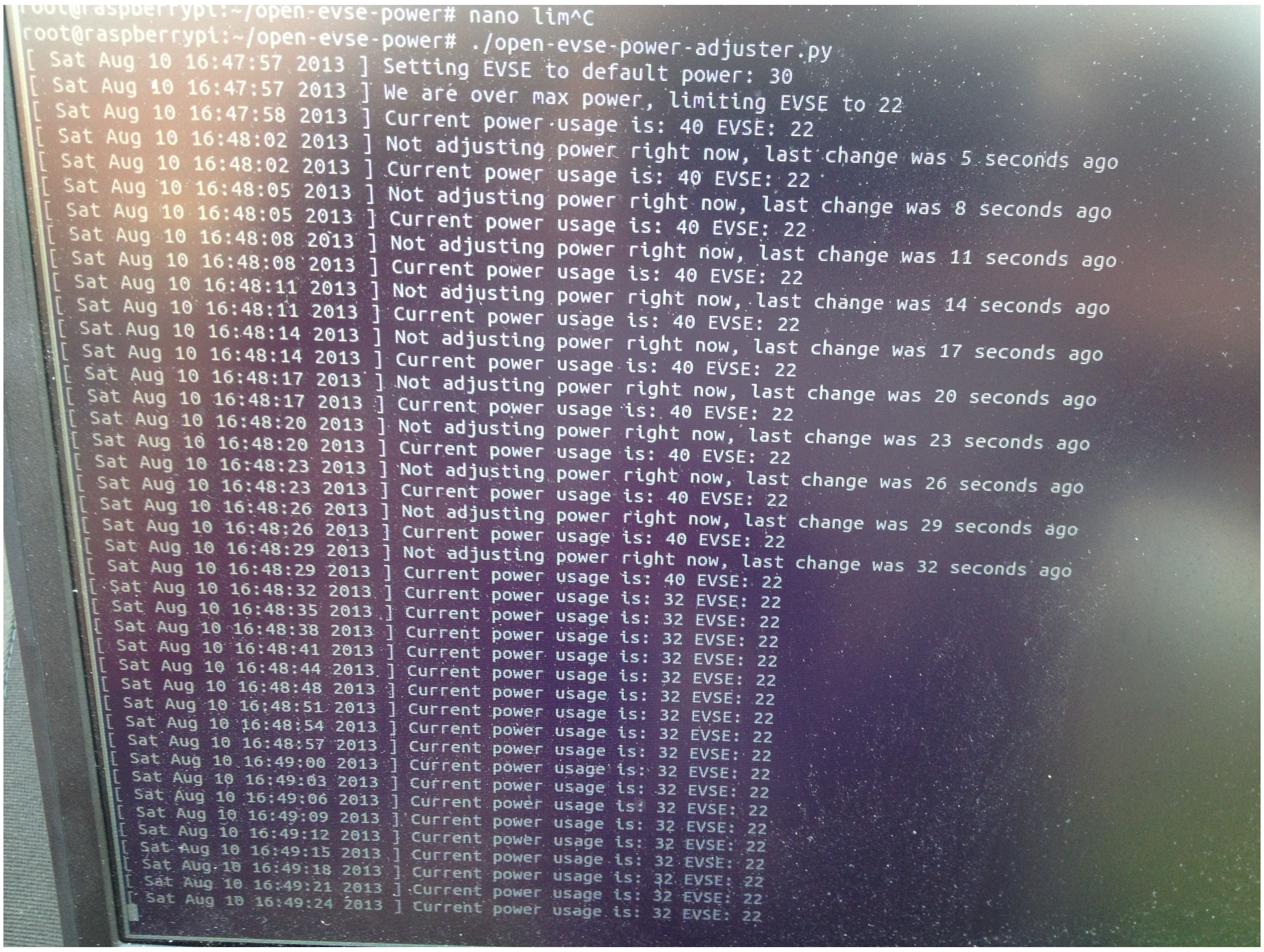
Automatic Power Adjustment with Open EVSE



```
root@raspberrypi:~/open-evse-power# nano lim^C
root@raspberrypi:~/open-evse-power# ./open-evse-power-adjuster.py
[ Sat Aug 10 16:47:57 2013 ] Setting EVSE to default power: 30
[ Sat Aug 10 16:47:57 2013 ] We are over max power, limiting EVSE to 22
[ Sat Aug 10 16:47:58 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:02 2013 ] Not adjusting power right now, last change was 5 seconds ago
[ Sat Aug 10 16:48:02 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:05 2013 ] Not adjusting power right now, last change was 8 seconds ago
[ Sat Aug 10 16:48:05 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:08 2013 ] Not adjusting power right now, last change was 11 seconds ago
[ Sat Aug 10 16:48:08 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:11 2013 ] Not adjusting power right now, last change was 14 seconds ago
[ Sat Aug 10 16:48:11 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:14 2013 ] Not adjusting power right now, last change was 17 seconds ago
[ Sat Aug 10 16:48:14 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:17 2013 ] Not adjusting power right now, last change was 20 seconds ago
[ Sat Aug 10 16:48:17 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:20 2013 ] Not adjusting power right now, last change was 23 seconds ago
[ Sat Aug 10 16:48:20 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:23 2013 ] Not adjusting power right now, last change was 26 seconds ago
[ Sat Aug 10 16:48:23 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:26 2013 ] Not adjusting power right now, last change was 29 seconds ago
[ Sat Aug 10 16:48:26 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:29 2013 ] Not adjusting power right now, last change was 32 seconds ago
[ Sat Aug 10 16:48:29 2013 ] Current power usage is: 40 EVSE: 22
[ Sat Aug 10 16:48:32 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:35 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:38 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:41 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:44 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:48 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:51 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:54 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:48:57 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:00 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:03 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:06 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:09 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:12 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:15 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:18 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:21 2013 ] Current power usage is: 32 EVSE: 22
[ Sat Aug 10 16:49:24 2013 ] Current power usage is: 32 EVSE: 22
```

Author: Wido den Hollander <wido@widodh.nl>
Date: August 2013

Table of Contents

The need for automatic power adjustment

Being able to quickly charge your Electric Vehicle (EV) is great, but charging a EV quickly can put a lot of load on the electrical infrastructure in your house.

I live in the Netherlands where we use 3-phase power. In my case my house has a 3x40A connection, which means I have 3 main fuses at 40Amp, one on each phase.

An electric vehicle like the Tesla Model S can draw 3x32A, so when charging at maximum power there will only be 8A left on each phase.

I could have my connection expanded from 3x40A to 3x50A or even 3x63A, but that would increase my monthly invoice with tens of Euros per month, something I'm not willing to do.

Wouldn't it be great if the car would use less power when demand in the house increases? If for example my girlfriend turns on the oven while the car is charging.

I've build exactly that using off the shelf components! (And some Python code)

Open EVSE

Before I start with summing up all the components I've used it's important to understand the key component in this.

EVs won't charge quickly on dumb sockets, they need some intelligence talking to them instructing them what to do. We call this a EVSE (Electric Vehicle Supply Equipment).

A EVSE talks to the car when it plugs in and will tell it how much current it is allowed to draw.

You can buy commercial EVSEs from small and large companies or you can build on yourself.

Since I'm a true believer in Open Source software and I love fiddling with electronics I build my EVSE using the Open EVSE project.

A document of how I build my EVSE is available here.

Automatic Power Adjustment

Open EVSE supports changing the MaxCurrent using the CLI. You can do this manually as I demonstrate in this video: http://www.youtube.com/watch?v=m-Q_ap7IETI

As you can see in the video, the Tesla Roadster (my colleagues) responds instantly to change in maximum current. With this support in Open EVSE I was able to build logic around this and do the adjustments automatically.

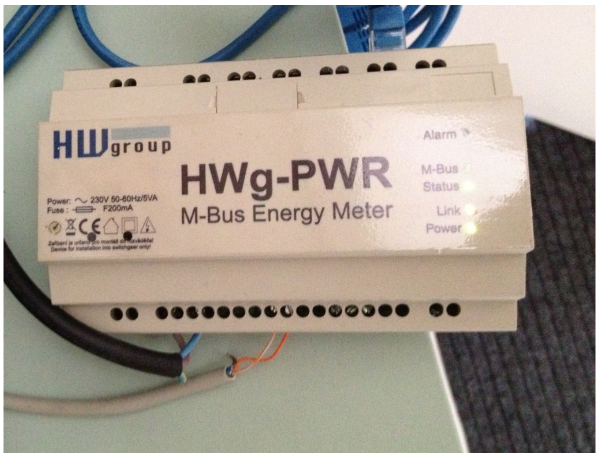
Measuring Power Usage

Before I could build any logic around this I had to know how much power my house was actually using. To do so I added two components to my main distribution panel:

- A kW/kWh/Amp meter with a M-Bus output (Finder 7E.46.8.400.0022)
- A M-Bus to SNMP (IP/Ethernet) poller (HWg-PWR Energy Meter)

These components enabled me to read out in almost real-time how much power my house is using.

The kW meter from Finder is installed in front of all:



So I now have a way to adjust the current that the car draws, but I'm also able to measure how much power the house is using in total.

The last step is to do something with this information.

Adjusting Max Current

I'm using a RaspberryPi mini-computer to do this for me. It uses only 5W and is running Raspbian, a special version of Debian GNU Linux.

Since the Open EVSE has a TTL serial port and the Raspberry Pi doesn't I'm using a USB to TTL cable purchased a DX.com (DealeXtreme)

Now it was a matter of writing some code which would read the values out of the HWg-PWR meter and based on that control the Open EVSE using the serial console.

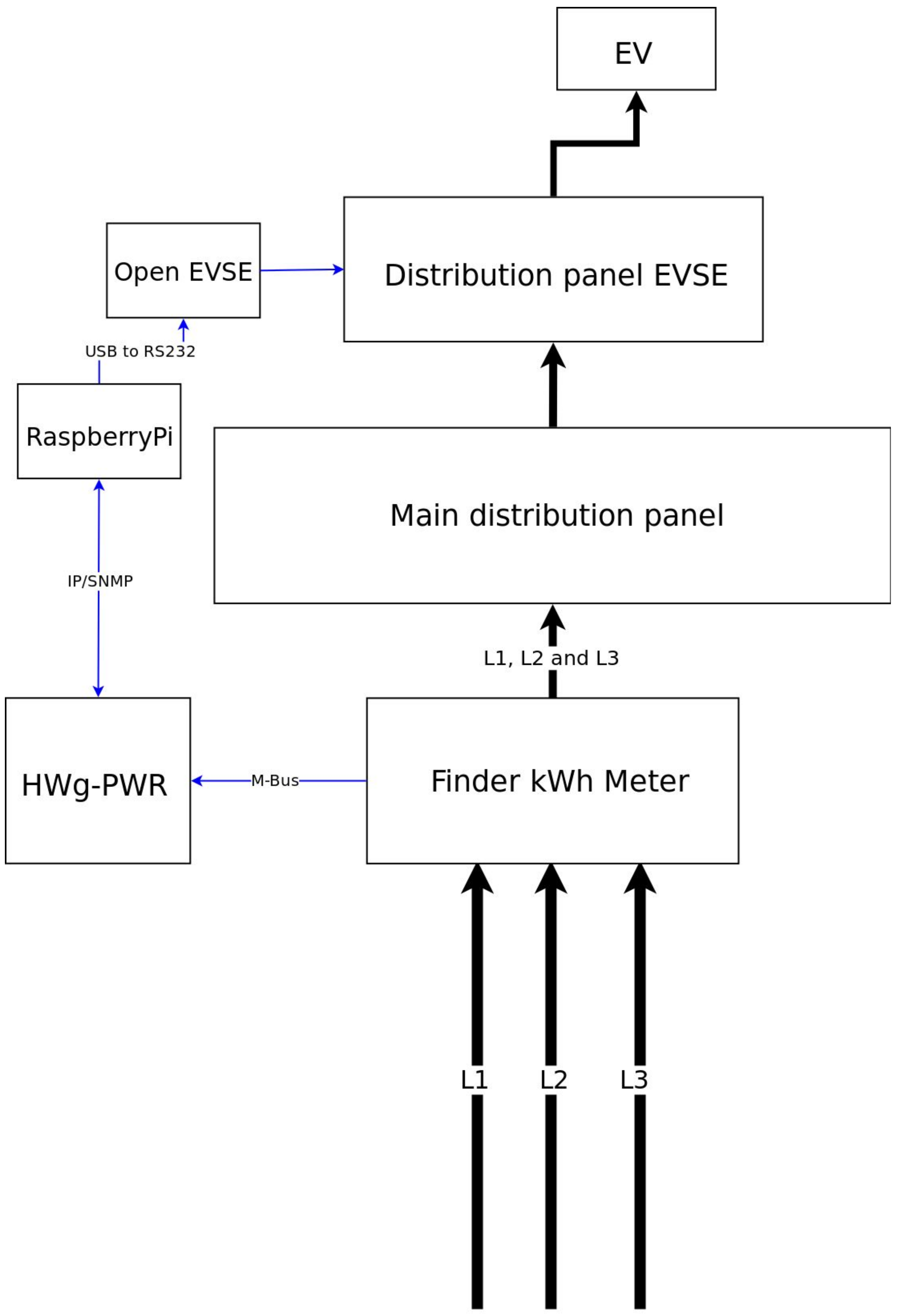
I used Python to read out the values from the HWg-PWR meter and do some calculations with that.

This code can be found at my Github page.

How it all works

Where it comes down to is that I'll never go over the maximum of 40Amps in my house when I'm charging my car and my girlfriend decides to turn on the oven.

A overview of the components and how they are tied together:



The Python code on the RaspberryPi polls the HWg-PWR meter every 10 seconds and compares the output of that meter to the maximum I defined.

If it is over the maximum allowed it will calculate how much I'm over the maximum and it will adjust the EVSE so that I'll be below max power again. For the sake of safety I adjust the EVSE to go 2A lower, just to be sure.

The code (for the most recent version, see Github):

```
#!/usr/bin/env python
import serial
import netsnmp
import time

serial_port    = '/dev/ttyUSB0'
baud_rate      = 38400
snmp_host      = 'XX.XX.XX.XX'
snmp_community = 'XXXX'
max_power      = 40
max_evse_power = 30
poll_interval  = 10

backoff_time   = 120

# No user configuration beyond this point
def logmsg(message):
    print "[ %s ] %s" % (time.ctime(time.time()), message)

def setEVSEPower(power):
    … write to serial console ..

def pollPowerUsage():
    … Do SNMP stuff....
    return int(max(power))

# At first we always reset the EVSE to max power
logmsg('Setting EVSE to default power: ' + str(max_evse_power))
setEVSEPower(max_evse_power)

current_evse_power = max_evse_power

limit_time = time.mktime(time.gmtime())
limited = False

while True:
    try:
        power = pollPowerUsage()

        if power > max_power:
            previous_evse_power = current_evse_power
            current_evse_power = max_power - (power - max_power) - 2
            last_power_change = time.mktime(time.gmtime()) - limit_time

            if (previous_evse_power <= current_evse_power) and (last_power_change < backoff_time):
                logmsg("Not adjusting power right now, last change was %d seconds ago" % last_power_change)
            else:
                limit_time = time.mktime(time.gmtime())
                limited = True
                logmsg('We are over max power, limiting EVSE to ' + str(current_evse_power))
                setEVSEPower(current_evse_power)

        if limited == True:
            now = time.mktime(time.gmtime());
            if ((now - limit_time) > backoff_time):
                if (power + (max_evse_power - current_evse_power) > max_power):
                    logmsg("Not bringing EVSE back to normal level, would go over %d" % max_power)
                else:
                    logmsg('Bringing EVSE back to normal level: ' + str(max_evse_power))
                    current_evse_power = max_evse_power
                    setEVSEPower(current_evse_power)
                    limited = False
            else:
                logmsg("Not considering changing level, still to early to change")

        logmsg('Current power usage is: ' + str(power) + ' EVSE: ' + str(current_evse_power))
        time.sleep(poll_interval)

    except (KeyboardInterrupt, SystemExit):
        break
```

(I snipped out some code)

When the power usage drops below *max_power* (40A) it will wait for two minutes (*backoff_time*) before increasing the maximum current in the EVSE. It will however never set the EVSE higher then 30A as I defined with *max_evse_power*

For me it is still work in progress, I might have to change some logic here and there, but this is the basic idea.

Component List

Below is a list of the component I used and where I purchased them.

Component	**SKU/Article number**	**Shop**	**Price (~)**	**Function**
Finder kW meter	7E.46.8.400.0022	Conrad.nl	EUR 250	kW/kWh/A meter
HWg-PWR	Ethernet M-Bus energy meter	Dynatronics.nl	EUR 400	M-Bus to IP/SNMP
Open EVSE	Open EVSE v6 board	Chris Howell	EUR 140	The EVSE itself
RaspberryPi	Pi Model A 256MB	Farnell	EUR 30	RaspberryPi computer
USB to TTL	PL2302HW 150928	DX.com	EUR 4	Comm with EVSE

I made a video which shows all the components in action: http://www.youtube.com/watch?v=QFfDsoljem8