L&G FUGGA PRV

Protokol L&G PRV Fugga

Podporované typy a verzie zariadení
Konfigurácia komunikanej linky
Konfigurácia komunikanej stanice
Konfigurácia meraných bodov
Literatúra
Zmeny a úpravy
Revízie dokumentu

Podporované typy a verzie zariadení

Komunikácia podporuje ítanie/zápis údajov do zariadení Landis&Gyr PRV 1.xx a 2.xx.

Konfigurácia komunikanej linky

- Kategória komunikanej linky: Serial.
- Prenosová rýchlos a prenosové parametre poda nastavenia parametrov komunikaného programu v PRV.

Konfigurácia komunikanej stanice

- Komunikaný protokol: **L&G FUGGA PRV**.
- Adresa stanice je v rozsahu 0 až 255 a zadáva sa v decimálnom tvare poda nastavenia ísla stanice v komunikanom programe v PRV (vi alej).

Parametre protokolu stanice

Môžu by zadané nasledovné parametre protokolu stanice:

Tab. . 2

Kúové slovo	Plný názov	Popis	Jednotka	Náhradná hodnota
MR	Max Retry	Poet opakovaní výzvy v prípade chyby komunikácie.	-	3
MWR	Max Wait Retry	Poet opakovaní ítania odpovede do jej skompletovania.	-	14
RT	Retry Timeout	Oneskorenie medzi opakovaním výzvy v prípade chyby komunikácie.	s.ms	1.000 s
WT	Wait Timeout	Oneskorenie medzi ítaniami odpovede do jej skompletovania.	ms	300 milisek.
WFT	Wait First Timeout	Prvé akanie po odvysielani výzvy pred ítaním odpovede.	ms	1200 milisek.
RD	Request Delay	Oneskorenie pred vysielaním výzvy.	ms	1000 milisek.
MBS	Maximum Buffer Size	Maximálna vekos výzvy v bytoch.	byte	60
DOFF	Date Offset	Korekcia posunu dátumu v interpetácií PRV a systému D2000. Systém D2000 interpretuje dátum ako poet dní od 1-1-1972. Pre testované PRV 1.64 (SW ver. 4.04) bolo nutné nastavi hodnotu 20089, pretože PRV implementuje dátum od 1-1-1917.	dni	4806

String s parametrami protokolu sa zapisuje poda pravidiel:

```
Kúové_slovo=hodnota;Kúové_slovo=hodnota; ...
```

Príklad:

```
MR=4;RD=500;
```

Ak nebolo v inicializanom stringu nájdené kúové slovo s platnou hodnotou, použije sa náhradná hodnota poda tabuky.

Konfigurácia meraných bodov

Možné typy bodov: **AI, AO, CI, CO, DI, DO, TIR, TOR, TIA, TOA.**

Konfigurácia vyžaduje adresu v textovom tvare. Oproti implementácií v D2000 ver.3.xx nie je rozsah typov nijako obmedzený. Formu zápisu niektorých typov znázoruje tabuka:

Typ v D2000 v.3.xx	Zápis v D2000 v.4.x	Príklad
I/O	$xxx.name	$120.HW
PLT	PLTx.name	PLT5.PAR1
ZON	ZONx.name	
RGB	RGBx.name	
VIP	VIPx.name	
CVP	CVPx.name	
@MGR	@MGRx.name	
@MGG	@MGGx.name	
@SEZ	@SEZx.name	
@AI	@AIx.name	
@AO	@AOx.name	
@DI	@DIx.name	
@DO	@DOx.name	
@CI	@CIx.name	
@PHON	@PHONx.name	
@RING	@RINGx.name	
TTY	TTYx.name	
@PBUS	@PBUSx.name	
@MBUS	@MBUSx.name	
@OS	@OSx.name	
@BLN	@BLNx.name	
@BPS	@BPSx.name	
TSK	[tsknr]name	[10]S2
SYS	name	DATE

Protokol PRV (Fugga)

Formát správy:

<STX><ADRESA><FUNKCIA><SEQ><OBSAH SPRAVY><ETX/ETB><CRC>

STX - Start of text, znak 0x02

ADRESA - dvojíslie v ASCII tvare - rozsah 01 - 99.

FUNKCIA - dvojíslie v ASCII tvare - íslo funkcie. V odpovedi sa k íslu funkcie pripoítava 50.

SEQ – poradové íslo správy - jeden bajt - hodnoty 0x41 – 0x5A

OBSAH SPRAVY - sprava v ASCII tvare - jednotlivé asti správy sú oddelené znakom “;”.

ETX - koniec správy

ETB - koniec bloku

CRC - kontrolná suma CRC-CCITT Cyclic Redundancy Check according to IBM

- Smerom do PRV - 6 bajtov ASCII
- Smerom z PRV 2 bajty binárne

Potvrdenie:

ACK - O.K. - obsah správy znak ACK

NAK - chyba - obsah správy znak NAK

Funkcie:

Funkcia	číslo	Obsah správy	Odpove
Config Check	01	Config ID	ACK/NAK
New Point Def	02	ID:NAME;0:ConfigId;	ACK/NAK(memory)
Delete Point	03	ID;ConfigId;	ACK
Read All	04	---	ID:VALUE;...
ReadAllContinue	05	INDEX	ID:VALUE;...
Set Output	06	ID=Value;	ACK
Reset PRV	07	---	ACK
Clear Config	08	---	ACK

Riešenie sporných situácií

Na strane D2000:

1. PRV neodpovedá v stanovenom limite

2. Zlá kontrolná suma

Správa sa opakuje N-krát - všetky potrebné parametre sa nastavujú parametrami protokolu stanice.

Na strane PRV:

Zlá kontrolná suma - PRV neodpovedá.

Zmeny komunikaného protokolu PRV (Fugga) ver 1.02

Urenie: Komunikácia s PRV1/PRV2 cez sériové linky a rádiomodem s OnLine konfiguráciou prenášaných hodnôt.

Rozšírenie oproti ver. 1.01:

- implementované asové body
- optimalizácia komunikácie - prenášajú sa len zmeny hodnôt

Poet prenášaných bodov z jednej stanice je max. 255.

Parametre komunikácie nastavované na strane PRV:

Task 3:

číslo komunikaného portu v PRV: premenná PRT <1..3>

Adresa stanice: Premenná AD$ - adresa sa skladá vždy z dvoch ísel napr. “01” - adresa 1.

číslo komunikaného tasku - default 3. Ak ho treba zmeni treba upravi tasky 254 a 253 - reštart po výpadku napájania a po WatchDog Error. Nutne upravi task 4 - odkazy na lokálne premenné tasku 3.

číslo Watch Dogu - default je 7. Ak ho treba zmeni treba upravi task 253 a zada nové číslo Watch Dogu vo volaní funkcie WADO na riadku 305.

Maximálna vekos vysielanej správy do PC - premenná BLK - nesmie by viac ako 160.

Prenosová rýchlos: Parameter TTYPRT.BD na riadku 60.

číslo tasku pre zber hodnôt: Použije sa task číslo 4. Ak ho treba zmeni - musí sa upravi číslo tasku na riadku 48 - štart tasku 4.

Task 4:

Parameter TSKFRM$ - uruje výstupný formát tasku. Definuje vlastne presnos prenášaných floatingových ísel.

Parameter WAIT DURING 5 , na riadku 70 uruje periodu zberu hodnôt v sekundách.

Výpis taskov pre ver. 1.02

```
.TSK 3
```

```
1 -- FUGGA PROTOCOL (C) IPESOFT 1997 V 1.02
```

```
5 ad$ := "21"; -- adresa stanice
```

```
10 er := 1 ; cf$ ="";RT$ := "";mx:=100
```

```
15 sl := 2; bl = 60; cn := 1; wx := 5;
```

```
20 DIM N$(mx); DIM V$(mx); DIM NW(mx);
```

```
25 FOR I := 1 TO mx
```

```
26 V$(I) := ""; NW(I) := 0;
```

```
27 NEXT I;
```

```
48 STOP 4; RUN 4;
```

```
40 TSKPRP:=3
```

```
50 TSKTTY:=1
```

```
55 TSKPRIO:=15
```

```
56 SYSDIAG:=0;
```

```
58 ON ERROR GOTO 7000
```

```
60 TTY1.BD=19200
```

```
61 -- TTY1.PARCHK = 0
```

```
62 TTY1.SSB = 0
```

```
70 TTY1.NOEC=1
```

```
72 TTY1.NCON=1
```

```
74 TTY1.MOD=0
```

```
76 TTY1.NWCR=1
```

```
78 TTY1.NFCR=1
```

```
79 TTY1.NOX=1;
```

```
80 INITTY1
```

```
90 BEGIN_INKEY
```

```
100 cm$ := "" ;C$ := ""; ax$ := "";fn$ := "";CR$ := "";WT:=0
```

```
200 ST := 1; fc := 0;
```

```
300 INKEY A$
```

```
305 WADO(7,60) ; -- 7 je cislo Watch Dogu
```

```
310 IF A$="" THEN WT := WT + 1 ELSE WT:= 0;
```

```
320 IF WT > wx THEN GOTO 100
```

```
330 IF WT > 0 THEN WAIT DURING 1; GOTO 300
```

```
400 FOR RI:= 1 TO LEN(A$)
```

```
450 C$ := A$[RI..RI]
```

```
500 ON ST GOSUB 600,800,1000,1200,1300,950,3000
```

```
540 cm$ := cm$ + C$
```

```
550 NEXT RI
```

```
560 GOTO 300
```

```
600 IF C$ = CHR$(2) THEN ST:= 2; ax$:="" ELSE ST:= 1
```

```
610 IF C$ = CHR$(24) THEN ST:=7;
```

```
620 cm$ := ""
```

```
700 RETURN
```

```
800 ax$ := ax$ + C$
```

```
820 IF LEN(ax$) < 2 THEN RETURN;
```

```
830 IF ax$ = ad$ THEN fn$:= "";ST := 3 ELSE ST := 1;
```

```
900 RETURN
```

```
950 RT$ := C$; ST := 4;
```

```
960 RETURN
```

```
1000 fn$ := fn$ + C$
```

```
1020 IF LEN(fn$) < 2 THEN RETURN
```

```
1025 fc := VAL(fn$); ST := 6;
```

```
1030 fn$ := STR$(fc+50);
```

```
1040 RETURN
```

```
1200 IF (C$ = CHR$(3)) OR (C$ = CHR$(23)) THEN ST := 5; CR$ := ""
```

```
1210 RETURN
```

```
1300 CR$ := CR$ + C$ ; C$ := "";C4$ := "";
```

```
1320 IF LEN(CR$) = 6 THEN GOTO 1400;
```

```
1340 RETURN
```

```
1400 C1 := VAL(CR$[1..3]); C2 := VAL(CR$[4..6]);C1$ := "";
```

```
1410 C1$ := CHR$(C1) + CHR$(C2);
```

```
1415 C4$ := CRC$(4,cm$)
```

```
1420 IF C1$ = C4$ THEN GOTO 1500 ELSE ST := 1;
```

```
1430 RETURN
```

```
1500 ON fc GOSUB 2100,2200,2300,2400,2500,2600,2700,2800
```

```
1505 ST := 1;cm$ := ""
```

```
1510 RETURN
```

```
2100 ET := INSTR(7,cm$,";");
```

```
2110 FOR I := 1 TO mx
```

```
2115 IF N$(I) <> "" THEN NW(I) := 1;
```

```
2117 NEXT I;
```

```
2120 IF cm$[7..ET-1] = cf$ THEN GOTO 2125 ELSE 2130
```

```
2125 SB(CHR$(6),3,ad$,fn$,RT$); RETURN
```

```
2130 SB(CHR$(21),3,ad$,fn$,RT$); RETURN
```

```
2200 S := 7
```

```
2210 FM := FREE
```

```
2220 IF FM < 1000 THEN SB(CHR$(21),3,ad$,fn$,RT$); RETURN
```

```
2230 E := INSTR(S,cm$,":")
```

```
2240 ix := VAL(cm$[S..E-1]);
```

```
2245 IF ix > mx THEN SB(CHR$(21),3,ad$,fn$,RT$); RETURN;
```

```
2250 ET := INSTR(E+1,cm$,";");
```

```
2255 IF ix = 0 THEN cf$ := cm$[E+1..ET-1]; GOTO 2270;
```

```
2260 N$(ix) := cm$[E+1..ET-1];
```

```
2270 S := ET+1;
```

```
2280 IF cm$[S..S] = CHR$(3) THEN GOTO 2290 ELSE GOTO 2210
```

```
2290 SB(CHR$(6),3,ad$,fn$,RT$); RETURN;
```

```
2300 E := INSTR(cm$,";")
```

```
2340 ix := VAL(cm$[7..E-1]);
```

```
2350 N$(ix) := ""
```

```
2360 ET := INSTR(E+1,cm$,';')
```

```
2370 cf$ := cm$[E+1..ET-1]
```

```
2380 SB(CHR$(6),3,ad$,fn$,RT$)
```

```
2390 RETURN
```

```
2400 cn := 1;
```

```
2402 SB$ := "" ;VL$ := ""
```

```
2405 FOR I:= cn TO mx
```

```
2407 IF NW(I) = 0 THEN GOTO 2420
```

```
2408 IF N$(I) = "" THEN GOTO 2420
```

```
2410 SB$:=SB$+STR$(I)+":"+V$(I)+";"; NW(I):=0; IF LEN(SB$)>bl THEN GOTO 2445
```

```
2420 NEXT I
```

```
2430 SB(SB$,3,ad$,fn$,RT$)
```

```
2440 RETURN
```

```
2445 SB(SB$,23,ad$,fn$,RT$);RETURN;
```

```
2500 cn := VAL(cm$[7..9]); GOTO 2402
```

```
2600 E := INSTR(cm$,"=")
```

```
2640 ix := VAL(cm$[7..E-1]);
```

```
2645 ET := INSTR(E+1,cm$,";");
```

```
2650 LVAL(N$(ix)) := VAL(cm$[E+1..ET-1])
```

```
2660 SB(CHR$(6),3,ad$,fn$,RT$)
```

```
2665 V$(ix) := STR$(VAL(N$(ix)));
```

```
2667 NW(ix) := 1;
```

```
2670 RETURN
```

```
2700 SB(CHR$(21),3,ad$,fn$,RT$)
```

```
2701 RETURN
```

```
2800 FOR I:= 1 TO mx
```

```
2810 N$(I) := ""; V$(I) := ""; NW(I) := 0;
```

```
2820 NEXT I
```

```
2825 cf$ := ""
```

```
2830 SB(CHR$(6),3,ad$,fn$,RT$)
```

```
2840 RETURN
```

```
3000 fn$ := fn$ + C$
```

```
3010 IF LEN(fn$) = 5 THEN GOTO 3100
```

```
3020 RETURN
```

```
3100 IF fn$ = "FUGGA" THEN GOTO 3200
```

```
3150 ST := 1; fn$ := ""; RETURN;
```

```
3200 WADO(7,-1);SDTTY1;SYSDIAG := 1; STOP;
```

```
7000 ON er GOTO 8100,8200
```

```
7100 RESUME 100
```

```
8100 RESUME 100;
```

```
8200 er := 1;
```

```
8300 VL$ := "???";
```

```
8400 RESUME 2410;
```

```
9999 END
```

```
.TSK 4
```

```
10 ON ERROR GOTO 200;
```

```
15 TSKFRM$ :=:(S:2) -- vystupny format - presnost floatingovych hodnot
```

```
20 VL$ := ""
```

```
30 FOR I:= 1 TO [3]mx
```

```
40 IF [3]N$(I) = "" THEN GOTO 60 ELSE VL$ := STR$(VAL([3]N$(I)));
```

```
50 IF VL$ <> [3]V$(I) THEN [3]V$(I) := VL$; [3]NW(I) := 1;
```

```
60 NEXT I
```

```
65 WADO(6,20); -- watch dog pre task 4
```

```
70 WAIT DURING 5; -- perioda zberu hodnot
```

```
80 GOTO 20
```

```
200 VL$ := "???";
```

```
210 RESUME 50;
```

```
500 END
```

```
.TSK 253
```

```
5 EA := EVADR
```

```
10 IF EA=$340.ADR AND $340.ERR = 20 AND $340.ER1=7 THEN GOTO 30 ELSE GOTO 100
```

```
20 ; -- 7 je cislo Watch Dogu, 3 je cislo komunikacneho tasku
```

```
30 WADO(7,-1); STOP 3; RUN 3,40; GOTO 200
```

```
100 IF EA=$340.ADR AND $340.ERR = 20 AND $340.ER1=6 THEN GOTO 130 ELSE GOTO 200
```

```
130 WADO(6,-1); STOP 4; RUN 4; GOTO 200
```

```
200 END
```

```
EXIT
```

```
TSK 254
```

```
5555 RUN 3,40
```

```
EXIT
```

\

```
.PROC SB
```

```
ENTRY (IN:B$,Ex,A$,F$,R$)
```

```
1050 BF$:=CHR$(2)+A$+F$+R$+B$+CHR$(Ex)
```

```
1060 CR$:=CRC$(4,BF$);
```

```
1070 BF$:=BF$+CR$
```

```
1076 FOR I:=1 TO 15 ; PRINT CHR$(255); NEXT I
```

```
1100 PRINT BF$
```

```
1200 WAIT DURING 1
```

```
1500 CLITTY1
```

```
7000 ENDPROC
```

```
EXIT
```

Literatúra

-

Zmeny a úpravy

- Máj 2000 Prenesenie protokolu do D2000 ver. 4.10

Revízie dokumentu

- Ver. 1.0 – 9. máj 2000 – Vytvorenie tohto dokumentu.

Súvisiace stránky:

Komunikané protokoly