

## 1. Einleitung

Mit der Prüfschrittfunktion „Geräte-Steuerung über RS232 / TCP/IP“ können Befehle an ein Gerät ( z.B. Messgerät, Motor, Steuerung, SPS, ... ) gesendet werden, welches über einen COM- oder TCP/IP-Port angeschlossen ist.  
Außerdem kann ein vom Gerät gesendeter Antwort-String empfangen und in einer TXT-Datei abgelegt werden.

Die Befehle müssen in eine Befehlsdatei ( Dateiendung \*.prg ) eingetragen werden.

### Wichtig:

Für die Ausführung dieser Funktion wird eine Lizenz für Modul 72 benötigt !

## 2. Befehlsdatei

Der Name der Befehlsdatei muss wie folgt lauten : <Prüfplan-Name>.prg

Die Befehlsdatei muss im ComGage-Verzeichnis für Prüfaufträge abgelegt werden.

### Dateiaufbau

#### a) Kommentarseiten :

Das erste Zeichen einer Kommentarseite ist ein Semikolon :

*Beispiel :*

```
;=====
;Command list for test scheme >>xxx<<
;
; Version :   V1.00
; Date    :   16.12.2007
;=====
```

#### b) Zeilen mit Konfigurations-Einstellungen ( z.B. COM-Port, Baudrate, ... ) :

Konfigurations-Einstellungen werden durch eckige Klammern gekennzeichnet.

Syntax :                      [<Parameter><Wert>]

#### • für COM-Ports :

<Parameter> :	C	➔	COM-Port
	B	➔	Baudrate
	P	➔	Parität
	D	➔	Anzahl Datenbits
	S	➔	Anzahl Stopbits
	T	➔	Timeout-Zeit
			( = Zeit, bis das Gerät die Antwort gesendet haben muss )
	E	➔	ComGage-Ergebnisregister, optional
			( gibt das Ergebnis einer Befehlsausführung in ein Register
			( siehe 2. d ) , statt über eine MessageBox aus )

<Wert> :	bei C	➔	1...8
	bei B	➔	1200, 2400, ..., 115200
	bei P	➔	No, Even, Odd, Space, Mark
	bei D	➔	7 oder 8
	bei S	➔	1 oder 2
	bei T	➔	Zeit in msec
	bei E	➔	0 ( = aus ) / 1 ( = Register 1 ) / ... / 2000 ( = Register 2000 )

### Beispiel für COM-Port :

```
; COM-Port = COM2
[C2]
; Baudrate = 9600
[B9600]
; Parity = no
[PN]
; Anzahl der Datenbits = 8
[D8]
; Anzahl der Stoppbits = 1
[S1]
; Timeout-Zeit = 10000 msec
[T10000]
```

### • für TCP/IP-Ports :

<Parameter> : I	→	IP-Adresse + Port ( Bei gesetzter IP-Adresse wird der COM-Port nicht verwendet. )
T	→	Timeout-Zeit ( = Zeit, bis das Gerät die Antwort gesendet haben muss )
E	→	ComGage-Ergebnisregister, optional ( gibt das Ergebnis einer Befehlsausführung in ein Register ( siehe 2. d ), statt über eine MessageBox aus )
<Wert> :		
bei I	→	<IP-Adresse>:<Port>, z.B. 127.0.0.1:27015
bei T	→	Zeit in msec
bei E	→	0 ( = aus ) / 1 ( = Register 1 ) / ... / 2000 ( = Register 2000 )

### Beispiel für TCP/IP-Port :

```
; IP-Adresse = 127.0.0.1:27015    →    Hierdurch werden die COM-Port-Einstellungen
[I127.0.0.1:27015]                ignoriert.
; Timeout time = 10000 msec
[T10000]
```

### c) Befehlszeilen :

Jede Befehlszeile enthält die Befehls-Zeilenummer, den Befehl und die erwartete Befehlsantwort vom Gerät.

Syntax :                    <Zeilenummer>:<Befehl>><Befehlsantwort>

<Zeilenummer> :    Nummer zwischen 0 ... 32767

Die Software sortiert die Befehle automatisch nach den Nummern.  
( Der erste Befehl muss nicht die Nummer 0 besitzen )

<Befehl> :            Der Befehlsstring kann aus einer Mischung von ASCII-Zeichen und  
                         ASCII-Codes bestehen.  
                         Die ASCII-Zeichen werden von " " eingeschlossen. Der ASCII-Code wird  
                         als Nummer eingetragen. Darüber hinaus stehen zusätzlich Platzhalter  
                         zur Ausgabe von Registerwerten zur Verfügung.  
                         ( Beispiel „Platzhalter für Register 5“ : {R5} )  
                         Alle Elemente werden durch Kommas getrennt.

Beispiele :

- Zur Ausgabe von **Move75<cr>Speed72<cr><lf>** sieht der

Befehlsstring wie folgt aus : "Move75",13,"Speed72",13,10

- Zur Ausgabe des Registerwerts von R6 als Parameter des Move-Befehls sieht der Befehlsstring wie folgt aus : "Move",{R6},13, ...

<Befehlsantwort> :    Der Antwortstring ist genauso wie der Befehlsstring aufgebaut.  
                         Wenn kein Antwortstring angegeben wird, dann wird auch keine Antwort  
                         vom Gerät erwartet.

**Beispiel : ( Hinweis : Bei Zeile 1 wird keine Befehlsantwort erwartet. )**

```
; Nullen des Systems  
1:"CLR",13>  
;  
; Verfahre relativ 1000 Schritte und erwarte als Antwort „OK“  
2:"MOVR 1000",13,10>"OK",13,10
```

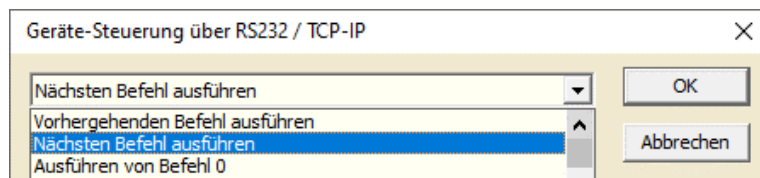
d) Ergebnisregister :

Wurde ein Ergebnisregister aktiviert ( siehe 2. b ) , erfolgt die Ergebnisausgabe der Funktion in dieses Register. Das Register kann folgende Werte annehmen :

- 0 = Kein Fehler
- 1 = Ausgewählter Befehl nicht verfügbar
- 2 = Befehlsantwort wurde nicht empfangen
- 3 = Fehlermeldung wurde empfangen
- 10 = Lizenz fehlt !
- 11 = Datei mit Befehlsliste wurde nicht gefunden
- 12 = Dateizugriffsfehler
- 13 = Syntax-Fehler in Befehlszeile
- 14 = Syntax-Fehler in Zeile mit Konfigurations-Einstellungen
- 15 = Fehler beim Öffnen des COM-Ports
- 16 = IP-Adresse / Port nicht korrekt konfiguriert
- 17 = TCP-Verbindung konnte nicht aufgebaut werden
- 18 = Fehler in IP-Adresse oder Host-Name

### 3. Konfiguration

Die Funktion wird innerhalb eines Prüfschritts angelegt. Über den Setup-Button öffnet sich folgender Dialog :



Der Dialog bietet folgende Konfigurationsmöglichkeiten :

#### **Nächsten Befehl ausführen**

Bei Aufruf der Funktion wird der Befehl mit der nächst höheren Zeilennummer ausgeführt.  
Wurde zuvor noch kein Befehl aus der Liste ausgeführt, wird der Befehl mit der niedrigsten Zeilennummer aufgerufen.

Wurde zuvor der Befehl mit der höchsten Zeilennummer ausgeführt, wird als nächstes der Befehl mit der niedrigsten Zeilennummer aufgerufen.

#### **Vorhergehenden Befehl ausführen**

Bei Aufruf der Funktion wird der Befehl mit der nächst niedrigeren Zeilennummer ausgeführt.  
Wurde zuvor noch kein Befehl oder der Befehl mit der niedrigsten Zeilennummer ausgeführt, wird als nächstes der Befehl mit der höchsten Zeilennummer aufgerufen.

#### **Ausführen von Befehl 0 ... 32767**

Bei Aufruf der Funktion wird immer der Befehl mit der gewählten Zeilennummer ausgeführt.

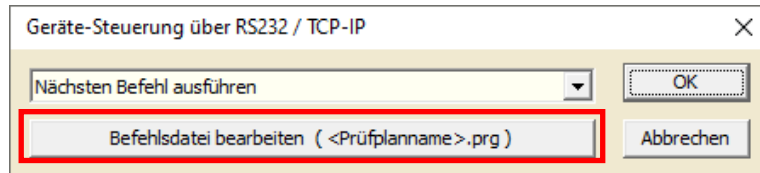
### Empfangenen Datenstring in TXT-Datei speichern

Wurde ein neuer Datenstring empfangen, wird dieser in eine TXT-Datei geschrieben.

Die TXT-Dateien werden im Datenverzeichnis für Prüfaufträge abgelegt.

Der Dateiname setzt sich wie folgt zusammen : <Prüfauftrags-Name>\_<Datum><Zeit>.txt

Über den Button „**Befehlsdatei bearbeiten ( <Prüfplanname>.prg )**“ kann die zum aktuellen Prüfplan gehörende Befehlsdatei mit einem Editor geöffnet werden :



## 4. Beispiele

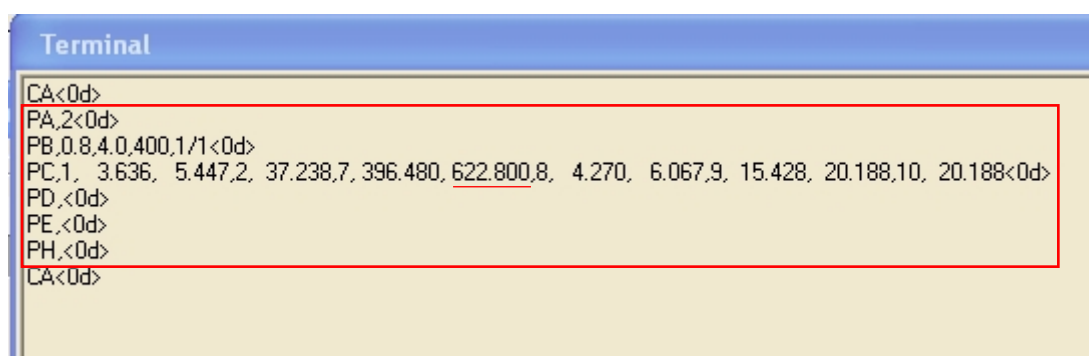
- Beispiel 1 : Faulhaber-Motor ansteuern

```
; Motor einschalten
1:"EN",13>"OK",13,10
; Motor ausschalten
2:"DI",13>"OK",13,10
; Motor mit Geschwindigkeit 500 verfahren
3:"V500",13>"OK",13,10
; Motor anhalten
4:"V0",13>"OK",13,10
; Motor an Position 8000 mit Geschwindigkeit 500 verfahren
5:"LA8000",13,"SP500",13,"M",13>"OK",13,10
```

- Beispiel 2 : Messwerte von Zeiss-Gerät empfangen

```
; Der folgende Befehl startet die Messung :
1:"CA",13>
```

Daraufhin wird der folgende String ( hier in einem Terminal dargestellt ) in eine Text-Datei geschrieben :



Der benötigte Wert ( hier unterstrichen ) kann nun mit der WGL018 als Merkmalswert aus der TXT-Datei eingelesen werden.

Hierzu kann das „PC“ am Zeilenanfang genutzt werden, um die richtige Zeile zu wählen.

Die Kommas können als Trennzeichen verwendet und die Spalte 9 als die Spalte angegeben werden, in der sich der Messwert befindet.

Es ist sogar möglich, die Werte in den verschiedenen Spalten verschiedenen Merkmalen im Prüfplan zuzuordnen.

Eine ausführliche Beschreibung der WGL018 finden Sie in der zugehörigen Dokumentation.