

9.6 Messwert Erfassung und Visualisierung im Internet

9.6.1 Inhalt – was wird behandelt:

Diese Projekte sind meistens nicht einfach realisierbar, weil einige Fachgebiete tangiert werden. Wenn allerdings etwas Einsatz vorhanden ist, können mit sehr einfachen Mitteln, online Daten verarbeitet werden. In den folgenden Fachbereichen werden

Grundkenntnisse benötigt:

Steuer & Regeltechnik, Elektronik, Micro Controller, C++/Pascal, Internet Protokolle, Server side Progaming z.B. PHP, Datenbank Technik, WWW HTML.

An einem Beispiel sollen alle obigen Bereiche und deren Funktion vorgezeigt und erläutert werden.

- Fühler / Sensoren	Messwertaufnehmer
- Messwert Empfangssystem	Hardware μ C
- Messwert Empfangssystem	Software C++/Pascal
- Daten übertragungs Protokoll	Modem/Internet z.B. FTP
- Datenbank auslesen	Darstellung in Text Form
- Datenbank auslesen	Darstellung in Text Grafik
- Datenbank auslesen	Darstellung in Klotz Grafik
- Illustriert an einem laufenden Beispiel	Leistungserfassung lokales Elektrizitätswerk (Verteiler)

Quelle dieses Dokumentes: http://www.clinch.ch/pls_messwerte/pls_messwerte.htm

Auftrag (Ziel):

Abschätzen und zusammenstellen der Infrastruktur so wie der zu erwartenden Arbeitsaufwendungen. Erstellen und Breitstellen von Hardware und Software für eine online Messwert Grafik auf dem Internet.

Gegeben:

Infrastruktur des EW Netzes

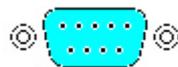
- Mittelspannungsnetz 16kVolt mit verschiedenen Zugängen.
- Aufbau als Ring, teils mehrfach vermascht.
- Trafo Stationen (ca. 16) in der Gemeinde verteilt.
- Zentrale Steuerung im Gebäude des Werkbetriebs
- Steuerung liefert Relais Ausgang, potentialfrei
- Jeder Puls 1kWh

9.6.2 Fühler / Sensor

Da die meisten PC's oder Micro Controller eine serielle RS232C Schnittstelle standardmässig eingebaut haben, empfiehlt es sich diese gleich als Eingang für die Pulszählung zu verwenden. Damit keine zusätzliche Stromversorgung für das Eingangssignal an der RS232 benötigt wird, wurden zwei Steuerleitungen entsprechend gesetzt. Der Dateneingang an der RS232 RxD muss +12 für ein logisches "1" oder -12 für ein logisches "0" haben. Um diese Spannungen bereit zu stellen wurden die Ausgänge (Steuerleitungen) wie folgt gesetzt: DTR (Data Terminal Ready) auf +12V, RTS (Request To Send)

RS232 Pin Belegung DB9-male (am PC)

D	Beschrieb	Name	Pin
I	Data Carrier Detect	DCD	1
I	Receive Data	RxD	2
O	Transmit Data	TxD	3
O	Data Terminal Ready	DTR	4
/	Signal Ground	GND	5



Pin	Name	Beschrieb	D
6	DSR	Data Set Ready	I
7	RTS	Request To Send	O
8	CTS	Clear To Send	I
9	RI	Ring Indicator	O
S	Shield	Chassis Ground	/

Schema des Pulsempfängers für die serielle Schnittstelle:

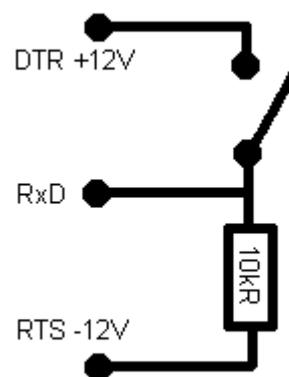
Der 10k Ohm Widerstand wird direkt im Steckergehäuse des DB9 Steckers verlötet, so dass nur noch das zwei adrige Kabel für den Schalt-Kontakt aus dem Stecker heraus kommt.

Berechnung Widerstand:

$$I = U / R = 24V / 10k = 2.4mA \text{ (bei geschaltetem Kontakt)}$$

$$P = U * I = 24V * 2.4mA = 60mWatt \text{ (Wiederst. hat 125mW)}$$

RTS und DTR sind auf 20mA strombegrenzt und Kurzschlussfest.



Die Pulse die von der Leistungserfassung kommen haben eine Zeit von ca. 50mSekunden. Der UART der seriellen Schnittstelle wird das schliessen des Kontaktes als beginn eines Seriell einlaufenden Bytes interpretieren. Als erstes ein Start Bit dann 8 Daten Bits und zum Schluss ein Stop Bit. Nun wird gewartet bis die Leitung wieder auf den Grundzustand zurück geht, da nur eine ansteigende Flanke als Start-Bit erkannt werden kann.

Daraus kann gefolgert werden, dass mit 50mS pro Puls nur maximal 19 Pulse pro Sekunde möglich sind. Also maximal 19 Bytes Pro Sekunde empfangen werden können. Daraus folgt 19Bytes/Sek. x10Bit's pro Byte gleich 190 Bits pro Sekunde muss der Empfänger maximal verarbeiten könne. Also alle Datenraten grösser als 300BPS erfüllen dies.

Ein weiterer enormer Vorteil der Seriellen Schnittstelle mit dieser Betriebsart liegt darin, dass der Kontakt auch automatisch entprellt ist. D.h. ein Schaltkontakt schwingt beim schliessen, das hätte jedoch nur zur Folge, dass das erste Bit eventuell nicht gesetzt ist. Wir prüfen aber nicht die Bit's sondern nur die viele Zeichen sind eingegangen. Ein Puls ist also ein Zeichen.

Noch ein High-Light - die heutigen UART's haben einen FIFO (First In First Out) Buffer von 16 Zeichen eingebaut. Das bedeutet es ist nicht zwingend erforderlich, dass jedes Zeichen sofort abgeholt wird.

Aus den obigen Gründen wurde auf die Interrupt Auslösung der seriellen Schnittstelle verzichtet. Die Abholung der Zeichen wird im 18Hz Takt durchgeführt, also im Timer Interrupt Zyklus des PC's, was maximal $18 \times 16 = 288$ Zeichen (Pulse) pro Sekunde möglich macht.

9.6.3 Messwert Empfangssystem - Hardware PC_mC

Klar die Signale einer Messtation lassen sich nicht einfach über einige Dutzend Kilometer übertragen. Die Signale müssen den Leitungseigenschaften angepasst werden. Weiter werden im Normalfall eine grössere Menge von verschiedenen Betriebsparametern übertragen. Bei solchen Anlagen die auch Fernwirk Anlagen genannt, werden läuft die Übertragung meistens via Telefon Leitung mittels Modems. Als Übertragungsprotokoll wird meistens HDLC verwendet (weit verbreitet und sehr simpel).

Bei unserer Messhardware drängt sich das Internet direkt auf, da am Ende die Visualisierung via Internet abläuft.

Am geeignetsten scheint also ein Mikro Controller oder ein PC. Da Mikro Controller mit Netzwerkanschluss eher teuer und komplex sind ist wohl ein PC vorzuziehen. Von der Leistung sind keine besonderen Anforderungen zu erwarten, alles ab PC-AT ist da wohl ausreichend. Benötigt wird lediglich eine serielle Schnittstelle (COM1), eine Netzwerkkarte Ethernet, ein Diskettenlaufwerk und 4MByte RAM.

Für unser Prototypen System wurde ein altes Systemboard im mikro-AT Format verwendet, ca. 20x20cm mit einem Intel Pentium-I Prozessor mit 166MHz. Zur Stromversorgung wurde ein Industrie Netzteil von Melcher eingesetzt mit drei Spannungen +5V 10A, +12V 2A, -12V 1A. Der Vorteil von Industrie Netzteilen ist, dass keine Ventilatoren benötigt werden, d.h. keine mechanischen Teile sind vorhanden.

Die Pin Belegung der AT Stromversorgung

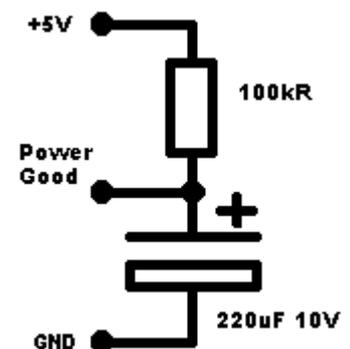
Pin	P8	Farbe
1	Power Good	Orange
2	+5V	Rot
3	+12V	Gelb
4	-12V	Blau
5	GND	Schwarz
6	GND	Schwarz

Pin	P9	Farbe
1	GND	Schwarz
2	GND	Schwarz
3	-5V	Weiss
4	+5V	Rot
5	+5V	Rot
6	+5V	Rot

Die Pin's sind in einer Reihe angeordnet und die 4 Schwarzen Kabel liegen in der Mitte zusammen

Reset / Power Good Signal:

Da die Industriespeisungen kein Power-Good Signal bereitstellen, muss dies noch konventionell erzeugt werden (wird für den Reset benötigt). Das Power Good Signal wird mit folgender Beschaltung erzeugt:



Natürlich muss auch das Diskettenlaufwerk mit Strom versorgt werden, die Farben der Drähte sind da ebenfalls genügend aussage kräftig.

Noch als kleines extra wurde die Power LED (grün) am System-Board angeschlossen. Es kann nichts schaden, wenn man sieht, dass das Gerät unter Strom ist.

Die rote LED die ja gar nicht in Gebrauch ist ohne Festplatte, wurde neu an die Pin's des Lautsprechers angeschlossen. Nun wird die rote LED anstelle des Lautsprechers

betrieben. D.h. mit einem Beep (ASC 7) oder [Ctrl] + [G] wird die LED für ca. ½ Sekunden aufleuchten. Bei beiden LED's ist auf die Polarität zu achten damit's funktioniert.

Nun wird das alles in ein kleines Gehäuse gebaut.

9.6.4 Messwert Empfangssystem - Software

Bei der Software wurde versucht, den einfachsten Weg zu gehen. D.h. die Software liegt nicht in einem Option ROM des PC's sondern wird ab Diskette geladen. Dies ist zwar ein hohes Betriebs Sicherheits Risiko, da die Laufwerke verschmutzen und dann eventuell den Dienst versagen. Aber der Einfachheit zuliebe ...

Aus lizenzrechtlichen Gründen wurde als Betriebssystem FreeDOS gewählt.

FreeDOS ist ein Open Source DOS
<http://www.freedos.org> das fast 100% zu MS-DOS kompatibel ist.



Für die effektive Zählung der Puls wurde ein Speicher residentes Programm (TSR) in Turbo Pascal 4 geschrieben.

Source Code: [PLS_MESS.PAS](#) [PLS_MESS.EXE](#)

Das Programm unterstützt COM1 bis COM4

{ COM1 := \$03F8, COM2 := \$02F8, COM3 := \$03e8, com4 := \$2e8 }

Interrupt der seriellen Schnittstelle wird nicht benutzt!!

Programm Aufruf: PLS_MESS.EXE 1

Der erste Parameter ist die Nummer des zu verwendenden COM Portes in diesem Falle COM1. Nach dem Starten des Programmes geht die Kontrolle sogleich wieder an den Kommando Prozessor zurück, da das Programm sich speicherresident eingeklinkt hat. Es wird nun im RAM auf der Adresse \$0:181 eine LongInteger Variable bei jedem Zeichen auf COM1 inkrementiert. Wobei \$0:181 Das nieder wertigste Byte ist und \$0:184 das höchst wertige. Auf der Adresse \$0:180 wird in jedem Messdurchlauf \$5A geschrieben. Damit kann jeder Zeit geprüft werden ob das TSR Programm läuft. Wird die Adresse \$0:180 mit \$00 überschrieben, so müsste spätestens 55mS später wieder \$5A stehen. Das TSR Programm arbeitet wie ein Maustreiber und wird im System-Timer Interrupt 18 mal pro Sekunde gestartet. Dies kann auch im Debug'er verifiziert werden:

Debug [Enter]

D 0:180 [Enter]

D 0:180 [Enter]

Quit [Enter]

Jetzt können die Summe der Pulse jeder Zeit, alle 15 Minuten oder jede volle Stunde ausgelesen werden.

Source Code: [PLS_READ.PAS](#) [PLS_READ.EXE](#)

Das Programm liest den Wert auf der Adresse \$0:181 und folgende und schreibt diesen in dezimal in die als Parameter angegebene Datei. Danach wird der Zähler wieder auf Null zurück gestellt. Natürlich wird der ausgelesene Wert so oder so auf dem Bildschirm ausgegeben, egal ob ein Parameter angegeben wurde.

Der ausgelesene Wert wird mit FTP an den FTP-Server gesendet. Dies wird per Batch Script erledigt:

Ausschnitt aus dem Batch-Script zum Senden der Messwerte

```
IF "%FTPNAME%" == "" SET FTPNAME=Anonymous           Internet Access Staff
IF "%FTPPASSWD%" == "" SET FTPPASSWD=XXYYZZ
IF "%FTPDIR%" == "" SET FTPDIR=\
::
PLS_READ PLS_READ.TMP                               Puls Zähler auslesen
TYPE PLS_READ.TMP | SETWORD 1 > PLS_READ.BAT
CALL PLS_READ.BAT PLS_READ > NUL                   Setze Variable PLS_READ
```

```

DEL PLS_READ.TMP > NUL
DEL PLS_READ.BAT > NUL
::
echo.>enter
DATE <enter >DATUM.tmp
TYPE DATUM.tmp | SETWORD 5 > DATUM.bat
CALL DATUM.BAT DATUM > NUL
DEL DATUM.BAT > NUL
::
ECHO %PLS_READ%,%ZEIT%>>DATUM%
ECHO %PLS_READ%,%ZEIT% == %DATUM%
::
ECHO OPEN DATA.CLINCH.CH>FTP.TMP
ECHO %FTPNAME%>>FTP.TMP
ECHO %FTPPASSWD%>>FTP.TMP
ECHO CD %FTPPDIR%>>FTP.TMP
ECHO PUT %DATUM% %DATUM%>>FTP.TMP
ECHO CD ..>>FTP.TMP
ECHO CLOSE>>FTP.TMP
ECHO BYE>>FTP.TMP
::
FTP -f ftp.tmp
DEL FTP.TMP>NUL

```

Variable mit dem aktuellen Datum erst.

Messwerte aneinander hängen

FTP Kommando Liste erstellen

FTP ausführen

Neben bei noch das mit der roten LED

Die Codes die die rote front LED zum Blinken bringen:

```
ECHO ^G
```

ASC Code für Beep, Code 7, LED anstatt Lautsprecher angeschlossen ☺

Da das Programm weitgehendst in Batch Script geschrieben ist, kann es einfach an alle Gegebenheiten angepasst werden.

Selbst verständlich muss für jede Netzwerkkarte der individuelle Paket Treiber geladen werden, dieser ist im Verzeichnis \LAN abgelegt und wird vom PLS_INIT initialisiert. In diesem Beispiel ist eine D-LINK DFE-530TX+ Netzwerkkarte im Rechner.

Als TCP/IP Stack für DOS wurde WhatTCP verwendet.

Weitere Infos: <http://www.smashco.com/wattcp.asp>

TCP/IP Treiber unterstützt Statische IP Adressierung als auch Dynamische mit DHCP. Hilfsprogramme wie Ping, Telnet usw... sind auch vorhanden. Interessant ist auch, dass der Netzwerkkarten Paket Treiber speicher Resident installiert wird, der TCP Stack ist jedoch in jedem TCP Programm separat enthalten. Dies hat zur Folge, dass das Gerät nur während einer FTP Übertragung oder während einer laufenden Telnet Session mit Ping von Extern gePing't werden kann.

Diskette: http://www.clinch.ch/clinch/pls_messwerte/pls_mess.img

Image2Disk: http://www.clinch.ch/clinch/pls_messwerte/dd.exe

Automlmg2Disk: http://www.clinch.ch/clinch/pls_messwerte/makedisk.bat

9.6.5 Daten Übertragungs Protokoll - Modem/Internet z.B. FTP

Wie immer bei der Wahl des Übertragungs Protokolls stellt sich die Frage, nehmen wir etwas standardisiertes oder machen wir was eigenes? Ein eigenes Protokoll hätte zur Folge, dass auf dem Server ein entsprechendes Gegenstück installiert werden müsste. Dies ist bei den meisten Providern nicht möglich. Ein Protokoll das bei jedem ISP möglich ist, ist wohl FTP, das dies für den Daten upload fast un verzicht bar ist. Einigen wir uns also auf FTP, ist zwar unverschlüsselt aber es sind ja keine heiklen Daten die übertragen werden und das Datenverzeichnis ist ein separater FTP Zugang und ist kontingentiert. Da wir zu möglichst vielen ISP Kompatibilität sein wollen, wurde auf einen SQL Server verzichtet. Kostet ja auch nur zusätzlich gebühren.

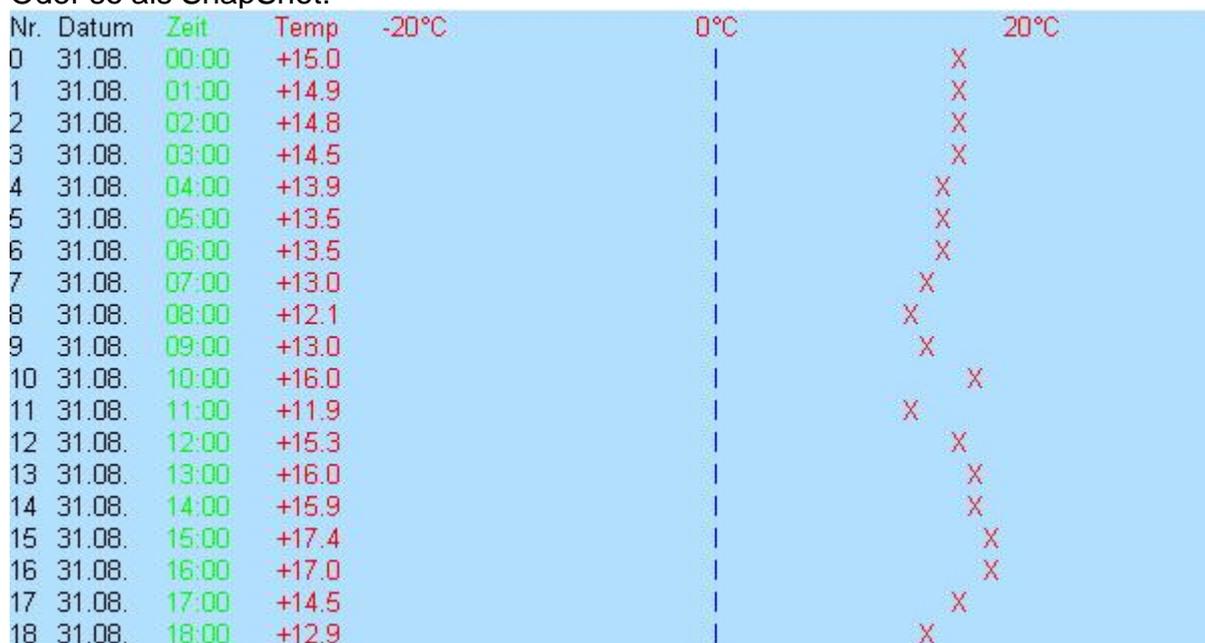
Als ist eine Text Datenbank gefragt. D.h. das Dateisystem dient als Datenbank. Dies ist bis 1000 Dateien je Ordner absolut kein Problem. Wenn wir also für jeden Tag eine Datei

Wie sollen jetzt die Daten im Web dargestellt werden auf einfachste Weise (Text Grafik)?

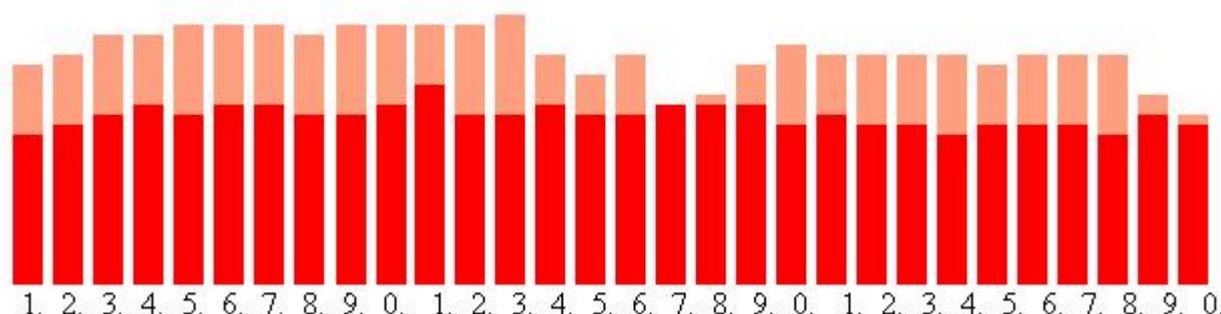
Nr.	Datum	Zeit	Temp	-20°C	-10°C	0°C	10°C	15°C
0	31.08.	00:00	+15.0			I		X
1	31.08.	01:00	+14.9			I		X
2	31.08.	02:00	+14.8			I		X
3	31.08.	03:00	+14.5			I		X
4	31.08.	04:00	+13.9			I		X
5	31.08.	05:00	+13.5			I		X
6	31.08.	06:00	+13.5			I		X
7	31.08.	07:00	+13.0			I		X
8	31.08.	08:00	+12.1			I		X
9	31.08.	09:00	+13.0			I		X
10	31.08.	10:00	+16.0			I		X
11	31.08.	11:00	+11.9			I		X
12	31.08.	12:00	+15.3			I		X
13	31.08.	13:00	+16.0			I		X
14	31.08.	14:00	+15.9			I		X
15	31.08.	15:00	+17.4			I		X
16	31.08.	16:00	+17.0			I		X
17	31.08.	17:00	+14.5			I		X
18	31.08.	18:00	+12.9			I		X

Die obige Darstellung lässt sich in HTML in 5 Spalten realisieren (Nr, Datum, Zeit, Temp und Graph). Damit die "X" die die Werte repräsentieren auf Position sind wurden die Punkte mit einer For-Schleife hochgezählt. Damit die Punkte nicht sichtbar sind wurde der Vordergrund und der Hintergrund mit der gleichen Farbe versehen. Dies ist zwar keine gewaltige Leistung, aber es ist schnell und einfach realisierbar.

Oder so als SnapShot:



Oder als Balken Grafik aus einzelnen kleinen Grafik Gif block's:



On-Line Messdaten finden in den folgenden Bereichen vorzugsmässigen Einsatz.

Die jeweilige Betreiber Firma kann ebenfalls via Internet auf die eigenen Daten zugreifen.

Einsatzgebiete:

WERKE: Gas, Elektro, Wasser, Solar, Generatoren, Treibstoffe, Heizung, Kühlung, Kommunikation, Strahlung, usw...

Fabrikations und Verteilungsanlagen: Abfüllmedien, Chemie, Statistik, Leistung, Aktuelle Pegel

KMU's: Seriennummern Abfrage, Auftragsstatus

Und das waren einige Ideen für die Messwert Erfassung