



## Abfrage & Steuerung von Viessmann Heizungen

Druckbare Version

40 Beiträge dieses Themas auf einer Seite anzeigen

Seite 3 von 8 [Erste](#) [1](#) [2](#) [3](#) [4](#) [5](#) ... [Letzte](#)

harry28

11.05.12, 22:34

Hallo Coyote, hallo Kronos und alle anderen die mitwirken.

Liste der Anhänge anzeigen (Anzahl: 1)

Tolle Skripte, wie überall im Forum - Ihr habt gute Sachen gemacht.

Angeregt durch die Threads und die erfolgreiche Anbindung an IPS habe ich diese Woche auch den USB-Optolink-Adapter für meine Viessmann Heizung bestellt. Angebunden, die Skripte von Coyote und es funktioniert auf Anhieb.

Danke an Euch und die Openv-Gemeinde, vor allem Dirk Eichel für die Platine.

Ich werde nun die Möglichkeiten durchtesten und dranbleiben.

Als Gerätekennung kommt bei mir **2904**.

Viele Grüße aus dem Unterallgäu

Harry

diamorph

24.05.12, 22:53

Falls noch relevant:

Ich schalte jeden Betriebsmodi jedes Heizkreises ohne Probleme:

PHP-Code:

```
<?
$zugriff = GetValue (30593 /*[Heizung\Heizung Output\Zugriff]*/ );
if($IPS_SENDER == "WebFront")

{
    include( "ViessmannDeviceTools.inc.php" );

    switch($IPS_VALUE)
    {

        case 0:
            if ($zugriff == 0)
            {
                ViessmannOpen();
                ViessmannSetData( "BetriebsartM2", chr(0x00) );
                IPS_Sleep(200);
                ViessmannClose();
                SetValueInteger(43443 /*[Heizung\Heizung Output\Betriebsart M2]*/ , 5);
                //SetValue($IPS_VARIABLE, $IPS_VALUE);
            }

            break;

        case 1:
            if ($zugriff == 0)
            {
                ViessmannOpen();
                ViessmannSetData( "BetriebsartM2", chr(0x02) );
                IPS_Sleep(200);
                ViessmannClose();
                SetValueInteger(43443 /*[Heizung\Heizung Output\Betriebsart M2]*/ , 0);
                //SetValue($IPS_VARIABLE, $IPS_VALUE);
            }

            break;

        case 2:
            if ($zugriff == 0)
            {
                ViessmannOpen();
                ViessmannSetData( "BetriebsartM2", chr(0x04) );
                ViessmannClose();
                SetValueInteger(43443 /*[Heizung\Heizung Output\Betriebsart M2]*/ , 3);
                //SetValue($IPS_VARIABLE, $IPS_VALUE);
            }

            break;

    }
}
?>
```

bastianstrauss

31.05.12, 21:27

Probleme mit dem Script

Hallo ,

versuche gerade das Script zum laufen zu bekommen. Leider scheitert es bei mir irgendwo. Bekomme über den Debugger den Hex 5 Wert übermittelt. Doch leider scheint das Script keine

Werte abzufragen. Ich sehe, dass das Script ViessmannDeviceOutputHandlingKW.php aufgerufen wird.

Hier mal ne Frage zu Zeile 20

PHP-Code:

```
if ( $ViessmannDeviceID == "" AND strlen( $Command ) == 0 )
{
    $Command = "R:KW:00F8:2: "; // retrieve Device ID
    SetValueString( VIESSMANN_DEVICE_COMMAND, $Command );
}
```

Die Definition VIESSMANN\_DEVICE\_COMMAND kennt IPS nicht. Ist da vielleicht ein Fehler unterlaufen? Weil so wird über SetValueString der \$Command Inhalt nicht geschrieben. Ist hier vielleicht VIESSMANN\_VARIABLE\_COMMAND gemeint?

Gruß Bastian

---

diamorph

31.05.12, 21:29

05 gibt die Heizung immer zurück wenn die Verbindung geöffnet ist.

Hast du die DeviceTool eingebunden?

PHP-Code:

```
include( "ViessmannDeviceTools.inc.php" );
```

MFG

---

bastianstrauss

31.05.12, 21:32

Hallo soweit war ich noch gar nicht. Bei Punkt 5 in der Anleitung scheitere ich bereits. Das heißt das Script liest gar nicht die ID der Anlage aus.

Zitat:

*Da die Variable "Steuerungstyp" nicht gefüllt ist, wird automatisch versucht, den Typ der Steuerung auszulesen und in dieser Variable abzulegen.*

Gruß

---

diamorph

31.05.12, 21:41

Sind die Variablen in der ViessmannVariables.inc.php denn richtig gesetzt und bist auch sicher, dass IPS alle Skripte erreichen kann?

---

bastianstrauss

31.05.12, 21:49

Hallo,

in der ViessmannVariables.inc.php sind alle Variablen gesetzt.

die ViessmannDeviceOutputHandlingKW.php ist mit der Register Variable verknüpft und wird auch beim Testöffnen des ComPorts aktualisiert (also wird bearbeitet). Die inkludierte ViessmannVariables.inc.php wird auch aufgerufen (habe einfach in der ViessmannVariables.inc.php einen weiteren Befehl eingebunden der auch ausgeführt wurde). Die ViessmannDeviceTools.inc.php brauche ich ja erst, wenn ich direkt Werte aus der Anlage abfragen will. Aber wie gesagt, es scheitert bereits im Schritt 5 bei der Einrichtung.

4 Variablen sind unter Register Variable angelegt (3x String 1x Boolean)

Com Port auf 4800 8 E 2 gesetzt. Über die OpenV Tools ist die Anlage zu erreichen. Auch im Debugger von IPS sendet die Anlage den Hex 5 zurück als Status. Somit gehe ich davon aus, dass die Kommunikation stimmt.

Wenn ich ViessmannDeviceOutputHandlingKW.php in IPS von Hand starte kommt halt die Fehlermeldung, dass VIESSMANN\_DEVICE\_COMMAND nicht definiert ist. Hier vermute ich einen Fehler.

---

diamorph

31.05.12, 21:59

Ohne da jetzt tiefer einzusteigen, (bin da übrigens auch kein Fachmann :) ) sieht das für mich so aus als ob der

VIESSMANN\_DEVICE\_COMMAND

VIESSMANN\_VARIABLE\_DEVICE

heißen müsste. Denn soweit ich das jetzt überblickt habe sollte dieses SetValue den Heizungstyp in die "VIESSMANN\_VARIABLE\_DEVICE" Variable welche unter der ViessmannVariables.inc.php hinterlegt wurde hineingeschrieben werden.

---

bastianstrauss

31.05.12, 22:04

Jo so sehe ich das auch.

Na ja werde noch mal eine Nacht darüber schlafen. Denn meistens lösen sich die Probleme am nächsten Tag wie von alleine.

Bei mir habe ich noch zusätzlich das Problem, das der Optolink via USB to Lan Server angebunden wurde und am virtuellen Rechner nur eine virtuelle Schnittstelle anliegt. \*Schauder\*  
Mit dem M-Bus von Relay funktioniert das aber super.

Trotzdem danke für die Unterstützung

Gruß

---

bastianstrauss

01.06.12, 22:41

Also nach langem probieren mit diversen Tools musste ich feststellen, dass unsere Anlage zu neu ist und scheinbar entgegen der Beschreibung aus openv nicht abwärtskompatibel ist.  
Sprich die Anlage versteht nur das 300er Protokoll. Also werde ich mit dem Script als Vorlage eine neue Version schreiben.

---

bastianstrauss

16.07.12, 21:18

So...

Unsere Anlage versteht doch das KW Protokoll, allerdings sind ein paar Änderung im Script notwendig gewesen.

In der ViessmannDeviceOutputHandlingKW.php Datei folgende Zeile ändern:  
Zeile 20

PHP-Code:

```
SetValueString( VIESSMANN_VARIABLE_COMMAND, $Command );
```

Zeile 37

PHP-Code:

```
$ComData .= $_IPS['VALUE'];
```

In der Datei ViessmannDeviceTools.inc.php folgende Änderungen:  
Zeile 165 - 166

PHP-Code:

```
if ( $Result != chr(0x05).chr(0x05)) $Run = -1;  
// $Run--;
```

Die letzte Änderung bewirkt, wenn sich das Script verschluckt, dann wird die Leerantwort der Heizungsanlage eingetragen (2x 0x05) das ergibt falsche Werte (12,8°C z.B.) Daher die Scriptanpassung.

---

Coyote

04.09.12, 14:29

Nur mal so als Ankündigung: Ich arbeite gerade an der Protokoll 300 Version. Abfrage läuft schon. Sollte demnächst kommen ;)

Leider nicht als Modul, sondern wieder als Script-Variante. Das dürfte den einen oder anderen aber ggf. auch lieber sein...

---

andreasypilon

05.09.12, 09:39

Hi,

hört sich klasse an... Kannst Du vielleicht für mich als Anfänger kurz die Vorteile zusammenfassen? Sind dann alle Merkmale ausles- und schreibbar? Wie Du ja weißt, kämpfe ich seit längerem schon mit der Suche passender Adressen für meine Heizung... Hat das dann ein Ende?

Wenn Du noch einen Beta Tester brauchst...

P.S. VDensH01 / 20C2

---

Coyote

06.09.12, 10:36

Hi,

der Vorteil eines echten Moduls (was ich eigentlich wollte) wäre gewesen:

- komplett in IPS integriert (Datei kopieren, danach über Instanzverwaltung einfach unter den ComPort und fertig)
- kein wirklichen manuellen Schritte vom Anwender nötig

Leider ist es aber so, das man dazu Delphi 2007 benötigt. Diese Version kostet knapp 1.000,-- €, welches es mir nicht wert ist. Ich hatte eine "halblegale" Kopie, die aber jetzt, bei einem Rechnerwechsel, verlorengegangen ist. Somit kann ich aktuell nicht Delphi nutzen.

Also, Back to the Roots...

Die Neue Version ähnelt eigentlich der alten Version. Hauptunterschied ist das verwendete Protokoll. Die alte Version hat das KW-Protokoll verwendet, die neue das 300er. Wer die Adressen seiner Heizung hat, kann damit die Daten auslesen und bei einzelnen Adressen (das geht ja nur z.B. bei den Modi) auch schreiben. Von komplexeren Dingen wie dem schreiben von Steuerzeiten würde ich dringend abraten. Die sind nicht trivial hinterlegt.

So kann ich z.B. mit der neuen Version den Partymodus aktivieren, was bei der alten komischerweise nicht ging. Beim Sparmodus will meine Vitotronic aber leider trotzdem nicht.... !?

Bzgl. der Adressen scheint es so zu sein, das es eine grössere Menge an Adressen gibt, die über die Steuerungen identisch ist. So wird z.B. die gedämpfte Aussentemperatur bei allen Steuerungen in 5525 mit 2 Bytes abgelegt. Insofern kann man sich darüber gut austauschen. Das wäre dann ggf. ein Folgeschritt. Ich werde aber einige Adressen mitliefern (was fehlt dir

denn?).

Wie genau die endgültige Version aussehen wird, muss ich mal schauen. Mein Ziel sollte aber eigentlich eine möglichst einfache Verwendung für "einfache" Nutzer sein. Also

- Skript(e) anlegen
- COMPort und RegisterVariable anlegen sowie der RegisterVariable ein Skript zuordnen
- Skript in eigene Skripte inkludieren und einfache, klare Befehle nutzen

Fehler sollten dann zentral korrigiert werden, so dass durch neue Versionen etwas davon haben.

Gruss  
Coyote

---

## andreasypilon

06.09.12, 13:18

Zitat:

Zitat von **Coyote** »

*Ich werde aber einige Adressen mitliefern (was fehlt dir denn?).*

Hallo,

erstmal vielen Dank für die ausführliche Antwort. Klasse....

Was mir fehlt bzw. woraus ich nicht schlau werde, ist alles was mit dem Brenner zusammenhängt. Brennerstarts, -zustand, -leistung und Laufzeit. Abfragen kann ich, bekomme auch Werte zurück, allerdings sind die nicht wirklich zu gebrauchen. Entweder bin ich an den falschen Adressen oder die Umrechnung passt nicht. Und da ich noch ziemlicher PHP Anfänger bin... Habe zwar den Umrechnungsblock in der viessmanndevicetools.inc.php gefunden, aber so richtig erhellt es mich noch nicht...

:confused:

edit: Worauf ich auch nicht gekommen bin: Wenn ich mit "v-control" die Daten kontrolliere, funktioniert das Skript innerhalb von IPS nicht mehr. Auch wenn ich den Com Port manuell schließe... Gibts da noch einen Trick, ohne den Server neu starten zu müssen?

---

## andreasypilon

06.09.12, 15:01

Die Brennerlaufzeit habe ich jetzt.

Die Adresse war korrekt, nur die Umrechnung falsch. Habe in der viessmanndevicetools.inc.php einen weiteren Case eingebaut:

Code:

```
case 8:
$Count = $Count ;
break;
```

Somit wird der Wert in \$Count nativ ausgegeben. Und das ist die Brennerlaufzeit in Sekunden. Das durch 3600 und wir haben Stunden. Der Case müsste doch auch so umzubauen sein, dass ich hier direkt rechnen kann, oder? Meine kläglichen PHP Kenntnisse scheitern hier aber...

---

## Coyote

07.09.12, 10:30

Hi,

den Typ "StundenInSekunden" kennt meine neue Implementierung ;)

Gruss  
Coyote

---

## mikewolf

24.10.12, 20:36

Funktioniert die Abfrage/Steu schon ?

Hallo hätte Interesse an der Heizungssteuerung, habe eine Vitotronic VScotH01

schaffe es aber derzeit nicht das "alte" Script zum laufen zu bekommen .

Wäre dankbar da ich derzeit mit Vcontrol arbeite aber manche stimmt nicht , da ich eine Hydraulische Weiche habe und daher ich keine normale Vorlauftemperatur habe sondern eine "gemeinsame" und würde auch gerne den Partymodus bzw den Sparmodus über Ips steuern können wegen Anwesenheit

Hätte auch noch adresen wenn Du eventuell welche benötigt ?

mfg  
mikewolf

---

## masterp

31.10.12, 18:52

Frage zur Viessmannsteuerung

Hallo

ich habe mal wieder ein Problem

ich habe die Viessmann Abfrage von coyote (vielen Dank dafür) eingerichtet, aber leider funktioniert sie bei mir nicht habe alles wie beschrieben(nun schon 3x) eingerichtet, Variablen angelegt, im script bekannt gemacht ...

doch leider scheitert es bei mir schon am Punkt 5. wenn ich den comPort öffne, sehe ich im Debug kein Datenfluss bzw. sehe ich nicht den periodischen hex Wert der da sein müsste, ich habe mir ein comPort Monitor zur Hilfe genommen, und siehe da, dort sehe ich den Wert, sobald ich aber in IPS den Port öffne kommen keine Daten mehr an, woran liegt das ??? :confused: hat irgend jemand eine Idee und kann einen ratlosen Neuling helfen

vielen Dank  
Remo

---

chris.moe

02.01.13, 18:04

Hallo,

habe einen USB-Optolink Adapter an die Fritz Box angeschlossen, USB Fernanschluss in der Fritz Box aktiviert und siehe da:

Device VScotHO1  
ID 20CB  
ExtID 002B  
Protocol 300

per VitoOpen Identifikation ausgelesen! Auch V-Control funktioniert!

Bin mir jetzt jedoch nicht ganz sicher, wie ich mit IP-SYMCON am Besten an die Daten rankomme? Mit dem Skript hier auf der ersten Seite oder mit dem "V\_Control\_to\_IPSymcon\_1\_0\_0\_0" Modul? Oder gibts da noch was neueres?

Gruß,

Chris

---

Spechtl

27.01.13, 09:43

Hi Coyote,

sorry für die Nachfrage, aber bist Du mit der Implementierung des 300er Protokoll weitergekommen, die Abfragen mit den KW Protokoll laufen Dank Deiner Arbeit zwar, allerdings nicht wirklich stabil. Würde jede Art von der 300er Implementierung begrüßen, leider auch mit dem Statement bedingt durch meine Fähigkeiten nicht wirklich beitragen zu können, außer Feedback zu geben und als Tester gerne mitzuarbeiten.

Grüße  
Spechtl

---

Schablone

05.02.13, 12:52

Ich komme sehr gut mit [dieser](#) Beschreibung zurecht.

Zum testen habe ich [das](#) genommen.

---

hcp

07.02.13, 22:42

Zitat:

---

Zitat von **Schablone** 

Ich komme sehr gut mit [dieser](#) Beschreibung zurecht.

Zum testen habe ich [das](#) genommen.

---

Habe versucht mal ein Testscript an eine Registervariable zu hängen und das 300er Protokoll auszutesten. Bin nicht weitergekommen. Hat jmd. schon ein php-Testscript zum Laufen gebracht?

Ich kann via php den Bus in den 300er Modus versetzen und auch wieder zurück. Ich kann dann einen Befehl zum Auslesen (z.B. Geräteart) schicken - es kommt aber nichts zurück.

Mit dem Tool zum Testen von oben kann ich auch auslesen ... das ist aber nicht php ...

Gruss, hcp

---

Schablone

08.02.13, 23:01

Hallo hcp,

ich bin selber noch in der Findungsphase. ;)  
Ich kann dir nur sagen was ich bis jetzt gemacht habe.

1. Ich habe mir den USB-Optolink aus dem ersten Post bestellt.
2. Den USB-Optolink habe ich in den PC eingesteckt.  
Danach hat sich das Teil selbstständig installiert mit einem USB - Serial Port Umsetzer.  
Da brauchte ich nichts für zu tun.
3. Diesen Umsetzer habe ich so eingestellt.

[Anhang 20164](#)

4. Danach habe ich in IPS einen Serial Port angelegt und so eingestellt.

Liste der Anhänge anzeigen (Anzahl: 4)

## Anhang 20165

5. An diesem Serial Port habe ich eine Register Variable gehängt.  
6. An der Register-Variable habe ich ein Script gehängt.

## Anhang 20166

7. In IPS eine String Variable anlegen die da heißt Receive o.ä.  
8. In dem Script das die Register Variable aufruft steht das hier.

PHP-Code:

```
// neu empfangene Daten ablegen
$data = $_IPS['VALUE'];
// Daten ausgeben
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, $data);
```

Erster Test:

In deiner Receive Variablen sollte nun über die Com-Schnittstelle das hier kommen

## Anhang 20167

Die Uhrzeit sollte sich alle 2-3 Sekunden aktualisieren.

Dieser Strich den Du in der Variablen siehst ist das Ascii-Zeichen für "0x05 Hex" der sogenannte Viessmann-Ping.

Wenn Du so weit bist geht es weiter!!! :)

hcp

09.02.13, 21:36

Zitat:

Zitat von **Schablone** 

Hallo hcp,

ich bin selber noch in der Findungsphase. ;)

Ich kann dir nur sagen was ich bis jetzt gemacht habe.

1. Ich habe mir den USB-Optolink aus dem ersten Post bestellt.
2. Den USB-Optolink habe ich in den PC eingesteckt.  
Danach hat sich das Teil selbstständig installiert mit einem USB - Serial Port Umsetzer.  
Da brauchte ich nichts für zu tun.
3. Diesen Umsetzer habe ich so eingestellt.

## Anhang 20164

4. Danach habe ich in IPS einen Serial Port angelegt und so eingestellt.

## Anhang 20165

5. An diesem Serial Port habe ich eine Register Variable gehängt.  
6. An der Register-Variable habe ich ein Script gehängt.

## Anhang 20166

7. In IPS eine String Variable anlegen die da heißt Receive o.ä.  
8. In dem Script das die Register Variable aufruft steht das hier.

PHP-Code:

```
// neu empfangene Daten ablegen
$data = $_IPS['VALUE'];
// Daten ausgeben
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, $data);
```

Erster Test:

In deiner Receive Variablen sollte nun über die Com-Schnittstelle das hier kommen

## Anhang 20167

Die Uhrzeit sollte sich alle 2-3 Sekunden aktualisieren.

Dieser Strich den Du in der Variablen siehst ist das Ascii-Zeichen für "0x05 Hex" der sogenannte Viessmann-Ping.

Wenn Du so weit bist geht es weiter!!! :)

Das klappt bei mir auch. Kann dann auch den Bus in den 300er Modus setzen indem ich 0x16 0x00 0x00 gleich nach einem 0x05 Empfang sende. Bekomme dann auch wie erwartet die 0x06 als Antwort. Wenn ich danach aber etwas sende bekomme ich keine vernünftige Antwort ... da hänge ich ...

Schablone

10.02.13, 21:17

Zitat:

Zitat von **hcp** 

*Das klappt bei mir auch. Kann dann auch den Bus in den 300er Modus setzen indem ich 0x16 0x00 0x00 gleich nach einem 0x05 Empfang sende. Bekomme dann auch wie erwartet die 0x06 als Antwort. Wenn ich danach aber etwas sende bekomme ich keine vernünftige Antwort ... da hänge ich ...*

Das ist klar. Die Antwort kommt in mehreren Häppchen.

Bei mir sieht das Empfangsscript so aus:

PHP-Code:

```
<?
/*=====
ViessmannOut - Martin Heinzel - 05.02.2013
Version: 0.1

Beschreibung:
Dieses Script wird von der Registervariablen gestartet.
Hier wird der Empfang der Daten gehandelt und in eine
Empfangsvariablen abgelegt.

Änderungen
-----

tt.mm.jjjj von Version x.x -> y.y

Beschreibung:
=====*/

// Variablen Deklaration -----
// ID's
//-----
$ID_Connect_Done           = 38994 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect_Done]*/;
$SID_ViessmannEmpfangen    = 33859 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\ViessmannEmpfangen]*/;

// Variablen
//-----
$Connect                   = GetValueBoolean(47954 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect]*/);
$Receive_lenght            = GetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/);

// Konstanten
//-----

//-----
require IPS_GetKernelDir(). "scripts\\ViessmannVariablen.ips.php";
//----- Main -----

// wenn das Skript von einer RegisterVariable-Instanz aus aufgerufen worden ist
if ($_IPS['SENDER'] == "RegisterVariable")
{
    // kontrolle der empfangenen Daten auf den Viessmann "PING"
    if($_IPS['VALUE'] == VIESS_PING)
    {
        // neu empfangene Daten ablegen
        $data = $_IPS['VALUE'];
        // Inhalt des Puffer der RegisterVariable-Instanz löschen
        RegVar_SetBuffer($_IPS['INSTANCE'], "");
        // Daten ausgeben
        SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, $data);
        // Verbindung zu Viessmann ist abgebaut
        SetValueBoolean($ID_Connect_Done, false);
        // Timeout Zeit zurücksetzen
        IPS_SetScriptTimer($SID_ViessmannEmpfangen , 0);
        // Wenn verbunden werden soll dann
        if($Connect)
        {
            // verbinden
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_CONNECT );
        }
    }
    else
    {
        if(substr($_IPS['VALUE'], 0, 1) == VIESS_CONNECTED)
        {
            // Verbindung zu Viessmann ist aufgebaut
            SetValueBoolean($ID_Connect_Done, true);
            // Inhalt des Puffer der RegisterVariable-Instanz löschen
            RegVar_SetBuffer($_IPS['INSTANCE'], "");
        }

        // bereits im Puffer der Instanz vorhandene Daten in $data kopieren
        $data = RegVar_GetBuffer($_IPS['INSTANCE']);
        // neu empfangene Daten an $data anhängen
        $data .= $_IPS['VALUE'];
        // Inhalt von $data im Puffer der RegisterVariable-Instanz speichern
        RegVar_SetBuffer($_IPS['INSTANCE'], $data);

        if(strlen($data) == $Receive_lenght or $Receive_lenght == 0)
        {
            // ausgeben der Daten
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, $data);
        }
    }
}
```

```
//----- Main ENDE -----
?>
```

Das ist das Script ViessmannVariablen.ips.php

PHP-Code:

```
<?
/*=====
Script - ViessmannVariablen - 03.02.2013
Version: 0.1

Beschreibung:
In diesem Script werden die Viessmann - Konstanten definiert.

Änderungen
-----

tt.mm.jjjj von Version x.x -> y.y

Beschreibung:

=====*/

// Variablen Deklaration -----

// Konstanten
//-----
// Empfangen
define( "VIESS_PING",                chr(0x05));
define( "VIESS_COMERR",              chr(0x15));
define( "VIESS_CONNECTED",           chr(0x06));

// Senden
define( "IPS_CONNECT",                chr(0x16).chr(0x00).chr(0x00));
define( "IPS_DISCONNECT",            chr(0x04));

// Senden Telegramme
define( "IPS_Temp_Aussen_ACT",         chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x55).chr(0x25). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0x82));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of

define( "IPS_Temp_Kessel_ACT",         chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x08).chr(0x10). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0x20));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of

define( "IPS_Temp_Kessel_SET",         chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x55).chr(0x5A). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0xB7));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of

define( "IPS_Temp_Speicher_ACT",       chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x08).chr(0x12). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0x22));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of

define( "IPS_Temp_Speicher_SET",       chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x63).chr(0x00). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0x6B));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of

define( "IPS_Temp_Vorlauf_ACT",        chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x29).chr(0x00). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0x31));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of

define( "IPS_Temp_Vorlauf_SET",        chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x25).chr(0x44). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0x71));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of

define( "IPS_Temp_Ruecklauf_ACT",      chr(0x41).                // Telegramm Start-Byte
                                     chr(0x05).                // Länge der Nutzdaten
                                     chr(0x00).                // 00 = Anfrage, 01 = Antwort, 03 = Fehler
                                     chr(0x01).                // 01 = ReadData, 02 = WriteData, 07 = Function
                                     chr(0x08).chr(0x08). // 2 byte Adresse der Daten oder Prozedur
                                     chr(0x02).                // Anzahl der Bytes, die in der Antwort erwartet
                                     chr(0x18));                // Prüfsumme = Summe der Werte ab dem 2 Byte (of
```



```
//-----
?>
```

hcp

13.02.13, 13:41

Skript läuft und empfängt die Pings. Was kommt als nächstes?

Thomas

18.02.13, 19:11

Adapter/Abfrage läuft

Hab nun gestern meinen Adapter in Betrieb genommen.  
Habe beide Scriptvarianten ausprobiert und verwende nun die Scripts aus dem ersten Post.  
Damit habe ich problemlos die Abfrage von Werten hinbekommen.  
Meine Anlage ist eine VPlusHO1 (20C8).  
Mache mich dann mal an das Logging von Daten und die ersten Einstellungen.  
Falls es updates gibt oder ich helfen kann, laßt es mich wissen.

Gruß Thomas

Spechtl

27.02.13, 20:08

Hi Schablone,

bin Deiner Anregung gefolgt und versuche auf diesen Weg das Protokoll meiner Abfrage umzustellen. Habe mich an Deine Instruktionen gehalten, aber nachdem das alles für mich immer noch ein großes Rätsel darstellt, bin ich wohl auf Deine Hilfe in Form von den weiteren Schritten angewiesen.

Danke  
Spechtl

Schablone

28.02.13, 04:34

Ich bin noch bei einer IBN in Holland (Bergen). Wenn ich es schaffe, werde ich am Wochenende das, was ich bis jetzt habe einstellen. Ich bin aber selber noch nicht ganz fertig!

Nur zur Info:

Als nächstes wird das Online-Bit gesetzt. Dann geht das Viessmann Out Script nach dem nächsten Ping auf "Verbinden". Danach Geht die Viessmann auf 06 hex.

Dann ist die Viessmann bereit Anfragen zu beantworten.

Die Anfragen die ich für meine Heizung senden kann, stehen in den Konstanten die ich bereits gepostet habe.

Gesendet von meinem GT-I9100 mit Tapatak 2

Schablone

03.03.13, 20:36

Hallo Spechtel,

wie versprochen meine Scripte:

Das Senden Script:

PHP-Code:

```
<?
/*=====
ViessmannSenden - Martin Heinzel - 05.02.2013
Version: 0.1

Beschreibung:
Mit diesem Script werden Daten gesendet

Änderungen
-----

tt.mm.jjjj von Version x.x -> y.y

Beschreibung:

=====*/

// Variablen Deklaration -----

// ID's
//-----
$ID_Connect          = 47954 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect]*/;
$ID_Connect_Done     = 38994 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect_Done]*/;
$ID_Token            = 24085 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Token]*/;

// Variablen
//-----
$Connect_Done        = GetValueBoolean($ID_Connect_Done);
$Connect             = GetValueBoolean($ID_Connect);
$Token               = GetValueInteger($ID_Token);
```

```

// Konstanten
//-----

//-----

require IPS_GetKernelDir(). "scripts\\ViessmannVariablen.ips.php";

//----- Main -----

// Verbindung zu Viessmann herstellen
if(!$Connect_Done and !$Connect)
{
    SetValueBoolean($ID_Connect, true);
}

// Verbindung zu Viessmann OK
if($Connect_Done)
{
    Switch($Token)
    {
        //##### Temperaturen #####
        case 1:
            // abfragen der Aussentemperatur
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Aussen_ACT );
            break;

        case 2:
            // abfragen der Kesseltemperatur ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Kessel_ACT );
            break;

        case 3:
            // abfragen der Kesseltemperatur SET
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Kessel_SET );
            break;

        case 4:
            // abfragen der Speichertemperatur ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Speicher_ACT );
            break;

        case 5:
            // abfragen der Speichertemperatur SET
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Speicher_SET );
            break;

        case 6:
            // abfragen der Vorlauftemperatur ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Vorlauf_ACT );
            break;

        case 7:
            // abfragen der Vorlauftemperatur SET
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Vorlauf_SET );
            break;

        case 8:
            // abfragen der Rücklauftemperatur ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Ruecklauf_ACT );
            break;

        case 9:
            // abfragen der Betriebsart ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Betriebsart_ACT );
            break;

        case 10:
            // abfragen der Brennerstarts ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 13);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Brenner_Starts_ACT );
            break;

        case 11:
            // abfragen der Brenner Stunden ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 13);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Brenner_Stunden_ACT );
            break;

        case 12:
            // Verbindung zu Viessmann abbauen
            SetValueBoolean($ID_Connect, false);
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 0);
    }
}

```

```

        SetValueInteger($ID_Token, 0);
        COMPort_SendText(43855 /*[Viessmann]*/, IPS_DISCONNECT );
        //IPS_SetScriptTimer($ _IPS['SELF'], 30);
    }
}

//----- Main ENDE -----

?>

```

Das Empfangene Script:

PHP-Code:

```

<?
/*=====
ViessmannEmpfangen - Martin Heinzel - 05.02.2013
Version: 0.1

Beschreibung:
Dieses Script wertet die Empfangenen Daten der Viessmann aus.

Änderungen
-----

tt.mm.jjjj von Version x.x -> y.y

Beschreibung:
=====*/

// Variablen Deklaration -----

// ID's
//-----
$ID_Receive           = 39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/;
$ID_Connect           = 47954 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect]*/;
$ID_Connect_Done      = 38994 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect_Done]*/;
$ID_Token             = 24085 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Token]*/;
$ID_TimeOutCounter    = 17928 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\TimeOutCounter]*/;
$ID_CommFaultCounter  = 34278 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\CommFaultCounter]*/;
$ID_TimeOut           = 38680 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\TimeOut]*/;
$SID_ViessmannSenden  = 25080 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\ViessmannSenden]*/;
$ID_Temp_Aussen_ACT   = 40248 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Aussen_ACT]*/;
$ID_Temp_Kessel_ACT   = 30466 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Kessel_ACT]*/;
$ID_Temp_Kessel_SET   = 12619 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Kessel_SET]*/;
$ID_Temp_Speicher_ACT = 27117 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Speicher_ACT]*/;
$ID_Temp_Speicher_SET = 22278 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Speicher_SET]*/;
$ID_Temp_Vorlauf_ACT  = 48976 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Vorlauf_ACT]*/;
$ID_Temp_Vorlauf_SET  = 35626 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Vorlauf_SET]*/;
$ID_Temp_Ruecklauf_ACT = 43689 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Ruecklauf_ACT]*/;
$ID_Betriebsart_ACT   = 50731 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Betriebsart_ACT]*/;
$ID_Brennerstarts_ACT = 40380 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Brennerstarts_ACT]*/;
$ID_Brenner_Stunden_ACT = 42523 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Brenner_Stunden_ACT]*/;

// Variablen
//-----
$Receive           = GetValueString($ID_Receive);
$Connect_Done      = GetValueBoolean($ID_Connect_Done);
$Connect           = GetValueBoolean($ID_Connect);
$Token             = GetValueInteger($ID_Token);
$Receive_Byte      = array();
$TimeOutCounter    = GetValueInteger($ID_TimeOutCounter);
$CommFaultCounter  = GetValueInteger($ID_CommFaultCounter);

// Konstanten
//-----

//-----

//----- Main -----

// Byteweise einlesen der Daten
for ($i = 0; $i < strlen($Receive); $i++)
{
    // wandeln der einzelnen Zeichen von String in Hex
    $Receive_Byte[$i] = dechex(ord(substr($Receive, $i)));
    //echo $Receive_Byte[$i]. "\r\n";
}

// Wenn die Checksumme OK ist
if(checksum($Receive_Byte) == $Receive_Byte[strlen($Receive)-1])
{
    //zurücksetzen des TimeOut Counters
    SetValueInteger($ID_TimeOutCounter, 0);
    //rücksetzen des TimeOut Flag
    SetValueBoolean($ID_TimeOut, false);

    Switch($Token)
    {
        //##### Temperaturen #####
        case 1:
            // Auslesen der Aussentemperatur
            ViessmannTemp($Receive_Byte, $ID_Temp_Aussen_ACT);
            SetValueInteger($ID_Token, 2);
            IPS_SetScriptTimer($ _IPS['SELF'], 5);
            break;

        case 2:
            // Auslesen der Kesseltemperatur ACT

```

```

ViesmannTemp($Receive_Byte, $ID_Temp_Kessel_ACT);
SetValueInteger($ID_Token, 3);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 3:
// Auslesen der Kesseltemperatur SET
ViesmannTemp($Receive_Byte, $ID_Temp_Kessel_SET);
SetValueInteger($ID_Token, 4);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 4:
// Auslesen der Speichertemperatur ACT
ViesmannTemp($Receive_Byte, $ID_Temp_Speicher_ACT);
SetValueInteger($ID_Token, 5);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 5:
// Auslesen der Speichertemperatur SET
ViesmannTemp($Receive_Byte, $ID_Temp_Speicher_SET);
SetValueInteger($ID_Token, 6);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 6:
// Auslesen der Vorlauftemperatur ACT
ViesmannTemp($Receive_Byte, $ID_Temp_Vorlauf_ACT);
SetValueInteger($ID_Token, 7);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 7:
// Auslesen der Vorlauftemperatur SET
ViesmannTemp($Receive_Byte, $ID_Temp_Vorlauf_SET);
SetValueInteger($ID_Token, 8);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 8:
// Auslesen der Rücklauftemperatur ACT
ViesmannTemp($Receive_Byte, $ID_Temp_Ruecklauf_ACT);
SetValueInteger($ID_Token, 9);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 9:
// Auslesen der Betriebsart ACT
Switch($Receive_Byte[8])
{
    case 0:
        SetValueString($ID_Betriebsart_ACT, "Aus");
        break;

    case 1:
        SetValueString($ID_Betriebsart_ACT, "Warmwasser");
        break;

    case 2:
        SetValueString($ID_Betriebsart_ACT, "Warmwasser + Heizen");
        break;
}
SetValueInteger($ID_Token, 10);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 10:
ViesmannCount($Receive_Byte, $ID_Brennerstarts_ACT);
SetValueInteger($ID_Token, 11);
IPS_SetScriptTimer($_IPS['SELF'], 5);
break;

case 11:
$Count_HEX = $Receive_Byte[11].$Receive_Byte[10].$Receive_Byte[9].$Receive_Byte[8];
$Counter = hexdec($Count_HEX)/3600;
SetValueInteger($ID_Brenner_Stunden_ACT, (int)$Counter);
SetValueInteger($ID_Token, 12);
IPS_SetScriptTimer($_IPS['SELF'], 5);
}
}

// Wenn die Verbindung zu Viessmann hergestellt ist
if($Connect_Done and $Connect and ($Token == 0))
{
    //Erste Abfrage starten
    SetValueInteger($ID_Token, 1);
    IPS_SetScriptTimer($_IPS['SELF'], 5);
    IPS_SetScriptTimer($_SID_ViessmannSenden, 0);
}

// Timeout
if($_IPS['SENDER'] == "TimerEvent")
{
    //Erhöhen des Timeout
    if($TimeoutCounter <= 2)
    {
        $TimeoutCounter += 1;
        SetValueInteger($ID_TimeOutCounter, $TimeoutCounter);
        IPS_RunScript(25080 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\ViessmannSenden]*);
    }
    else
    {
        //zurücksetzen des Timeout Counters
        SetValueInteger($ID_TimeOutCounter, 0);
    }
}

```

```

//erhöhen des gesamten Fault counters
$CommFaultCounter += 1;
SetValueInteger($ID_CommFaultCounter, $CommFaultCounter);
//setzen des TimeOut Flag
SetValueBoolean($ID_TimeOut, true);
//erhöhen des Token
if($Token <= 12)
{
    $Token += 1;
    SetValueInteger($ID_Token , $Token);
}
else
{
    // Verbindung zu Viessmann beenden
    SetValueBoolean($ID_Connect_Done, false);
}
}
}

//----- Main ENDE -----

//----- Funktionen -----

/**
 * Returns the calculated checksum.
 * @param Byte_array $Receive_Byte Array of the received Bytes
 * @return Calculated checksum in hex
 */
function checksum($Receive_Byte)
{
    // Prüfen der Länge
    $Receive_lenght = count($Receive_Byte);
    // initialisieren der Checksumme
    $checksum = 0;

    // Aufaddieren der Byte's
    for ($i = 2; $i < $Receive_lenght-1; $i++)
    {
        $checksum += hexdec($Receive_Byte[$i]);
    }

    // wandeln in hex und ausgeben der ersten beiden Zeichen
    $checksum_hex = substr(dechex($checksum), -2);
    //if(substr($checksum_hex, 0) == 0)
    //{
    //    $checksum_hex = substr(dechex($checksum), -1);
    //}
    return $checksum_hex;
}

function ViesmannTemp($Receive_Byte, $ID)
{
    $Temp_HEX = $Receive_Byte[9].$Receive_Byte[8];
    $Temperatur = hexdec($Temp_HEX)/10.0;
    SetValueFloat($ID, $Temperatur);
}

function ViesmannCount($Receive_Byte, $ID)
{
    $Count_HEX = $Receive_Byte[11].$Receive_Byte[10].$Receive_Byte[9].$Receive_Byte[8];
    $Counter = hexdec($Count_HEX);
    SetValueInteger($ID, $Counter);
}

//----- Funktionen ENDE -----

?>

```

Das sind noch keine fertigen Scripte!!

Es sind nur Beispiele wie man Sendet und Empfängt.

---

Thomas

24.05.13, 20:24

Anbei auch die aktuelle Version meiner Scripte inkl. einer XML für die 20C8.  
 Basieren immer noch auf der Version von coyote aus dem ersten Post.  
 Habe bei der Behandlung von Count noch was geändert, damit die Betriebsstunden richtig übergeben werden.  
 Ebenso hatte ich öfter Viessmann\_Ping (0x05) statt Werten in meinen Variablen und werfe die nun weg.  
 Damit lese ich alle 5 min Betriebswerte aus (im Script Viessmann\_Send).  
 Zum Schreiben von Werten bin ich noch nicht gekommen.

mfg  
 Thomas

Liste der Anhänge anzeigen (Anzahl: 1)

---

bastianstrauss

27.05.13, 19:03

Also ich nutze auch die Version aus dem ersten Post. Hatte auch ähnliche Probleme wie Thomas.  
 Bei mir läuft das Script in 4 Anlagen.  
[Anhang 21228](#)

Liste der Anhänge anzeigen (Anzahl: 1)

---

Schablone

12.06.13, 06:33

Guten Morgen!!

Ich habe ein paar Fragen zu meiner Viessmann Model "Vitodens 300-W".

Ich arbeite am 300er Protokoll.  
Meine Gerätekenung ist die "20CB".

1.

Ich habe nicht die richtige Adresse für den Warmwasser-Sollwert.  
Ich benutze die "chr(0x63).chr(0x00)" und bekomme immer nur einen Wert von 5°C.  
Hat da jemand eine andere Adresse?

2.

Die Zirkulationspumpe (Adresse chr(0x65).chr(0x15)) ist mir nicht ganz klar.  
a. Ist das die Externe Pumpe für Warmwasser oder die Heizungs(Heizkreise)pumpe?  
b. Stimmt meine Adresse chr(0x65).chr(0x15)?

Für eure Hilfe besten Dank!!

hcp

02.07.13, 14:46

Zitat:

Zitat von [Schablone](#)

Hallo Spechtel,

wie versprochen meine Scripte:

Das Senden Script:

PHP-Code:

```
<?
/*=====
ViessmannSenden - Martin Heinzel - 05.02.2013
Version: 0.1

Beschreibung:
Mit diesem Script werden Daten gesendet

Änderungen
-----

tt.mm.jjjj von Version x.x -> y.y

Beschreibung:
=====*/

// Variablen Deklaration -----

// ID's
//-----
$ID_Connect           = 47954 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect]*/;
$ID_Connect_Done      = 38994 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect_Done]*/;
$ID_Token              = 24085 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Token]*/;

// Variablen
//-----
$Connect_Done         = GetValueBoolean($ID_Connect_Done);
$Connect              = GetValueBoolean($ID_Connect);
$Token                = GetValueInteger($ID_Token);

// Konstanten
//-----

//-----
require IPS_GetKernelDir()."scripts\ViessmannVariablen.ips.php";

//----- Main -----

// Verbindung zu Viessmann herstellen
if(!$Connect_Done and !$Connect)
{
    SetValueBoolean($ID_Connect, true);
}

// Verbindung zu Viessmann OK
if($Connect_Done)
{
    Switch($Token)
    {
        //##### Temperaturen #####
        case 1:
            // abfragen der Aussentemperatur
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
            SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
            COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Aussen_ACT );
            break;

        case 2:
            // abfragen der Kesseltemperatur ACT
            SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
```

```

SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Kessel_ACT );
break;

case 3:
// abfragen der Kesseltemperatur SET
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Kessel_SET );
break;

case 4:
// abfragen der Speichertemperatur ACT
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Speicher_ACT );
break;

case 5:
// abfragen der Speichertemperatur SET
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Speicher_SET );
break;

case 6:
// abfragen der Vorlauftemperatur ACT
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Vorlauf_ACT );
break;

case 7:
// abfragen der Vorlauftemperatur SET
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Vorlauf_SET );
break;

case 8:
// abfragen der Rücklauftemperatur ACT
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Temp_Ruecklauf_ACT );
break;

case 9:
// abfragen der Betriebsart ACT
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 11);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Betriebsart_ACT );
break;

case 10:
// abfragen der Brennerstarts ACT
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 13);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Brenner_Starts_ACT );
break;

case 11:
// abfragen der Brenner Stunden ACT
SetValueString(39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/, "");
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 13);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_Brenner_Stunden_ACT );
break;

case 12:
// Verbindung zu Viessmann abbauen
SetValueBoolean($ID_Connect, false);
SetValueInteger(21524 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_lenght]*/, 0);
SetValueInteger($ID_Token, 0);
COMPort_SendText(43855 /*[Viessmann]*/, IPS_DISCONNECT );
//IPS_SetScriptTimer($_IPS['SELF'], 30);
}
}

//----- Main ENDE -----

?>

```

Das Empfangen Script:

PHP-Code:

```

<?
/*=====
ViessmannEmpfangen - Martin Heinzel - 05.02.2013
Version: 0.1

Beschreibung:
Dieses Script wertet die Empfangenen Daten der Viessmann aus.

Änderungen
-----

tt.mm.jjjj von Version x.x -> y.y

Beschreibung:
=====*/

```

```
// Variablen Deklaration -----

// ID's
//-----
$ID_Receive           = 39800 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive]*/;
$ID_Connect           = 47954 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect]*/;
$ID_Connect_Done      = 38994 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Connect_Done]*/;
$ID-Token             = 24085 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Token]*/;
$ID_TimeOutCounter    = 17928 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\TimeOutCounter]*/;
$ID_CommFaultCounter  = 34278 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\CommFaultCounter]*/;
$ID_TimeOut           = 38680 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\TimeOut]*/;
$SID_ViessmannSenden  = 25080 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\ViessmannSenden]*/;
$ID_Temp_Aussen_ACT   = 40248 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Aussen_ACT]*/;
$ID_Temp_Kessel_ACT   = 30466 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Kessel_ACT]*/;
$ID_Temp_Kessel_SET   = 12619 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Kessel_SET]*/;
$ID_Temp_Speicher_ACT = 27117 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Speicher_ACT]*/;
$ID_Temp_Speicher_SET = 22278 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Speicher_SET]*/;
$ID_Temp_Vorlauf_ACT  = 48976 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Vorlauf_ACT]*/;
$ID_Temp_Vorlauf_SET  = 35626 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Vorlauf_SET]*/;
$ID_Temp_Ruecklauf_ACT = 43689 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Temp_Ruecklauf_ACT]*/;
$ID_Betriebsart_ACT   = 50731 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Betriebsart_ACT]*/;
$ID_Brennerstarts_ACT = 40380 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Brennerstarts_ACT]*/;
$ID_Brenner_Stunden_ACT = 42523 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\Receive_Data\Brenner_Stunden_ACT]*/;

// Variablen
//-----
$Receive              = GetValueString($ID_Receive);
$Connect_Done         = GetValueBoolean($ID_Connect_Done);
$Connect              = GetValueBoolean($ID_Connect);
$Token                = GetValueInteger($ID-Token);
$Receive_Byte         = array();
$TimeOutCounter        = GetValueInteger($ID_TimeOutCounter);
$CommFaultCounter     = GetValueInteger($ID_CommFaultCounter);

// Konstanten
//-----

//-----

//----- Main -----

// Byteweise einlesen der Daten
for ($i = 0; $i < strlen($Receive); $i++)
{
    // wandeln der einzelnen Zeichen von String in Hex
    $Receive_Byte[$i] = dechex(ord(substr($Receive, $i)));
    //echo $Receive_Byte[$i]. " \r\n";
}

// Wenn die Checksumme OK ist
if(checksum($Receive_Byte) == $Receive_Byte[strlen($Receive)-1])
{
    //zurücksetzen des TimeOut Counters
    SetValueInteger($ID_TimeOutCounter, 0);
    //rücksetzen des TimeOut Flag
    SetValueBoolean($ID_TimeOut, false);

    Switch($Token)
    {
        //##### Temperaturen #####
        case 1:
            // Auslesen der Aussentemperatur
            ViessmannTemp($Receive_Byte, $ID_Temp_Aussen_ACT);
            SetValueInteger($ID-Token, 2);
            IPS_SetScriptTimer($IPS['SELF'], 5);
            break;

        case 2:
            // Auslesen der Kesseltemperatur ACT
            ViessmannTemp($Receive_Byte, $ID_Temp_Kessel_ACT);
            SetValueInteger($ID-Token, 3);
            IPS_SetScriptTimer($IPS['SELF'], 5);
            break;

        case 3:
            // Auslesen der Kesseltemperatur SET
            ViessmannTemp($Receive_Byte, $ID_Temp_Kessel_SET);
            SetValueInteger($ID-Token, 4);
            IPS_SetScriptTimer($IPS['SELF'], 5);
            break;

        case 4:
            // Auslesen der Speichertemperatur ACT
            ViessmannTemp($Receive_Byte, $ID_Temp_Speicher_ACT);
            SetValueInteger($ID-Token, 5);
            IPS_SetScriptTimer($IPS['SELF'], 5);
            break;

        case 5:
            // Auslesen der Speichertemperatur SET
            ViessmannTemp($Receive_Byte, $ID_Temp_Speicher_SET);
            SetValueInteger($ID-Token, 6);
            IPS_SetScriptTimer($IPS['SELF'], 5);
            break;

        case 6:
            // Auslesen der Vorlauftemperatur ACT
            ViessmannTemp($Receive_Byte, $ID_Temp_Vorlauf_ACT);
            SetValueInteger($ID-Token, 7);
            IPS_SetScriptTimer($IPS['SELF'], 5);
            break;

        case 7:
    }
}

```



```

// Auslesen der Vorlauftemperatur SET
ViesmannTemp($Receive_Byte, $ID_Temp_Vorlauf_SET);
SetValueInteger($ID_Token, 8);
IPS_SetScriptTimer($IPS['SELF'], 5);
break;

case 8:
// Auslesen der Rücklauftemperatur ACT
ViesmannTemp($Receive_Byte, $ID_Temp_Ruecklauf_ACT);
SetValueInteger($ID_Token, 9);
IPS_SetScriptTimer($IPS['SELF'], 5);
break;

case 9:
// Auslesen der Betriebsart ACT
Switch($Receive_Byte[8])
{
    case 0:
        SetValueString($ID_Betriebsart_ACT, "Aus");
        break;

    case 1:
        SetValueString($ID_Betriebsart_ACT, "Warmwasser");
        break;

    case 2:
        SetValueString($ID_Betriebsart_ACT, "Warmwasser + Heizen");
}
SetValueInteger($ID_Token, 10);
IPS_SetScriptTimer($IPS['SELF'], 5);
break;

case 10:
ViesmannCount($Receive_Byte, $ID_Brennerstarts_ACT);
SetValueInteger($ID_Token, 11);
IPS_SetScriptTimer($IPS['SELF'], 5);
break;

case 11:
$Count_HEX = $Receive_Byte[11].$Receive_Byte[10].$Receive_Byte[9].$Receive_Byte[8];
$Counter = hexdec($Count_HEX)/3600;
SetValueInteger($ID_Brenner_Stunden_ACT, (int)$Counter);
SetValueInteger($ID_Token, 12);
IPS_SetScriptTimer($IPS['SELF'], 5);
}
}

// Wenn die Verbindung zu Viessmann hergestellt ist
if($Connect_Done and $Connect and ($Token == 0))
{
    //Erste Abfrage starten
    SetValueInteger($ID_Token, 1);
    IPS_SetScriptTimer($IPS['SELF'], 5);
    IPS_SetScriptTimer($SID_ViessmannSenden, 0);
}

// Timeout
if($IPS['SENDER'] == "TimerEvent")
{
    //Erhöhen des Timeout
    if($TimeoutCounter <= 2)
    {
        $TimeoutCounter += 1;
        SetValueInteger($ID_TimeOutCounter, $TimeoutCounter);
        IPS_RunScript(25080 /*[Schäferweg\Dachgeschoss\Heizungsraum\Viessmann Heizung\ViessmannSenden]*);
    }
    else
    {
        //zurücksetzen des Timeout Counters
        SetValueInteger($ID_TimeOutCounter, 0);
        //erhöhen des gesamten Fault counters
        $CommFaultCounter += 1;
        SetValueInteger($ID_CommFaultCounter, $CommFaultCounter);
        //setzen des Timeout Flag
        SetValueBoolean($ID_TimeOut, true);
        //erhöhen des Token
        if($Token <= 12)
        {
            $Token += 1;
            SetValueInteger($ID_Token, $Token);
        }
        else
        {
            // Verbindung zu Viessmann beenden
            SetValueBoolean($ID_Connect_Done, false);
        }
    }
}

//----- Main ENDE -----
//----- Funktionen -----

/**
 * Returns the calculated checksum.
 *
 * @param Byte_array $Receive_Byte Array of the received Bytes
 * @return Calculated checksum in hex
 */
function checksum($Receive_Byte)
{
    // Prüfen der Länge
    $Receive_lenght = count($Receive_Byte);

```

```

// initialisieren der Checksumme
$checksum      = 0;

// Aufaddieren der Byte's
for ($i = 2; $i < $Receive_lenght-1; $i++)
{
    $checksum    +=  hexdec($Receive_Byte[$i]);
}

// wandeln in hex und ausgeben der ersten beiden Zeichen
$checksum_hex   = substr(dechex($checksum), -2);
//if(substr($checksum_hex, 0) == 0)
//{
//    $checksum_hex   = substr(dechex($checksum), -1);
//}
return $checksum_hex;
}

function ViesmannTemp($Receive_Byte, $ID)
{
    $Temp_HEX = $Receive_Byte[9].$Receive_Byte[8];
    $Temperatur = hexdec($Temp_HEX)/10.0;
    SetValueFloat($ID, $Temperatur);
}

function ViesmannCount($Receive_Byte, $ID)
{
    $Count_HEX   = $Receive_Byte[11].$Receive_Byte[10].$Receive_Byte[9].$Receive_Byte[8];
    $Counter     = hexdec($Count_HEX);
    SetValueInteger($ID, $Counter);
}

//----- Funktionen ENDE -----
?>

```

Das sind noch keine fertigen Skripte!!

Es sind nur Beispiele wie man Sendet und Empfängt.

Das Skript läuft bei mir prima und ich würde das gerne als Basis nehmen um weiterzubauen. Das Empfangsskript triggert alle 5 sec und hört damit nicht mehr auf. Sollte das nicht stoppen nachdem alle Daten einmal abgefragt wurden?

Gruss, hcp

## Schablone

03.07.13, 19:11

Hallo hcp,

die Vissmann sendet alle 2-3 Sekunden einen Vissmann-Ping.  
 Ich bin selber noch nicht wesentlich weiter mit dem Scrip.  
 Mir fehlen noch einige Datenpunkte in der Steuerung. Siehe Post über dir.  
 Ich bin momentan auf Teneriffa. Nächste Woche bin ich wieder zu Hause.

## Martin76

20.10.13, 19:10

Hallo Schablone,

Deine Skripte für das 300er Protokoll sehen gefallen mir recht gut. Bevor ich anfangen sie für mich zu adaptieren wollte ich mal nachhören ob Du mit den Skripten weitergekommen bist,

Danke und Grüsse,

Martin

## Schablone

21.10.13, 18:44

Hallo Martin,

Zitat:

Zitat von **Martin76** 

Deine Skripte für das 300er Protokoll sehen gefallen mir recht gut.

Besten Dank.

Zitat:

Zitat von **Martin76** 

Bevor ich anfangen sie für mich zu adaptieren wollte ich mal nachhören ob Du mit den Skripten weitergekommen bist.

Nein bin ich nicht. Ich bin beruflich immer noch unter Dampf.  
 Des weiteren fehlen mir immer noch ein paar Datenpunkte (siehe oben).

Das Prinzip ist aber klar. Wenn ich nachher alles zusammen habe, möchte ich jede Minute(als Beispiel) alle Datenpunkte per "Token ring" abfragen.

Wenn nur einer meine Frage beantworten könnte!! :o

Ich finde den verdammten Datenpunkt nicht.

---

## bastianstrauss

21.10.13, 19:05

Moin zusammen,

also eine korrekte Datenpunktliste zu bekommen ist unglaublich schwierig, selbst mit einigen Kontakte direkt bei Viessmann rücken die diese Liste nicht heraus. Die einzige Chance wäre wirklich die Originalsoftware von Viessmann mit einem Optolinkadpater einzusetzen und dazwischen einen Sniffer zu stecken der das Protokoll mitplottet.

Ich habe mir bei <http://openv.wikispaces.com> jede Menge Adressen rausgesucht und per Trail & Error Werte rausgesucht. Ich habe bei uns drei Anlagen per LON verbunden wie auch ein Modem für die Fernüberwachung. Das gibt mir die Möglichkeit eine Adresse auszuprobieren und am Modem bzw. an der Viessmann HP die Werte zu kontrollieren. Ich habe versucht über ein Optolink auf alle Anlagen zuzugreifen. Scheinbart geht das aber nicht. Also gibt es bei mir 3 Optolinkadapter die über ein Lan/USB Device an meinen virtuellen Server angebunden werden. Ich nutze das Script aus dem ersten Post mit einigen Modifikationen. Kann ich gerne mal posten! Wie auch meine funktionierende XML für verschiedene Anlagen.

Gruß Bastian

---

## Schablone

21.10.13, 19:52

Hallo Bastian,

Zitat:

Zitat von **bastianstrauss** »

Ich habe mir bei <http://openv.wikispaces.com> jede Menge Adressen rausgesucht und per Trail & Error Werte rausgesucht.

Genau so mache ich das auch.

Zitat:

Zitat von **bastianstrauss** »

Wie auch meine funktionierende XML für verschiedene Anlagen.

Das wäre echt lieb von Dir! :)