

Waschmaschinen Online im Wohnheim

Online Nachvollziehbarkeit der Waschmaschinennutzung

Ein ungeeichtes Strommessgerät für mehrere Geräte und die Anzeige auf der Homepage

Zweck

Einige hundert Bewohner teilen sich im Wohnheim vier Waschräume mit insgesamt 20 Waschmaschinen und Trocknern. Durch das Waschmaschinen Online Projekt können alle Bewohner unnötige Wege in die temporär voll ausgelasteten Waschräume vermeiden und genau sehen, wann ihre Wäsche fertig ist.

Aus Betreibersicht hat dies das Potential die Auslastung zu verbessern bzw. die Zufriedenheit der Nutzer zu verbessern, ohne die Anzahl der Maschinen zu erhöhen. Falsche Defektmeldungen durch Nutzer lassen sich schneller aussortieren und defekte Heizungen an Phase 1 erkennen.

Funktionsweise

Hardware

Stromwandler werden direkt im Sicherungskasten auf den Leiter der Motor/Elektronik-Phase der Maschinen gesteckt. Dadurch beschränkt sich die Installation auf einen zentralen Ort pro Waschaum, während die Installation von Gerätewechseln unabhängig bleibt.

Zusätzlich wird im Sicherungskasten ein Messmodul auf der Hutschiene installiert, an dem alle Stromwandler angeschlossen werden. Der Mikrocontroller dieses Moduls misst die Spannung der Stromwandler und gibt die Werte über eine USB Schnittstelle aus.

Ein softwaremodifizierter OpenWrt Router nimmt die Werte entgegen. Der Webserver, der die Waschmaschinen Online-Seite für die Nutzer ausgibt, holt die Daten von dem OpenWrt Router.

Anforderungen

- Etwas Platz im Sicherungskasten für das Messmodul
- Steckdose am/im Sicherungskasten
- Netzwerkzugang am/im Sicherungskasten
- OpenWrt-tauglicher Router mit 1x Ethernet und 1x USB

Software

Der Mikrocontroller im Hutschiene Modul wird so programmiert, dass er für je eine Sekunde mit 400Hz die Quadratsummen der ADC-Messwerte für die Stromwandler aufsummiert und an der seriellen Schnittstelle ausgibt. Ein USB/Seriell-Wandler übernimmt das Umsetzen auf USB.

Der OpenWrt Router muss den USB/Seriell-Wandler unterstützen. Er nimmt mithilfe des Programms ser2net die Daten von dem USB/Seriell Wandler und gibt sie auf einem festen

Netzwerkport (TCP) aus.

Der Webserver des Wohnheims holt die Daten von dem OpenWrt Router (mit bash-Skripten) und filtert simple Spitzen (mit einem awk-Skript).

Die Wohnheimshomepage ist eine Dokuwiki. In ihr werden die Plots der Daten über php-includes in eingebunden. Ein Aufruf der Seite stößt über php-include → bash-skript (Datenaufbereitung und fertige Plot-Bilder zur Wiki kopieren) → Gnuplot-Skript das erstellen aktueller Nutzungsplots an.

Benötigte Software

- Firmware für das Messmodul
- OpenWrt mit ser2net, kmod-usb-serial-cp210x
- Webserver mit Gnuplot, bash, awk, netcat, Unterstützung für phpincludes

Hardware

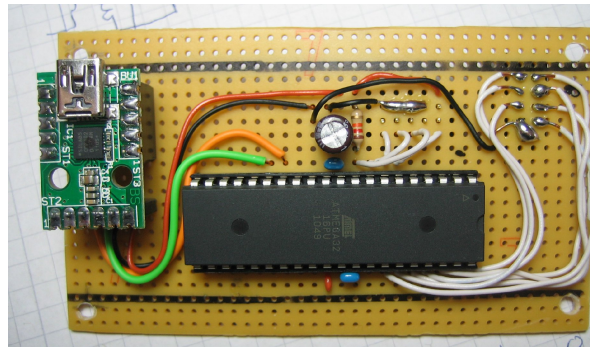
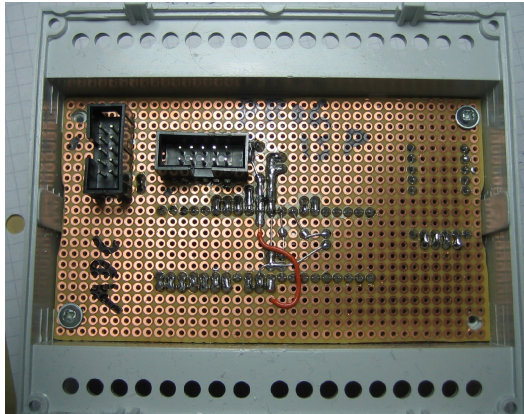
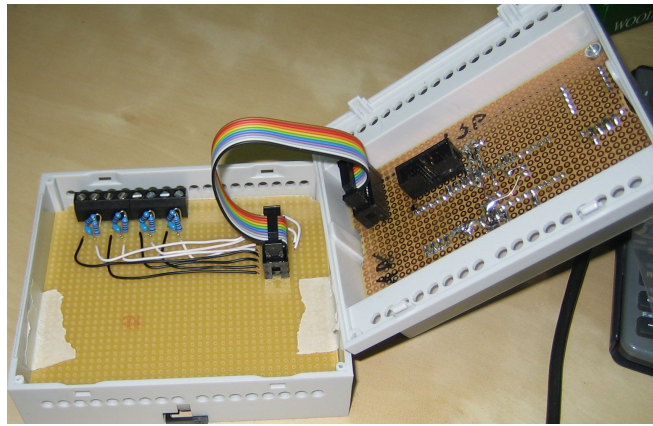
Stromwandler

Stromwandler AC1020, angelötet an eine 2x0,75mm² Schlauchleitung (Kabel) mit 0,5-1m Länge, je nach Sicherungskasten (Verteiler/Schaltschrank/...). Zwecks Zugentlastung wird das Kabel vor dem Stromwandler mit zwei Kabelbindern versehen und – auch zur Isolierung – mit Schrumpfschlauch überzogen. Das andere Ende wird mit Aderendhülsen versehen.



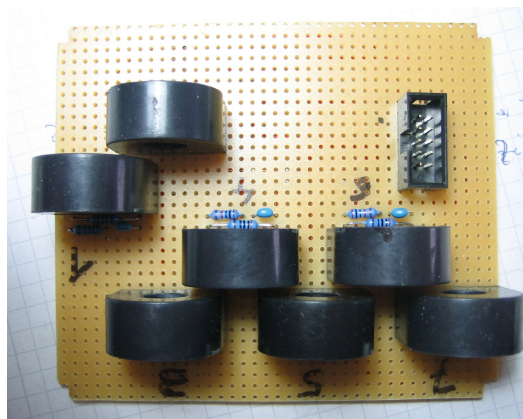
Messmodul für die Hutschiene mit μ C/ADC-Wandler

Das Gehäuse enthält zwei Lochrasterplatten von unten 34x39 Loch (~10x8,5cm) und oben 22x39 Loch (~10x5,7cm). Der USB/Seriell Wandler von ELV wird mit Buchsen/Pfostenleisten gesteckt und sitzt damit hoch genug, um den USB-Port im Sicherungskasten immer erreichen zu können. Obere und untere Platine werden über ein kurzes Flachbandkabel verbunden. Die Funktion der unteren Platine beschränkt sich auf das Tragen der Anschlussklemmen für die Stromwandler, deren Lastwiderstände und RC-Filter. Die Öffnung für den USB-Port wurde von Hand nach Augenmaß gedremelt. Schaltplan siehe Anhang.



Abweichung in Haus XX

Hier wurde die untere Platine mit den Stromwandlern bestückt. Die jeweiligen zu messenden Leiter wurden von der Sicherung gelöst, verlängert, durch das Messmodul und zurück zur Sicherung geführt. Diese Bauweise erschwert die Installation beträchtlich!

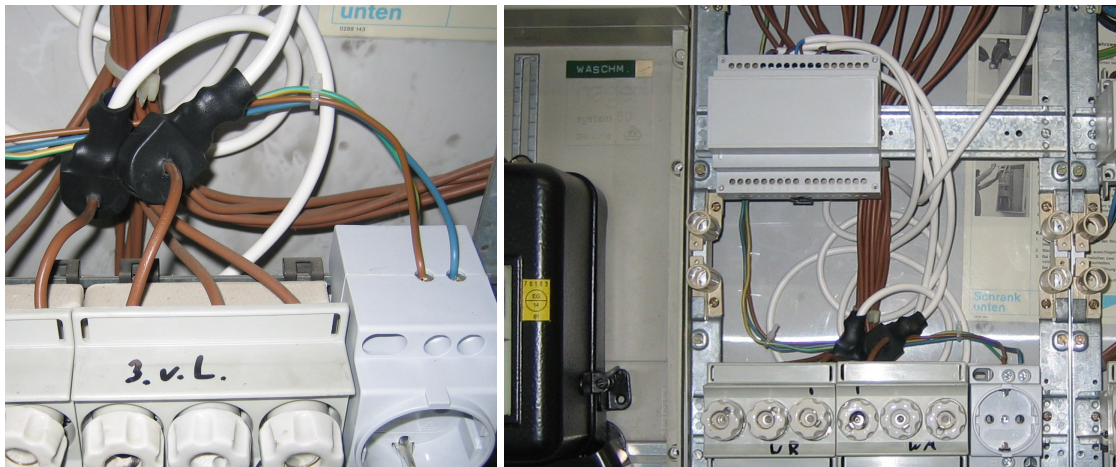


OpenWrt Router

Edimax BR-6104KP aus Bestand. In die Sicherungskästen wurde jeweils hinter eine willkürliche Waschmaschinensicherung für den Router eine Hutschienen-Steckdose montiert. Neben die Sicherungskästen wurden Netzwerkdosen gesetzt, in Haus YY steht das noch aus – dort ist ein anderes OpenWrt-Routermodell Wlan Client.

Einbau

Der Einbau ist bei unsachgemäßer Handhabung **lebensgefährlich** und nur bei entsprechendem Kenntnisstand durchzuführen. Korrekt montiert ist das Messmodul von den zu messenden Leitungen galvanisch getrennt. Siehe Installationsplan.



Software

Firmware

Der AVR-Mikrocontroller wird über mit einem In-System-Programmer (ISP) programmiert. Da Netzstrom gemessen werden soll, wird mit einem Vielfachen von 50Hz gesampelt, hier 400Hz.

Zur Strommessung wird allgemein das Mittlere Quadrat (Root Mean Square, RMS) gebildet. Die Aufgabe des Wurzelziehens und Teilens wird an den Webserver abgetreten, die Software gibt mit 1Hz die Quadratsummen jedes ADC von ADC0..6 aus.

Um mit 32-bit Datentypen arbeiten zu können, und dabei die ltoa-Funktion nutzen zu können:

Der ADC liefert 10bit Werte, die quadriert werden \rightarrow max. 20bit. ltoa() erwartet signed-Werte \rightarrow -1bit. Bei einem ausgegebenen Wert pro Sekunde bleibt eine Samplingfrequenz pro ADC von maximal 2^{11} Hz. Ggf. würde die Prozessorgeschwindigkeit dies weiter limitieren, da das Quadrieren durch $x*x$ in C programmiert etwa 100 (Core-)Cycles dauert.

Unabhängig davon läuft die ADC-Wandlung: Eine Wandlung dauert etwa 13 ADC-Cycles, die Frequenz des ADC sollte im Bereich von 50..200KHz liegen.

Durch die simple Schaltung werden nur positive Halbwellen gemessen, was aber keine Auswirkung haben sollte (außer halbierte Messwerte). Die Schaltung ist so ausgelegt, dass bei Nutzung der internen Referenzschaltung von 2.56V die Signale erst ab 11A abgeschnitten werden – was bei reinen Sinussignalen durch eine entsprechende Eichkurve auf der Auswerteseite korrigierbar wäre.

Ein Sinussignal ist bestimmt wenn man mit minimal $>2f$ sampled. Hier werden nur positive Halbwellen gemessen $\rightarrow >4f$ (? Theorie/Richtigkeit unklar). RC-Filter mit 20kOhm/100nF hat eine Grenzfrequenz von 80Hz \rightarrow bei Störungen ist $f >80$ Hz und nicht gleich 50Hz.

OpenWrt

Die Installation von OpenWrt ist unter openwrt.org hinreichend beschrieben. Unter `/etc/rc.d/` lassen sich eigene Startskripte hinzufügen. Bei den Edimax im sind die Start-/Configskripte aufgrund des Speichermangels fest einkompiliert. (Dies war nur nötig weil weil WLAN-Treiber und Software mit einkompiliert wurden da der Betrieb eines WLAN-USB-Sticks möglich sein sollte. Diese Lösung erwies sich aber als instabil.) Die Router sind leicht unterfordert – bieten aber immerhin einen Switch und Strom via USB für das Messmodul. Außerdem ist ihr Netzwerkstack auch erprobt in großen Netzen mit vielen Broad-/Multicast-Paketen – zu AVR-Ethernet-Lösungen waren keine Erfahrungsberichte zu finden und der μ C durch das Quadrieren und Samplen von 7 ADCs (in XX stehen 7 Maschinen) potentiell bereits zu stark ausgelastet.

Nicht notwendig, aber im diesem Wohnheim sind in den Edimax fest einkompiliert ein root-Passwort und ein ssh-allowed-key für sebastian.

Webserver

Daten holend:

Die `/etc/crontab` führt unter dem Namen Sebastian einmal täglich das Skript `/home/sebastian/wama/wama-datenerhebung.sh` aus. Aus unbekanntem Gründen läuft das Skript nur, wenn es durch `nohup` aufgerufen wird. Das Skript prüft ob es schon läuft und beendet sich ggf.. Um es neu zu starten muss man vorher den entsprechenden Prozess killen.

Die Daten werden durch netcat abgeholt. Die IPs sind `x.x.x.202`, `.203`, `.204`, `205`. Da dies u. U. instabil sein kann, wird regelmäßig geprüft ob die Datenlogs noch wachsen.

Der Ordner `/home/sebastian/wama`:

<code>daten</code>	alle gesammelten Daten, die Waschräume nach IPs, eine File/Tag
<code>wama-datenerhebung.sh</code>	holt die Daten ab, mit <code>nohup</code> aufrufen!
<code>nohup.out</code>	Ausgabe von <code>wama-datenerhebung.sh</code> , da durch <code>nohup</code> aufzurufen

Daten auswertend:

Wird die entsprechende Seite aufgerufen (Dokuwiki), stößt `<phpinc=wama.php>` das Auswerten/Plotten an und bindet den Plot für XX ein.

`<phpinc=wama-203.php>`, `<phpinc=wama-204.php>` und `<phpinc=wama-205.php>` binden die Bilder ein. Das Bildereinbinden passiert durch `
`.

Die php-Dateien liegen unter `/var/www/WIKINAME/phpincludes/`. Die Bilder selbst liegen unter `/var/www/WIKINAME/`. Die Dateirechte müssen die Manipulation durch den Webserver, also durch alle, erlauben (`rw-rw-rw`).

`wama.php` ruft `/home/sebastian/wama/plot.sh` auf, welches zwischenzeitlich `plot.plt` aufruft.

Der Ordner /home/sebastian/wama (fortgesetzt):

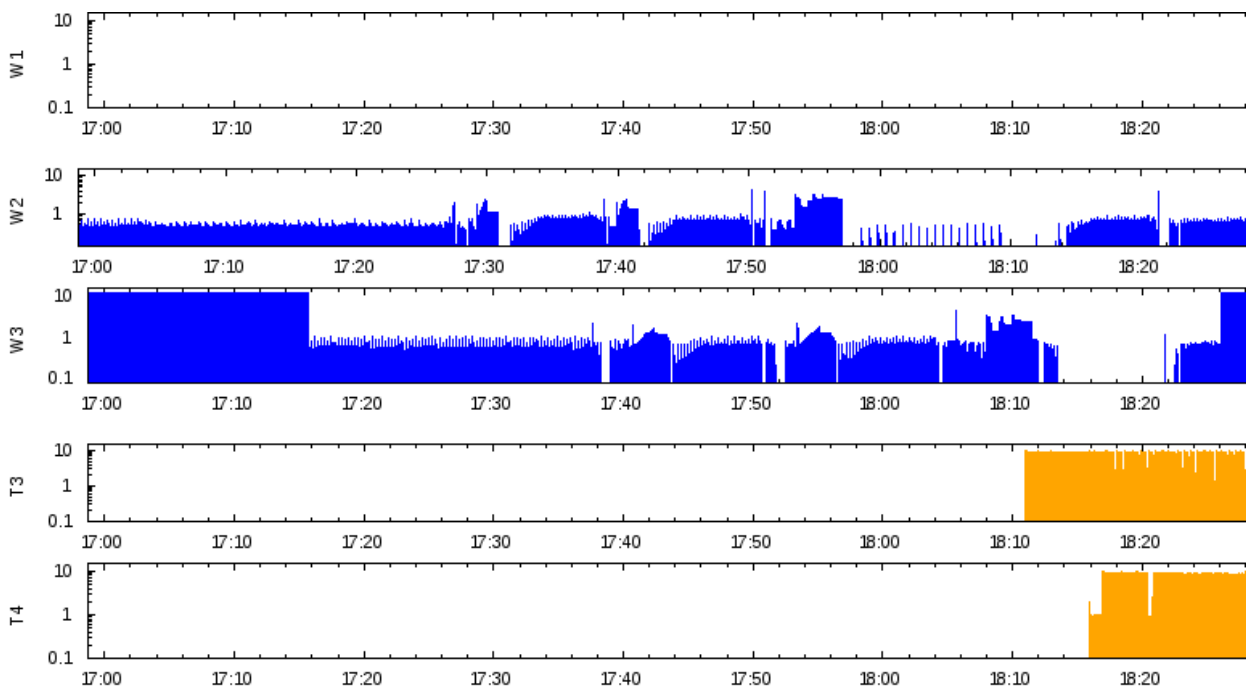
plot.sh	Bereitet Daten auf und stößt das Plotten durch Gnuplot an und kopiert die neuen Plots nach /var/www/WIKINAME
letzterplot	dient als Datumsstempel für die letzte Ausführung - plot.sh beendet sich falls es innerhalb der letzten 5sec bereits lief
filter.awk	aufgerufen von plot.sh, filtert einfache Störungs-Peaks
plot_IP.png	Gnuplot-Ausgabedateien
plt_daten_IP	aus ./daten werden von plot.sh die letzten 5400 Werte in diese Dateien kopiert (die letzten 1,5 Stunden)
plot.plt	Gnuplot-Skript, die Skalierungsfunktionen wurden empirisch bestimmt

Leider sind die IPs der Waschräume in fast allen Skripten eingetragen, genauso wie in vielen Dateinamen.

Die Mindestrechte für alle von plot.sh geänderten Dateien: ****rw-** sein; für die php-Dateien ****r--**; für plot.sh ****r-x**, da der apache mit dem user www-data läuft.

Ergebnis und Ausblick

Letztendlich hat meine eine Seite auf seiner Homepage die Grafiken wie diese zeigt:



Die Strommessung ist erwartungsgemäß ungenau, aber für jeden ist erkennbar welche Maschine beschäftigt ist. Wer ungefähr weiß was eine Waschmaschine so macht liest noch mehr heraus.

Eine weitergehende Auswertung oder Smartphone-Apps die aufmerksam machen wann eine bestimmte Maschine fertig ist wären sicher schön.

Die Umsetzung in ein reines ist an / ist aus Signal ist nicht ganz einfach, da es gelegentlich Störungen oder mitten im Waschgang Phasen gibt, in denen die Maschine fast keinen Strom zieht. Auch an eine Abrechnungskontrolle könnte man denken, sobald man einzelne Waschgänge zuverlässig trennen kann.

Anmerkungen

Die Eichung ist sehr dürftig an einem einzelnen Testaufbau mit wenigen Verbrauchern (Fön, Glühlampe, Wasserkocher, ...) vorgenommen worden. Die Messwerte streuen wenn man die verschiedenen ADCs vergleicht. Die Stromwandler u./o. ADCs beeinflussen sich in dem Aufbau zeitweise gegenseitig. Praktisch nicht schlimm, in der Darstellung werden nur Werte größer $\sim 100 \mu\text{mA}$ geplottet.

Würde man SMD-Löten oder nur bis zu 6 Maschinen an einem Messmodul haben wollen würde auch ein Mega8 reichen.

Auf den Edimaxen läuft OpenWrt Backfire 10.03.1. Sollten die einmal kaputtgehen ist ein Modellwechsel empfohlen, da die Edimax 6104KP gelegentlich beim booten hängen bleiben (LEDs bleiben aus → Strom ab- und anstecken).

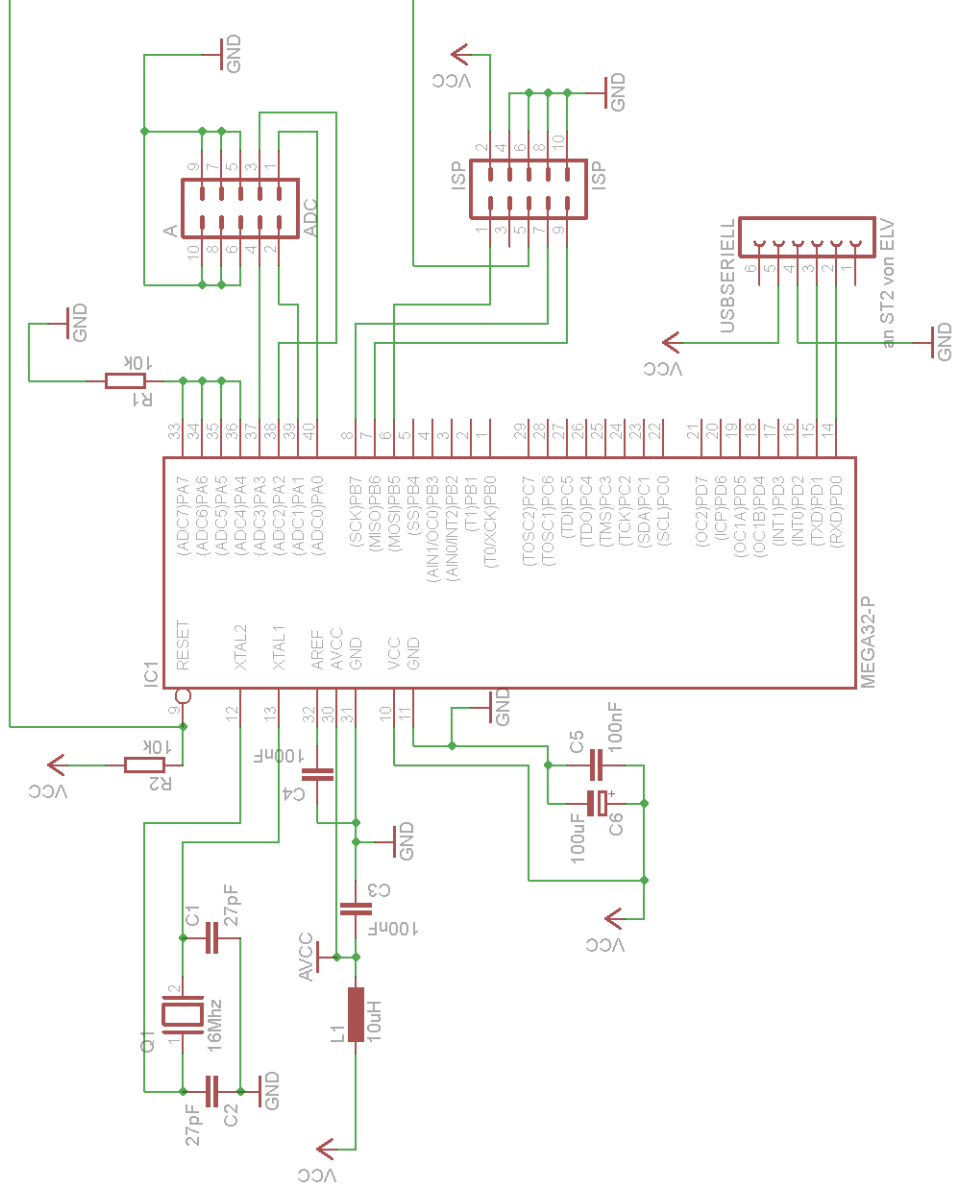
Die Hutschienengehäuse mussten noch zurechtgedremelt werden, damit sie auf die Hutschienen passen.

Die Edimaxe erhalten ihre IP per dhcp und sind in der hostlist/dhcpd.conf des Wohnheimrouters eingetragen.

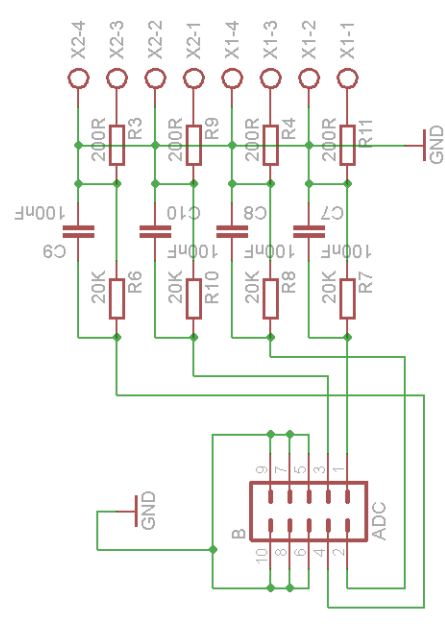
Anhang

- Teileliste für das Messmodul
- Quellcode für den AVR / Firmware des Messmoduls (für AVR-Studio)
- Startskript der Edimaxe
- Webserver-Skripte
- Eichdaten und -Skripte für Gnuplot

Platine oben



Platine unten



Messmodul Waschmaschinen Online

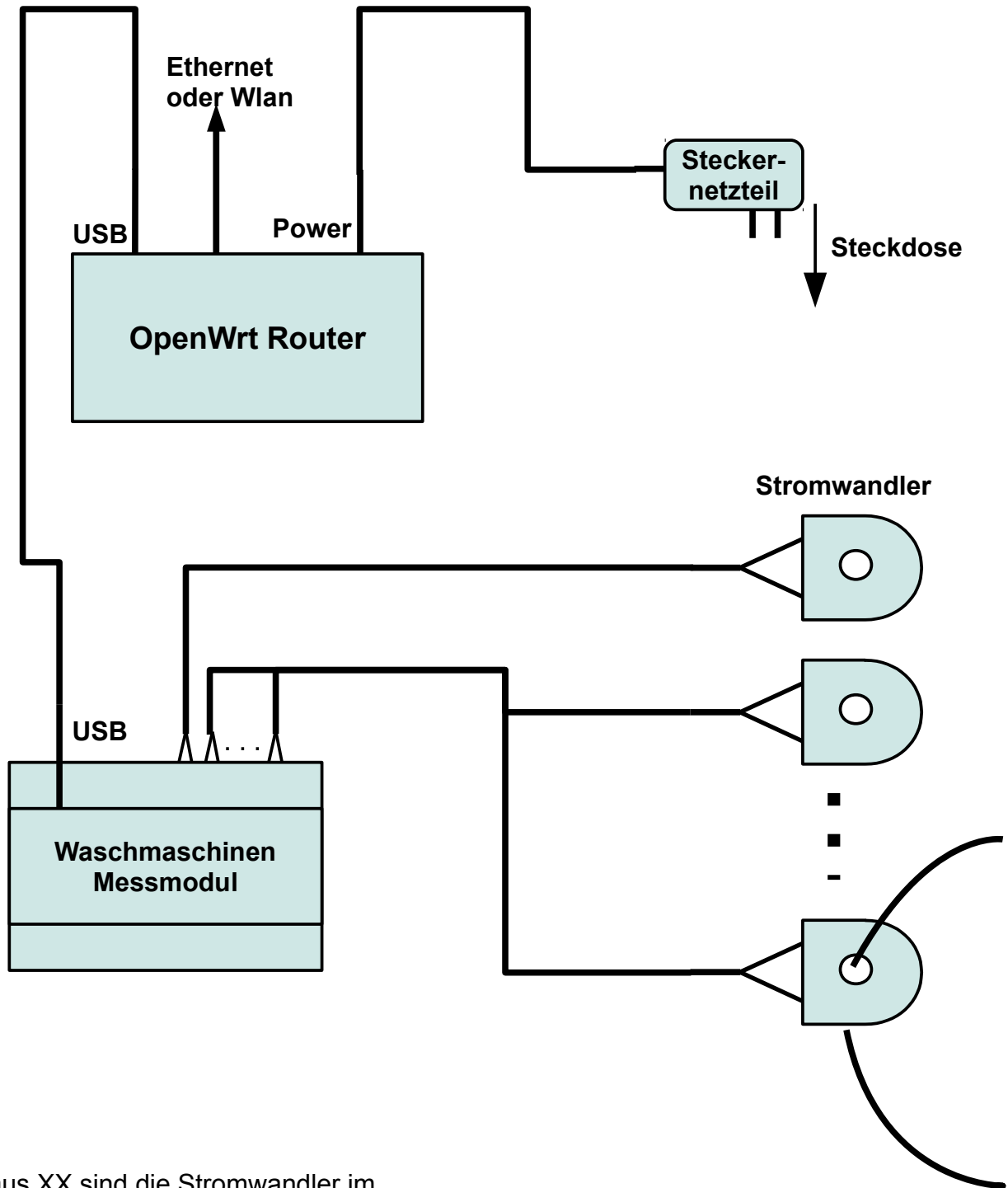
TITLE:

Document Number:

REV:

Date: nicht gespeichert!

Sheet: 1/1



In Haus XX sind die Stromwandler im Messmodul integriert.
 Die Phasen der Maschinen werden entsprechend kontaktfrei und isoliert durch das Modul hindurchgeführt.

Die maximale Anzahl anschließbarer Stromwandler ist bereits ausgereizt.
 Änderung nur durch Lötarbeiten.

Phase der Waschmaschine
 - diejenige die die Elektronik und den Motor der Maschine versorgt, wird durch den Stromwandler geführt

Teileliste Messmodul

Bezeichnung	Preis einzel	Anz.	Artikelnummer	Beschreibung	Link
Steckdose					
	6,35	6,35	1 EL SKSD H	Steckdose Hutschiene	http://www.reic
Messmodul					
	5,95	5,95	1 68-09 18 59		http://www.elv
	1,00	1,00	1 AK 673-A	USB-A/USB-Mini-B-5Pol Kabel, 1,5m	http://www.reic
	6,65	6,65	1 HUT 6-C	Hutschienenleergehäuse	http://www.reic
	2,35	2,35	1 H25PR200	Lochrasterplatine	http://www.reic
USBSERIELL	0,31	0,31	1 BL 1X20G8 2,54	Buchsenleiste	http://www.reic
	0,15	0,15	1 SL 1X40G 2,54	Stiftleiste	http://www.reic
A, B, ISP	0,24	0,08	3 WSL 10G	Wannenstecker 10pol	http://www.reic
C1, C2	0,12	0,06	2 KERKO 27P	Kerko 27pF	http://www.reic
Q1	0,17	0,17	1 16,0000-HC49U-S	Quarz 16Mhz	http://www.reic
C3-5,C7-10	0,42	0,06	7 KERKO 100N	Kerko 100nF	http://www.reic
L1	0,17	0,17	1 SMCC 10µ	Induktivitaet	http://www.reic
IC1	3,90	3,90	1 ATMEGA 32-16 DIP	ATmega32 DIL	http://www.reic
	0,10	0,10	1 GS 40	IC-Sockel 40-pol	http://www.reic
C6	0,04	0,04	1 RAD 105 100/35	Elko 100uF	http://www.reic
R1, R2	0,16	0,08	2 METALL 10,0K	Widerstand 1% 10K	http://www.reic
R3,4,9,11	0,32	0,08	4 METALL 200	Widerstand 1% 200R	http://www.reic
R6,7,8,10	0,32	0,08	4 METALL 20,0K	Widerstand 1% 20K	http://www.reic
	0,16	0,08	2 PFL 10	Pfostenbuchse 10pol	http://www.reic
	3,00	3,00	1 AWG 28-10F 3M	Flachbandkabel 10pol, nur 10cm benötigt	http://www.reic
X1, X2	0,76	0,38	2 AKL 101-04	Anschlussklemme	http://www.reic
Stromwandler					
	13,96	3,49	4 20863	Stromwandler AC1020	http://csd-elec
	2,55	2,55	1 H03VV F275W 5M	Schlauchleitung 2x0,75mm2 5m	http://www.reic
	0,38	0,38	1 KAB 100-2,5	Kleine Kabelbinder, 8 benoetigt	http://www.reic
	0,48	0,48	1 AEH 0,75-100	Adernendhuelsen 0,75mm2, 8 benoetigt	http://www.reic
	2,90	1,45	2 SDH 25-4	Schrumpfschlauch 25,4mm, 4x6cm benoetigt	http://www.reic
	52,91			Für ein Messmodul mit vier Stromwandlern, zzgl. Versand und OpenWrt-Router	

Listings

Messmodul C-Code für die AVR Firmware

```
// Programm fuer Waschmaschinen online Projekt

// Peter Fleury's Uart Lib ist nicht enthalten aber schnell ergooglet.

// Soll einmal/Sekunde fuer jeden ADC0..6 die Summe der quadrierten Werte ausgeben

// Eine eichende Normierung der Werte erfolgt auf Auswerteseite, also nicht auf dem
uC
// Aus diesen Werten kann man in der Auswertung Root-Mean-Square (RMS) errechnen

// Als uC: ein ATmega32, 5V, 16Mhz
// Ausgabe ueber UART
// ADC mit interner V_Ref 2.56V

// Definitionen und Einstellungen
#define UART_BAUD_RATE      9600
#define UART_TX_BUFFER_SIZE 128
#define F_CPU 16000000
#define F_SAMPLE (50*16*7) // Signalfrequenz * Samples pro Welle * Anzahl der
ADC

// Includes
#include <stdlib.h> // Zeug wie z.B. ltoa()
#include <avr/io.h> // Definitionen für das spezifische Device
#include <avr/interrupt.h> // Interrupt Krams
#include <avr/pgmspace.h> // Zugriffe auf den Programmspeicher
#include <stdint.h> // Definitionen der Integertypen fixer Länge
#include <avr/wdt.h> // Watchdog
// #include <avr/sleep.h> // Definitionen für Powersaving sleep modes
// #include <util/delay.h> // Stupide Delay Funktionen
#include "uart.h" // Peter Fleury's Uart Lib

// Funktionsdeklarationen
void adc_init(void);
void timer1_init(void);
ISR(TIMER1_COMPA_vect);

// Globale Variablen
volatile uint16_t   adcmesswert = 0; // volatile da quasi heimlich von ISR
geandert
volatile uint8_t   adc_ctr      = 0; // schaltet die ADC Kanale durch

// der main loop muss schneller sein als F_SAMPLE!
int main(void) {
    uint32_t   ergebnis[7]   = {0,0,0,0,0,0,0};
    int32_t   zwischensumme[7]= {0,0,0,0,0,0,0};
    uint16_t   zeit           = 0;
    uint8_t   temp_ctr        = 0;
    uint16_t   zeit2          = 0;
    char       buffer[12];
    uint16_t   temp_adcmesswert= 0;

    // Watchdog an machen mit laengster Zeit (~2.1s)
    wdt_enable(WDTO_2S);
    wdt_reset();
}
```

```

// Initialisierung des UARTs nach Peter Fleury
uart_init( UART_BAUD_SELECT(UART_BAUD_RATE,F_CPU) );

// Initialisierung des ADC und des Timers
adc_init();
timer1_init();

// Interrupts ermöglichen fuer UART, ADC, Timer usw.
sei();

// main loop
while(1){
    if(adc_ctr != temp_ctr){
        temp_ctr = adc_ctr; // Immer wenn adc_ctr sich aendert
        temp_adcmesswert = adcmesswert;
        zeit++; // Zeit zaehlen bis 1 sec
        // Aktuellen Messwert zum Zwischenergebnis usw
        zwischensumme[(temp_ctr==0 ? 6 : temp_ctr-1)] += (int32_t)
temp_adcmesswert*temp_adcmesswert;
        // Ausgabe nach je einer Sekunde
        if(zeit==(uint16_t) (F_SAMPLE-1+0.5)){
            zeit=0; // Zeit zuruecksetzen
            zeit2++; // Zeit fuer Ausgabe laeuft weiter

            for(uint8_t i=0; i<7; i++){
                // Zwischensumme ins Ergebnis kopieren
                ergebnis[i]=zwischensumme[i];
                // Zwischensumme zuruecksetzen
                zwischensumme[i]=0;
            }
            uart_puts(ltoa(zeit2, buffer, 10));
            // Gesamtzeit ausgegeben
            for(uint8_t i=0; i<7; i++){
                // Ergebnis ausgegeben
                uart_putc(' ');
                uart_puts(ltoa(ergebnis[i], buffer, 10));
            }
            uart_putc('\n');
        }
        wdt_reset();
    }
    return 0;
}

void adc_init(void){
    uint16_t wegwerfergebnis;

    // interne 2,56V als Referenz
    // Die Bits ADLAR und MUXx bleiben Null --> kein left adjust
    // ADC0 Single Ended Input
    ADMUX = (1<<REFS0 | 1<<REFS1);

    // ADC einschalten, Interrupts bleiben aus, Prescaler auf 128 --> 125KHz
    // mit Prescaler 64 --> 250KHz wuerde man die obere Grenze von 200KHz
    // ueberschreiten (Datenblatt), Ergebnis waere ungenauer
    ADCSRA = (1<<ADEN | 1<<ADPS2 | 1<<ADPS1 | 1<<ADPS0);

    // Erste Wandlung wegwerfen weil die erste Wandlung doppelt lang braucht,
    // da intern die allererste Wandlung weggeworfen wird
    ADCSRA |= (1<<ADSC); // Wandlung starten
    while (ADCSRA & (1<<ADSC)){ // warten bis Wandlung fertig
        wegwerfergebnis = ADCW; // Auslesen und ADCW damit wieder freigeben
    }
}

```

```

void timer1_init(void){
    TCCR1A = 0; //Normaler Modus, keine komischen Pins connected

    // Waveform 4 = CTC, TOP=OCR1A, TOV1 Flag nur bei MAX=0xFFFF
    // Deshalb laesst sich der ADC darueber nicht triggern
    // CTC ist aber einfacher exakt einzustellen
    // Timer wird ADC starten muessen
    // Alternative waere eine regelmaessige Anpassung des compare-B gewesen
    TCCR1B = (1<<WGM12 | 1<<CS10); // clk/1,
    OCR1A = (uint16_t) (F_CPU/F_SAMPLE-1+0.5); // compare-match, +0,5 fuer
korrekte Rundung
    TIMSK |= (1<<OCIE1A); // Output Compare A Interrupt
Enable
}

// Timer Interrupt
ISR(TIMER1_COMPA_vect){
    uint8_t tmp_adc_ctr; // fuer den Interrupt ctr in eine nicht-
volatile Variable kopieren
    tmp_adc_ctr = adc_ctr;
    // dass der erste Wert Schrott ist ingorier ich / mittelt sich weg
    adcmesswert = ADCW; // letzten ADC Wert auslesen
    tmp_adc_ctr++;
    if(tmp_adc_ctr==7) tmp_adc_ctr=0; // Wir nutzen 7 ADC Kanale
    ADMUX = (ADMUX & 0b11111000) | tmp_adc_ctr; // ADC Eingang einen weiter
    ADCSRA |= (1<<ADSC); // naechste Wandlung starten
    adc_ctr = tmp_adc_ctr;
}

```

Startskript des OpenWrt-Routers „S99xConfig“

```
#!/bin/sh

# OpenWRT start-config-script, abzulegen unter /etc/rc.d/

# Init LED-states, LED act like usual connection indicator
echo "port_state"> /sys/class/leds/wan_speed/trigger
echo "port_state"> /sys/class/leds/wan_lnkact/trigger
echo "speed"> /sys/class/leds/wan_speed/port_state
echo "link_act"> /sys/class/leds/wan_lnkact/port_state
echo "port_state"> /sys/class/leds/lan1_speed/trigger
echo "port_state"> /sys/class/leds/lan1_lnkact/trigger
echo "speed"> /sys/class/leds/lan1_speed/port_state
echo "link_act"> /sys/class/leds/lan1_lnkact/port_state
echo "port_state"> /sys/class/leds/lan2_speed/trigger
echo "port_state"> /sys/class/leds/lan2_lnkact/trigger
echo "speed"> /sys/class/leds/lan2_speed/port_state
echo "link_act"> /sys/class/leds/lan2_lnkact/port_state
echo "port_state"> /sys/class/leds/lan3_speed/trigger
echo "port_state"> /sys/class/leds/lan3_lnkact/trigger
echo "speed"> /sys/class/leds/lan3_speed/port_state
echo "link_act"> /sys/class/leds/lan3_lnkact/port_state
echo "port_state"> /sys/class/leds/lan4_speed/trigger
echo "port_state"> /sys/class/leds/lan4_lnkact/trigger
echo "speed"> /sys/class/leds/lan4_speed/port_state
echo "link_act"> /sys/class/leds/lan4_lnkact/port_state

# Init Switch
admswconfig eth0 "1234c"
admswconfig eth1 "0c"
admswconfig eth2
admswconfig eth3
admswconfig eth4

# voodoo fehler unterdruecken mit wifi
wireless.@wifi-iface[0].network=wlan

# Lan Config proto=dhcp mit einkompiliert
uci set network.lan.macaddr=00:0E:2E:FD:79:33

# Wan Config - als Diagnoseport
uci set network.wan=interface
uci set network.wan.ifname=eth1
uci set network.wan.proto=static
uci set network.wan.ipaddr=192.168.1.1
uci set network.wan.netmask=255.255.255.0
uci set network.wan.macaddr=00:0E:2E:FD:79:34
uci commit

# Init Lan/Wan - neccessary after admswconfig!
ifdown lan
ifup lan
ifdown wan
ifup wan

sleep 2
```



```

# Paketliste update
while ! ping downloads.openwrt.org -c 1 >/dev/null 2>/tmp/errooor
do
    logger "No connection to downloads.openwrt.org"
    sleep 5s
done
logger "ping to downloads.openwrt.org possible"

# opkg update Package List
while ! opkg update >/dev/null 2>/tmp/errooor
do
    logger "opkg update failed"
    sleep 5s
done
logger "opkg update successful"

# Install NTP client, mtd, USB-Seriell-Treiber, ser2net=Ausgabe der seriellen
Schnittstelle ueber Netzwerk; Ziel: RAM
while ! opkg -d ram install ntpclient mtd kmod-usb-serial-cp210x ser2net >/dev/null
2>/tmp/errooor
do
    logger "tools install failed"
    sleep 5s
done
logger "tools install successful"

# Zeit einstellen
while ! ( /tmp/usr/sbin/ntpclient -s -h ptbtime2.ptb.de >/dev/null 2>/tmp/errooor ||
/tmp/usr/sbin/ntpclient -s -h ptbtime1.ptb.de >/dev/null 2>/tmp/errooor )
do
    logger "ntp client got/set no time yet"
    sleep 2s
done
logger "ntp got/set time successful"

# Zeit regelmaessig updaten
/tmp/usr/sbin/ntpclient -s -i 40000 -h ptbtime1.ptb.de &

# Treiber fuer den usb-seriell-wandler laden
insmod /tmp/lib/modules/2.6.32.27/usbserial.ko
insmod /tmp/lib/modules/2.6.32.27/cp210x.ko

# Serielle Schnittstelle ausgeben auf Port 33333
/tmp/usr/sbin/ser2net -C 33333:raw:10:/dev/ttyUSB0:9600,NONE,1STOPBIT,8DATABITS,-
XONXOFF,-RTSCTS,LOCAL

```

Eichdaten

von mangelhafter Qualität. Werte von Glühbirne, Haarfön und Wasserkocher

#0	0.024
0.0233742201	0.0299
0.0786728865	0.0405
0.251926103	0.115
0.4418669806	0.1775
3.1548812489	1.01
5.79094906	1.91
10.0248433981	3.07
10.7527808507	3.285
13.4287143984	4.06
15.4402635066	4.9
21.1253493772	6.34
30.4260126329	9.13
33.8704132587	10.13
35.39754755	10.93
39.7051012113	12.35

Eichskript für Gnuplot

reset

file = "eichung-daten.txt"

g0=g1=g2=g3=1

$g(x) = g0 + g1*x + g2*x**2 + g3*x**3$

fit [x=0:40] g(x) file using 1:2 via g0,g1,g2,g3

plot file using (\$1):(\$2) with linespoints, g(x)

PHP-Includes für die Wiki

wama.php:

```
<meta http-equiv="refresh" content="30" >
<?php
system ("/home/sebastian/wama/plot.sh");
?>
<br>
<br>
<br>
```

wama-YY.php:

```
<br>
```

AWK-SKRIPT „filter.awk“

```
# Funktion verwirft Werte, wenn in der Zeile drueber und gleichzeitig in der Zeile drunter
# der Wert Null war. Filter dadurch Spikes.
```

```
function filter(x) {
    old3[x]=old2[x]
    old2[x]=old1[x]
    old1[x]=$x
    if( old3[x]==0 && old2[x]!=0 && old1[x]==0 ) {
        old2[x]=0
    }
    printf old2[x] " "
}
}
```

```
# Hier steigt awk ein
# Es wird fuer die Anzahl der Spalten der Filter aufgerufen
# und jeweils ein Zeilenumbruch angehaengt
BEGIN { } {
```

```
    for(i=1;i<=NF;i++) {
        filter(i)
    }
    print " "
```

```
} END { }
```

Gnuplot-Skript „plot.plt“

```
#!/usr/bin/gnuplot

reset

set term png font "Arial,8" size 800,600

# Pseudostromstaerke
f(a) = sqrt(a)/400.0

# Korrektur nach der "eichung"
g0      = 0.0173401
g1      = 0.333759
g2      = -0.00282192
g3      = 5.6405e-005
g(x)    = g0+g1*x +g2*x**2 + g3*x**3
h(x)    = x<0.02 ? g(0.02)/0.02*x : g(x)

set ylabel 'Strom [A]'
set yrange[0.05:15]
set logscale y
set ytics nomirror
unset key

set timefmt "%s"
set xdata time
set format x "%H:%M"

zeitzone=int(system('date +%:::z'))
max=int(system("date +%s")) + zeitzone*3600
min=max-5400+1
set xrange[min:max]

set output "/home/sebastian/wama/plot_202.png"
file = "/home/sebastian/wama/plr_daten_202"

set sample 5400
set multiplot
set size 1,0.1/0.7

set ylabel 'W2'
set origin 0,0.6/0.7
set yrange[0.1:15]
plot file using ($1+zeitzone*3600):(h(f($2))) with filledcurves x1 lc rgb 'blue'
title 'W2'

set ylabel 'W3'
set origin 0,0.5/0.7
plot file using ($1+zeitzone*3600):(h(f($3))) with filledcurves x1 lc rgb 'blue'
title 'W3'

set ylabel 'W4'
set origin 0,0.4/0.7
set yrange[0.05:15]
plot file using ($1+zeitzone*3600):(h(f($4))) with filledcurves x1 lc rgb 'blue'
title 'W4'

set ylabel 'W5'
set origin 0,0.3/0.7
plot file using ($1+zeitzone*3600):(h(f($5))) with filledcurves x1 lc rgb 'blue'
title 'W5'
```

```

set ylabel 'T7'
set origin 0,0.2/0.7
plot file using ($1+zeitzone*3600):(h(f($6))) with filledcurves x1 lc rgb 'orange'
title 'T7'

set ylabel 'T8'
set origin 0,0.1/0.7
plot file using ($1+zeitzone*3600):(h(f($7))) with filledcurves x1 lc rgb 'orange'
title 'T8'

set ylabel 'T9'
set origin 0,0.0/0.7
plot file using ($1+zeitzone*3600):(h(f($8))) with filledcurves x1 lc rgb 'orange'
title 'T9'

unset multiplot

## Haus xx

set output "/home/sebastian/wama/plot_203.png"
file = "/home/sebastian/wama/pld_daten_203"

set term png font "Arial,8" size 800,428
set multiplot
set size 1,0.1/0.5

set ylabel 'W1'
set origin 0,0.4/0.5
set yrange[0.1:15]
plot file using ($1+zeitzone*3600):(h(f($2))) with filledcurves x1 lc rgb 'blue'
title 'W1'

set ylabel 'W2'
set origin 0,0.3/0.5
set yrange[0.15:15]
plot file using ($1+zeitzone*3600):(h(f($3))) with filledcurves x1 lc rgb 'blue'
title 'W2'

set ylabel 'W3'
set origin 0,0.2/0.5
set yrange[0.08:15]
plot file using ($1+zeitzone*3600):(h(f($4))) with filledcurves x1 lc rgb 'blue'
title 'W3'

set ylabel 'T3'
set origin 0,0.1/0.5
set yrange[0.10:15]
plot file using ($1+zeitzone*3600):(h(f($5))) with filledcurves x1 lc rgb 'orange'
title 'T3'

set ylabel 'T4'
set origin 0,0.0/0.5
set yrange[0.10:15]
plot file using ($1+zeitzone*3600):(h(f($6))) with filledcurves x1 lc rgb 'orange'
title 'T4'

unset multiplot

## Haus xx

set output "/home/sebastian/wama/plot_204.png"
file = "/home/sebastian/wama/pld_daten_204"

```

```

set term png font "Arial,8" size 800,342
set multiplot
set size 1,0.1/0.4

set ylabel 'W links'
set origin 0,0.3/0.4
set yrange[0.05:15]
plot file using ($1+zeitzone*3600):(h(f($2))) with filledcurves x1 lc rgb 'blue'
title 'W links'

set ylabel 'Trockner'
set origin 0,0.2/0.4
set yrange[0.05:15]
plot file using ($1+zeitzone*3600):(h(f($3))) with filledcurves x1 lc rgb 'orange'
title 'Trockner'

set ylabel 'W mitte'
set origin 0,0.1/0.4
set yrange[0.05:15]
plot file using ($1+zeitzone*3600):(h(f($4))) with filledcurves x1 lc rgb 'blue'
title 'W mitte'

set ylabel 'W rechts'
set origin 0,0.0/0.4
set yrange[0.1:15]
plot file using ($1+zeitzone*3600):(h(f($5))) with filledcurves x1 lc rgb 'blue'
title 'W rechts'

unset multiplot

## Haus xx
set output "/home/sebastian/wama/plot_205.png"
file = "/home/sebastian/wama/pld_daten_205"

set term png font "Arial,8" size 800,342
set multiplot
set size 1,0.1/0.4

set ylabel 'T links'
set origin 0,0.3/0.4
set yrange[0.05:15]
plot file using ($1+zeitzone*3600):(h(f($2))) with filledcurves x1 lc rgb 'orange'
title 'T links'

set ylabel 'T rechts'
set origin 0,0.2/0.4
plot file using ($1+zeitzone*3600):(h(f($3))) with filledcurves x1 lc rgb 'orange'
title 'T rechts'

set ylabel 'W links'
set origin 0,0.1/0.4
set yrange[0.08:15]
plot file using ($1+zeitzone*3600):(h(f($4))) with filledcurves x1 lc rgb 'blue'
title 'W links'

set ylabel 'W rechts'
set origin 0,0.0/0.4
set yrange[0.05:15]
plot file using ($1+zeitzone*3600):(h(f($5))) with filledcurves x1 lc rgb 'blue'
title 'W rechts'

unset multiplot

```


„plot.sh“

```
#!/bin/bash

# Skript zum Plotten
# fuehrt sich nur aus wenn letzter plot mindestens MINTIME Sekunden her ist

PFAD=/home/sebastian/wama
MINTIME=5
WAMAS="202 203 204 205"
WEB=/var/www/WIKINAME

# Beenden wenn der letzte Plot zu jung ist
if (( $(($(date +%s)-$(ls -l --time-style=%s $PFAD/letzterplot | awk '{print $6}')) < $MINTIME )); then exit 3; fi

# Zeitpunkt des letzten Plots markieren
touch $PFAD/letzterplot

# Daten der letzten anderthalb Stunden in die plot_daten_x kopieren
# beruecksichtigt dass jeden Tag eine neue Datei angefangen wird
for i in $WAMAS; do
    VORHANDEN=$(grep -c "" $PFAD/daten/$i.$(date +%Y%m%d))
    if (( $VORHANDEN < 5400 )); then
        FEHLEND=$((5400-$VORHANDEN))
        tail -n $FEHLEND $PFAD/daten/$i.$(date +%Y%m%d --date="-1 day") | awk
-f $PFAD/filter.awk > $PFAD/plt_daten_$i
        tail -n $VORHANDEN $PFAD/daten/$i.$(date +%Y%m%d) | awk -f
$PFAD/filter.awk >> $PFAD/plt_daten_$i
    else
        tail -n 5400 $PFAD/daten/$i.$(date +%Y%m%d) | awk -f $PFAD/filter.awk
> $PFAD/plt_daten_$i
    fi
done

# Gnuplot anstossen - Gnuplot liefert die png Dateien
$PFAD/plot.plt

# Grafiken zur Wiki / in den www ordner kopieren
# Falls seit 10 Minuten keine neuen Daten mehr reingekommen sind wird stattdessen das
Fehler-png kopiert
# wahrscheinlich gibt das zur zeitungstellung kurzzeitig fuer maximal 1 stunde 10
minuten selbstloesende probleme
for i in $WAMAS; do
    cp $PFAD/plot_$i.png $WEB/plot_$i.png

    if test -e $PFAD/daten/$i.$(date +%Y%m%d); then
        if test $(($(date +%s) - $(date --reference=$PFAD/daten/$i.$(date +
%Y%m%d) +%s))) -gt 600; then
            cp $PFAD/fehler-keineaktuellendaten.png $WEB/plot_$i.png
        fi
    fi
done

exit 0
```

„wama-datenerhebung.sh“

```
#!/bin/bash
#
# Dieses Skript ist leider mit nohup aufzurufen, sonst pausiert es im Background
# bloed und keine Ahnung warum
#
# Skript, dass fuer Aufzeichnung der Daten der Waschmaschinen sorgt
# Die Daten werden per nc geholt
#
# In der /etc/crontab wird dieses Skript einmal pro Tag aufgerufen
# Es faengt fuer jeden Tag und Waschraum eine neue Datei an

# Nur eine Instanz erlauben
# Da das skript, die subshell mit ps und grep mitzaehlen greater 3
TESTOBSCHONAN=$(ps aux |grep -c $0)
if [ $TESTOBSCHONAN -gt 3 ] ; then
    echo laeuft schon
    exit 3
else
    echo noch nicht an
fi

# Speicherordner:
ZIEL="/home/sebastian/wama/daten"
# Liste der Waschmaschinen:
WAMAS="202 203 204 205"

# nc / Streamen beenden:
# falls noch haengengeblieben
wama_kill() {
    kill -term $(ps aux | grep "nc x.x.x.$i 33333" | grep -v grep | awk '{print
$2}') ;
}

# Streamen starten:
wama_start() {
    nc x.x.x.$i 33333 | awk '{system("echo -n $(date +%s)"); print " "$2" "$3"
"$4" "$5" "$6" "$7" "$8}' >> $ZIEL/$i.$(date +%Y%m%d) &
}

# Ausgabedatei anlegen wenn nicht vorhanden, Groesse zurueckgeben
wama_dateicheck() {
    if [ ! -e $ZIEL/$i.$(date +%Y%m%d) ]; then
        touch $ZIEL/$i.$(date +%Y%m%d)
    fi
    ret_groesse=$(ls -l $ZIEL/$i.$(date +%Y%m%d) | awk '{ print $5 }')
}

while true; do
    for i in $WAMAS; do
        echo while am anfang #debug
        # Dateigroesse jetzt und 5 sec spaeter
        wama_dateicheck; s1=$ret_groesse
        sleep 5
        wama_dateicheck; s2=$ret_groesse
        # Wenn die Dateigroesse nicht waechst neu starten
        echo s1=$s1 s2=$s2 #debug
        if (( $s2 <= $s1 )); then
            echo neustart #debug
        fi
    done
done
```

```
wama_kill  
wama_start
```

```
fi
```

```
done
```

```
done
```

Inhaltsverzeichnis

Zweck.....	1
Funktionsweise.....	1
Hardware.....	1
Anforderungen.....	1
Software.....	1
Benötigte Software.....	2
Hardware.....	2
Stromwandler.....	2
Messmodul für die Hutschiene mit μ C/ADC-Wandler.....	2
Abweichung in Haus XX.....	3
OpenWrt Router.....	3
Einbau.....	4
Software.....	4
Firmware.....	4
OpenWrt.....	5
Webserver.....	5
Daten holend:.....	5
Der Ordner /home/sebastian/wama:.....	5
Daten auswertend:.....	5
Der Ordner /home/sebastian/wama (fortgesetzt):.....	6
Ergebnis und Ausblick.....	6
Anmerkungen.....	7
Anhang.....	7
Schaltplan des Messmoduls.....	8
Installationsplan.....	9
Teileliste Messmodul.....	10
Listings.....	11
Messmodul C-Code für die AVR Firmware.....	11
Startskript des OpenWrt-Routers „S99xConfig“.....	14
Eichdaten.....	16
Eichskript für Gnuplot.....	16
PHP-Includes für die Wiki.....	17
wama.php:.....	17
wama-YY.php:.....	17
AWK-SKRIPT „filter.awk“.....	17
Gnuplot-Skript „plot.plt“.....	18
„plot.sh“.....	21
„wama-datenerhebung.sh“.....	22
Inhaltsverzeichnis.....	24