



AT : Microteaching

Wegen der besonderen Situation in Corona-Zeiten sind diese Übungen für den reinen "remote-teaching"-Betrieb in winzige, gut überschaubare Paketen gegliedert. Jede Einheit sollte Sie etwa 10 bis 20 Minuten beschäftigen.

Hier in AT werden wir zunächst keine neuen Inhalte mehr einführen, sondern das bisherige durch Übungen sichern. (Einige haben schon die umfangreichen, anspruchsvollen „normalen“ Übungen aus dem AT-Skript durchgeführt : dann können Sie sich diese Pakete sparen ...)

Bitte melden Sie mir ihre Erfahrungen (Aufwand, Schwierigkeitsgrad usw.) intensiv zurück, damit ich weiter optimieren kann.

Ich freue mich auf Ihre Fragen!

Reiner Doll, Technikerschule München

<http://portal.ts-muenchen.de>

mail@reinerdoll.de



Bisheriger Inhalt

Installieren des digitalen Zwillings

Paket 1 : Handshake mit Python

Paket 2 : Starre Kopplung zweier Module

Digitaler Zwilling



Sie haben ja alle in der guten, alten Zeit, als es noch richtigen Unterricht gab, schon mit dem digitalTwin gearbeitet.

Ich denke, daß einige das auch schon Zuhause installiert haben. Mittlerweile sind (fast ?) alle Fehler beseitigt und alle Module sind aktiviert. Die Dateien für die Installation finden Sie hier :

<http://portal.ts-muenchen.de/index.php/digitale-fabrik-2018/digitaler-zwilling>

Im AT-Skript auf dem Portal finden Sie ab Seite 145 Hilfen zur Installation und Bedienung des digitalTwin und der Klassenbibliothek twinClass.



Paket 1

Essentiell für die Steuerung von Anlagen mittels flow-execution im MES-System ist das Verständnis von Handshakeprotokollen.

Schauen Sie sich den Anlagenhandshake Seite 135, 136 nochmal an. Dann lesen Sie die Doku der Klasse PlcModule der Klassenbibliothek twinClass2020. Kopieren Sie die aktuelle Version des digitalen Zwilling vom Portal.

- a) Schreiben Sie eine kleine Python-Routine, die das Band des digitalTwin startet.
- b) Schreiben Sie eine Routine, die die Testfunktion an Modul 1 auslöst.



Lösungsvorschlag : Paket 1

a)

```
from twinClass2020_v20 import *
```

```
sps = PlcModule()  
sps.StartBand()
```

Laden der Software und Installation finden Sie auf den Seiten 144-146 im AT-Skript beschrieben. Welche Version (hier _v20) der Klassenbibliothek sie haben, müssen sie ggf. anpassen. Am Besten immer die Aktuellste vom Portal holen :

<http://portal.ts-muenchen.de/index.php/digitale-fabrik-2018/digitaler-zwilling>

b)

Bibliothek einbinden, testen mit Band :

```
from twinClass2020_v20 import *  
sps = PlcModule()  
sps.StartBand()
```

Eine Möglichkeit wäre, Ready und Busy durch polling zu prüfen.

Man könnte bei Ready=0 auch genauso gut abbrechen :

```
ready = sps.ReadOPCTag("module1","Ready")  
busy = sps.ReadOPCTag("module1","Busy")  
while ready==0 or busy==1:  
    ready = sps.ReadOPCTag("module1","Ready")  
    busy = sps.ReadOPCTag("module1","Busy")
```

Jetzt die Order und den Start setzen :

```
sps.WriteOPCTag("module1","Order",10)  
sps.WriteOPCTag("module1","Start",1)
```

Dann mit polling auf die Quittung warten :

```
ack = sps.ReadOPCTag("module1","Acknowledge")  
while ack==0:  
    ack = sps.ReadOPCTag("module1","Acknowledge")
```

Und wenn die eingetroffen ist, Start wieder rücksetzen :

```
sps.WriteOPCTag("module1","Start",0)
```



Paket 2

Nun lassen Sie Modul 1 und Modul 2 in starrer Kopplung laufen. Geben Sie eine feste Order (1 oder 2), starten Sie beide Module gleichzeitig, warten Sie bis beide fertig sind, und dann starten Sie (gleichzeitig) wieder.