

1. Einleitung

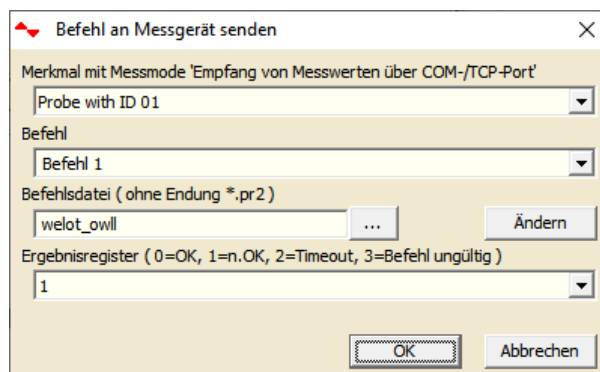
Mit der Prüfschrittfunktion „Befehl an Messgerät senden“ kann ein Befehl an ein Messgerät gesendet werden, welches von einem Merkmal mit dem Sonder-Messmode „Empfang von Messwerten über COM-/TCP-Port“ (wgl029) verwendet wird.

Wichtige Hinweise :

- Diese Prüfschrittfunktion kann nur zusammen mit dem kostenpflichtigen Sonder-Messmode „Empfang von Messwerten über COM- / TCP-Port“ verwendet werden, da der Befehl über die Verbindung des Sonder-Messmodes mit dem Messmittel übertragen wird.
- Damit diese Prüfschrittfunktion konfiguriert werden kann, sind zuvor Merkmale anzulegen, welche diesen Sonder-Messmode verwenden.
- Die Befehle müssen in einer Befehlsdatei (Dateiendung *.pr2) gespeichert werden. Die Befehlsdatei muss im ComGage-Verzeichnis für Prüfpläne abgelegt sein.
- Diese Funktion ist abhängig von der Auswahl der wgl029 „Empfang von Messwerten über COM-/TCP-Port“ als Messmode des ausgewählten Merkmals. Werden Änderungen am Merkmal vorgenommen, so muss diese Funktion entsprechend angepasst werden.
- Befehle können mit dieser Prüfschrittfunktion nur über einen COM-Port gesendet werden.

2. Konfiguration

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



Der Dialog bietet folgende Konfigurationsmöglichkeiten :

- **Merkmal mit Messmode 'Empfang von Messwerten über COM-/TCP-Port'**
Hier ist das Merkmal auszuwählen, in dem der Messmode „Empfang von Messwerten über COM- / TCP-Port“ angelegt wurde.
- **Befehl**
Hier ist die Nummer des auszuführenden Befehls auszuwählen (Zeilennummer → siehe 3. c)).
- **Befehlsdatei (ohne Endung *.pr2)**
Dateiname der Befehlsdatei (ohne Dateiendung). Die Befehlsdatei muss im ComGage-Verzeichnis für Prüfpläne abgelegt sein. Über den Button ... kann eine Befehlsdatei ausgewählt werden. Der Button **Ändern** öffnet die Befehlsdatei in einem Editor zum Bearbeiten.
- **Ergebnisregister**
In das ausgewählte Ergebnisregister wird das Funktionsergebnis geschrieben.
Mögliche Werte sind:
 - (0) Befehl wurde erfolgreich ausgeführt
 - (1) Ausführung des Befehls nicht erfolgreich
 - (2) Timeout beim Ausführen des Befehls
 - (3) Befehlsformat ungültig

3. Dateiaufbau der Befehlsdatei :

a) Kommentarzeile :

Das erste Zeichen einer Kommentarzeile ist ein Semikolon :

Beispiel :

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

b) Zeilen mit Konfigurations-Einstellungen :

Konfigurations-Einstellungen werden durch eckige Klammern gekennzeichnet.

Syntax : [<Parameter><Wert>]

<Parameter> : T ➔ Timeout-Zeit (aktuell ist nur dieser Parameter verfügbar)
 = Zeit bis das Gerät die Antwort gesendet haben muss

<Wert> : ➔ Zeit in msec

Beispiel :

```
; Timeout time = 5000 msec
[T5000]
```

c) Befehlszeile :

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 Kommata 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 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 :

```
;
; Command for disabling RS485 lock on sensor 1
1:":01W010;0;E9C3",13,10>":01a;89EE",13,10
```