

Datentransmitter

Übertragungsprotokoll

„GoCo“



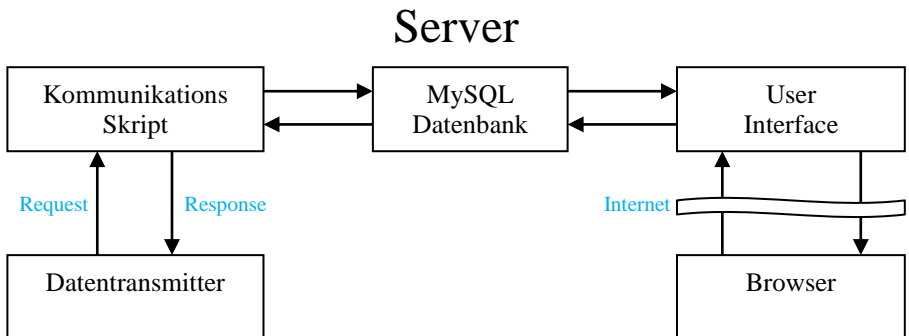
ConiuGo GmbH, Berliner Str. 4a, 16540 Hohen Neuendorf

Inhaltsverzeichnis

	Seite
1. Einführung	3
2. Kommunikation	4
2.1. Übertragungsparameter im Request	5
2.2. Modul spezifische Daten	6
2.3. Parameter in der Response	9
2.3.1. Schaltbefehle für Digital-Out Module	10
2.3.2. SMS Aufträge	11
2.3.3. Übertragungsintervall	12
2.3.4. Zeitereignisse	12
2.3.5. Schwellwertüberwachung Analog-In Pt	13
2.3.6. Schwellwertüberwachung Analog-In	15
2.3.7. Schaltzeiten der 8 Relais	16
2.3.8. Übertragungskey	17
2.3.9. Ident	18
2.3.10. Adresse	18
2.3.11. Verzeichnis/PHP Skript	19
3. Unterstützte Werte für den Parameter „action“	19
3.1. Hole Datum und Uhrzeit vom Server (001)	19
3.2. Sende Daten zum Server (002)	20
4. Kommunikations-Skript	21
5. Anmerkungen	22
Anhang	24

1. Einführung

Dieses Dokument beschreibt das Übertragungsprotokoll zwischen einem Datentransmitter, wie beispielsweise der *Go Zentrale blueline SCADA + Alarmierung* und einem Server.



Datentransmitter

Der Datentransmitter überträgt aufgenommene Messwerte sowie Zustände digitaler Eingänge zu einem Server.

Der Datentransmitter führt Aufträge vom Server, wie beispielsweise das Schalten von Relais und den Versand von Meldungs-SMS aus.

Diese Beschreibung bezieht sich auf einen Datentransmitter vom Typ **Go Zentrale blueline SCADA + Alarmierung**.



Die hier beschriebene Funktionalität wird nicht von allen Datentransmittern unterstützt!

Kommunikationsskript

Das Kommunikationsskript bearbeitet Anfragen (Request) vom Datentransmitter, speichert empfangene Daten vom Datentransmitter in die Datenbank, stellt Aufträge für den Datentransmitter zusammen und überträgt diese als Response zum Datentransmitter.

MySQL Datenbank

In die Datenbank werden empfangene Daten vom Datentransmitter gespeichert, sowie Konfigurationseinstellungen des User Interfaces gespeichert.

User Interface

Das User Interface stellt die Schnittstelle zwischen dem Benutzer und der Datenbank dar. Hier werden beispielsweise Messdaten visualisiert und auch Konfigurationseinstellungen zusammengestellt.

Browser

Über einen Internetbrowser erfolgt der Zugriff auf das User Interface.

2. Kommunikation

Die Kommunikation zwischen dem Datentransmitter und dem Server erfolgt über das **Hypertext Transfer Protokoll (http)** auf Port 80, bzw. kann auch über das **Hypertext Transfer Protokoll Secure (https)** auf Port 443 erfolgen.

Verwendung findet dabei die Request Methode „**Post**“.

Request:

ident=<ident>&device=<device>&address=<address>&key=<key>&action=<action>[&date=<date>&time=<time>][&<Modul spezifische Daten>]

Beispiel für ein Request:

```
POST /portal/dbmod0001_001_01.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: SW/com1/
Accept-Encoding: identity
Host: www.mein-server.de
Content-Length: 146
```

```
ident=1234&device=002&address=00001&key=1234567&action=002
&date=2011-08-30&time=13:37:31&di1=1:1:1:0:1:0:0:1&.....
```

Beispiel für eine Response:

```
BOF000....002....04092015....083705EOF
```

2.1. Übertragungsparameter im Request

Alle Parameter sind vom Typ char. Verwendung finden folgende Parameter:

ident

Bereich: 4 Ziffern 0000 ... 9999

Identifikation einer möglichen Gerätegruppe, beispielsweise die Identifikation für einen bestimmten Kunden.

device

Bereich: 3 Ziffern 000 ... 999

Dieser Parameter kennzeichnet den Gerätetyp.

Zur Zeit sind hierfür folgende Typen reserviert:

001	Fußscanner
002	Go Zentrale blueline SCADA + Alarmierung
003	LDA
004	GPRS Wireless Connect
005	SmartMeter Reader



Der Gerätetyp wird vom Gerät vorgegeben und kann nicht verändert werden!

address

Bereich: 5 Ziffern 00000 ... 99999

Dieser Parameter kennzeichnet die Adresse des Datentransmitter.



Die Adresse muss einmalig in einer Gerätegruppe sein!

key

Bereich: Der Übertragungskey besteht aus maximal 32 ASCII Zeichen zwischen 0 ... 9

A ... Z

a ... z

Die Verwendung eines Übertragungskey erhöht zusätzlich die Sicherheit bei der Übertragung der Daten.

action

Bereich: 3 Ziffern 000 ... 999

Über diesen Parameter teilt der Datentransmitter dem Server die auszuführende Aktion mit.

Zur Zeit finden hierfür folgende Werte Verwendung:

001 Datentransmitter fordert vom Server die aktuelle Zeit und das Datum an.

002 Datentransmitter überträgt Daten zum Server

email

Dieser Parameter ist für die Registrierung des Datentransmitters bei EasySCADA notwendig.

Beispiel: „mein_name@mail_server.de“

date

Das Datum hat folgenden Aufbau: YYYY-MM-DD

Beispiel: 2012-04-27

time

Die Uhrzeit hat folgenden Aufbau: hh:mm:ss

Beispiel: 13:07:54

data

Dieser Parameter ist für Datentransmitters vom Device Typ 005 notwendig.

Der Aufbau wird in diesem Dokument nicht weiter ausgeführt.

2.2. Modul spezifische Daten

In Abhängigkeit vom Modultyp ist der Aufbau wie folgt:

<Modultyp><Modulnr. 1..10>=

Beispiel: di1=

Modul Digital-In

Dieses Modul hat 8 Eingänge. Der Zustand ist immer „0“ für „aus (offen)“ und „1“ für „ein (auf GND)“.

di[1..10]=<Zustand Eingang 1>:<Zustand Eingang 2>: ... :<Zustand Eingang 8>
Ein Zustandswechsel von „0“ nach „1“ führt ggf. zu einer Alarmierung – z.B. über SMS.

Beispiel:

di1=1:0:0:1:0:1:0:1&di2=0:0:1:1:1:0:0:0

Modul Digital-In inv

Dieses Modul hat 8 Eingänge. Der Zustand ist immer „0“ für „aus (offen)“ und „1“ für „ein (auf GND)“.

dv[1..10]=<Zustand Eingang 1>:<Zustand Eingang 2>: ... :<Zustand Eingang 8>
Ein Zustandswechsel von „1“ nach „0“ führt ggf. zu einer Alarmierung – z.B. über SMS.

Beispiel:

dv1=1:0:0:1:0:1:0:1&dv2=0:0:1:1:1:0:0:0

Modul Analog-In

Dieses Modul hat 4 Eingänge. Der Wert wird immer in Digits angegeben und liegt zwischen 0 und 1023 (10 bit Wandler).

ai[1..10]=<Wert Eingang 1>:< Wert Eingang 2>: ... :< Wert Eingang 4>

Beispiel:

ai1=100:2395:8002:12&ai2=200:1234:195:20

Modul Analog-In Pt

Dieses Modul hat 4 Eingänge. Der Wert wird immer in 1/10 °C angegeben.

ap[1..10]=<Wert Eingang 1>:< Wert Eingang 2>: ... :< Wert Eingang 4>

Beispiel:

ap1=100:239:-342:12&ai2=200:234:-195:20
(entspricht 10,0 °C, 23,9 °C, -34,2 °C, 1,2 °C,)

Modul M-Bus Zähler

An dieses Modul können max. 8 M-Bus Zähler angeschlossen werden.



Stellen Sie bitte sicher, dass der von Ihnen verwendete Zähler auch vom Modul unterstützt wird!

Der Wertebereich beträgt 30 bit, also von 0 bis 1073741824. Unabhängig davon, ob die max. Anzahl von Zählern angeschlossen ist, werden immer 8 Werte übertragen.

`mc[1..10]=<Wert Zähler 1>:< Wert Zähler 2>: ... :< Wert Zähler 8>`

Beispiel:

`mc1=100:2345329:1322342:112:0:123456789:34:2`

Modul Digital-Out

Dieses Modul kann sowohl 2 (230V Variante) als auch 4 Ausgänge (Relais) haben. Es wird der Status der Relais als „0“ = nicht angezogen bzw. „1“ = angezogen übertragen. Unabhängig davon ob 2-oder 4 Relais wird immer der Status von 4 Relais übertragen, wobei bei der 230V Variante die Relais 3 und 4 keine Funktion haben.

`do[1..10]=<Status Relais 1>:<Status Relais 2>: ... :<Status Relais 4>`

Beispiel:

`do1=1:0:0:0&do2=1:1:0:1`



Der Status der Relais wird nur dann übertragen, wenn vom Datentransmitter aus selbsttätig Schaltvorgänge aktiviert werden (Brennzeitkalender)!

Beispiel:

Eine Anlage besteht aus einem Digital-In Modul, sowie 3 Digital-Out Modulen. Ein Brennzeitkalender ist aktiv.

Der Request würde wie folgt aussehen:

`di1=1:0:0:0:1:0:0:1&do1=1:0:0:1&do2=0:0:1:0`

Der Zustand der Relais vom 3. Digital-Out-Modul würde nicht übertragen werden, da nur do1 und do2 zum Brennzeitkalender gehört.

Modul Optoport (optischer Zähler)

An dieses Modul können max. 8 Impulsgeber angeschlossen werden.

Der Wertebereich beträgt 30 bit, also von 0 bis 1073741824. Unabhängig davon, ob die max. Anzahl von Zählern angeschlossen ist, werden immer 8 Werte übertragen.

op[1..10]=<Wert Zähler 1>:< Wert Zähler 2>: ... :< Wert Zähler 8>

Beispiel:

op1=100:2345329:1322342:112:0:123456789:34:2

Eine Sonderart des Optoport ist das SmartMeter Reader (Device 005), deren Beschreibung in dieser Doku nicht weiter ausgeführt wird.

2.3. Parameter in der Response

Alle Parameter sind vom Typ char.

In Abhängigkeit des „Request“ vom Datentransmitter sendet der Server folgende Response:

BOF

<Return Code>

....<Action>

[....<Liste mit den Schaltbefehlen für Digital-Out Module>]

[...s;<SMS Aufträge>;]

[...i;<Übertragungsintervall>;]

[...z;<Liste der Zeitereignisse>;]

[...p;<Schwellwertüberwachung für die Analog-In Pt Module>;]

[...a;<Schwellwertüberwachung für die Analog-In Module>;]

[...b;<Schaltzeiten für die 8 Relais des Brennzeitkalenders>;]

[...k;<Übertragungskey>;]

[...d;<Ident>;]

[...g;<Address>;]

[...h;<Verzeichnis und PHP Scriptname>;]

....<Datum>

....<Uhrzeit>

EOF

Darin bedeutet:

BOF	Anfang (Begin Of File)
Return Code	Return Code
000	OK
001	Error: Action unbekannt
002	Error: Ident unbekannt
003	Error: Device unbekannt
004	Error: Adresse unbekannt
005	Error: Syntax
006	Error: Gerät inaktiv
007	Error: Key unbekannt
008	OK, aber Datensatz bereits in der Datenbank vorhanden
100	OK
108	OK, aber Datensatz bereits in der Datenbank vorhanden
999	Error: Sammelfehler

Anmerkung: Die Return Codes 1xx werden immer dann gesendet, wenn in der Response auch Schaltbefehle für Digital-Out Module vorhanden sind !

Action	Action
001	Datum und Uhrzeit vom Server holen
002	Daten zum Server übertragen
Datum	Datum im Format TTMMJJJJ, z.B. 27052013 (27.05.2013)
Uhrzeit	Uhrzeit im Format hhmmss, z.B. 130537 (13:05:37)
EOF	Ende (End Of File)

2.3.1. Schaltbefehle für Digital-Out Module

Schaltzustand:

0 Relais aus

1 Relais ein

....

<Schaltzustand Modul 1, Ausgang 1>:<Schaltzustand Modul 1, Ausgang 2>:

<Schaltzustand Modul 1, Ausgang 3>:<Schaltzustand Modul 1, Ausgang 4>:

<Schaltzustand Modul 2, Ausgang 1>:<Schaltzustand Modul 2, Ausgang 2>:

<Schaltzustand Modul 2, Ausgang 3>:<Schaltzustand Modul 2, Ausgang 4>:
bis

<Schaltzustand Modul 10, Ausgang 1>:<Schaltzustand Modul 10, Ausgang 2>:

<Schaltzustand Modul 10, Ausgang 3>:<Schaltzustand Modul 10, Ausgang 4>

Der Schaltbefehl wird unmittelbar nach Empfang im Datentransmitter ausgeführt.

Beispiele:

Datentransmitter mit einem Digital-Out Modul

....1:0:1:1

Datentransmitter mit zwei Digital-Out Modulen

....1:0:1:1:0:0:1:0

2.3.2. SMS Aufträge

Der SMS Text kann eine Länge von maximal 160 Zeichen haben.

....s;

<Zielrufnr. 1>;<SMS Text 1>;

<Zielrufnr. 2>;<SMS Text 2>;

bis

<Zielrufnr. 10>;<SMS Text 10>;

Besteht ein SMS Text aus ### so wird keine SMS versendet. Es wird in diesem Fall ein Sprachruf ausgeführt. Es erfolgt aber keine Sprachnachricht o.ä.



Die SMS Aufträge werden unmittelbar nach Empfang vom Datentransmitter ausgeführt!

Beispiele:

Ein SMS Auftrag

....s;015712345678;Dies ist ein Test;

Zwei SMS Aufträge

....s;+4917212345678;Dies ist ein Test;016833333;Test 2;

Ein Auftrag für einen Sprachruf
....s;015712345678;###;

2.3.3. Übertragungsintervall

....i;<Intervall 0..86400 Sekunden>;



Dieser Parameter überschreibt den internen, gespeicherten Wert im Datentransmitter!

Beispiel: Übertragungsintervall 5 Minuten
....i;300;

2.3.4. Zeitereignisse

....Z;
JJJJMMTThhmmss;
JJJJMMTThhmmss;
bis zu 10 Zeitereignisse

Dabei bedeutet

JJJJ	Jahr
MM	Monat
TT	Tag
hh	Stunde
mm	Minute
ss	Sekunde

Jede Ziffer in einem Zeitereignis kann durch ein **X** ersetzt werden. Dieses steht dann für „wird nicht berücksichtigt“. Tritt ein Zeitereignis ein, so verbindet sich der Daten-

transmitter mit dem Server und überträgt die Daten und Zustände der eingesetzten Module.

Mitz;Reset; werden alle im Datentransmitter gespeicherten Zeitereigniss gelöscht.



Dieser Parameter wird im nicht flüchtigen Speicher des Datentransmitters gespeichert und entsprechend ausgeführt!

Beispiele:

Im Jahr 2015 soll immer am 1. Tag eines Monats um 18:23:00 ein Ereignis aktiviert werden.

....z;2015XX01182300;

Es soll alle 5 Minuten ein Ereignis aktiviert werden.

....z;XXXXXXXXXXXX000;XXXXXXXXXXXX500;

2.3.5. Schwellwertüberwachung Analog-In Pt

Die anzugebenen Werte sind in 1/10 °C.



Die Schwellwertüberwachung für einen bestimmten Eingang ist deaktiviert wenn die zugehörigen Werte auf 0 gesetzt sind!

....p;

<Hysterese Modul 1, Eingang 1>;<Oberer Schwellwert Modul 1, Eingang 1>;

<Unterer Schwellwert Modul 1, Eingang 1>;

<Hysterese Modul 1, Eingang 2>;<Oberer Schwellwert Modul 1, Eingang 2>;

<Unterer Schwellwert Modul 1, Eingang 2>;

<Hysterese Modul 1, Eingang 3>;<Oberer Schwellwert Modul 1, Eingang 3>;

<Unterer Schwellwert Modul 1, Eingang 3>;

<Hysterese Modul 1, Eingang 4>;<Oberer Schwellwert Modul 1, Eingang 4>;

<Unterer Schwellwert Modul 1, Eingang 4>;

bis

<Hysterese Modul 10, Eingang 1>;<Oberer Schwellwert Modul 10, Eingang 1>;

<Unterer Schwellwert Modul 10, Eingang 1>;
 <Hysterese Modul 10, Eingang 2>;<Oberer Schwellwert Modul 10, Eingang 2>;
 <Unterer Schwellwert Modul 10, Eingang 2>;
 <Hysterese Modul 10, Eingang 3>;<Oberer Schwellwert Modul 10, Eingang 3>;
 <Unterer Schwellwert Modul 10, Eingang 3>;
 <Hysterese Modul 10, Eingang 4>;<Oberer Schwellwert Modul 10, Eingang 4>;
 <Unterer Schwellwert Modul 10, Eingang 4>;

Mitp;Reset; werden alle im Datentransmitter gespeicherten Werte für die Schwellwertüberwachung gelöscht.



Dieser Parameter wird im nicht flüchtigen Speicher des Datentransmitters gespeichert und entsprechend ausgeführt!

Wird ein Schwellwert über-bzw. unterschritten, so verbindet sich der Datentransmitter mit dem Server und überträgt die Daten und Zustände der eingesetzten Module.

Beispiele:

Datentransmitter mit einem Analog-In Pt Modul

....p;10;300;100;0;0;5;50;-40;0;0;0;

Eingang 1: Hysterese: 1,0 °C, oberere Schwelle: 30,0 °C, untere Schwelle: 10,0 °C

Eingang 2: Deaktivierte Schwellwertüberwachung

Eingang 3: Hysterese: 0,5 °C, obere Schwelle: 5,0 °C, untere Schwelle: -4,0 °C

Eingang 2: Deaktivierte Schwellwertüberwachung

Datentransmitter mit zwei Analog-In Pt Modulen

....p;10;300;100;0;0;5;50;-40;0;0;0;0;0;4;10;-100;0;0;0;0;0;0;

Modul 1:

Eingang 1: Hysterese: 1,0 °C, oberere Schwelle: 30,0 °C, untere Schwelle: 10,0 °C

Eingang 2: Deaktivierte Schwellwertüberwachung

Eingang 3: Hysterese: 0,5 °C, obere Schwelle: 5,0 °C, untere Schwelle: -4,0 °C

Eingang 4: Deaktivierte Schwellwertüberwachung

Modul 2:

Eingang 1: Deaktivierte Schwellwertüberwachung

Eingang 2: Hysterese: 0,4 °C, oberere Schwelle: 1,0 °C, untere Schwelle: -10,0 °C

Eingang 3: Deaktivierte Schwellwertüberwachung

Eingang 4: Deaktivierte Schwellwertüberwachung

2.3.6. Schwellwertüberwachung Analog-In

Die anzugebenen Werte sind in Digit. Das Analog-In Modul besitzt einen 10 bit Analog/Digital-Wandler. Der Bereich liegt somit zwischen 0 und 1023.



Die Schwellwertüberwachung für einen bestimmten Eingang ist deaktiviert wenn die zugehörigen Werte auf 0 gesetzt sind!

....a;

<Hysterese Modul 1, Eingang 1>;<Oberer Schwellwert Modul 1, Eingang 1>;

<Unterer Schwellwert Modul 1, Eingang 1>;

<Hysterese Modul 1, Eingang 2>;<Oberer Schwellwert Modul 1, Eingang 2>;

<Unterer Schwellwert Modul 1, Eingang 2>;

<Hysterese Modul 1, Eingang 3>;<Oberer Schwellwert Modul 1, Eingang 3>;

<Unterer Schwellwert Modul 1, Eingang 3>;

<Hysterese Modul 1, Eingang 4>;<Oberer Schwellwert Modul 1, Eingang 4>;

<Unterer Schwellwert Modul 1, Eingang 4>;

bis

<Hysterese Modul 10, Eingang 1>;<Oberer Schwellwert Modul 10, Eingang 1>;

<Unterer Schwellwert Modul 10, Eingang 1>;

<Hysterese Modul 10, Eingang 2>;<Oberer Schwellwert Modul 10, Eingang 2>;

<Unterer Schwellwert Modul 10, Eingang 2>;

<Hysterese Modul 10, Eingang 3>;<Oberer Schwellwert Modul 10, Eingang 3>;

<Unterer Schwellwert Modul 10, Eingang 3>;

<Hysterese Modul 10, Eingang 4>;<Oberer Schwellwert Modul 10, Eingang 4>;

<Unterer Schwellwert Modul 10, Eingang 4>;

Mita;Reset; werden alle im Datentransmitter gespeicherten Werte für die Schwellwertüberwachung gelöscht.



Dieser Parameter wird im nicht flüchtigen Speicher des Datentransmitters gespeichert und entsprechend ausgeführt!

Wird ein Schwellwert über-bzw. unterschritten, so verbindet sich der Datentransmitter mit dem Server und überträgt die Daten und Zustände der eingesetzten Module.

Beispiele:

Datentransmitter mit einem Analog-In Modul

....a;10;300;100;0;0;5;1010;50;0;0;0;

Eingang 1: Hysterese: 10, oberere Schwelle: 300, untere Schwelle: 100

Eingang 2: Deaktivierte Schwellwertüberwachung

Eingang 3: Hysterese: 5, obere Schwelle: 1010, untere Schwelle: 50

Eingang 4: Deaktivierte Schwellwertüberwachung

Datentransmitter mit zwei Analog-In Modulen

....a;10;300;100;0;0;5;580;40;0;0;0; 0;0;0;4;190;120;0;0;0;0;0;

Modul 1:

Eingang 1: Hysterese: 10, oberere Schwelle: 300, untere Schwelle: 100

Eingang 2: Deaktivierte Schwellwertüberwachung

Eingang 3: Hysterese: 5, obere Schwelle: 580, untere Schwelle: 40

Eingang 4: Deaktivierte Schwellwertüberwachung

Modul 2:

Eingang 1: Deaktivierte Schwellwertüberwachung

Eingang 2: Hysterese: 4, oberere Schwelle: 190, untere Schwelle: 120

Eingang 3: Deaktivierte Schwellwertüberwachung

Eingang 4: Deaktivierte Schwellwertüberwachung

2.3.7. Schaltzeiten der 8 Relais

Mit diesem Parameter werden die Schaltzeiten der ersten 8 Relais des Brennzeitkallenders übertragen.

....b;

<Datum JJJJ-MM-TT Vortag>;<Schaltzeit „Aus“ Vortag>;<Schaltzeit „Ein“ Vortag>;

<Zustand „Ein“ der Relais Vortag – Relais 8 ... Relais 1>;

<Datum JJJJ-MM-TT>;<Schaltzeit „Aus“>;<Schaltzeit „Ein“>;

<Zustand „Ein“ der Relais – Relais 8 ... Relais 1>;

Beispiele:

....b;2016-05-02;16:20;12:00;10001010;2016-05-03;17:00;05:40;00111100;

Vortag ist der 02.05.2016

Ausschaltzeit des Vortages ist 16:20

Einschaltzeit des Vortages ist 12:00

Zustand der Relais, eingeschaltet ist Relais 8 = 1 Relais 1 = 0

Aktueller Tag ist der 03.05.2016

Ausschaltzeit ist 17:00

Einschaltzeit ist 05:40

Zustand der Relais, eingeschaltet ist Relais 8 = 0 Relais 1 = 0

Der 03.05.2016 beginnt mit ausgeschalteten Relais.

Um 5:40 werden die 8 Relais mit 00111100 eingeschaltet

Um 17:00 werden alle Relais ausgeschaltet

....b;2016-05-02;16:20;20:10;10001011;2016-05-03;12:31;14:20;10110100;

Vortag ist der 02.05.2016

Ausschaltzeit des Vortages ist 16:20

Einschaltzeit des Vortages ist 20:10

Zustand der Relais, eingeschaltet ist Relais 8 = 1 Relais 1 = 1

Aktueller Tag ist der 03.05.2016

Ausschaltzeit ist 12:31

Einschaltzeit ist 14:20

Zustand der Relais, eingeschaltet ist Relais 8 = 1 Relais 1 = 0

Der 03.05.2016 beginnt mit eingeschalteten Relais mit 10001011

Um 12:31 werden alle 8 Relais ausgeschaltet

Um 14:20 werden die Relais mit 10110100 eingeschaltet

2.3.8. Übertragungskey

....k;<Max. 32 Zeichen (0..9, A..Z, a..z)>;



Dieser Parameter überschreibt den internen, gespeicherten Key im Datentransmitter!

Dieser Parameter ist notwendig für die Registrierung des Datentransmitters bei EasySCADA.

Beispiel:

....k;aH7kL942bbMFk845;

2.3.9. Ident

....d;<Ident (0000..9998)>;



Dieser Parameter überschreibt den internen, gespeicherten Ident im Datentransmitter!

Dieser Parameter ist notwendig für die Registrierung des Datentransmitters bei EasySCADA.

Beispiel:

....d;0024;

2.3.10. Adresse

....g;<Adresse (00001..99999)>;



Dieser Parameter überschreibt die interne, gespeicherte Adresse im Datentransmitter!

Dieser Parameter ist notwendig für die Registrierung des Datentransmitters bei EasySCADA.

Beispiel:

....g;00002;

2.3.11. Verzeichnis/PHP Skript

....h;<Verzeichnis/PHP Skript>;



Dieser Parameter überschreibt den internen, gespeicherten Eintrag im Datentransmitter!

In diesem Verzeichnis befindet sich das PHP Skript, welches vom Datentransmitter aufgerufen wird. Die Angabe dafür ist ohne .php

Dieser Parameter ist notwendig für die Registrierung des Datentransmitters bei EasySCADA.

Beispiel: Das Skript mit dem Namen „dbmod0056_002_01.php“ befindet sich auf dem Server unter dem Verzeichnis „<ihr server>/skripte>“.

....h;skripte/dbmod0056_002_01;

3. Unterstützte Werte für den Parameter „action“

Über den Parameter “action” können verschiedene Anfragen an den Server gesendet werden.

Zur Zeit gibt es für „action“ folgende Definitionen:

- 001 Datum und Uhrzeit vom Server holen
- 002 Daten zum Server übertragen

3.1. Hole Datum und Uhrzeit vom Server (001)

Request

ident=<ident>&device=<device>&address=<address>&key=<key>&action=001

Beispiel:

```
POST /portal/dbmod0001_001_01.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: SW/com1/
Accept-Encoding: identity
Host: www.mein-server.de
Content-Length: ...
```

```
ident=1234&device=002&address=00001&key=1234567&action=001
```

Response

```
BOF<Return Code>....001....<Datum>....<Uhrzeit>EOF
```

Die Response kann ebenso auch Parameter für die Konfiguration enthalten. Siehe dazu Kapitel 2.3.

Beispiel:

```
BOF000....001....04092015....083705EOF
```

Es ist der 04.09.2015 um 08:37:05

3.2. Sende Daten zum Server (002)

Request

```
ident=<ident>&device=<device>&address=<address>&key=<key>&action=002
[&date=<date>&time=<time>][&<Modul spezifische Daten>]
```

Beispiel:

```
POST /portal/dbmod0001_001_01.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: SW/com1/
Accept-Encoding: identity
Host: www.mein-server.de
Content-Length: 146
```

```
ident=1234&device=002&address=00001&key=1234567&action=002
&date=2011-08-30&time=13:37:31&di1=1:1:1:0:1:0:0:1&.....
```

Response

```
BOF
```

```
<Return Code>
```

```
....<Action>
```

```
[....<Liste mit den Schaltbefehlen für Digital-Out Module>]
```

```
[...s;<SMS Aufträge>;]
[...i;<Übertragungsintervall>;]
[...z;<Liste der Zeitereignisse>;]
[...p;<Schwellwertüberwachung für die Analog-In Pt Module>;]
[...a;<Schwellwertüberwachung für die Analog-In Module>;]
[...b;<Schaltzeiten für die 8 Relais des Brennzeitkalenders>;]
[...k;<Übertragungskey>;]
[...d;<Ident>;]
[...g;<Address>;]
[...h;<Verzeichnis und PHP Scriptname>;]
....<Datum>
....<Uhrzeit>
EOF
```

Beispiel für eine Response:

```
BOF100...002...1:0:0:0:1:1:0:1...s;01453209;Alarm;...04092015...083
705EOF
```

4. Kommunikations-Skript

Über das Kommunikations-Skript kommuniziert der Datentransmitter mit dem Server.

Um die Arbeitsweise zu verdeutlichen, hier Auszüge aus einem einfachen Beispiel:

```
<?php

//Datenbank Login
include("../inc/config.php");

//Übergebene Parameter
$ident = $_REQUEST['ident'];
$device = $_REQUEST['device'];
$address = $_REQUEST['address'];
$action = $_REQUEST['action'];
$date_in = $_REQUEST['date'];
$time_in = $_REQUEST['time'];
$key_in = $_REQUEST['key'];
```

```
//DigIn - Modul
$dil = $_REQUEST['dil'];

//AnalogIn - Modul
$a1l = $_REQUEST['a1l'];

//AnalogIn PT - Modul
$ap1 = $_REQUEST['ap1'];

// .. mit Datenbank verbinden
$return = mysqli_connect (DBHOST,DBUSER,DBPASS,DBASE);

//Hier folgt nun der Vergleich der eingelesenen Parameter mit
den Datenbankwerten

//Wenn alles OK ist, werden die Messwerte in die Datenbank
eingetragen

//hier wird die Antwort an den Datentransmitter ausgegeben
$response = "BOF000....001....".$date."....".$time."EOF";
echo $response;

mysqli_close ($return); //DB schliessen

?>
```

5. Anmerkungen

Nachdem Einschalten des Datentransmitter, registrieren beim Provider und aktivieren eines Datenservices, wie beispielsweise GPRS holt der Datentransmitter im 1. Connect mit dem Server, das Datum und die Uhrzeit und stellt seine interne Systemzeit. Es wird empfohlen, in diesem Ablauf die Konfigurationseinstellungen, wie Übertragungsintervall, Zeitereignisse, Schwellwertüberwachung in der Response zurückzuliefern. Um diese Funktionalität zu gewährleisten, sollte das Kommunikations-Skript entsprechend programmiert sein.

Anhang

Version	Änderung	Datum	Name
02.07	SMS Auftrag um „###“ erweitert Diverse Anpassungen, ein Device 005 hinzugefügt Kleinere Korrekturen	16.11.2020	Skrebba
02.06	Modul „Optoport“ hinzugefügt. Für die Registrierungsprozedur wur- den die Parameter k;d;g;h; in der Response hinzugefügt. Ebefalls für die Registrierung der Parameter „email“	05.03.2018	Skrebba
02.05	Inhaltlich mit weiteren Beispielen aufbereitet	13.12.2016	Skrebba
02.04	Go Wireless Connect umbenannt in Go Zentrale blueline SCADA + Alar- mierung Digital-In invers aufgenommen	16.11.2016	Skrebba
02.03	Änderung der Statusübertragung im Request des Datentransmitters. Schaltzeiten werden im Request mit übertragen.	03.05.2016	Skrebba
02.02	Status der Digital-Out Module wird nun im Request mit übertragen.	21.04.2016	Skrebba
02.01	Korrekturen	30.03.2016	Skrebba
02.00	Dieses Protokoll wird ab „Go Wirel- ess Connect“, V 02.04 unterstützt.	07.09.2015	Skrebba