin

```
pi@raspberrypi:~ $ sudo apt-get install phpmyadmin
```

```
pi@raspberrypi:~ $ sudo mysql_secure_installation
```

### 4.2.3. PECL

```
pi@raspberrypi:~ $ sudo pear channel-discover  
pear.swiftmailer.org
```

```
pi@raspberrypi:~ $ sudo pecl channel-update  
pecl.php.net
```

```
pi@raspberrypi:~ $ sudo pecl install swift/swift redis
```

```
pi@raspberrypi:~ $ sudo nano /etc/php/7.0/apache2/php.ini
```

```
pi@raspberrypi:~ $ sudo a2enmod rewrite
```

```
pi@raspberrypi:~ $ printf "extension=redis.so" | sudo  
tee /etc/php/7.0/mods-available/redis.ini 1>&2
```

```
pi@raspberrypi:~ $ sudo phpenmod redis
```

```
pi@raspberrypi:~ $ sudo nano /etc/apache2/apache2.conf
```

```
pi@raspberrypi:~ $ sudo systemctl restart apache2
```

## 4.3. emoncms code

```
pi@raspberrypi:~ $ sudo chown $USER /var/www
```

```
pi@raspberrypi:~ $ cd /var/www && git clone -b stable  
https://github.com/emoncms/emoncms.git
```

```
pi@raspberrypi:~ $ ls -al
```

### 4.3.1. Configure emoncms

```
pi@raspberrypi:~ $ sudo mysql --user=root --  
password=<your password>
```

```
MariaDB [(none)]> DELETE FROM mysql.user WHERE
User='root' AND Host NOT IN ('localhost', '127.0.0.1',
'::1'); DELETE FROM mysql.user WHERE User=''; DROP
DATABASE IF EXISTS test; DELETE FROM mysql.db WHERE
Db='test' OR Db='test\_%'; FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> CREATE DATABASE emoncms DEFAULT
CHARACTER SET utf8;
```

```
MariaDB [(none)]> CREATE USER 'emoncms'@'localhost'
IDENTIFIED BY '<new_secure_password>'; GRANT ALL ON
emoncms.* TO 'emoncms'@'localhost'; flush privileges;
```

```
pi@raspberrypi:~ $ sudo mkdir
/var/lib/{phpfiwa,phpfina,phptimeseries}
```

```
pi@raspberrypi:~ $ sudo chown www-data:root
/var/lib/{phpfiwa,phpfina,phptimeseries}
```

```
pi@raspberrypi:~ $ cd /var/www/emoncms && cp
default.settings.php settings.php
```

```
pi@raspberrypi:/var/www/emoncms $ nano settings.php
```

```
pi@raspberrypi: $ cd /var/www/html && sudo ln -s
/var/www/emoncms
```

```
pi@raspberrypi: $ sudo touch /var/log/emoncms.log &&
sudo chmod 666 /var/log/emoncms.log
```

```
pi@raspberrypi: $ sudo visudo
```

### 4.4.1. Dashboard

```
pi@raspberrypi: $ cd /var/www/emoncms/Modules
pi@raspberrypi:/var/www/emoncms/Modules $ git clone
https://github.com/emoncms/dashboard
```

```
pi@raspberrypi:/var/www/emoncms/Modules $ ls -al
```

#### 4.4.2. Graph

```
pi@raspberrypi: $ cd /var/www/emoncms/Modules
pi@raspberrypi:/var/www/emoncms/Modules $ sudo git
clone https://github.com/emoncms/graph
```

```
pi@raspberrypi:/var/www/emoncms/Modules $ ls -al
```

```
pi@raspberrypi: $ nano /var/www/emoncms/settings.php
```

#### 4.4.3. Apps

```
pi@raspberrypi: $ cd /var/www/emoncms/Modules
pi@raspberrypi: $ git clone
https://github.com/emoncms/app.git
```

## 5. MQTT (mosquitto) installation on a Raspberry Pi

### 5.1. Installation of the Mosquitto broker.

```
pi@raspberrypi:~ $ wget
http://repo.mosquitto.org/debian/mosquitto-
repo.gpg.key
```

```
pi@raspberrypi:~ $ sudo apt-key add mosquitto-
repo.gpg.key
```

```
pi@raspberrypi:~ $ cd /etc/apt/sources.list.d/
```

```
pi@raspberrypi:~ $ sudo wget
http://repo.mosquitto.org/debian/mosquitto-jessie.list
```

```
pi@raspberrypi:~ $ sudo wget
http://repo.mosquitto.org/debian/mosquitto-stretch.list
```

```
pi@raspberrypi:~ $ sudo apt-get update
```

```
pi@raspberrypi:~ $ sudo apt-get install mosquitto
```

```
pi@raspberrypi:~ $ sudo mosquitto_passwd -c  
/etc/mosquitto/passwd doorlock  
pi@raspberrypi:~ $ ps -A | grep mosquitto
```

### 5.2. Installation of the Mosquitto client (RPI)

```
pi@raspberrypi:~ $ sudo apt-get install mosquitto-clients
```

### 5.3. Verification of the Mosquitto installation.

```
pi@raspberrypi:~ $ mosquitto_sub -d -t lock/esp8266 -u  
doorlock -P area#51
```

```
pi@raspberrypi:~ $ mosquitto_pub -d -t lock/esp8266 -m  
"D3:CLOSED" -u doorlock -P area#51
```

## 9. ESP8266 Over The Air – update/programming (OTA).

### 9.1.3.1. Firmware build.

### 9.1.3.2. Firmware update

```
D:\New_IKEAHack_wOTA> mkspiffs -p 256 -b 4096 -s 0x0FB000 -  
c data/ spiffs.bin
```

### 9.1.4. CLI OTA

To upload through terminal you can use: `curl -F  
"image=@firmware.bin" esp8266-webupdate.local/update`

```
c:\Users\Hans Henrik>curl -v
```

```
c:\Users\Hans Henrik> curl -F  
"image=@New_IKEAHack_wOTA_20210228_1.ino.d1_mini.bin"  
"172.16.0.11:8148/firmware"
```

```
c:\Users\Hans Henrik> curl -F  
image=@New_IKEAHack_wOTA_20210228_1.ino.d1_mini.bin -u  
admin:admin "172.16.0.11:8148/firmware"
```

## 12. VPN gateway.

### 12.2.1. Basic configuration.

```
pi@raspberrypi:~ $ sudo raspi-config
```

```
pi@ RPIVPNGateway:~ $ sudo nano /etc/dhcpd.conf
```

```
pi@ RPIVPNGateway:~ $ sudo reboot
```

### 12.2.2. OpenVPN installation

```
pi@ RPIVPNGateway:~ $ sudo apt-get update
```

```
pi@ RPIVPNGateway:~ $ sudo apt-get upgrade
```

```
pi@ RPIVPNGateway:~ $ sudo apt-get install openvpn
```

```
pi@ RPIVPNGateway:~ $ cd /etc/openvpn
```

```
pi@ RPIVPNGateway:~ $ sudo cp /boot/<xx>.ovpn  
RPIVPNGateway.ovpn
```

```
pi@ RPIVPNGateway:~ $ ls -al
```

```
pi@ RPIVPNGateway:~ $ sudo openvpn --config  
RPIVPNGateway.ovpn --daemon
```

```
pi@ RPIVPNGateway:~ $ sudo ifconfig
```

#### 12.2.3.1. OpenVPN Authorization file

```
pi@ RPIVPNGateway:/etc/openvpn $ sudo nano vpngwlogin
```

```
pi@ RPIVPNGateway:/etc/openvpn $ sudo nano <xx>.ovpn
```

```
pi@ RPIVPNGateway:/etc/openvpn $ sudo killall openvpn
```

```
pi@ RPIVPNGateway:~ $ sudo openvpn --config  
RPIVPNGateway.ovpn --daemon
```



### 12.2.3.2. Start at boot time.

```
sleep 5
sudo openvpn --config /etc/openvpn/RPIVPNGateway.ovpn --daemon
--keepalive 2 60
exit 0
```

```
pi@ RPIVPNGateway: $ sudo nano /etc/rc.local
```

```
pi@ RPIVPNGateway: $ sudo reboot
```

### 12.2.3.3. Configuration of the VPN gateway (fun with IP tables)

```
pi@ RPIVPNGateway: $ sudo iptables -F
```

```
pi@ RPIVPNGateway: $ sudo iptables -t nat -F
```

```
pi@ RPIVPNGateway: $ sudo iptables -X
```

```
pi@ RPIVPNGateway: $ sudo iptables -t nat -A
POSTROUTING -o tun0 -j MASQUERADE
```

```
pi@ RPIVPNGateway: $ sudo iptables -A FORWARD -i tun0
-o etho0 -m state --state RELATED,ESTABLISHED -j
ACCEPT
```

```
pi@ RPIVPNGateway: $ sudo iptables -A FORWARD -i eth0
-o tun0 -j ACCEPT
```

```
pi@ RPIVPNGateway: $ sudo iptables -L -v
```

```
pi@ RPIVPNGateway: $ sudo sh -c "iptables-save >
/etc/iptables/rules.v4"
```

```
pi@ RPIVPNGateway: $ sudo apt-get install netfilter-
persistent iptables-persistent
```

```
pi@ RPIVPNGateway: $ sudo sh -c "iptables-save >
/etc/iptables/rules.v4"
pi@ RPIVPNGateway: $ sudo sh -c "ip6tables-save >
/etc/iptables/rules.v6"
```

```
pi@ RPIVPNGateway: $ sudo netfilter-persistent save
```

```
pi@ RPIVPNGateway: $ sudo dpkg-reconfigure iptables-
persistent
```

## 13. Software/firmware design.

### 13.2.1. Installation of the utility functions.

```
C:\Appl\arduino\libraries\hhs>doxygen hhs
```

## 25. Software listing.

### 25.1.1. PHP

```
pi@raspberrypi: $ sudo apt-get install php-common php-cli
```

```
pi@raspberrypi: $ php --version
```

### 25.1.2. c-code

```
pi@raspberrypi: $ sudo apt-get install libcurl4-gnutls-dev
```

```
pi@raspberrypi: $ sudo gcc http-post_curl_1.c -lcurl -o ard
```