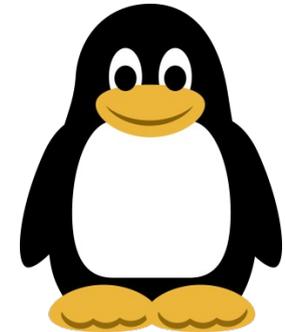


RasPi

Das Wetterdings, ein kleines Wochenendprojekt.

Zum Lernen, Üben und Anwenden von Linux.

Linux! Warum nichts anderes?



Das Beispiel heute läuft unter Linux mit Datenbank und WebServer auf einem RaspberryPi B+ oder RaspberryPi2, BananaPi, CubieBoard, auf Notebooks, Laptops oder Desktops und auch auf „big iron“ wie IBMs Großrechnern der Serie „Z“ und in Virtuellen Maschinen.

Funktioniert das auch mit anderen Betriebssystemen?

RasPi

Das Wetterdings, ein kleines Wochenendprojekt.

Wir wollen mit diesem Beispiel Wetterdaten

wie Temperatur, rel. Feuchte, Luftdruck und Windgeschwindigkeit

aus dem Internet holen,

pfiffig in einer Datenbank speichern,

mit dem Werkzeug „emoncms“ präsentieren

und jederzeit erweitern können.

Die Software ist frei zu nutzen, zu ändern und/oder anzupassen.

RasPi

Das Wetterdings, ein kleines Wochenendprojekt.

1. Teil

- Benötigte Hard- und Software
- Installation der Software „LAMP“



2. Teil

- Anmelden beim Wetterdaten-Lieferanten
- Registrieren und „api-key“ für den Datenzugang erhalten



3. Teil

- „php-script“ zum Abholen der Wetterdaten
- crontab einrichten



4. Teil

- Einrichten von „emoncms“



RasPi



Was benötigen wir an Hardware?

- 1 RaspberryPi hier Ver. B+ mit 512 MB RAM
- 1 PC mit Internetzugang
- 1 SD-Card mit 8 GB
- 1 Möglichkeit SDCards zu beschreiben
- 1 switch um den PC und den RasbPi mit dem Internet zu verbinden



RasPi



Was benötigen wir an Software?

| | | |
|----------|---------|---|
| L | inux | Betriebssystem, hier Raspbian Wheezy |
| A | pache2 | Webserver |
| M | ySQL | Datenbank |
| P | HP5 | Programmiersprache für Web-Anwendungen |
| | cron | Zeitsteuerung, Teil des Betriebssystems |
| | emoncsm | Die Anwendung |

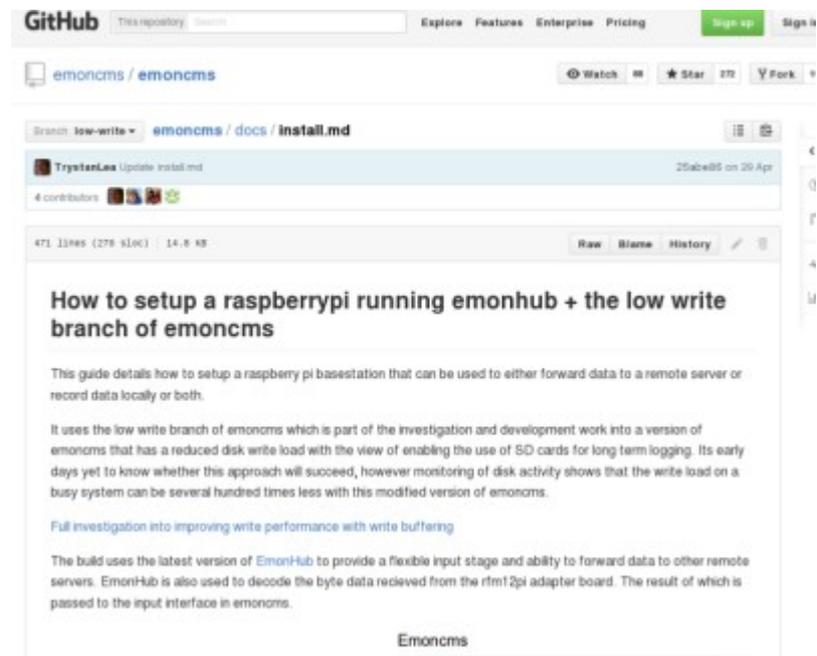
Wir finden alles im Internet passend für unser Projekt vorbereitet.

RasPi



Jetzt geht es los!

<https://github.com/emoncms/emoncms/blob/low-write/docs/install.md>



und arbeiten dann von hier alles bis zum

bis zu dem Punkt:

„Disable Serial console“ ab.

Zuerst also das Image holen.

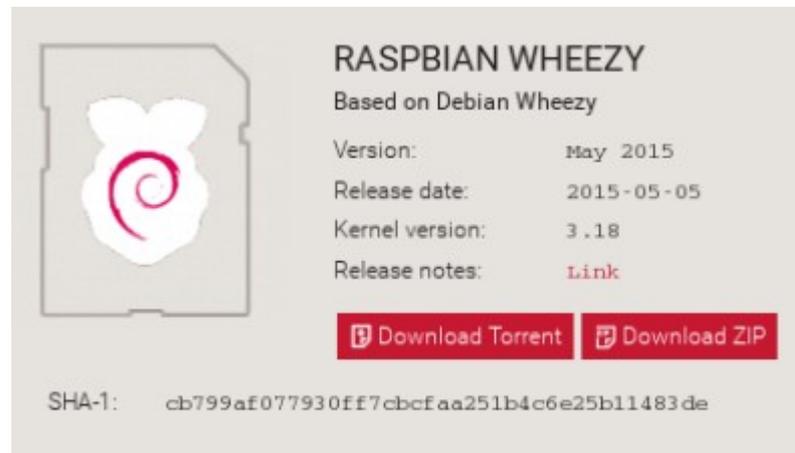
RasPi



Jetzt wird es ernst!

Auf <http://www.raspberrypi.org/downloads> wählen wir die Version

Raspbian Wheezy 2015-05-05



und schreiben das Image auf eine 8 GB SD-Card (größer funktioniert auch).

Wie man das macht ist dort beschrieben und es klappt auch unter anderen

Betriebssystemen, die nicht Linux heißen. (Siehe auch Tipps.)

Achtung: Auf die SD-Card schreiben!!! Nicht auf eine andere Partition!!!

RasPi



Weiter geht es!

Ohne Englischkenntnisse kann es recht schwierig werden.

Da hilft man sich einmal gegenseitig.

Wirklich jeden Schritt abarbeiten bis zu „Disable Serial Console“!

Siehe „Tipps“ auf den letzten Seiten dieses Vortrags .

Die Tipps zusammen mit den Installationsanweisungen benutzen.

Wer sich vertippt, der muß nochmal!

Daher auch die Beschreibung des Projekts:

Ein kleines Wochenendprojekt!

RasPi



Jetzt haben wir

ein Linuxsystem (Raspbian Wheezy),
einen Webserver (Apache2),
eine Datenbank (MySQL),
PHP5,
und das Programm emoncms installiert.

Probieren wir unsere Installation aus!

RasPi



http://IP-von-unserem-RasPi/emoncms

und weiter



Username:

Password:

Remember me

or [register](#)



Username:

Email:

Password:

Confirm password:

or [cancel](#)

RasPi



Input Feeds Vis Dashboard

My account

Username

Paul [Edit](#)

Email

pp@paul.home.de [Edit](#)

[Change Password](#)

Write API Key

a32aa438be5f3e170fab8f8b6adea9a0

Read API Key

a2988b9270fc19d2bad0d0d1548b2777

My Profile

Gravatar

Name N/A

Location N/A

Timezone UTC +0:00

Language en_EN

Bio N/A

Achtung! Diese Information benötigen wir noch später!

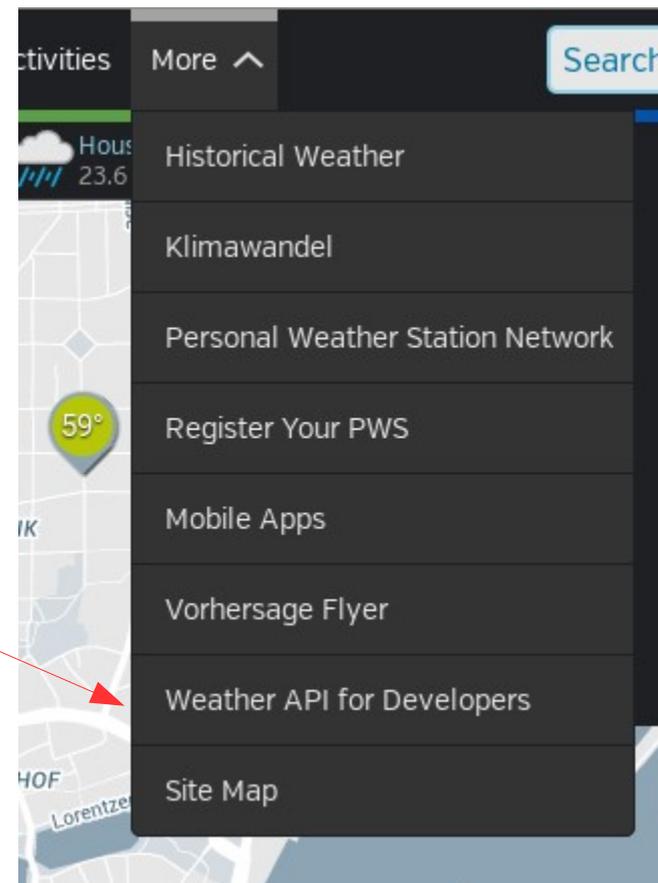
Es funktioniert und der wichtigste Schritt ist getan!

Ausloggen und nun zum Wetter.

RasPi



Wie kommen wir jetzt an die Wetterdaten?



Hier sind wir richtig, denn wir benötigen das Entwickler-API um kostenlos 500 mal am Tag die ausgewählten Daten abzuholen:

Für den Anfang: Temperatur, Luftdruck, rel. Luftfeuchtigkeit, Windgeschwindigkeit

RasPi

Das ist die richtige Wahl



A WEATHER API DESIGNED

Luftdruck

[API Home](#)

[Pricing](#)

[Featured Applications](#)

[Document](#)

Get in touch if you would like access to our history data.

How much will you use our service?

| | Monthly Pricing | Calls Per Day | Calls Per Minute |
|--|---|---------------|------------------|
| <input checked="" type="radio"/> Developer | \$0 | 500 | 10 |
| <input type="radio"/> Drizzle | \$20 | 5000 | 100 |
| <input type="radio"/> Shower | \$200 | 100,000 | 1000 |
| <input type="radio"/> Downpour | Get in touch for more than 100,000 calls per day. | | |

Your Selected Plan: Stratus Developer

[Purchase Key >>](#)

Monthly Pricing**

\$0

Calls Per Day

500

Calls Per Minute

10

+ History

Not
Included

TOTAL

\$0 USD per month

RasPi

wunderground.com möchte natürlich wissen wer einen api-key braucht und was er damit macht.

Hier einige Beispielangaben:

Alle Angaben notieren oder einen screen-shot machen!

*All fields are required

Contact Name
Mein Name

Project Contact Email
pp@paul.home.de

Project Name
A RaspberryPi Weekender

Project Website
None

Where will the API be used?
 Website Mobile Both Other

Will the API be used for commerical use?
 Yes No

Will the API be used for manufacturing mobile chip processing?
 Yes No

What country are you or your company based in?
Germany

Please give a brief description of how you will be using our API
255 character max length (200 remaining characters...)
A RaspberryPi project to display weather data at home.

I understand that usage of the Weather Underground API requires proper attribution

I agree to the [Terms of Service](#)

[Purchase Key >>](#)



RasPi



ACHTUNG!
Bei der Angabe „handle“ nur Buchstaben

START NOW - DEVELOP FOR FREE!

[API Home](#) [Pricing](#) [Featured Applications](#) [Documentation](#) [Forums](#)

Create Your Free Account!

*All fields are required

[What's a Handle?](#)

I agree to the [Terms of Service](#).

Sign Up >>

Already have a wunderground.com account?

[forgot password?](#)

Login >>

RasPi



Jetzt warten wir auf die email mit den Daten für die Aktivierung unseres Zugangs und loggen uns dann mit dem Passwort ein, um den api-key zu erhalten. Auch hier ist eine gute Idee, den api-key zu notieren.

Die Datei: weather.php

wird auf den beiden folgenden Seiten vorgestellt, abgeschrieben und bei unserem RasPi in das Verzeichnis /var/www kopiert.

Auch hier gilt:

Wer sich vertippt, der muß nochmal!



RasPi



```
<html>  
<?php
```

```
//http://www.wunderground.com/weather/api/d/documentation.html
```

```
//Change the weather underground API and city below
```

```
$json_string = file_get_contents("http://api.wunderground.com/api/api-key-hier-  
eintragen/geolookup/conditions/q/Kiel-Holtenau.json");
```

```
$parsed_json = json_decode($json_string);
```

```
$location = $parsed_json->{'location'}->{'city'};
```

```
$temp_c = $parsed_json->{'current_observation'}->{'temp_c'};
```

```
$relative_humidity = $parsed_json->{'current_observation'}->{'relative_humidity'};
```

```
$pressure_mb = $parsed_json->{'current_observation'}->{'pressure_mb'};
```

```
$wind_kph = $parsed_json->{'current_observation'}->{'wind_kph'};
```

RasPi



```
echo "Current temperature in ${location} is: ${temp_c}";  
echo "Current relative humidity in ${location} is: ${relative_humidity}";  
echo "Current pressure in ${location} is: ${pressure_mb}";  
echo "Current wind speed in ${location} is: ${wind_kph}";
```

```
//Change the URL and EmonCMS API below
```

```
$url = 'http://IP-von-unserem-RasPi/emoncms/input/post.json?json={humidity:' .  
$relative_humidity . ',pressure:' . $pressure_mb . ',wind:' . $wind_kph . ',temp:' .  
$temp_c .}'&node=1&apikey=emoncms-write-api-key-hierher-kopieren';  
echo $url;  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL,$url);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
$contents = curl_exec ($ch);  
curl_close ($ch);
```

```
?>  
</html>
```

RasPi



Test der Wetterdaten mit:

```
/usr/bin/php /var/www/weather.php
```

```
pi@raspberrypi ~ $  
pi@raspberrypi ~ $ /usr/bin/php /var/www/emoncms/weather.php  
<html>  
Current temperature in Kiel-Holtenau is: 10.2Current relative humidity in Kiel-Holtenau is: 81%Current pressure in Kiel-Holtenau  
is: 1013Current wind speed in Kiel-Holtenau is: 0http://192.168.102.152/emoncms/input/post.json?json={humidity:81%,pressure:1013,  
wind:0,temp:10.2}&node=1&apikey=51469073f8643be8f1f77430869e23d6</html>  
pi@raspberrypi ~ $ █
```

...und so könnte das Ergebnis aussehen.

RasPi



Damit regelmäßig alle 10 Minuten neue Daten von wunderground.com geholt werden, ist noch ein cronjob anzulegen.

```
rpi-rw
```

```
sudo crontab -e
```

Als letzte Zeile hinzufügen:

```
*/10 * * * * /usr/bin/php /var/www/weather.php
```

und speichern mit

[Ctrl + X] then [Y] then [Enter] to Save and exit.

RasPi



Nachdem bis hierher alles geklappt hat

wollen wir den Test wagen:

<http://IP-von-unserem-RasPi/emoncms>

und einloggen:

RasPi

Inputs

| Node: | Key | Name | Process list | last updated | value | | | |
|-------|----------|------|---------------------|--------------|-------|--|--|--|
| 1 | humidity | | log | 46s ago | 72.0 | | | |
| 1 | temp | | log | 46s ago | 17.0 | | | |
| 1 | pressure | | log | 46s ago | 1024 | | | |

Node1: humidity config

Input processes are executed sequentially with the result being passed back for further processing by the next processor in the input processing list.

| Order | Process | Arg | Actions |
|-------|-------------|---|---------|
| 1 | Log to feed | Node:1: 1:humidity (feedvalue:100.00) | |

Add process:

Log to feed Feed engine: Fixed Interval No Averaging (

Log to feed: This processor logs the current selected input to a timeseries feed which can then be used to explore historic data. This is recommended for log temperature, humidity, voltage and current data.

Feed interval: When selecting the feed interval select an interval that is the same as, or longer than the update rate that is set in your monitoring equipment. An interval rate to be shorter than the update rate of the equipment causes un-needed disk space to be used up.

RasPi

Input Feeds Vis Dashboard Admin Account Logout 

Inputs

[Input API Help](#)

| Node | Key | Name | Process list | last updated | value | | | |
|------|----------|------|---------------------|--------------|-------|--|--|--|
| 1 | humidity | | log | 46s ago | 72.0 | | | |
| 1 | temp | | log | 46s ago | 17.0 | | | |
| 1 | pressure | | log | 46s ago | 1024 | | | |
| 1 | wind | | log | 46s ago | 13.0 | | | |

Input Feeds Vis Dashboard Admin Account Logout

Feeds

[Feed API Help](#)

| Id | Name | Tag | Engine | Public | Size | Updated | Value | | | | |
|----|------------|--------|---------|--------|-------|----------|-------|--|--|--|--|
| 4 | 1:humidity | Node:1 | PHPFINA | | 0.0kb | 136s ago | 72.0 | | | | |
| 5 | 1:temp | Node:1 | PHPFINA | | 0.0kb | 136s ago | 17.0 | | | | |
| 6 | 1:pressure | Node:1 | PHPFINA | | 0.0kb | 136s ago | 1024 | | | | |
| 7 | 1:wind | Node:1 | PHPFINA | | 0.0kb | 136s ago | 13.0 | | | | |

Refresh feed size

RasPi

Input Feeds Vis Dashboard

Visualisations

1) Select visualisation:

2) Set options:

No multigraphs created yet, click new to create one:

[New multigraph](#)

Note: If a feed does not appear in the selection box, check that the type has been set on the feeds page.

3) [Full screen](#)

Embed in your website:

Input Feeds Vis Dashboard

Visualisations

1) Select visualisation:

2) Set options:

[New multigraph](#)

Note: If a feed does not appear in the selection box, check that the type has been set on the feeds page.

3) [Full screen](#)

Embed in your website:

RasPi

Input Feeds Vis Dashboard

Visualisations

1) Select visualisation: multigraph

2) Set options:

1: null

Name: null

| Feed | Left | Right | Fill |
|------------|--------------------------|--------------------------|--------------------------|
| 1:humidity | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Floating time

Saved

Note: If a feed does not appear in the selection box, check that the type has been set on the feeds page.

3)

Embed in your website:

Input Feeds Vis Dashboard 

Visualisations

1) Select visualisation: multigraph

2) Set options:

1: null

Name: null

| Feed | Left | Right | Fill |
|------------|--------------------------|--------------------------|--------------------------|
| 1:humidity | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Line Colour:

1:temp

Floating time

Saved

Note: If a feed does not appear in the selection box, check that the type has been set on the feeds page.

3)

RasPi



Input Feeds Vis Dashboard

Visualisations

1) Select visualisation: multigraph

2) Set options:

1: null

Name: null

| Feed | Left | Right | Fill | |
|--------------|-------------------------------------|--------------------------|--------------------------|----------------------------------|
| 1:humidity | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="button" value="x"/> |
| Line Colour: | <input type="text" value="black"/> | | | |
| 1:temp | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="button" value="x"/> |
| Line Colour: | <input type="text" value="black"/> | | | |

1:humidity

Floating time

Note: If a feed does not appear in the selection box

RasPi

Input Feeds Vis Dashboard

Dashboards: [no name](#)

Dashboard

| Id | Name | Alias | Main | Public |
|----|---------|-------|------|--------|
| 1 | no name | | ☆ | 🔒 |



Input Feeds Vis Dashboard

Dashboards: [no name](#)

Text ▾ Containers ▾ Widgets ▾ Visualisations ▾ [Configure](#) [Delete](#)

RasPi



Configure element

Feed: Feed value

Max value: Max value to show

Scale: Value is multiplied by scale before display

Units: Units to show

Offset: Static offset. Subtracted from value before computing

Type: Type to show

Graduations: Should the graduation limits be shown

Text ▾ Containers ▾ Widgets ▾ Visualisations ▾

72.0

Title

RasPi

Text ▾ Containers ▾ Widgets ▾ Visualisations ▾  Configure  Delete

13

72.0
rel.
Feuchte

1 hour 30 min 15 min 5 min 1 min

Please select a feed

Das Wetterdings, ein kleines Wochenendprojekt und so könnte es aussehen.

Wetterdaten aus Holtenau



Temp



Rel Feuchte

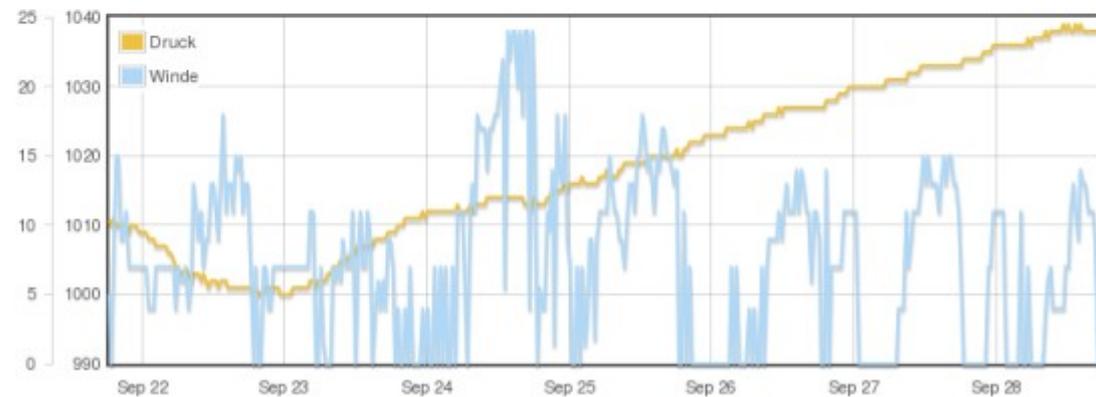
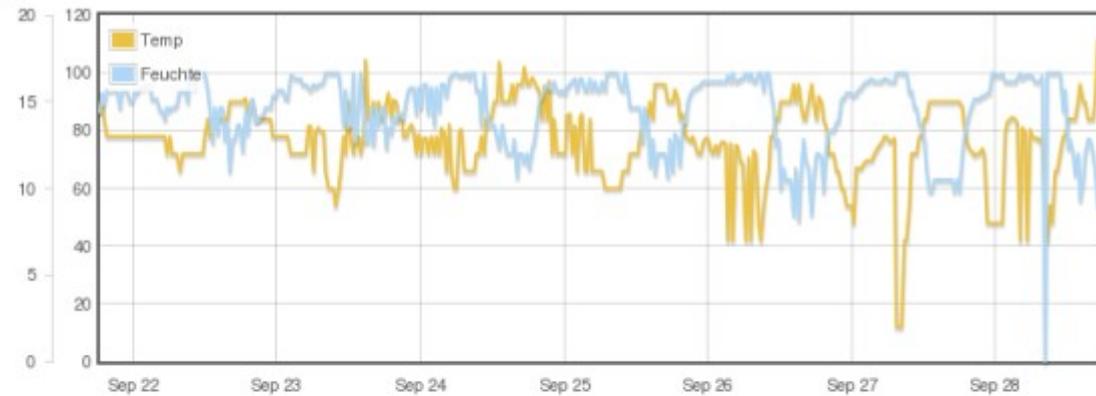


Luftdruck



Wind

50



Hier starten und eine 8 GB Sd-Card benutzen:

<https://github.com/emoncms/emoncms/blob/low-write/docs/install.md>

Auf <http://www.raspberrypi.org/downloads>

wählen wir die Version

Raspbian Wheezy 2015-05-05

und schreiben das Image auf eine 8 GB SD-Card, es kann auch eine größere sein.

Alle Schritte konsequent abarbeiten und es kann und es wird dauern!!!!

Schluss bei „Disable Serial Console“! Benötigen wir nicht!

Hier mein Beispiel für die Partitionen auf der 8 GB SD-Card.

```
pi@raspberrypi ~ $ sudo fdisk -l
Disk /dev/mmcblk0: 7948 MB, 7948206080 bytes
4 heads, 16 sectors/track, 242560 cylinders, total 15523840 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa6202af7

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1      8192        122879     57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2     122880     6399999    3138560   83   Linux
pi@raspberrypi ~ $ █
pi@raspberrypi ~ $ sudo fdisk /dev/mmcblk0
Command (m for help): n
Partition type:
  p   primary (2 primary, 0 extended, 2 free)
  e   extended
Select (default p): p
Partition number (1-4, default 3): 3
First sector (2048-15523839, default 2048): 6400000
Last sector, +sectors or +size{K,M,G} (6400000-15523839, default 15523839): +1G

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
pi@raspberrypi ~ $ █
```

Hier anpassen!

Passwort beim Installieren von MySQL notieren!

Vor dem Punkt „Install Dependencies“ schaffen wir noch etwas Platz:

```
sudo apt-get remove --purge wolfram-engine triggerhappy
```

Weiter mit „Install Dependencies“:

comment out all redis saving

```
# save 900 1  
# save 300 10  
# save 60 10000
```

```
sudo /etc/init.d/redis-server start
```

[Ctrl + X] then [Y] then [Enter] to Save and exit.

Hier funktionierte es bei mir nur mit „restart“, da der Server bereits lief.

```
?php
/*
Database connection settings
*/
$username = "root";
$password = "emoncms";
$server   = "localhost";
$database = "emoncms";

$redis_enabled = true;
```

/var/www/emoncms/settings.php

Hier das Passwort des MySQL-Servers eintragen. Der Rest kann so bleiben.

Im Programm „emoncms“ führt das Schraubenschlüsselsymbol immer zu weiteren Einstellmöglichkeiten.

Alle Daten werden auf dem Raspberry Pi gespeichert und nicht auf die Server von „emoncms.org“ hoch geladen.

Selbstverständlich soll so ein kleiner Server hinter einer Firewall oder einem Router betrieben werden.

Die Wetterdaten kommen von: „<http://www.wunderground.com>“

Es ist eine gute Idee sich die Informationen, die wunderground abfragt, zu notieren.

Es passiert, dass kein api-key bei der Anmeldung erzeugt wird.

Deshalb muß manchmal alles 2 mal eingegeben werden!

„handle“ (Spitzname) akzeptiert nur Buchstaben!

Hier findet sich die Beschreibung der Datei „weather.php“

<http://openenergymonitor.org/emon/node/589>

Der Standort der Wetterstation, die Anzahl der geholten Werte, es kann alles später noch angepasst werden.

Start mit: <http://IP-von-unserem-RasPi/emoncms>

Stopp mit: „sudo shutdown -hP now“, sonst werden die Daten aus dem RAM nicht weggeschrieben und fehlen hinterher!

Die SD-Card überlebt auf diese Weise auch!!!!

RasPi

Danke an

Glyn und Trystan von openerenergymonitor.org und emoncms.org

Martin Harizanov für “Software only solution for feeding weather data“

die Macher und Betreiber von „www.raspberrypi.org“

und an Alle, die Beiträge und Ideen für dieses Projekt geliefert haben.

Gibt es noch Fragen?