

Digitale Messdatenerfassung mit TP-Link WR703N, OneWire und webspaces

2013/11/23 – 2015/01/03

Benötigte Hardware:

- Router TP-Link TL-WR703N (amazon.de 22,80€, ebay 26,10€), Doku: [OpenWRT](#)
- OneWire-USB-Dongle DS9490R# (fuchs-shop.com 18,90€) DS2490 , alternative Adapter mit DS2480 sollten auch funktionieren
- Temperatursensor 1Wire DS18S20 / DS18B20 (ebay 11,69€ für 10 Stück) B erlaubt 9 (94ms 0,5°C) bis 12 bit (750ms 0,0625°C) Temporaufauflösung

Software

Neuestes Build von OpenWrt auf den WR703N flashen.

(z.B. attitude adjustment r33212 mit kernel 3.3.8 von 2012-08-21)

Am LANport ist jetzt 192.168.1.1, dorthin mit telnet connecten.

passwd eingeben und damit das rootpassword ändern.

Neue Verbindung auf 192.168.1.1 über ssh

Ändern der IP in /etc/config/network

Einrichten als Client im WLAN:

LAN-Kabel anschließen

IP ist jetzt 192.168.1.2

Connect per putty als ssh mit root und password

WLAN-Einstellungen unter: /etc/config/wireless [IP von dhcp]

Pakete nachinstallieren: opkg

Einrichten von OWFS:

```
# opkg update
# opkg install owfs
```

Ordner:

/tmp/1wire/ Das owfs mountet alle 1wire-Devices in dieses Verzeichnis

/tmp/owdata/ Hierher werden alle Messdaten geschrieben

Gespeicherte Scripte (auf NVRAM): **/usr/1wire/** **Inhalt weiter unten↓**

startowfs - erstellt Dateiordner /tmp/1wire und startet owfs zu diesem Verzeichnis

tempscript - schreibt Temperaturdaten in /tmp/owdata/\$datum.log

uploadscript - erstellt /tmp/owdata/data.js mit Temperaturdaten und startet upload wput nach hofkoh.saiht.de

cron: Eintrag erfolgt in **/etc/crontabs/root**, die Einträge bedeuten:

cron ruft alle 10 Minuten das Script auf /usr/1wire/tempscript

cron ruft in der ersten Minute jeder Stunde das Script auf /usr/1wire/uploadscript

cron ruft 2 Minuten vor jeder Stunde das /uploadday Script auf, ist upload der heutigen Daten

automatischer Start durch Eintrag: **/etc/init.d/S60cron**

Ab und zu gibt es einen Stromausfall. Dann sind alle Daten in /tmp gelöscht. Auch das Dateisystem ist nicht automatisch gemountet. Daher muss ein Startscript direkt nach dem Stromausfall ausgeführt werden: **/etc/init.d/example**

Beispielconfig auf /etc/config/wireless

```
config wifi-device 'radio0'
    option type 'mac80211'
    option channel '11'
    option macaddr 'ec:17:2f:9d:a0:c0'
    option hwmode '11ng'
    option htmode 'HT20'
    list ht_capab 'SHORT-GI-20'
    list ht_capab 'SHORT-GI-40'
    list ht_capab 'RX-STBC1'
    list ht_capab 'DSSS_CCK-40'

config wifi-iface
    option device 'radio0'
    option network 'wan'
    option ssid 'Hofkoh5e'
    option encryption 'psk2'
    option mode 'sta'
    option key 'Hofkoh5e'
```

Beispiel für die crontab in /etc/crontabs/root

```
*/10 * * * * sh /usr/1wire/tempscript
1 * * * * sh /usr/1wire/uploadscript
58 * * * * sh /usr/1wire/uploadday
```

Aufruf alle 10 Minuten

Aufruf erste Minute jede Stunde

Aufruf 58ste Minute jede Stunde

cron-Daemon starten durch /etc/init.d/S60cron

```
#!/bin/sh
# start crond
/usr/sbin/crond -c /etc/crontabs
```

cron-Daemon starten durch /etc/init.d/example

```
#!/bin/sh /etc/rc.common
START=11
STOP=15
start() {
    echo start
}
stop() {
    echo stop
}
boot() {
    sh /usr/1wire/startowfs
}
```

/usr/1wire/startowfs

```
#!/bin/bash
if [ ! -d /tmp/1wire ]; then
    mkdir /tmp/1wire
fi
owfs -allow_other -u /tmp/1wire
mkdir /tmp/owdata
```

/usr/1wire/tempscript – Temperaturmessung wird im Logfile geschrieben, alle 10 Minuten

```
#!/bin/bash
datum=$(date +%Y%m%d)
logfile=/tmp/owdata/$datum.log
if [ ! -f /tmp/owdata/$datum.log ]
then
    echo -e "YYYYMMDD HH:MM measured temperature" >> $logfile
fi

messwert=$(date +%Y%m%d) " "$(date +%R) " "
messwert=$messwert$(cat /tmp/1wire/28.09390E020000/temperature)

echo -e $messwert >> $logfile
```

Hier fehlt noch eine Abfrage,
wenn ./temperature einen leeren
Wert oder nach dem Booten +85
liefert

/usr/1wire/upload – Messdaten hochladen – Aufruf mit uploadday

```
#!/bin/bash
if [ -f /tmp/owdata/$1 ]
then
    wput -u /tmp/owdata/$1 ftp://ftpuser0815:Mf4jsZ7fb8d)m-T2@hofkoh.de/data/$1
fi
```

/usr/1wire/uploadscript – Messdaten des aktuellen Zeitpunktes werden hochgeladen, zu jeder ersten Minute

```
#!/bin/bash
if [ -f /tmp/owdata/data.js ]
then
    rm /tmp/owdata/data.js
fi

up='uptime_now="`${uptime}`";'
echo $up >> /tmp/owdata/data.js
tempe='temperatur="`${date +%Y%m%d}`" "`${date +%R}`" "`${cat /tmp/1wire/28.09390E020000/temperature}`";'
echo -e $tempe >> /tmp/owdata/data.js
wlan='wlan="`iwconfig wlan0 | grep 'Link Quality`";'
echo -e $wlan >> /tmp/owdata/data.js

wput -u /tmp/owdata/data.js ftp://ftpuser0815:Mf4jsZ7fb8d)m-T2@hofkoh.de/data.js
```

/usr/1wire/uploadday – Messdaten des aktuellen Tages werden hochgeladen, zu jeder 58sten Minute

```
#!/bin/bash
datum=$(date +%Y%m%d)
sh /usr/1wire/upload $datum.log
```

Auswertung auf der Webseite mit JavaScript

ausgewertete Dateien:

- /data/JJJJMMTT.log mit Einträgen JJJJMMTT HH:MM Temperatur alle 10 Minuten
- /data.js mit Einträgen für die Variablen uptime_now, tempe und wlan

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width; initial-scale=1.0;" />

    <link rel="stylesheet" type="text/css" href="style.css">

    <script type="text/javascript" src="data.js"></script> <!-- Daten der letzten Stunde in Variablen -->
    <script type="text/javascript" src="jquery-1.4.1.min.js"></script><!-- um Datei laden zu können -->
    <script type="text/javascript" src="https://www.google.com/jsapi"></script><!-- für graph/charts -->
    <script type="text/javascript" src="tempdata.js"></script><!-- nur vorübergehend!-->
    <script type="text/javascript" src="dynamic.js"></script><!-- Visualisierung, Aktualisierung Daten-->
</head>
<body>
    <p>Datum und Zeit: <span id="tempea">[script loading]</span></p>
    <p>Temperatur:</p>
    <p style="text-align:center;">
        <span style="text-align:center; margin:10px; padding:10px; font-size: x-large; background:lightgreen;">
            <span id="tempeb">10</span> °C
        </span>
    </p>
    <p><span id="uptime">0d 00:00:01</span></p>
    <p><span id="wlan">00/00</span>
    <span style="float:right;"><a href="info.pdf">Information</a></span></p>

    <div id="chart_div" style="width: 100%; height: 500px;"></div>

    <p>heute:<pre style="margin:10px;"><span id="heute">text wird ersetzt</span></pre></p>
    <p>gestern:<pre style="margin:10px;"><span id="gestern">text wird ersetzt</span></pre></p>
    <div id="Fuss"><small><a name="bottom">© 2009-2015</a> Matthias Kreier</small></div>
</body>
</html>
```

Beispiel /dynamic.js

```
google.load("visualization", "1", {packages:["corechart"]});
$(document).ready(function() {
    // Daten aus der data.js einfügen
    document.getElementById("tempea").firstChild.nodeValue = temperatur.slice(0,14);
    document.getElementById("tempeb").firstChild.nodeValue = temperatur.slice(15);
    document.getElementById("uptime").firstChild.nodeValue = uptime_now;
    document.getElementById("wlan").firstChild.nodeValue = wlan;

    // Datenarray data erstellen und mit Daten füllen
    var data = new google.visualization.DataTable();
    data.addColumn('string','Uhrzeit');
    data.addColumn('number', 'heute');
    data.addColumn('number', 'gestern');
    data.addRows(144);

    // heute laden
    var jetzt = new Date();
    var dh = String(jetzt.getFullYear());
    if (parseInt(jetzt.getMonth()+1,10) < 10) {
        dh = dh + "0" + String(jetzt.getMonth()+1)
    } else {
        dh = dh + String(jetzt.getMonth()+1)
    }
    if (parseInt(jetzt.getDate(),10) < 10) {
        dh = dh + "0" + String(jetzt.getDate());
    } else {
        dh = dh + String(jetzt.getDate());
    }
    var datei = "data/" + dh + ".log"; //Dateiname ist YYYYMMDD.log
    $('#heute').load(datei); // pre mit Messdaten von heute füllen
    $.get( datei, function( hdata ) {
        hdata=hdata.replace(/\n/g, " ");
        var harray = hdata.split(" ");
        var messende = (harray.length+1)/3 - 3 //- 6
        for (var i = 0; i < messende ; ++i)
            data.setValue(i, 1, parseFloat(harray[i*3+6])); // Temperatur heute
    });

    // Datum auf gestern setzen
    jetzt.setDate(jetzt.getDate() - 1);
    var dh = String(jetzt.getFullYear());
    if (parseInt(jetzt.getMonth()+1,10) < 10) {
        dh = dh + "0" + String(jetzt.getMonth()+1)
    } else {
        dh = dh + String(jetzt.getMonth()+1)
    }
    if (parseInt(jetzt.getDate(),10) < 10) {
        dh = dh + "0" + String(jetzt.getDate());
    } else {
        dh = dh + String(jetzt.getDate());
    }
    datei = "data/" + dh + ".log"; // jetzt entspricht Dateinamen gestern
    $('#gestern').load(datei); // pre mit Messdaten von gestern füllen
    $.get( datei, function( gdata ) {
        gdata=gdata.replace(/\n/g, " ");
        var garray = gdata.split(" ");
        for (var i = 0; i < 144 ; ++i) {
            data.setValue(i, 0, String(garray[i*3+5])); // Uhrzeit
            data.setValue(i, 2, parseFloat(garray[i*3+6])); // Temperatur gestern
        }
        var options = {          title: 'Temperaturverlauf in Hofkoh' };
        var chart = new google.visualization.AreaChart(document.getElementById('chart_div'));
        chart.draw(data, options);
    });
    return false;
});
```

Nach dem regulärem Ausdruck /\n/g,
" " ist die Unterscheidung leerer
Zeilen bzw. Fehlender
Temperaturwerte entfallen – somit
verschieben sich Datum, Uhrzeit und
Temperatur um einen Wert – liefert
Temperaturen von 18000000 °C

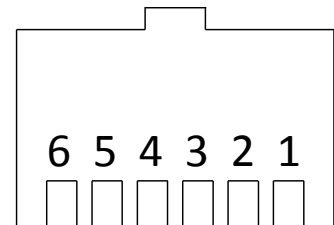
Beispiel [/data.js](#)

```
uptime_now=" 19:01:01 up 382 days, 18:57, load average: 0.00, 0.01, 0.04";  
temperatur="20150103 19:01 4.25";  
wlan=" Link Quality=51/70 Signal level=-59 dBm ";
```

Mechanisches und Elektrisches

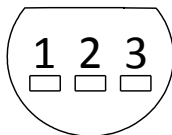
Belegte Pins am DS9490R mit RJ12 Plug (6 Pin)

PIN	Signalname	Beschreibung
1	V _{DD}	5VDC Output (5V Gleichspannung)
2	GND	Power Ground
3	OW	1-Wire Data
4	GND_OW	1-Wire Return (Rückkanal)
5	N. C.	No Connection
6	N. C.	No Connection



Blick auf weiblichen RJ11

Pins am DS18B20



Blick von unten
TO-92

- 1 – GND
- 2 – DQ – Data Input/Output, also power in parasite power mode
- 3 – VDD – optional, must be grounded for operation in parasite power mode