

Fortschritte des ROBOT-XTS-

Kommunikationsprojekts

### Prof. Dr. Armando Walter Colombo

Yahya Fakhet:7012464

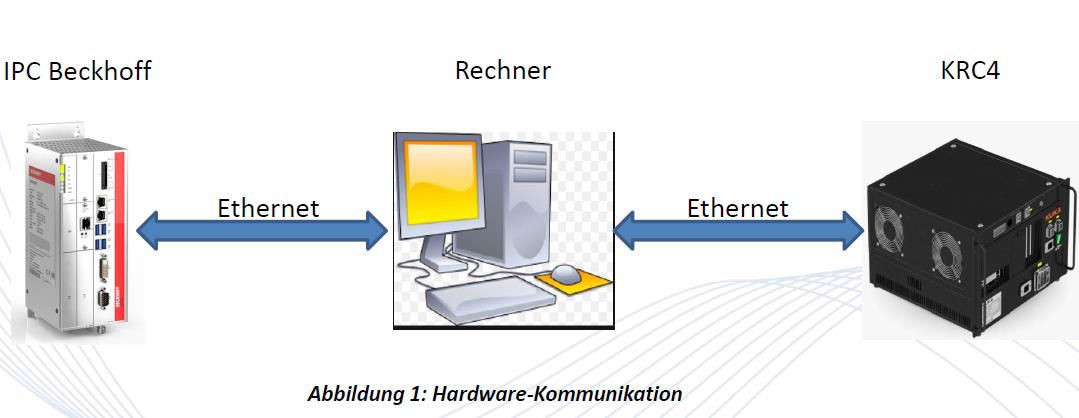
Faissal Hammouda:7012301 **WS/2021**



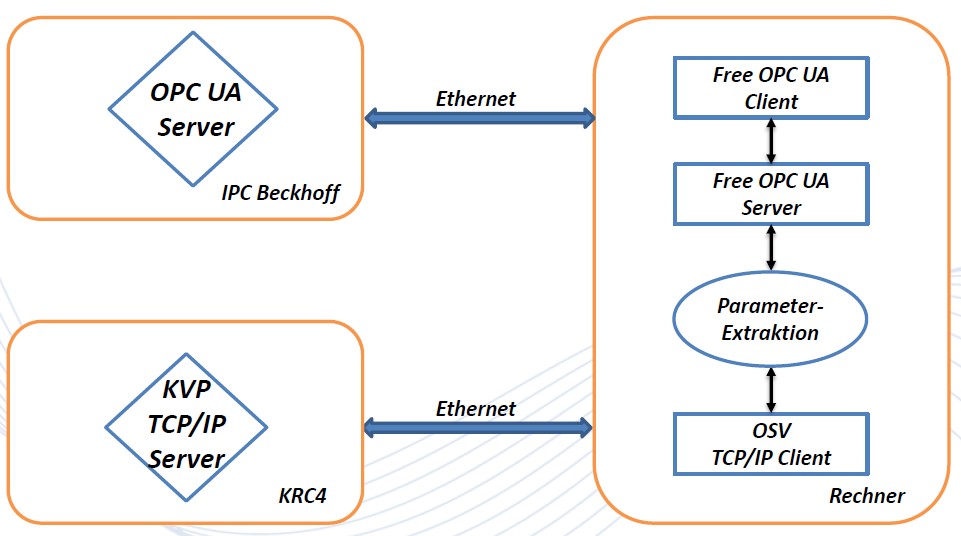
###### Kommunikations-Architektur

* **TwinCat 3-Projekt**
* **HMI-Projekt**
* **KUKAVARPROXY und OpenShowVar**
* **Applikation**
* **Inhaltsverzeichnis**
* **Antrag**

**2**



**3**

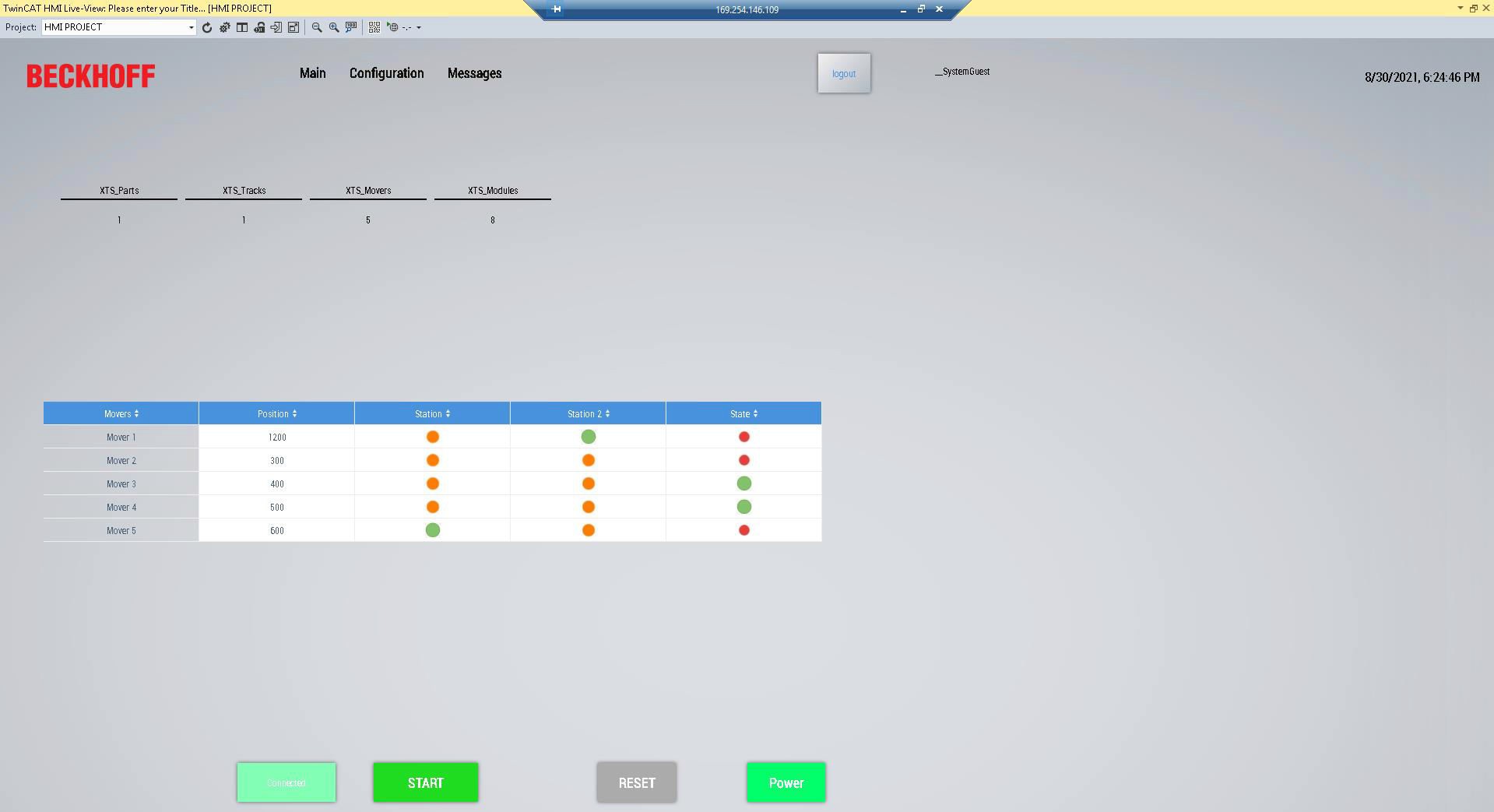


**4**



* Der TwinCat 3 Teil des Projekts ist in 3 Teile aufgeteilt:
* ***HMI-Projekt:*** Dieser Teil ist mit Hilfe der TF2000-Lizenz der Firma Beckhoff realisiert.
* ***OPC UA Server Projekt:*** Dieser Teil ist mit Hilfe der TF6100-Lizenz von Beckhoff realisiert.
* ***PLC-Projekt:*** Dieser Teil ist mit Hilfe der TF1200-Lizenz der Firma Beckhoff realisiert.

**5**



* Das HMI-Projekt enthält :

1 2 3

8

9.1 9.2 9.3 9.4 9.5

9

7 6 5 4

###### Main-Seite

**6**



***1. HMI-Navigation-Bar:*** ermöglicht das Navigieren zwischen den 3 Seiten des HMI. ***2.Login-Logout***: identifiziert die Verwendung der HMI je nach Art des Benutzers. ***3.Date-Time-Display:*** Anzeige Datum und Uhrzeit auf dem HMI.

###### 4.Power-Button 5.Reset-Button 6.Start-Stop-Button 7.Connect-Button

1. ***XTS-Informations:*** informiert vom Programm die Anzahl der Parts, Tracks, Movers und

Modules.

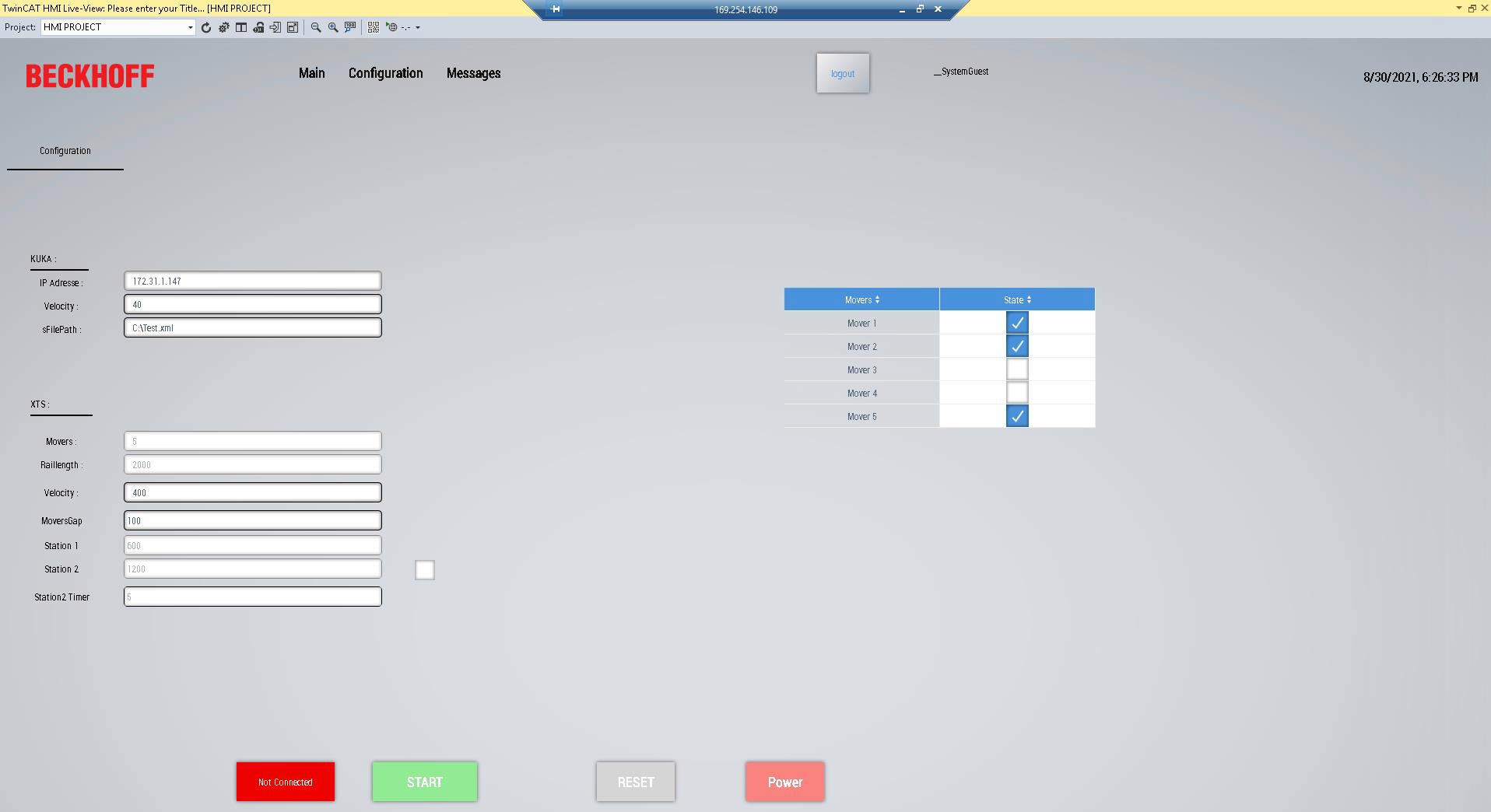
1. ***Datagrid:*** enthält die folgenden Informationen:
   1. ***Movers:*** identifizieren die Anzahl der jeweiligen Mover.
   2. ***Position:*** Identifizieren der Position des Mover im XTS in Millimetern.

**7**



* 1. ***Station:*** an der die Befüllung oder Entleerung der Mover erfolgt.
* *Grün Farbe:* der Mover ist im Station*.*
* *Orange Farbe:* der Mover ist nicht in der Station.
* *Rote Farbe:* es liegt ein Fehler vor.
  1. ***Station 2:*** Wartestation, an der die Mover für eine bestimmte Zeit anhalten.
* *Grün Farbe:* der Mover ist im Station*.*
* *Orange Farbe:* der Mover ist nicht in der Station.
* *Rote Farbe:* es liegt ein Fehler vor.
  1. ***State:*** zeigt den Status des Mover an.
* *Grün Farbe:* der Mover ist leer*.*
* *Rote Farbe:* der Mover ist voll.

**8**



1 12

2

3

4

5

6

7

9 8

11

10

###### Configuration-Seite

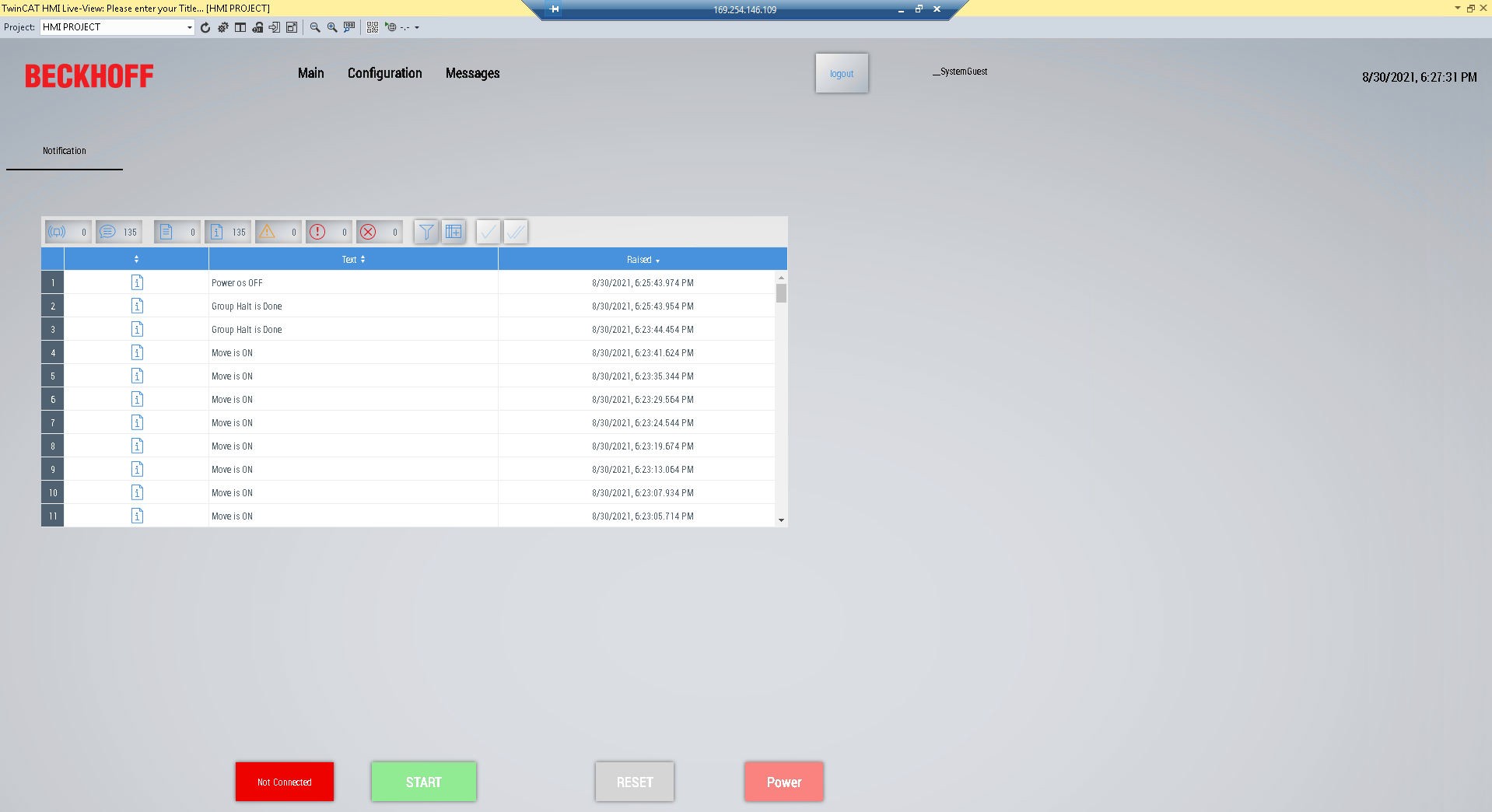
**9**



1. ***KUKA-IP-Adress:*** schreiben die IP-Adresse des Roboters.
2. ***KUKA-Velocity:*** Eingabe der Geschwindigkeit des Roboters (zwischen 0 und 100). ***3.sFilePath:*** Geben den Speicherort der XML-Datei für den Fall eines Stromausfalls an. ***4.Movers:*** Lesen die Anzahl der Movers.
3. ***Raillength:*** Lesen die Länge des XTS.
4. ***XTS-Velocity:*** Eingabe der Geschwindigkeit der Bewegung von Movers.
5. ***Movers-Gap:*** Eingabe des Sicherheitsabstands zwischen Movers.
6. ***Station-1:*** Position, an der der Mover darauf wartet, dass der Roboter ihn füllt oder leert.
7. ***Station-2:*** Geben die Warteposition in Millimetern an (fakultativ).
8. ***Station-2-Timer:*** Geben die Wartezeit an Station 2 in Sekunden ein (fakultativ).
9. ***CheckBox:*** eingeben, ob der Benutzer Station-2 verwenden möchte oder nicht.
10. ***Movers-State-Initialisation:*** Eingabe des Status der einzelnen Mover (voll oder leer) beim

Start.

**10**



2 3 4

1

###### Messages-Seite

**11**



1. ***Event-Grid:*** ermöglicht die Anzeige von Informationen, Warnungen oder Fehlern zu verschiedenen Funktionen und Schritten des Programms.
2. ***Symbol:*** Anzeige des Nachrichtensymbols (Information, Warnung oder Fehler).
3. ***Text:*** die Nachricht anzeigen.
4. ***Raised:*** das Datum und die Uhrzeit der Nachricht anzeigen.

**12**



* ***KUKAVARPROXY:*** ist ein TCP/IP-Server, der Robotervariablen über das Netzwerk liest und

schreibt.

* ***OpenShowVar:*** *Eine Python-Ports des KUKAVARPROXY -Clients.*

**13**



XTS KUKA

MoveXTS

MoveKuka

###### Applikationsprozess

**14**



* Der Datenaustausch wird wie folgt durchgeführt:
* ***MoveXTS:*** Variabel ermöglicht den Austausch vom Roboter zum XTS.
* ***MoveKUKA:*** Variabel ermöglicht den Wechsel vom XTS zum Roboter.
* für jeden Variablentausch kennt das Programm die Anzahl der Mover und deren Zustand

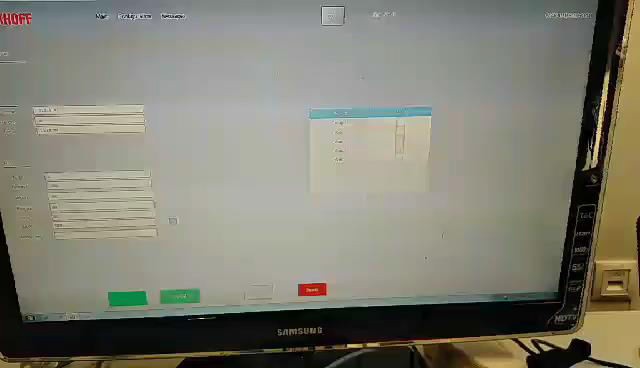
(ob voll oder leer)

* Die Videos erklären besser die Applikation.

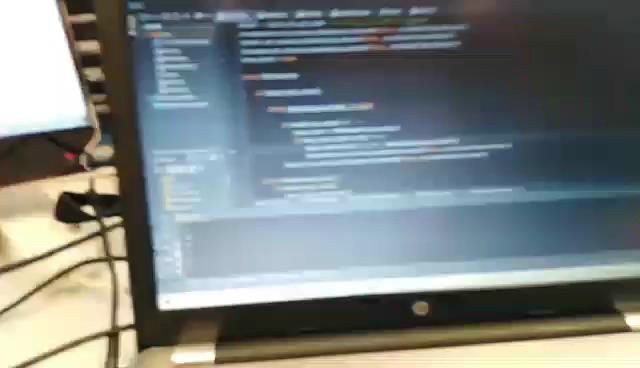
**15**



**16**



**17**



**18**



* 1. Allgemeine Einführung

1. Definition des Projekts 2.Ziel des Projekts

* 1. Literaturübersicht
     1. Industrie 4.0
     2. OPCUA (Server/Client)
     3. Python. 4.KUKA Roboter

5.lineares Transportsystem Beckhoff(XTS)

* 1. Analyse der Kommunikation über das OPCUA-Protokoll 1.a.Kommunikation des Roboters über OPCUA 1.a.Problem gefunden

1.b.Lösungen

1.c. Definition von KUKAVARPROXY und OpenShowVar

2.2 XTS-Kommunikation über OPCUA 2.a.OPCUA Server BECKHOFF(TF6100) 3.python Applikation

* 1. OPCUA Client(OpenShowVar)
  2. XTS-OPCUA-Server **19**
  3. OPCUA Server(KUKAVARPROXY)



* 1. Applikation

1. Überblick
2. Programmierung des KUKA Roboters 2.a.Robotersicherheit 2.b.Datenkonfiguration

2.c. Roboter-Programm

3.3 Programmierung des XTS

3.a.lizenz 3.b.Bibliotheken 3.c.Virtuelle Maschine

* 1. Reale Maschine
  2. Sicherheit von Maschinen und Movers
  3. Stationen
     1. Station-1 3.f.b.Warte-Station

4.python-Programmierung

**20**



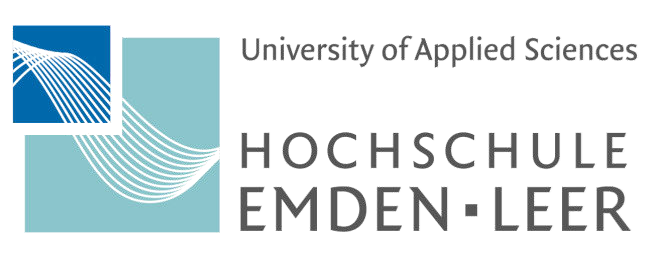
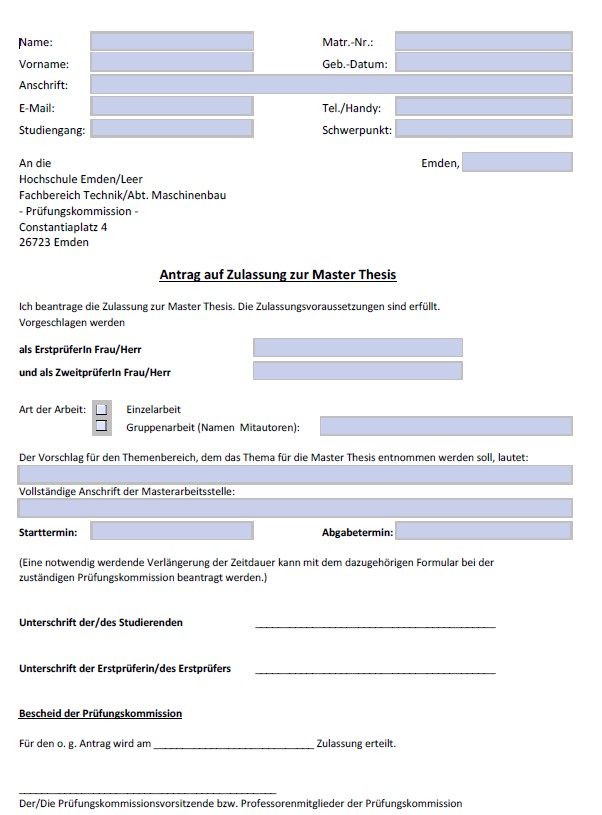
* 1. Human Machine Interface (HMI)
     1. Grundlage der Auswahl 2.HMI BECKHOFF(TF2000)

1. Architektur der Human Machine Interface
   1. Main-Seite
   2. Configuration-Seite 3.c.Notification-Seite
2. Allgemeine Zusammenfassung
3. Anhang
4. Roboter-Programm
5. XTS-Programm
6. Python-Programm
7. technische Zeichnung der Objektträger auf dem Mover
8. technische Zeichnung des Objekts
9. technische Zeichnung der Box
10. Informationsquellen

**21**

## Antrag

**22**



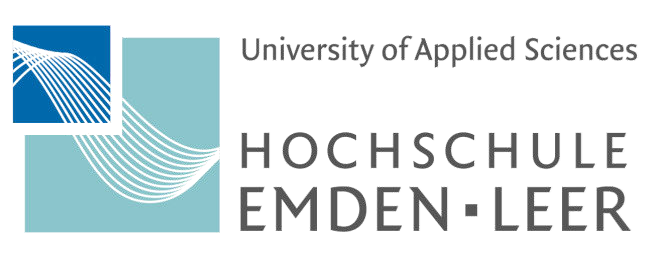
*Vielen Dank für Ihre Aufmerksamkeit*

# Studie zur Kommunikation

*und Konfiguration*

### Prof. Dr. Elmar Wings

Yahya Fakhet:7012464 Faissal Hammouda:7012301



**WS/2021**



* ***Problemanalyse***
* ***EthernetKRL***
* ***Konfiguration einer Ethernet-Verbindung***
* ***Datenaustausch***
* ***Client-Server-Betrieb***
* ***Protokollarten***
* ***EthernetKRL installieren***
* ***Konfiguration***
* ***Programmierung***

**2**



* ***OPC-UA-Server-Konfiguration***
* ***XTS Software***

##### Grundkonfiguration der XTS-Hardwarekomponenten

* ***XTS-Programmierung***
* ***Quellen***

**3**



* + Der Fehler, der beim Versuch der Kommunikation zwischen der KUKA Robotersteuerung und dem Computer aufgetreten ist, war die Änderung der IP-Adresse des Roboters

###### Problemstellung

• Änderung der IP-Adresse von "192.168.0.253" auf "172.3.1.254"

* + Die IP-Adresse des KUKA Roboters ist eine **statische IP-Adresse** und darf nicht verändert werden

**4**



###### Problemlösung

* Mit dem **KUKA USB-Stick** kann ein ‚***Reset‘*** über die Datei ***‚Image.sys‘*** durchgeführt werden
* Vor dem Einstecken des USB-Schlüssels muss der Roboter gestoppt werden, es ist auch ratsam, die Stromversorgung des Roboters auszuschalten
* Es gibt zwei Möglichkeiten, den USB-Stick in den Roboter zu stecken, entweder in die Robotersteuerung oder in das SmartPad. Es wird jedoch empfohlen, ihn an den Steuerung anzuschließen

**5**



###### Funktionen

EthernetKRL ist ein nachladbares Technologiepaket mit folgenden Funktionen

* Datenaustausch über die EthernetKRL-Schnittstelle
* Empfangen von XML-Daten eines externen Systems
* Senden von XML-Daten an ein externes System

###### Eigenschaften

* Robotersteuerung und externes System als Client oder Server

**6**



* Konfiguration von Verbindungen über XML-basierte Konfigurationsdatei
* Konfiguration von "Ereignismeldungen"
* Überwachen von Verbindungen durch einen Ping auf das externe System
* Lesen und Schreiben von Daten aus dem Submit-Interpreter
* Lesen und Schreiben von Daten aus dem Roboter-Interpreter

###### Kommunikation

* Daten werden über das **TCP/IP-Protokoll** übertragen

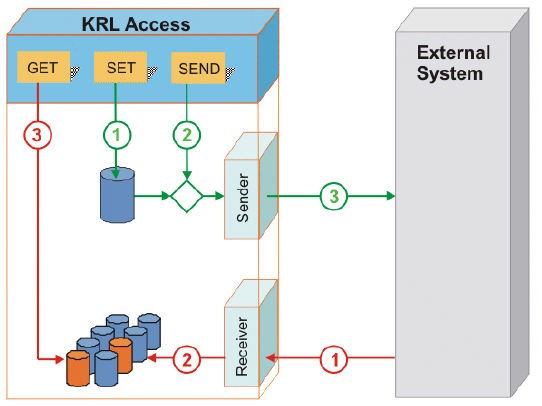
**7**



* Die Ethernet-Verbindung wird über eine **XML-Datei** konfiguriert
* Für jede Verbindung muss im Verzeichnis **C:\KRC\ROBOTER\Config\User\Common\EthernetKRL** der Robotersteuerung eine Konfigurationsdatei definiert sein
* Die Konfiguration wird beim Initialisieren einer Verbindung eingelesen
* Ethernet-Verbindungen können vom Roboter- oder Submit Interpreter angelegt und

bedient werden

**8**



* Über **EthernetKRL** kann die Robotersteuerung sowohl Daten von einem externen System empfangen als auch Daten an ein externes System senden

**9**



###### Datenempfang

1. Das externe System sendet Daten, die über ein Protokoll übertragen und von der EKI empfangen werden
2. Die Daten werden strukturiert in einem Datenspeicher abgelegt
3. Aus einem KRL-Programm heraus wird strukturiert auf die Daten zugegriffen Mithilfe von KRL-Anweisungen werden die Daten gelesen und in KRLVariablen kopiert

**10**

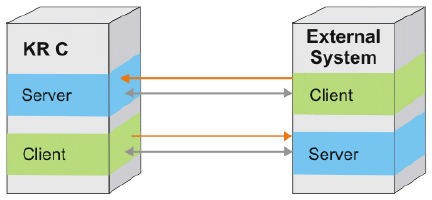


###### Datenversand

1. Mithilfe von KRL-Anweisungen werden die Daten strukturiert in einen Datenspeicher geschrieben
2. Mit einer KRL-Anweisung werden die Daten aus dem Speicher gelesen
3. EKI sendet die Daten über ein Protokoll an das externe System

* *Es ist möglich Daten direkt zu versenden, ohne dass die Daten zuvor in einem Speicher abgelegt werden*

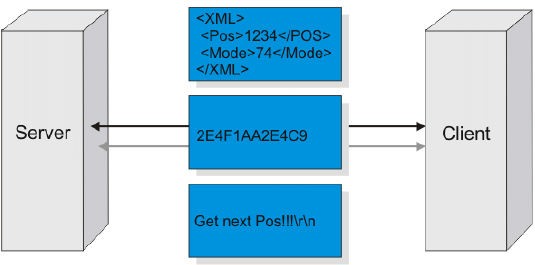
**11**



* Robotersteuerung und externes System verbinden sich als Client und Server
* Dabei kann das externe System Client oder Server sein
* Wenn die EKI als Server konfiguriert wird, kann sich nur ein einzelner Client mit dem

Server verbinden

**12**



* Die übertragenen Daten können in verschiedene Formate verpackt werden
* Folgende Formate werden unterstützt
* XML-Struktur frei konfigurierbar *(****das gewählte Protokoll****)*
* Binär-Datensatz mit fester Länge
* Binär-Datensatz variabel mit Endzeichenfolge

**13**



###### Vorbereitung

* Die Software von der CD auf einen USB-Stick kopieren
* Die Software muss so auf den Stick kopiert werden, dass die Datei **Setup.exe** in der obersten Ebene liegt

###### Vorgehensweise

1. Den USB-Stick an der Robotersteuerung oder am smartPAD anstecken
2. Im Hauptmenü **Inbetriebnahme** > **Zusatzsoftware** wählen
3. Auf **Neue Software** drücken: In der Spalte **Name** muss der Eintrag **EthernetKRL**

angezeigt werden und in der Spalte **Pfad** das Laufwerk **E:\** oder **K:\**

**14**



1. Den Eintrag **EthernetKRL** markieren und auf **Installieren** drücken. Die Sicherheitsabfrage mit **Ja** beantworten
2. Die Aufforderung zum Neustart mit **OK** bestätigen
3. Den Stick abziehen
4. Die Robotersteuerung neu starten

* *Es wird eine* ***LOG-Datei*** *unter* ***C:\KRC\ROBOTER\LOG*** *erstellt*

**15**



###### Netzwerkverbindung über das KLI der Robotersteuerung

* Für den Datenaustausch über Ethernet muss eine Netzwerkverbindung über das KLI der Robotersteuerung hergestellt werden
* Je nach Spezifikation stehen an der Kundenschnittstelle der Robotersteuerung folgende Ethernet-Schnittstellen als Option zur Verfügung
* Schnittstelle X66 (1 Steckplatz)
* Schnittstelle X67.1-3 (3 Steckplätze)

**16**



###### Konfigurieren einer Netzwerkverbindung

1. Im Hauptmenü Wählen **Inbetriebnahme** > **Wartung** > **HMI minimieren**
2. Wählen im Windows-Startmenü **Alle Programme** > **EKI-Netzwerk**

* Das Fenster **Netzwerk-Setup** erscheint. Die bereits eingerichteten Netzwerkverbindungen werden in der Baumstruktur unter **Andere installierte Schnittstellen** angezeigt

1. Markieren in der Baumstruktur unter **Ethernet KRL** den Eintrag **Neu** und drücken

**Bearbeiten**

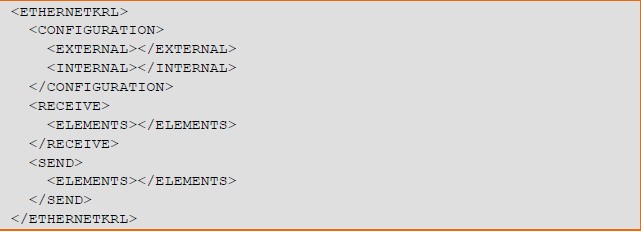
**17**



1. Geben die IP-Adresse ein und bestätigen mit **OK**
2. Starten die Robotersteuerung mit einem Kaltstart neustarten

* *Der IP-Adressbereich 192.168.0.x ist für die Konfiguration der Netzwerkverbindung blockiert*

**18**



###### Ethernet-Verbindung konfigurieren

* Eine Ethernet-Verbindung wird über eine XML-Datei konfiguriert
* Für jede Verbindung muss im Verzeichnis **C:\KRC\ROBOTER\Config\User\Common\ EthernetKRL** der Robotersteuerung eine Konfigurationsdatei definiert sein

**19**



* <CONFIGURATION>
* Konfiguration der Verbindungsparameter zwischen externem System und Schnittstelle
* <RECEIVE>
* Konfiguration der Empfangsstruktur, die von der Robotersteuerung empfangen wird
* <SEND>
* Konfiguration der Sendestruktur, die von der Robotersteuerung gesendet wird

**20**



###### XML-Struktur für Verbindungseigenschaften

* Im Abschnitt **<EXTERNAL> … </EXTERNAL>** werden die Einstellungen für das externe

System definiert

* TYPE
* Legt fest, ob das externe System als Server oder als Client mit der Schnittstelle kommuniziert
* IP
* IP-Adresse des externen Systems, wenn dieses als Server definiert ist (TYPE = Server)

**21**



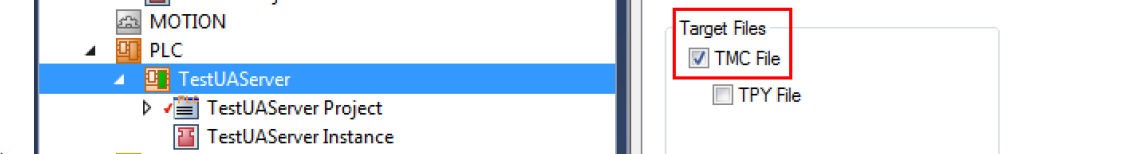
* PORT
* Port-Nummer des externen Systems, wenn dieses als Server definiert ist (TYPE = Server)
* Im Abschnitt **<INTERNAL> … </INTERNAL>** werden die Einstellungen für die Schnittstelle definiert
* ENVIRONMENT
* Löschen der Verbindung an Aktionen koppeln
* BUFFERING
* Mode: Verfahren, nach dem alle Datenspeicher abgearbeitet werden

**22**



* + Limit: Maximale Anzahl an Datenelementen, die ein Datenspeicher aufnehmen kann
* BUFFSIZE
  + Limit: Maximale Anzahl an Bytes, die empfangen werden können, ohne dass sie interpretiert werden
* TIMEOUT
  + Connect: Zeit, nach der der Versuch eine Verbindung aufzubauen abgebrochen wird
* PROTOCOL
  + Übertragungsprotokoll

**23**

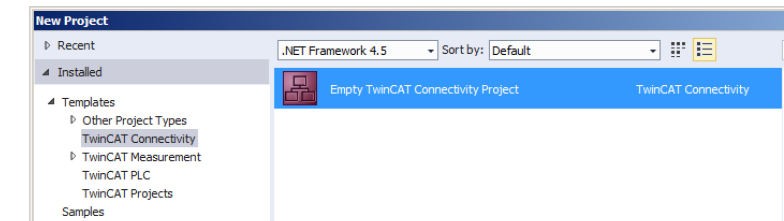
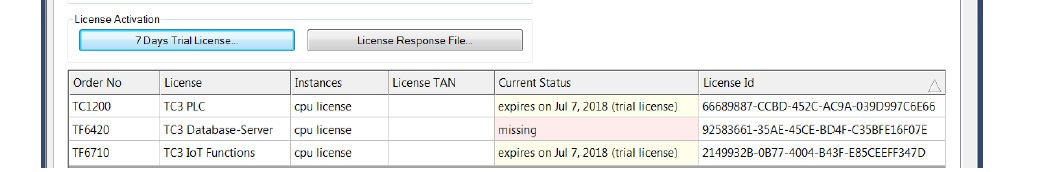


* In diesem Teil stellen die Schritte für den OPC-UA-Server zum Beckhoff IPC konfigurieren vor

###### TwinCAT OPC UA Server zur Verfügung zu stellen

* Importieren die Daten, die zwischen Beckhoff SPS und der Kuka-Steuerung übertragen möchten, in den OPC-UA-Server
* müssen den Download der Symboldatei in den Einstellungen des SPS-Projekts aktivieren.

**24**

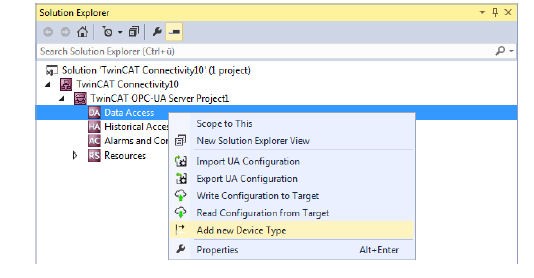


* TF6100-Lizenz für den Server aktivieren : Eine 7-Tage-Testlizenz reicht für einen schnellen Start aus

###### Konfigurieren die verschiedenen Facetten des OPC-UA-Servers

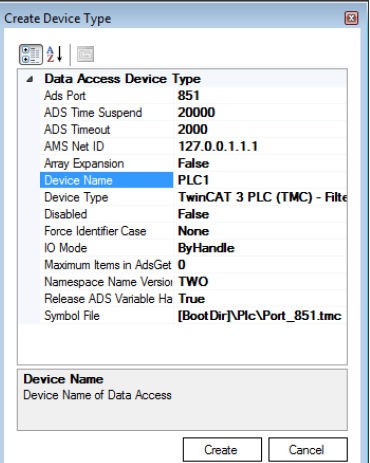
* müssen ein neues Projekt erstellen: Das Projektpaket des OPC-UA-Konfigurators ist in das sogenannte **Connectivity-Paket** integriert.

**25**



* Nachdem ein neues Projekt fertiggestellt wurde, stellen eine Verbindung zwischen dem OPC UA Server und ADS-Geräten her.
* Nach dem Ausführen des Befehls öffnet sich ein Dialogfenster, in dem Sie die Verbindungsparameter für dieses Gerät konfigurieren können, z. B. AMS Net ID, ADS-Port oder auch die Symboldatei

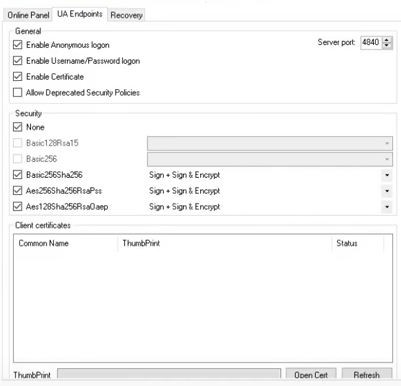
**26**



müssen die AMS-Net-ID für den Computer einstellen

***AMS-Net ID*:** Die AMS Net ID ist die Adresse des lokalen Rechners im TwinCAT-Netzwerk. Die AMS Net ID besteht aus 6 Bytes und wird in Punktschreibweise dargestellt. Die "Net IDs" müssen vom Projektierer vergeben werden und dürfen sich im TwinCAT-Netzwerk nicht wiederholen. Standardmäßig generiert die Installation eine AMS NetID aus der IP-Adresse des Systems (falls vorhanden) + "1.1"

**27**

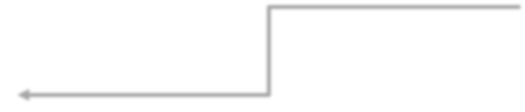
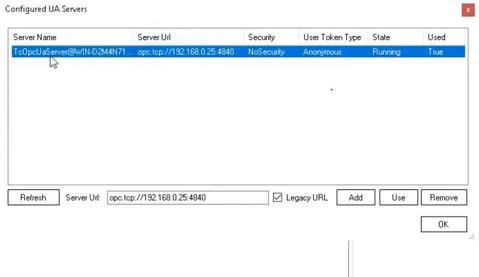
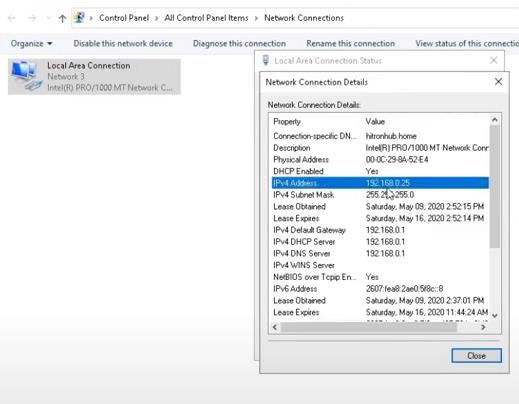


###### Endpunkte und Zertifikatsvertrauenseinstellungen konfigurieren

Die Endpunkte des OPC-UA Servers legen fest, welche Sicherheitsmechanismen beim Verbindungsaufbau eines Clients verwendet werden sollen. Diese reichen von "unverschlüsselt" bis "verschlüsselt und signiert".

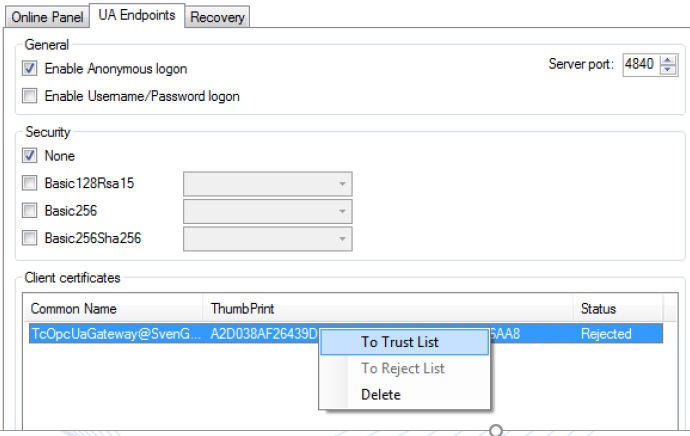
"verschlüsselt und signiert", basierend auf unterschiedlichen Schlüsselstärken.

**28**



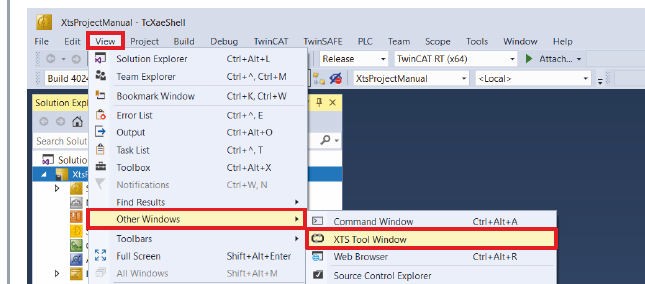
* Aufbau einer Verbindung zum OPC-UA-Server : Um eine Verbindung von einem OPC UA Client aufzubauen, muss der Client eine Verbindung mit der URL des OPC UA Servers aufbauen, z. B. opc.tcp://CX-12345:4840 oder opc.tcp://192.168.1.1:4840

**29**



Nachdem ein OPC-UA-Client zum ersten Mal versucht hat, sich mit einem sicheren Server- Endpunkt zu verbinden, wird das Zertifikat des Clients auf dem Server hinterlegt und als "abgelehnt" deklariert. Nächste, kann der Server-Administrator das Zertifikat freigeben. Ein anschließender Versuch, den Client mit einem sicheren Endpunkt zu verbinden, ist dann erfolgreich.

**30**



* Dieser Teil hilft uns bei der XTS-Programmierung und der Grundkonfiguration der XTS-Hardwarekomponenten in TwinCAT 3

##### Grundkonfiguration der XTS-Hardwarekomponenten in TwinCAT 3

* *Nach erfolgreicher Installation der* ***TF5850 TC3 XTS Extension*** *und dem Öffnen eines*

*neuen Projekts in TwinCAT 3 XAE muss das* ***XTS Tool Window*** *aktiviert werden.*

* Die Komponenten des eXtended Transport System (XTS) sind ausschließlich für die Verwendung durch die Automatisierungssoftware TwinCAT (TF5850 TC3 XTS Extension) vorgesehen.

**31**



* Vor der Verwendung von XTS muss es mit allen Komponenten und Konfiguration

vorbereitet werden

###### Die Komponenten

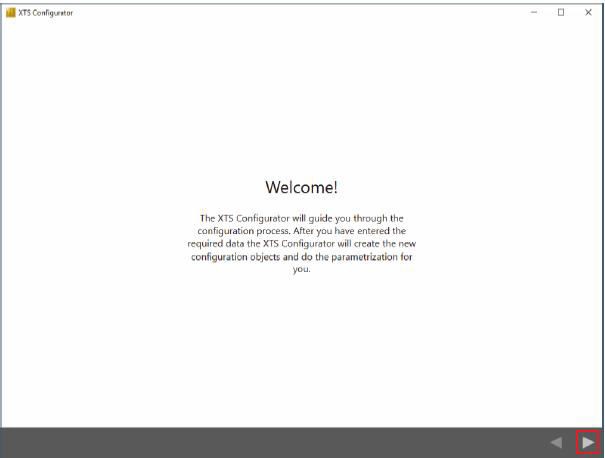
* ***XTS Processing Unit* :**ist das Zentrum des XTS Systems. In der XTS Processing Unit laufen alle erforderlichen Objekte zusammen und werden logisch miteinander verknüpft.
* ***XTS Part* :** ist die mindestens erforderliche Komponente eines XTS Systems. Der kleinste mögliche XTS Part ist ein einzelnes Einspeisemodul

**32**



* ***XTS Track* :** ist eine mit XTS Movern befahrbare Strecke, die aus einem oder mehreren XTS Parts besteht.
* ***XTS-Station*:** Die Station hat eine definierte Start- und Endposition. Diese Positionen bilden den Rahmen der XTS- Station und können sich über mehrere Teile der XTS-Station erstrecken. Die Stoppositionen können innerhalb des Rahmens konfiguriert werden.
* ***Die XTS Mover* :** sind zusammen mit den Modulen die Hauptkomponenten des XTS Systems. Die XTS Mover fahren auf den XTS Parts im XTS System, indem die entsprechenden XTS Tracks mit Hilfe der standardmäßigen NC Funktionalitäten von TwinCAT 3 aktiviert werden

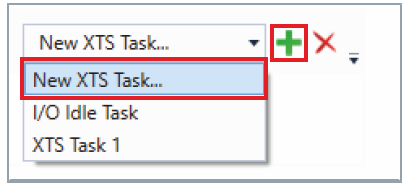
**33**



###### XTS Konfigurator :

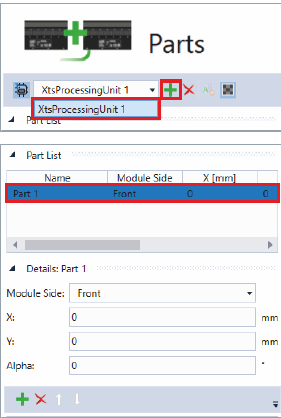
* klicken auf die Schaltfläche "XTS Configurator" im XTS Tool Window

**34**



* + legen eine neue XTS Processing Unit an
* muss man neue XTS Task hinzufügen

**35**



* legen XTS Part an

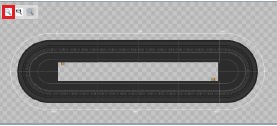
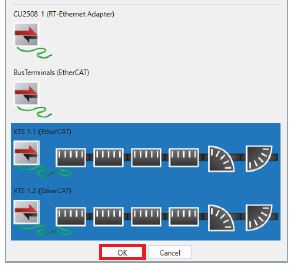
einen neuen XTS-Teil hinzufügen und die XTS- Processing für jeden Teil definieren.

Definieren die Eigenschaften der Parts

► Position des *XTS Parts* über die Felder **X** und **Y** eingeben

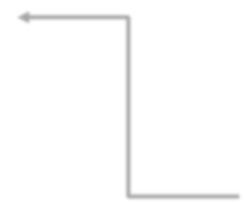
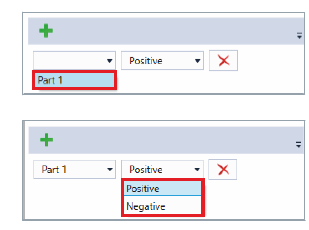
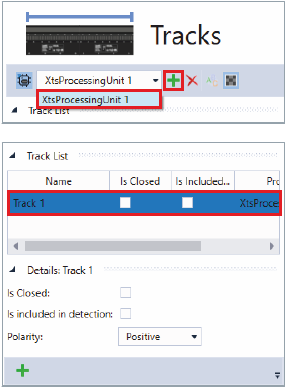
► Einbaulage des *XTS Parts* über das Feld **Alpha** eingeben

**36**



* Wählen die Reihenfolge der Devices entsprechend ihrem Auftreten im XTS System aus
* Der konfigurierte XTS Part wird angezeigt

**37**



* Erstellen einer XTS Track
* XTS Part auswählen, der dem aktuellen XTS Track hinzugefügt werden soll
* Polarität des XTS Parts auswählen : ob die Polarität des XTS Tracks in aufsteigender Zählrichtung positiv oder negativ ist.

**38**

* Der Parameter **Is Closed** bestätigt, dass die Strecke des XTS Systems geschlossen ist
* Der Parameter **Is included in detection** bestätigt, dass alle XTS Mover von einem XTS

Modul erkannt werden

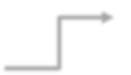
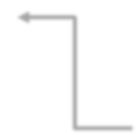
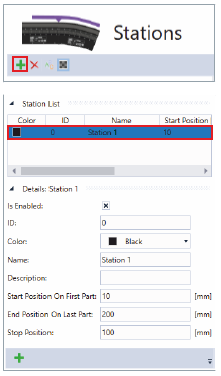
* müssen prüfen, ob die ausgewählte XTS-Tracks gültig ist und ob die XTS-Parts in der

richtigen Reihenfolge hinzugefügt wurden.

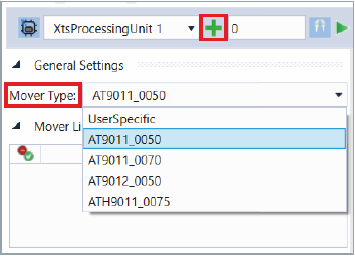
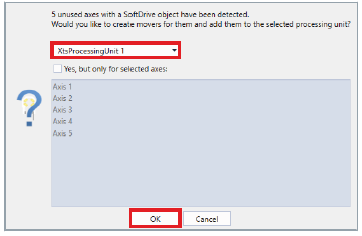
* Legen XTS Station an

Start Position On First Part , End Position On Last Part und Stop Position in Relation zur Länge des XTS Parts eingeben

Für jede XTS Station können Eigenschaften wie ID, Farbe, Name und Beschreibung vergeben.



**39**

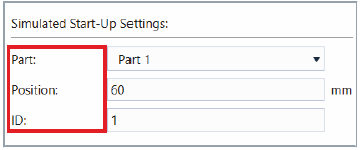
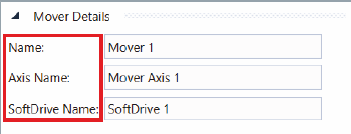
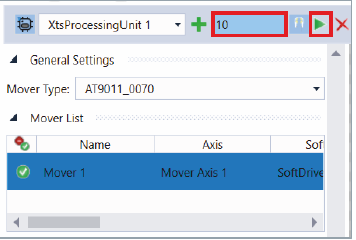


* Legen XTS Mover an
* **XTS Processing Unit** klicken, zu der *XTS Mover*

hinzugefügt werden sollen

* Einzelne Achsen auswählen
* **Mover Type** den entsprechenden XTS Mover klicken, der auf Ihrem XTS System vorhanden ist

**40**

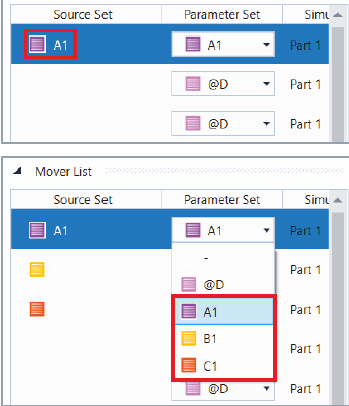


* Geben die Anzahl der benötigten XTS-Mover im

XTS-System ein.

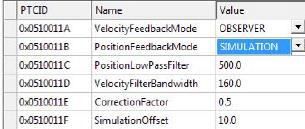
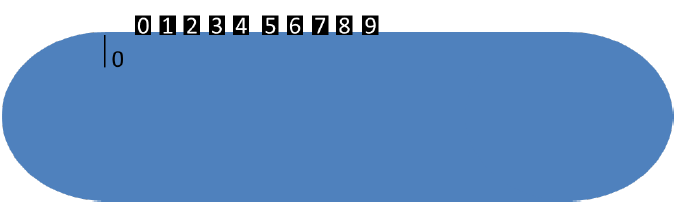
* Name für die jeweiligen Objekte vergeben
  + Wählen das Part, an dem der XTS Mover starten soll.
  + Bestimmen Sie die genaue Position des XTS Mover auf dem XTS Part.

**41**



* können **Source Sets** für mehrere XTS Mover festlegen. Source Sets werden fortlaufend nach den Buchstaben des Alphabets benannt.
* können Source Sets über das Dropdown-Menü der Parameter Sets auswählen und auf andere XTS Mover übertragen

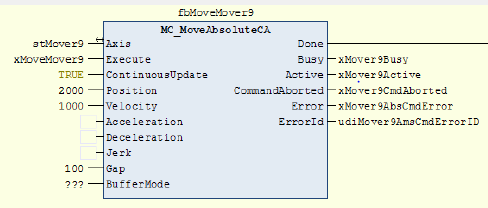
**42**



##### XTS-Programmierung

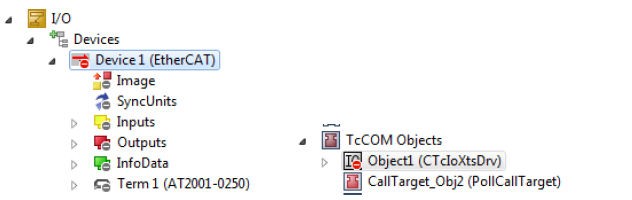
* Nachdem Konfiguration des TwinCAT-Systems, kann der Run-Modus gestartet werden.
* Alle Mover werden am Nullpunkt positioniert (Einspeise-Motormodul)
* Der Simulations-Vorgang wird im **Solution Explorer** durch den SoftDriveEncoder\_Obj2 gestartet.
* müssen alle Mover mit einer Startposition für die Simulation deklariert werden.
* muss man die Distanz von einem Mover zum anderen auf 100 mm einzustellen

**43**



* müssen über die SPS-Bausteine der Mover verschiedene Einstellungen vornehmen, die dafür sorgen, dass die Mover nicht miteinander kollidieren.
* ***Position* :** Vorgegebene relative Strecke für das Kommando z.B. 2000 ms
* ***Velocity*:** Die Geschwindigkeit ist durch die maximale Geschwindigkeit der Achse beschränkt
* ***Gap*:** Dieser Wert bestimmt den Mindestabstand zum Vorgänger für die **Collision Avoidance**.
* ***ContinuousUpdate*:** In dieser Version steht die kontinuierliche Aktualisierung nur für den Gap zur Verfügung.

**44**

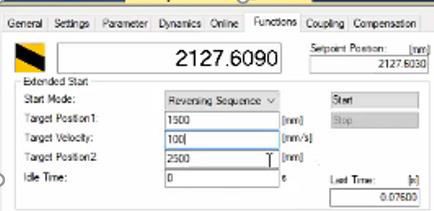
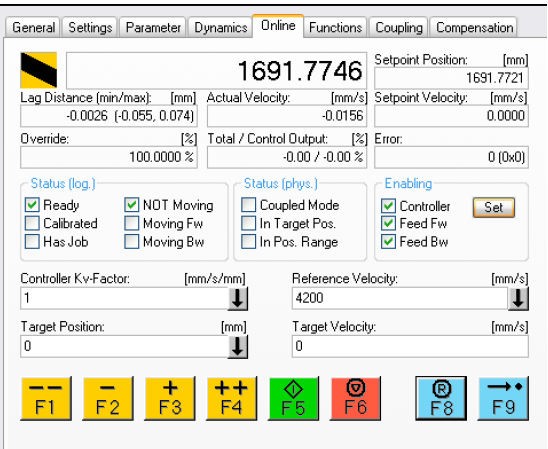


* müssen im **Solution Explorer** die Bäume Device 1 (EtherCAT) und Object1 (CTcIoXtsDrv)

ausgestellt werden

* muss das System neu gestartet werden, wenn alle Einstellungen vorgenommen worden sind und die Simulation gestartet werden kann

**45**



* die Mover auf dem XTS-System bewegen.
  + Mit den Funktionen F1 - F9 können Mover auf der Anlage bewegt werden
  + muss der Wert im Feld „Override“ auf > 0 eingestellt werden
* kann mann den Startpunkt, die Endpunkte und die Geschwindigkeit für jedes Mover festlegen

**46**



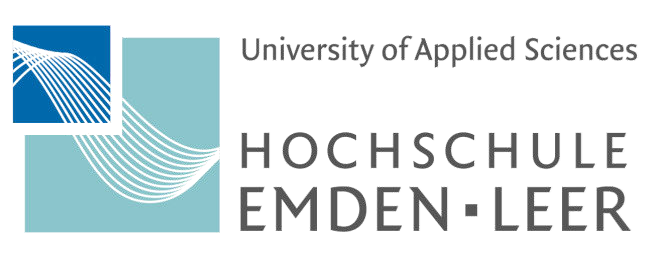
###### KUKA.EthernetKRL 2.2

***TwinCAT 3 OPC UA Handbuch***

***XTS Software Handbuch***

***infosys.beckhoff.com***

**47**



# Vielen Dank für Ihre

*Aufmerksamkeit*