

“Effiziente Vernetzung dezentraler Intelligenz”

Arbeitsgruppe AUT an der Ostbayerischen Technischen Hochschule Amberg-Weiden
& aia automations institut GmbH

“I4.0 Small Smart Factory Demonstrator”

M.Sc. Maximilian Bauer
M.Eng. Sebastian Schaffenroth
B.Eng. Katharina Lutter
M.Eng. Oleksandr Volodin

F&E Arbeiten & Ergebnisse aus dem Forschungsprojekt „[ISAC@OTH-AW](#)“

Überblick

- Konzept des Smart Factory Demonstrators
- Kommunikation (OPC UA, PROFINET, MQTT...)
- Digitaler Zwilling
- Mongo-Datenbank
- Live-Vorführung des Demonstrators
- Bildverarbeitung mit RaspberryPi und OpenSource Software
- „Nachbarschaftserkennung“ der Produktionseinheiten via Barcodescanner
- NFC Scanner
- PROFINET OPC UA Companion Standard
- Fazit

Konzept des Smart Factory Demonstrators

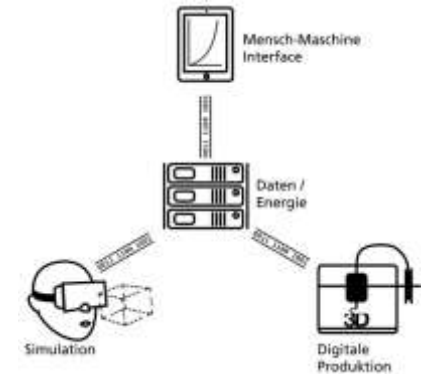
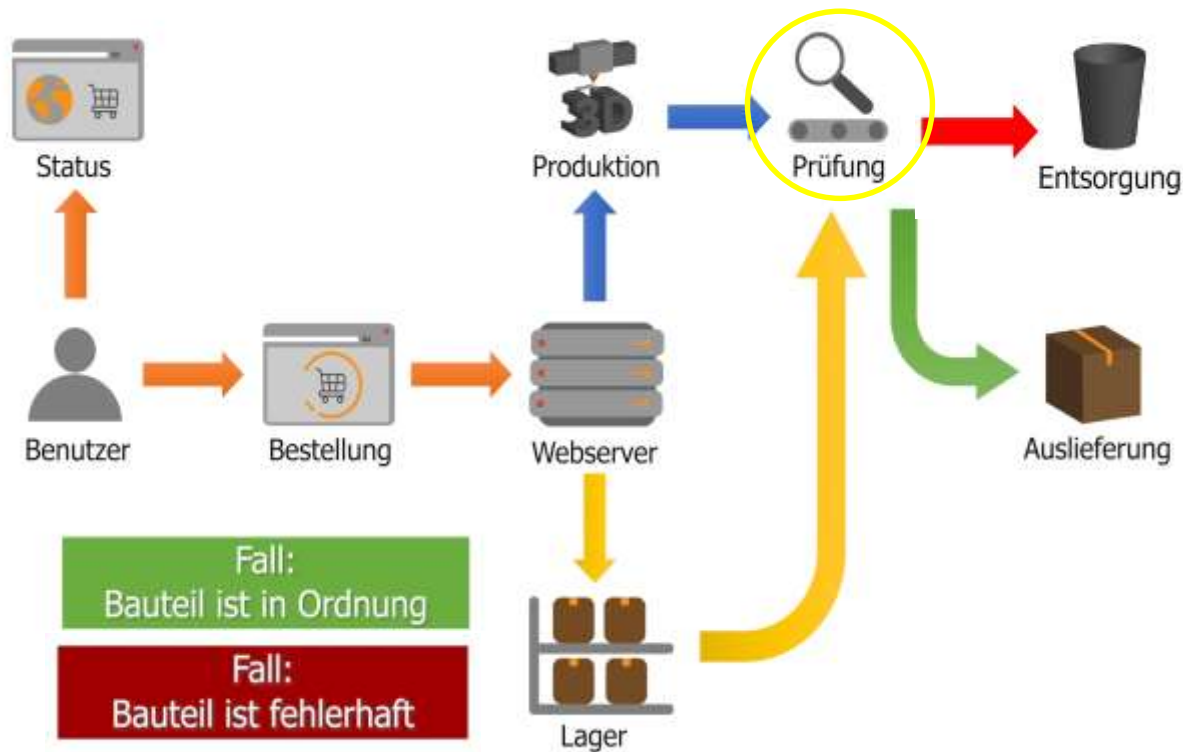


Konzept des Smart Factory Demonstrators

- Modular aufgebaute Produktionsanlage bestehend aus 10 Zellen und 2 Roboterzellen (frei positionierbar)
- Idee Industrie 4.0: Fertigung in Losgröße 1
- Nutzung desselben SPS Programms für alle Zellen:
 - Festlegung vorhandener Module pro Zelle (z.B. NFC Scanner, Kamera, Förderbänder....)
 - Aktivierung/Deaktivierung der Profinet Devices je nach Bedarf



Konzept des Smart Factory Demonstrators



<https://www.isac-oth.de/ergebnispraesentation/>

- Gemeinsamer Demonstrator aller Teilprojekte des ISAC Projektes
- Beitrag Teilprojekt 3: „Werkstückprüfung“ als Beispiel einer digitalen Industrie 4.0 Produktion
- Werkstückdaten aufnehmen (NFC), Qualitätskontrolle durchführen (Bildverarbeitung), Werkstückdaten um Ergebnis ergänzen (NFC) und Werkstück ausschleusen

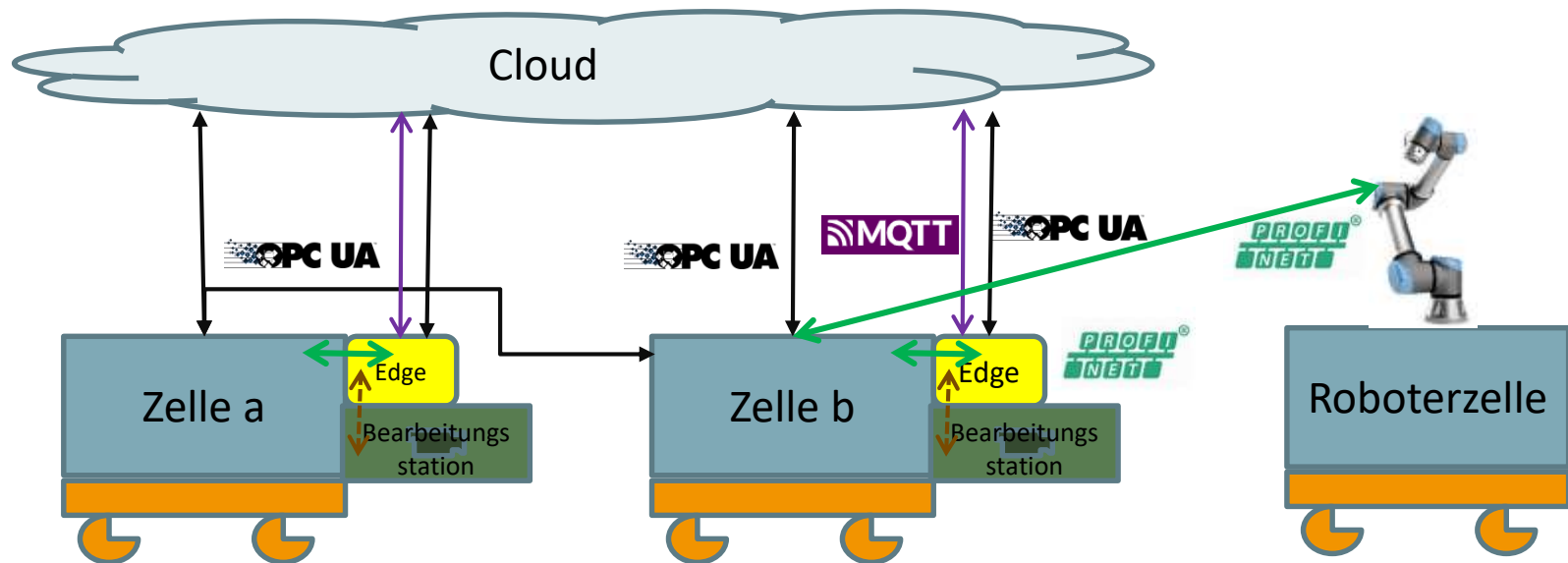


Kommunikation



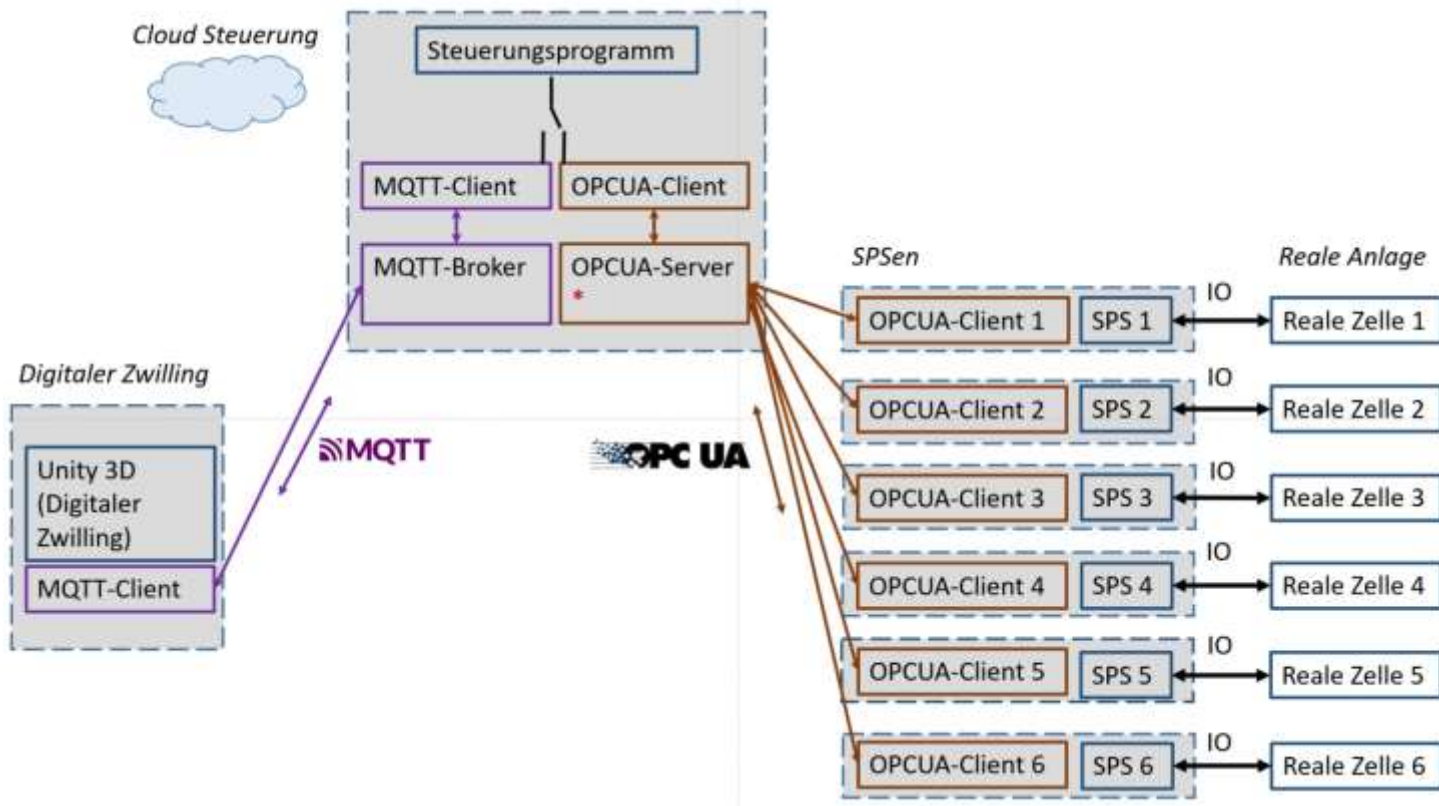
Kommunikation – Grundlegende Struktur

- **OPC UA** Server auf der SPS (eine ET200SP pro Zelle) erlaubt Zugriff auf Daten aus der Cloud heraus
- Kommunikation mit Peripherie (Roboter, Raspberry Pi 4 mit Bildverarbeitungssoftware, Raspberry Pi mit NFC Scanner usw...) über **Profinet**
- Kommunikation zwischen den einzelnen SPSen über **OPC UA**
- Ein Raspberry Pi 4 auf jeder Zelle fungiert als Edge Gateway
→ Kommunikation via **OPC UA**, wahlweise **MQTT**



Kommunikation – Steuerungskonzepte

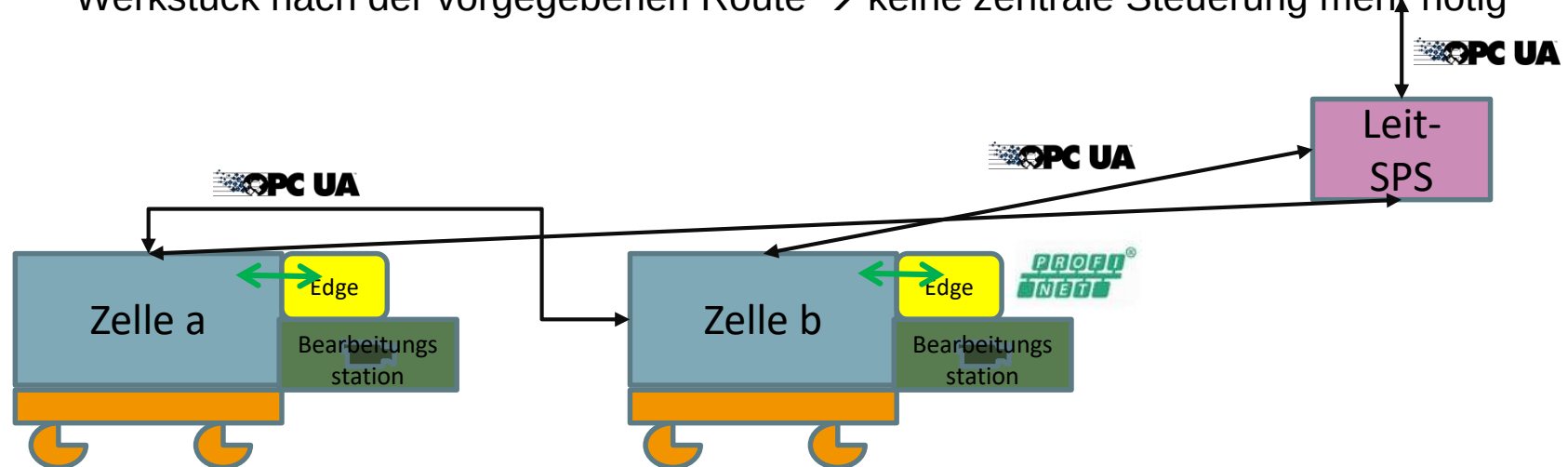
- Zentrale Steuerung aus der Cloud
 - OPCUA Clients auf den SPSen
 - SPSen als reine IO-Devices
 - Wahlweise auch Steuerung des digitalen Zwillings mit demselben Programm



Kommunikation – Steuerungskonzepte

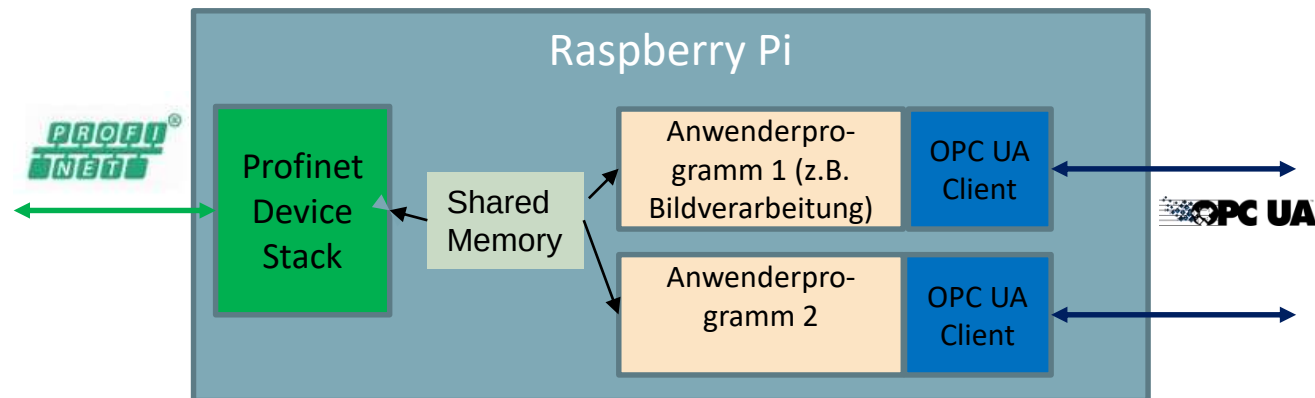
• Dezentrale Steuerung

- Zellen erkennen jeweils ihre Nachbarzellen via Barcodescanner-Profinet-Devices
- Leit-SPS/Leitrechner erhält Informationen zum Betriebszustand und zur Konfiguration (Nachbarschaftsbeziehungen, ausführbare Prozesse) der Zellen von allen SPSen via OPC UA
- Bediener gibt Reihenfolge der Prozesse (z.B. für Anwendungsfall „Qualitätskontrolle“) vor
- Leit-SPS/Leitrechner berechnet Werkstückträgerroute und teilt diese allen Zellen via OPC UA mit
- Zellen kommunizieren über OPC UA miteinander und bearbeiten Werkstück nach der vorgegebenen Route → keine zentrale Steuerung mehr nötig



Kommunikation - Implementierung

- **Raspberry Pi 4:**
 - open62541 Stack für OPC UA (open source)
 - mosquitto Bibliothek für MQTT (open source)
 - Phoenix Contact Stack für Profinet
→ Programm mit Shared Memory Konzept für Zugriff von Anwenderprogrammen (z.B. Bildverarbeitung) auf Profinet-Schnittstelle



- **SPS:**
 - Nutzung von OPC UA Server- und Client-Anwenderbausteinen aus dem TIA Portal von Siemens

Digitaler Zwilling

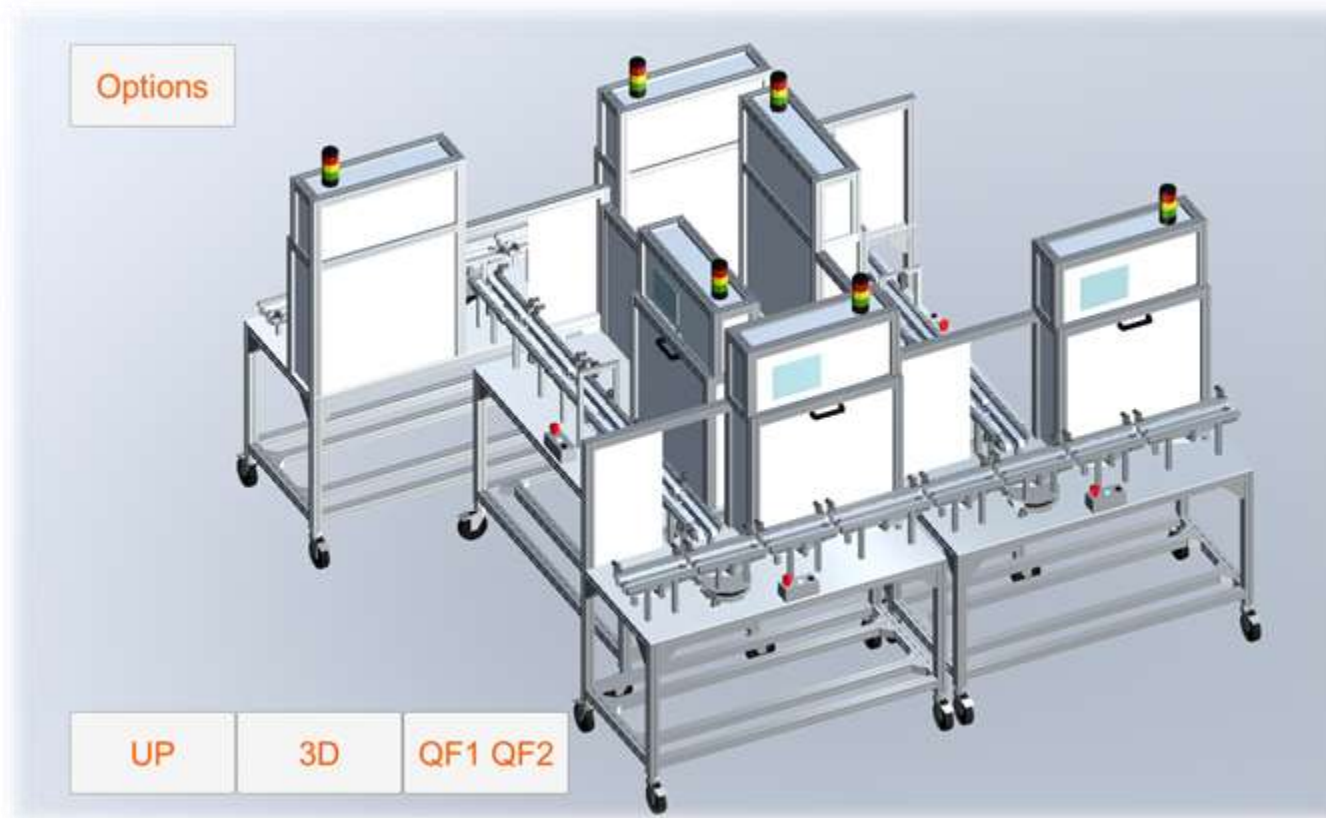


AUT

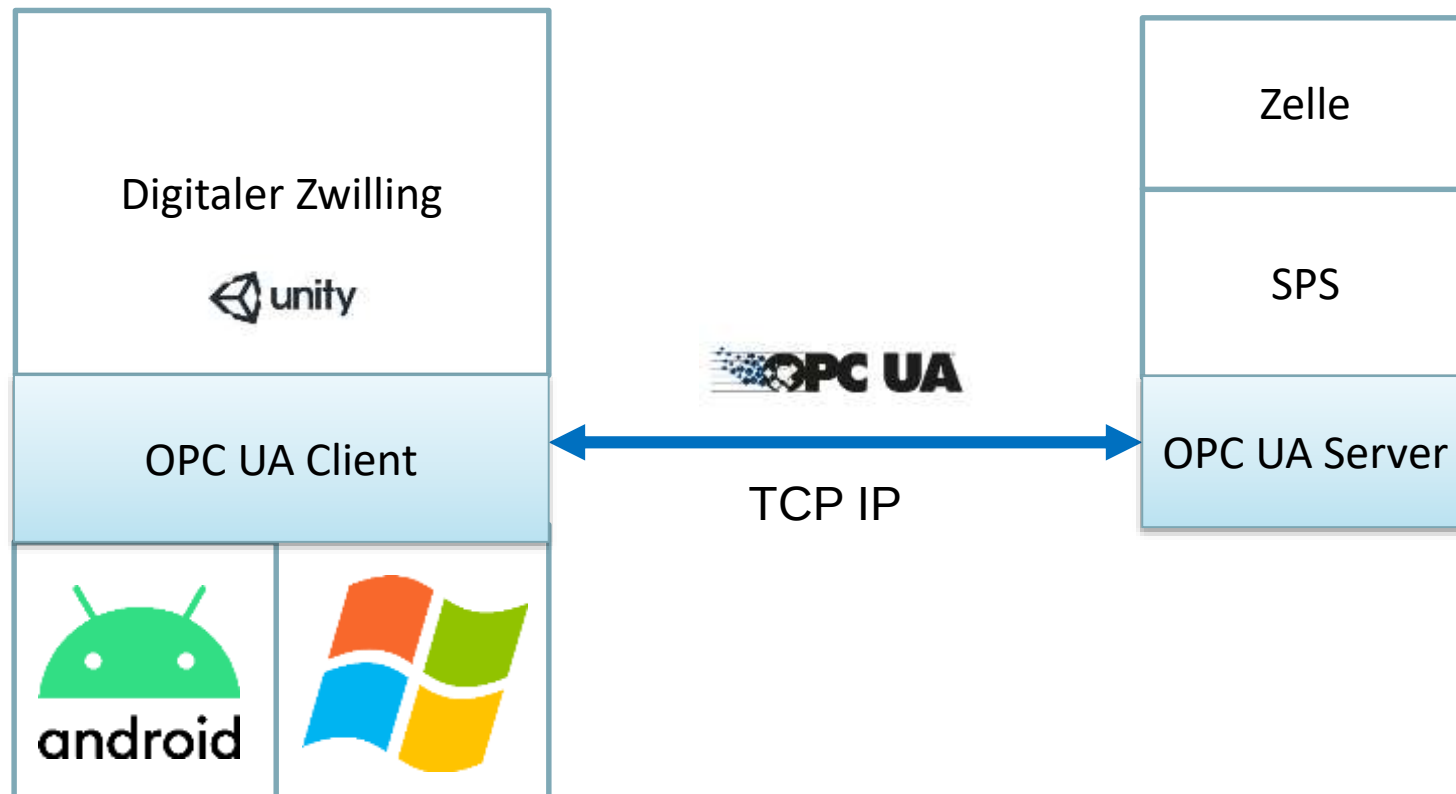
Arbeitsgruppe Automation Amberg



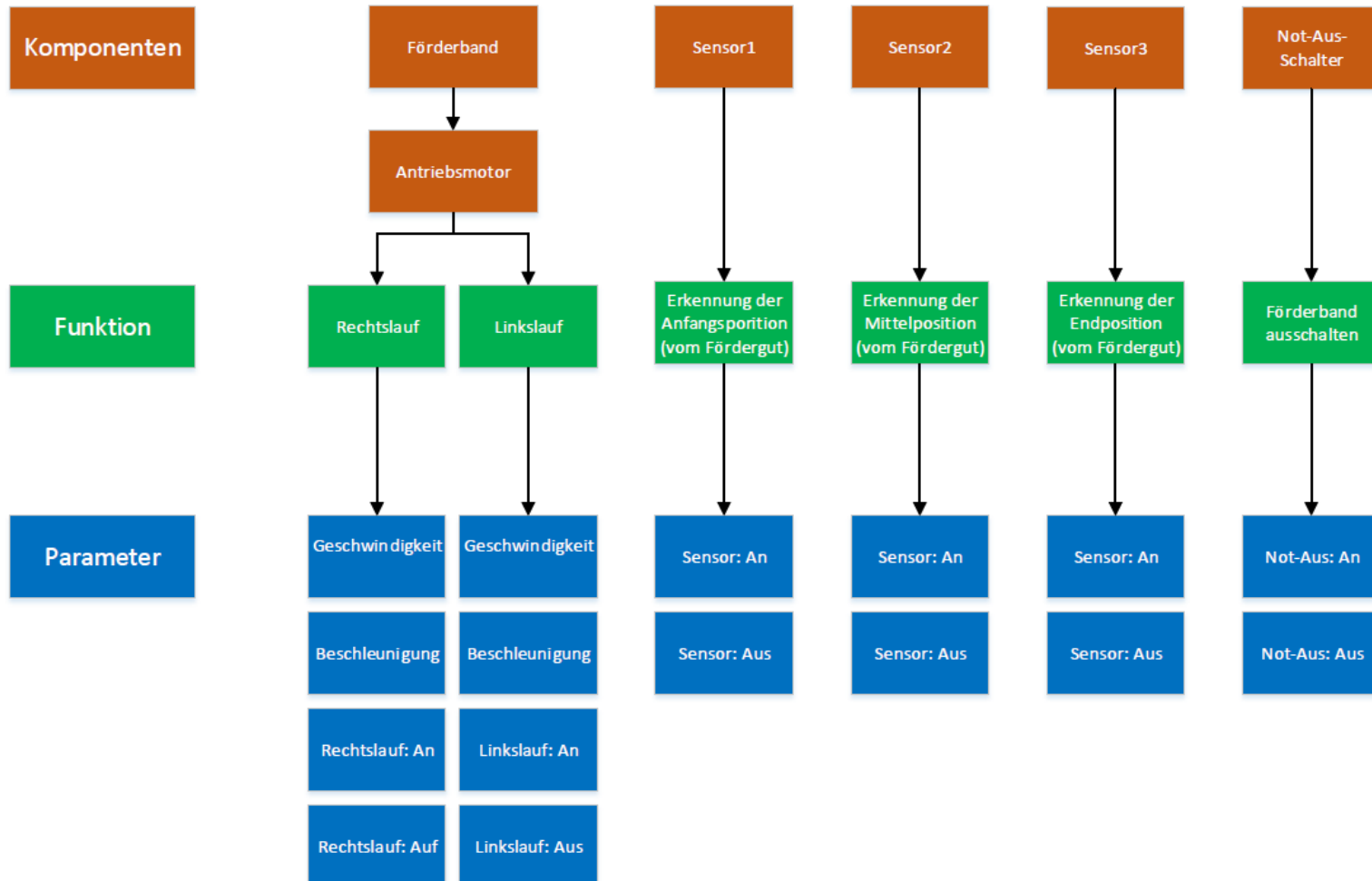
Ostbayerische Technische Hochschule
Amberg-Weiden



Digitaler Zwilling-Konzept



Informationsmodell für Zellen



Digitaler Zwilling Programm

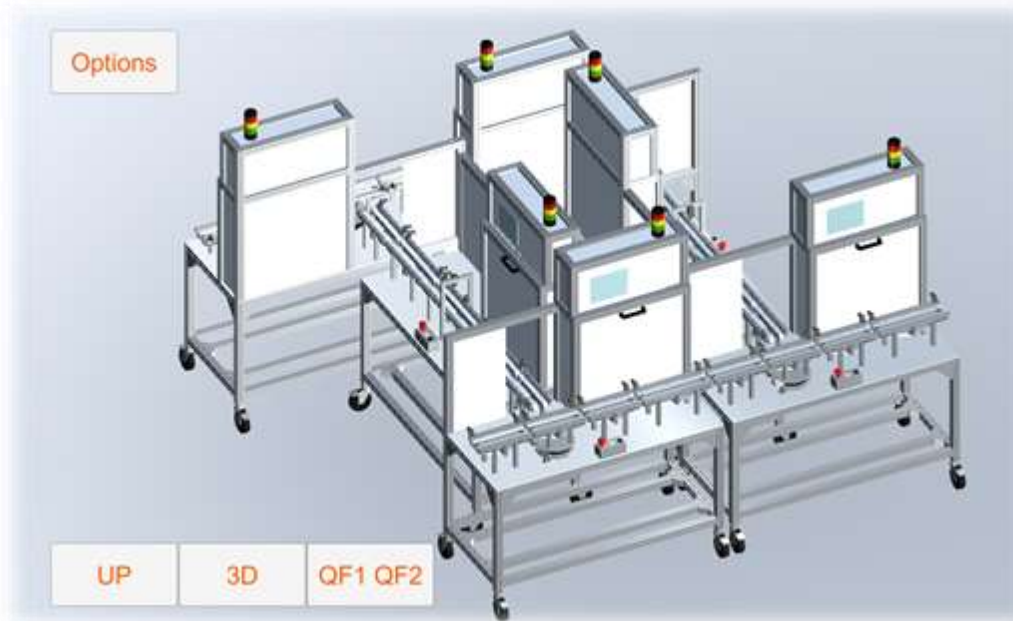


AUT

Arbeitsgruppe Automation Amberg



Ostbayerische Technische Hochschule
Amberg-Weiden



Options

UP

3D

QF1 QF2

Back

QF1

QF2

QF3

QF4

L1

L2



Green Light	True	Sensor 1 L	False	FB 1 ON	False
Yellow Light	False	Sensor 1 R	False	FB 1 RL	False
Red Light	True	Sensor 2 L	False	FB 2 ON	False
		Sensor 2 R	False	FB 2 RL	False
Server status	Connected	Sensor AS	False	Rotation RL	False
		Sensor GS	True	Rotation ON	False
		Sensor 3 L	False	FB 3 ON	False
		Sensor 3 R	False	FB 3 RL	False
		Sensor 4 L	False	FB 4 ON	False
		Sensor 4 R	False	FB 4 RL	False
		Sensor 5 L	False	FB 5 ON	False
		Sensor 5 R	False	FB 5 RL	False

Start

Stop

IP write

Default

Enter IP-address...

Mongo-Datenbank

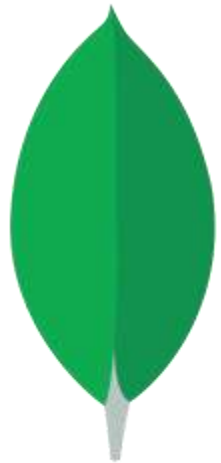


AUT

Arbeitsgruppe Automation Amberg

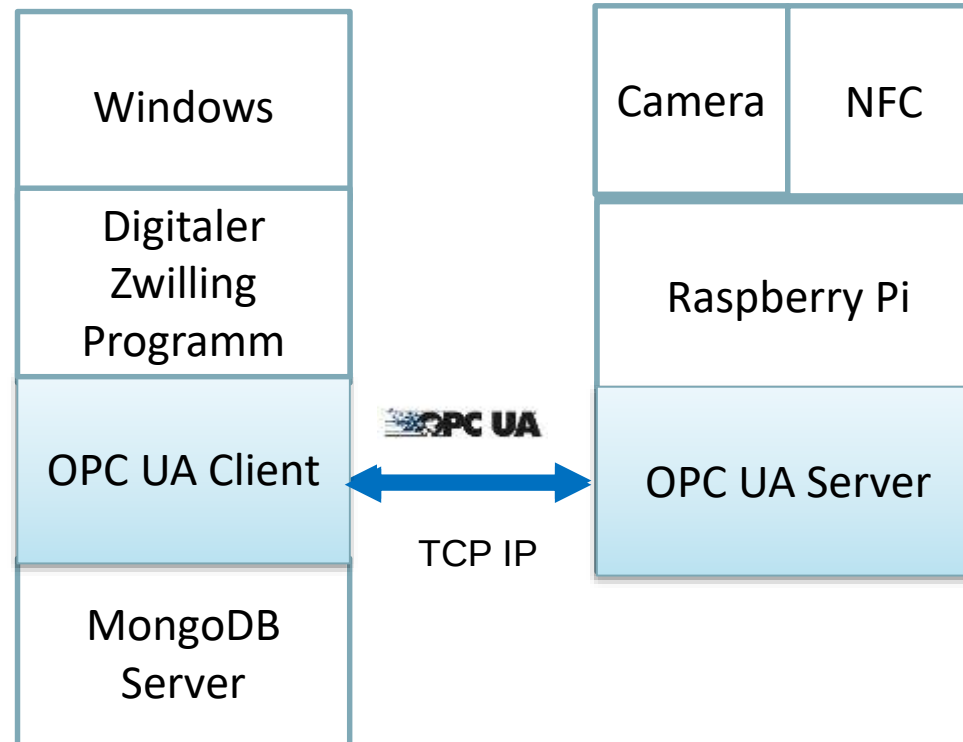


Ostbayerische Technische Hochschule
Amberg-Weiden



mongoDB®

Mongo-Datenbank



Mongo-Datenbank

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER [field: 'value'] OPTIONS FIND RESET ↺ ...

ADD DATA VIEW {}

Displaying documents 1 - 2 of 2 < > C REFRESH

```
{
  "_id": ObjectId("614c64b8b22c86eb6069b11c"),
  "NFC number": "16",
  "Bestellnummer": "vs-92439-191216-gr-tri-2",
  "Farbe Soll-Wert Bestell...": "Orange",
  "Form Soll-Wert Bestell...": "GuyFawkes",
  "Datenbank-ID": "0",
  "E-Mail": "k.lutter@oth-aw.de",
  "Nachname": "Blue",
  "PLZ": "Kathi",
  "Vorname": "92439",
  "Farbe Ist-Wert": "Orange",
  "Farbe Soll-Wert": "Orange",
  "Form Ist-Wert": "MattPohl",
  "Form Soll-Wert": "MattPohl",
  "Status-ID": "3",
  "Status-Meldung": "AufdemWegzumLager",
  "Farbe": "1",
  "Funktionstest": "1",
  "Geometrie": "1",
  "Handshake": "1",
  "Pruefabschluss-Zeitpun...": "2021.09.23 01:27:26",
  "Pruefbeginn-Zeitpunkt": "2021.09.23 01:25:54",
  "System Zeit": "2021.09.23 01:25:57"
}
```

Raspberry Pi OPC-UA Server



AUT

Arbeitsgruppe Automation Amberg



Ostbayerische Technische Hochschule
Amberg-Weiden

Back

RP Start

RP Camera

RP End

MongoDB

Server status **Connected**

Pruefablauf

Bestellung

Pruefbereit 0

Bestellnummer vs-92224-191216-gr-tri-1

Farbe Soll-Wert Blau

Form Soll-Wert GuyFawkes

Kunde

E-Mail d.trump@oth-aw.de

Nachname Trump

Vorname 92224

PLZ Donald

Datenbank-ID 0

Farbe Ist-Wert 0

Form Ist-Wert 0

Farbe 0

Geometrie 0

Abschlusszeit 0

Beginnzeit 0

System Zeit 2021.09.27 07:34:49

Status-ID 1

Status Meldung Wirdgeprüft

Werkstueck-Id 0



IP write

Default

Enter IP-address...

Live-Vorführung des Demonstrators



Bildverarbeitung mit Raspberry Pi und OpenSource Software

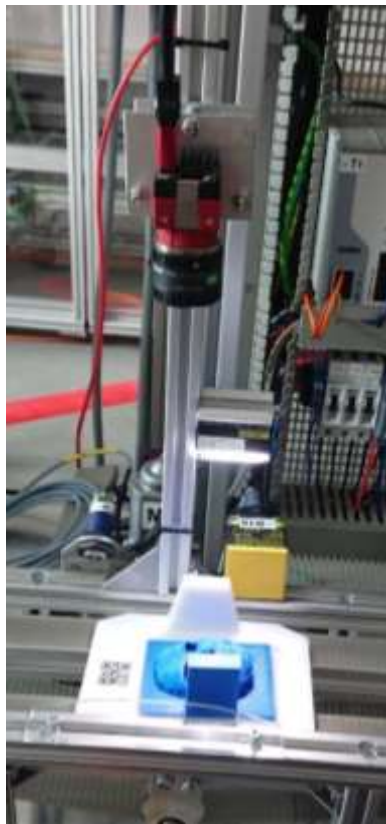


AUT

Arbeitsgruppe Automation Amberg

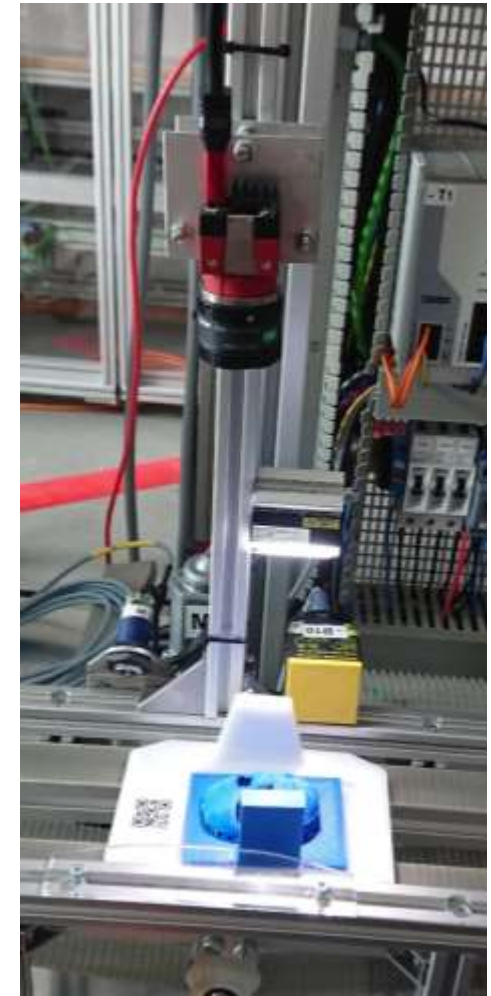
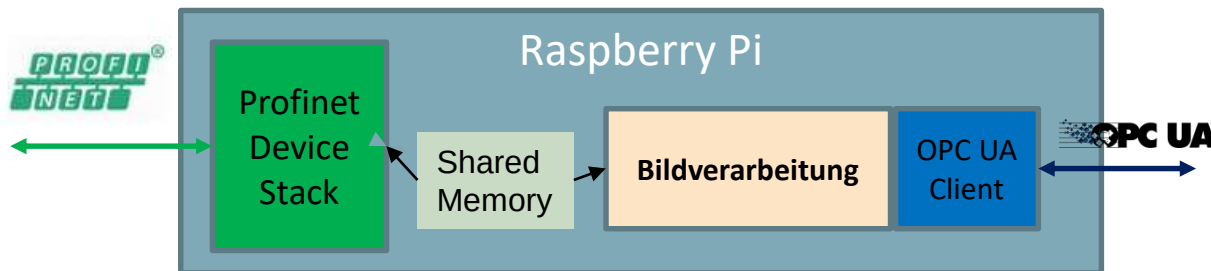


Ostbayerische Technische Hochschule
Amberg-Weiden



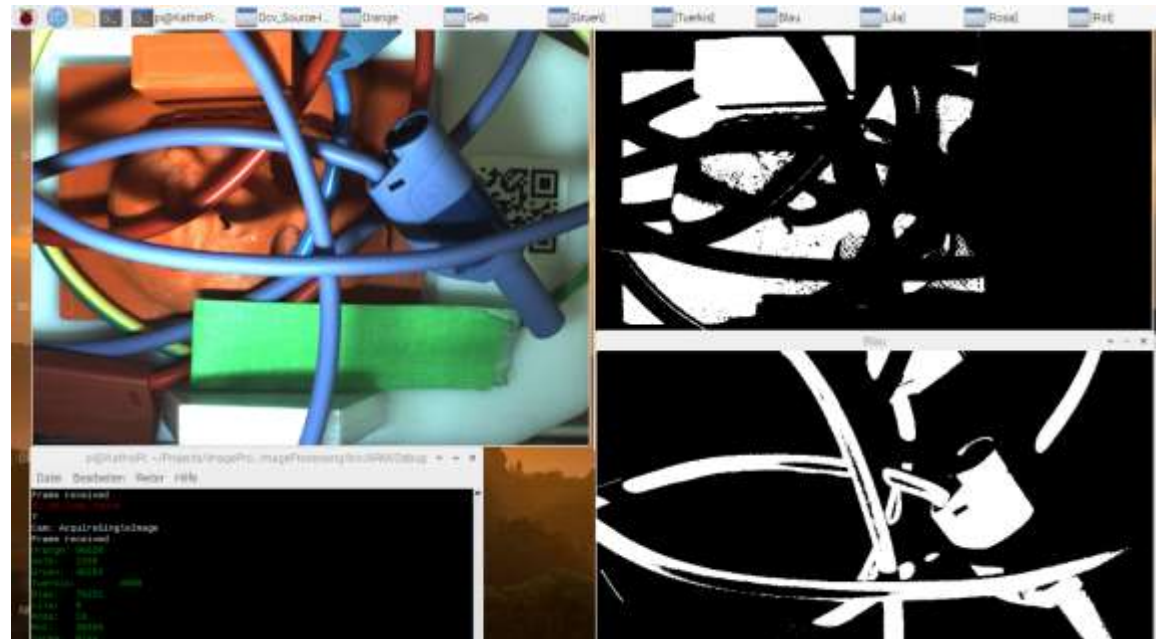
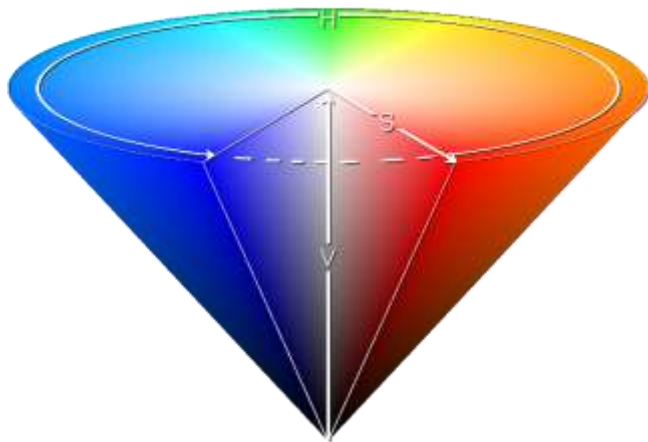
Bildverarbeitung mit Raspberry Pi und OpenSource Software - Grundlagen

- Raspberry Pi Modell 4 mit Linux Betriebssystem
- Kamera von Allied Vision mit USB-Anbindung
- C++ Programm für Zugriff auf Kamera (unter Nutzung der Vimba SDK von Allied Vision)
- Bildverarbeitung mit opencv Bibliothek (kostenlos und open source)
- Nutzung als Smart Sensor mit Profinet und OPCUA Schnittstelle



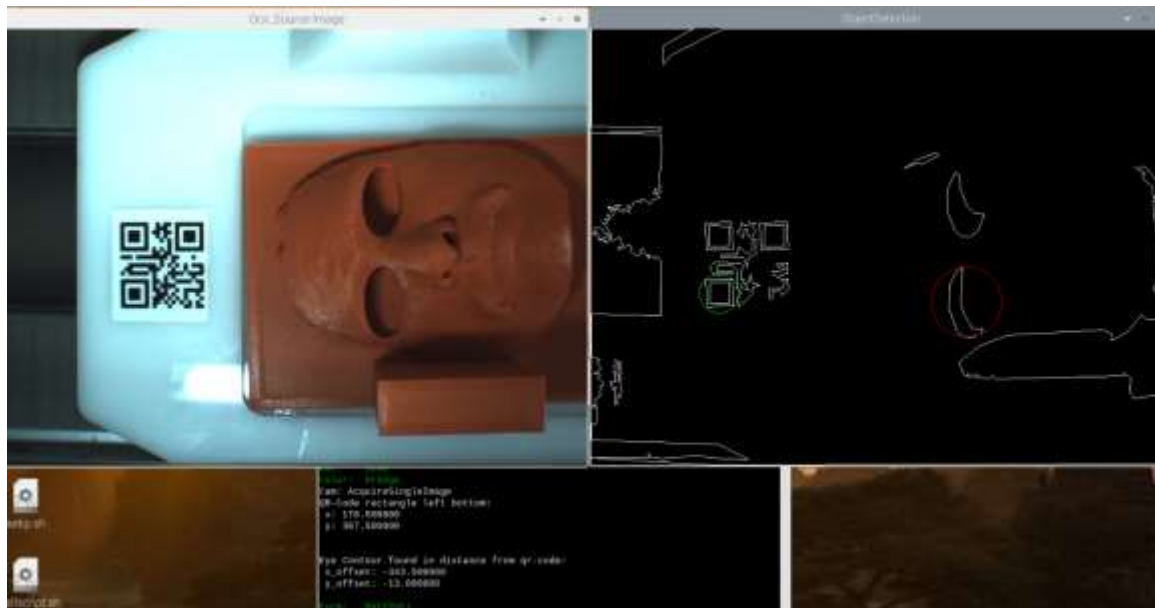
Bildverarbeitung mit Raspberry Pi und OpenSource Software - Anwendungsbeispiele

- Farberkennung Werkstück
 - Ein Filter je „Grundfarbe“, Nutzung des HSV-Farbraums für Filtergrenzen
 - Zählen der Pixel
 - Ermittlung der Hauptfarbe



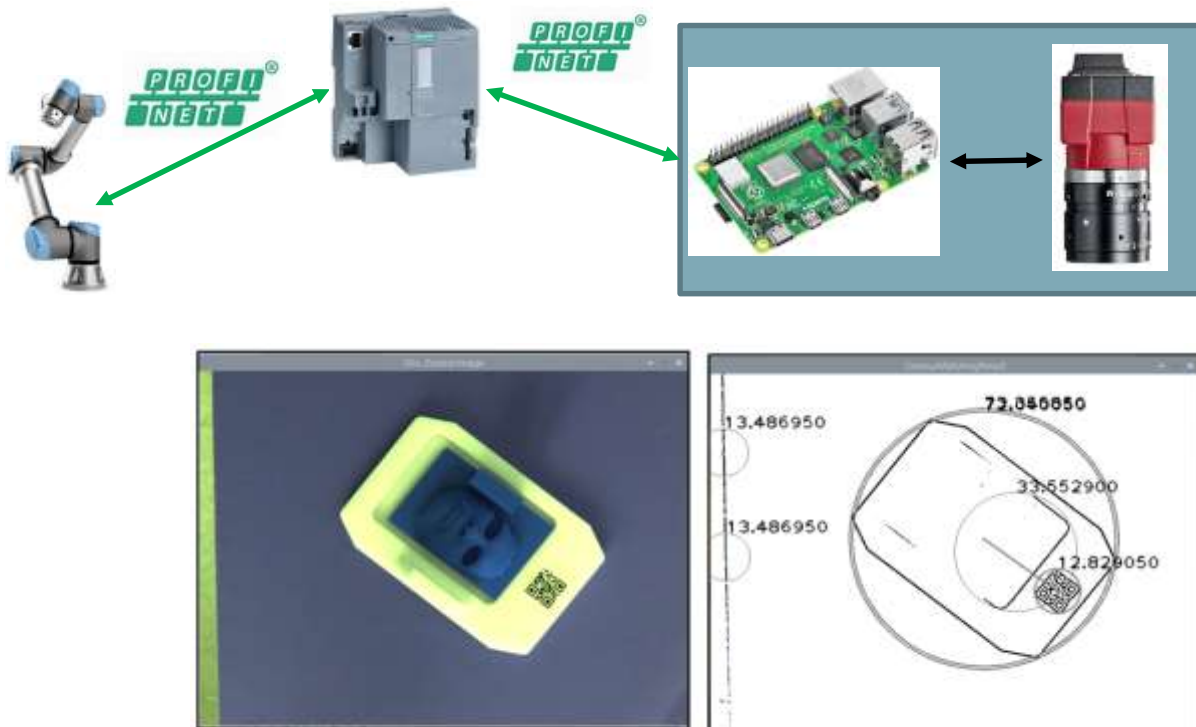
Bildverarbeitung mit Raspberry Pi und OpenSource Software - Anwendungsbeispiele

- Formerkennung Werkstück
 - Unterscheidung zweier unterschiedlicher „Masken“
 - Erkennung der „MattPohl-Maske“ aufgrund der Kontur des unteren Auges
 - Identifizierung charakteristischer Konturen



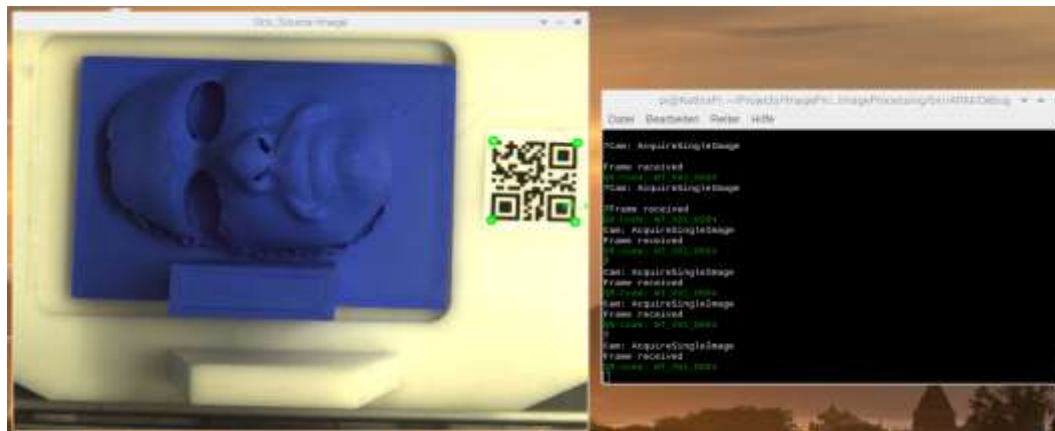
Bildverarbeitung mit Raspberry Pi und OpenSource Software - Anwendungsbeispiele

- Lokalisierung Werkstückträger
 - Roboter-Kamera Kollaboration
 - Alternative für High-Tech Roboter Kamera
 - Ermittlung der x und y Koordinaten sowie des Winkels

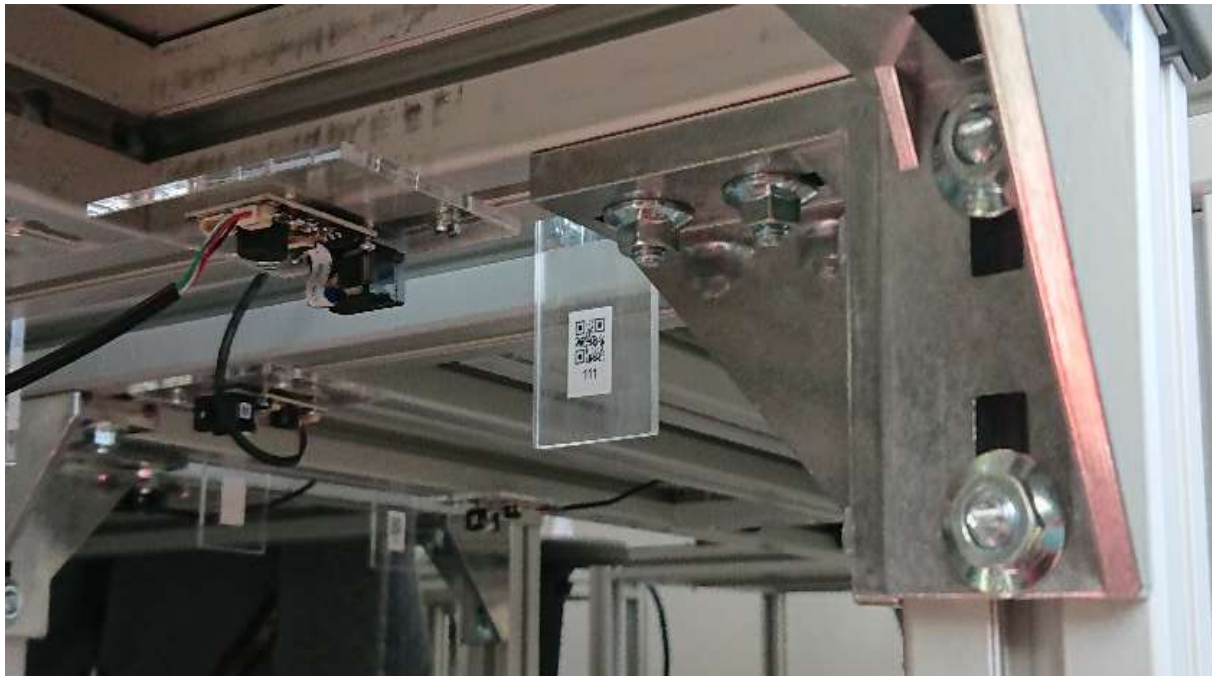


Bildverarbeitung mit Raspberry Pi und OpenSource Software - Anwendungsbeispiele

- QR-Code detektieren und lesen (zbar open source Bibliothek)
- Vergleich zweier Bilder auf Ähnlichkeit (z.B. genutzt zur Unterscheidung, ob Werkstückträger leer oder nicht)
- Template Matching mit Werkstückträger als Template (erkennt aber Rotation nicht)

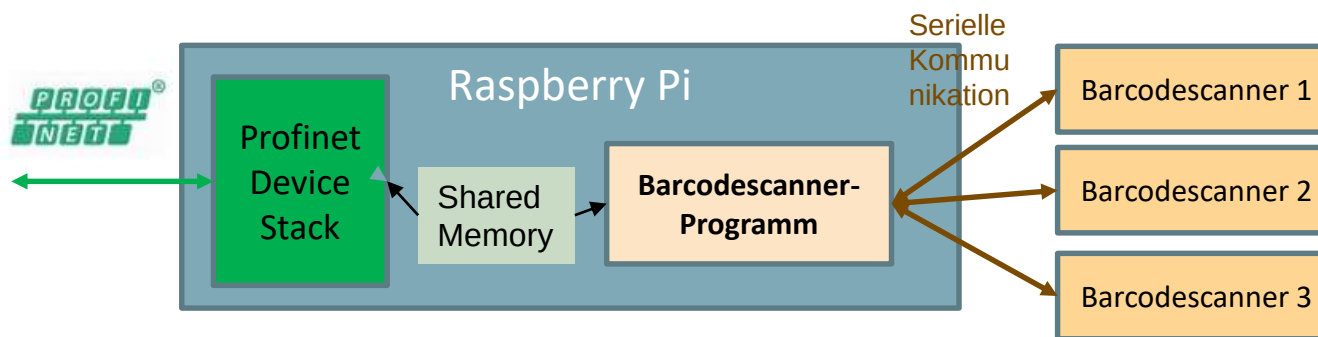
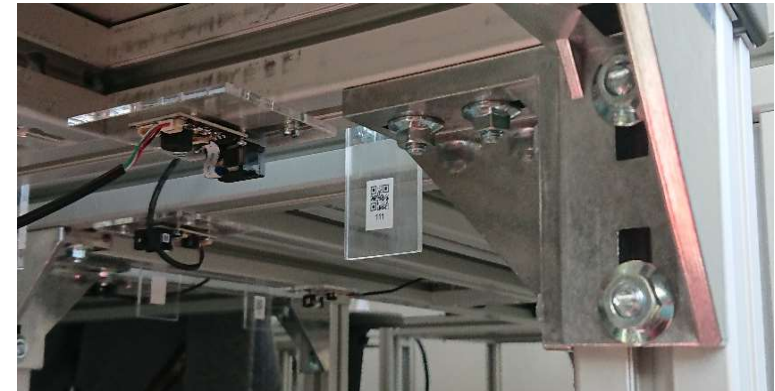


„Nachbarschaftserkennung“ der Produktionseinheiten via Barcodescanner

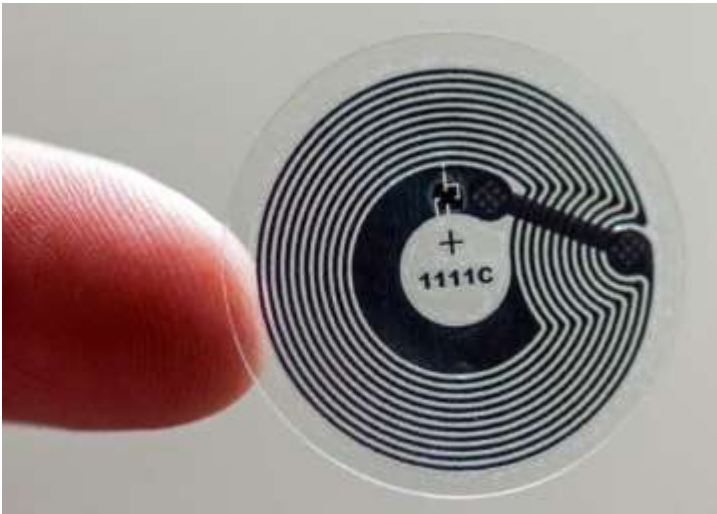


„Nachbarschaftserkennung“ der Produktionseinheiten via Barcodescanner

- GM 65 Barcodescanner mit USB 2.0 und UART Schnittstelle (Low-Cost: < 30 €)
- Anbindung via USB an Raspberry Pi 4
- Anwenderprogramm realisiert Kommunikation mit Barcodescanner über virtuelle serielle Schnittstelle (USB)
- Nutzung als Smart Sensor mit Profinet Schnittstelle

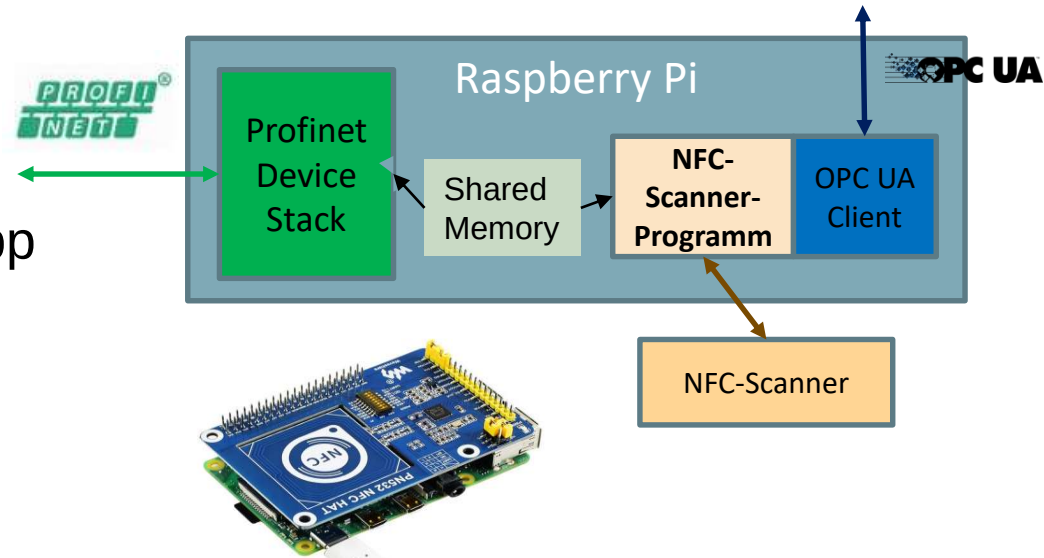


NFC Scanner



NFC-Scanner

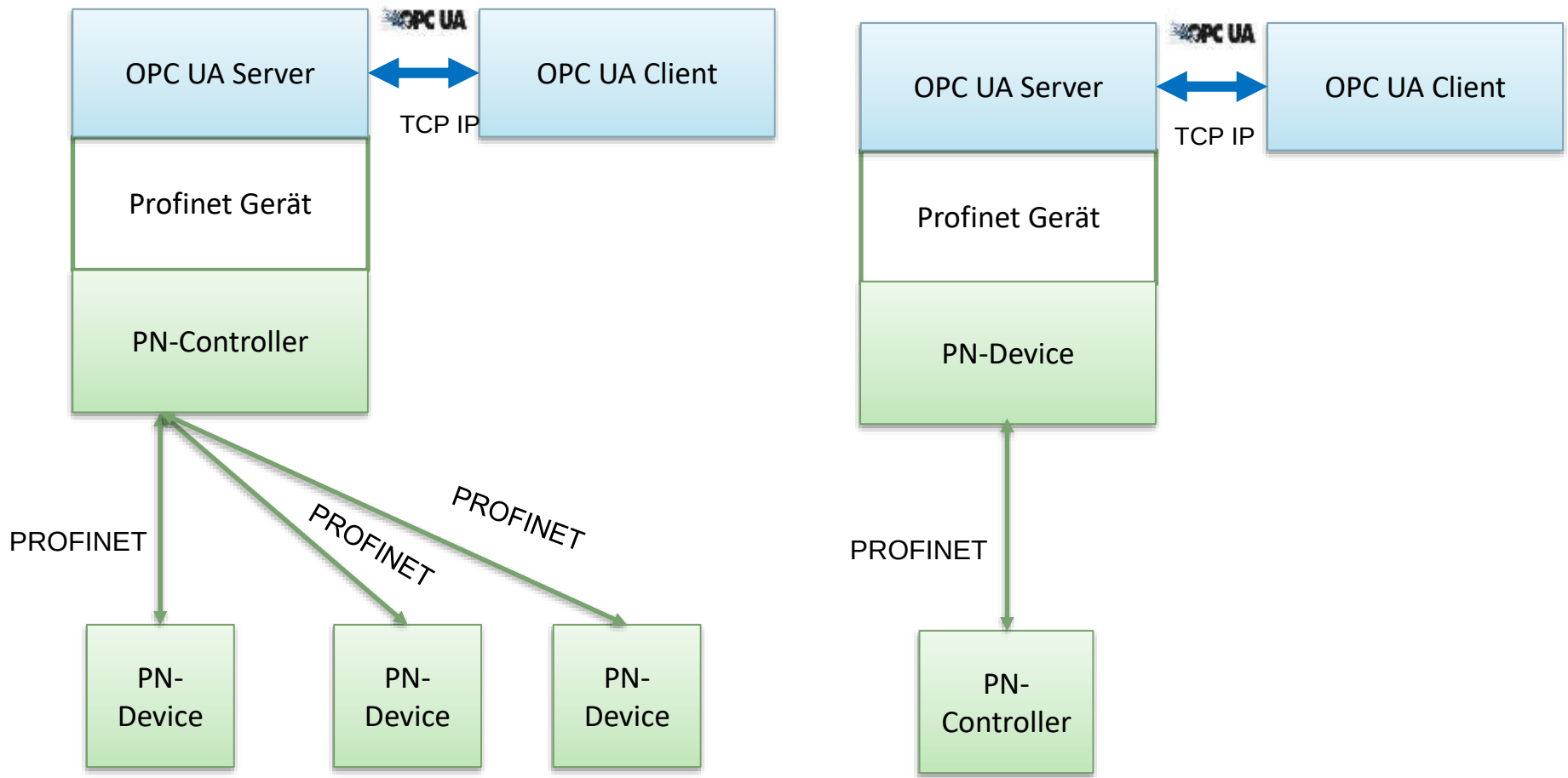
- Werkstückidentifikation durch eindeutige ID (auch bekannt durch SPS)
- Werkstückdaten befinden sich auf NFC Chip am Werkstück
- MongoDB-Datenbank zur Anbindung an überlagerte IT
- Nutzung eines Raspberry Pi 4 mit Erweiterungsboard „PN532 NFC HAT“ für das Lesen und Schreiben von Daten von/auf NFC Tags
- Codierung der Daten nach dem NFC Data Exchange Format (NDEF)
 - Daten per Smartphone App auslesbar
 - Nutzung der ndeflib Bibliothek (open source) für NFDEF Codierung



PROFINET OPC UA Companion Standard



PROFINET OPC UA Companion Standard Konzept



OPC UA for PROFINET Companion Specification



OPC 30140

OPC UA for PROFINET

Release 1.00

2020-01-20

OPC UA Companion-Specification

Table 59 – EthernetInterfaceType Definition

Attribute	Value				
BrowseName	EthernetInterfaceType				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule
Subtype of <i>NetworkComponentType</i> .					
HasComponent	Variable	MacAddress	Byte [6]	BaseDataVariableType	Mandatory
CommLinkTo	Object	<PortName>		EthernetPortType	MandatoryPlaceholder

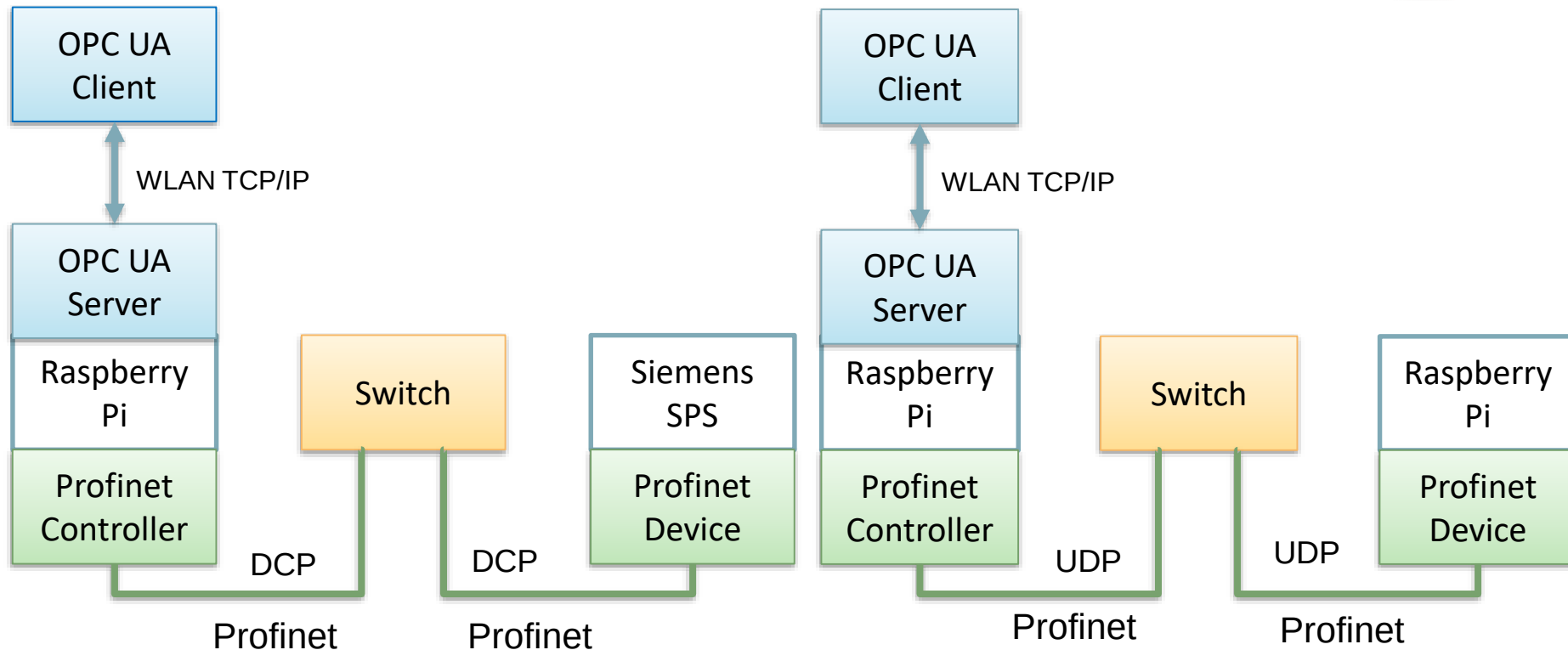
Table 60 – EthernetPortType Definition

Attribute	Value				
BrowseName	EthernetPortType				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule
Subtype of <i>NetworkComponentType</i> .					
HasComponent	Variable	PhysAddress	Byte [6]	BaseDataVariableType	Optional
CommLinkTo	Object	<EthernetPort>		EthernetPortType	Optional

6.3.3.3.2 PnARStateEnumeration

Name	Description
CONNECTED_0	The AR connection to the device is established
UNCONNECTED_1	The AR connection to the device is not established
UNCONNECTED_ERR_DEVICE_NOT_FOUND_2	The AR connection to the device is not established because the device is not available in the network
UNCONNECTED_ERR_DUPLICATE_IP_3	The AR connection to the device is not established because the IP address of the device exists multiple times
UNCONNECTED_ERR_DUPLICATE_NOS_4	The AR connection to the device is not established because the Name of Station of the device exists multiple times

Raspberry Pi als Profinet Controller

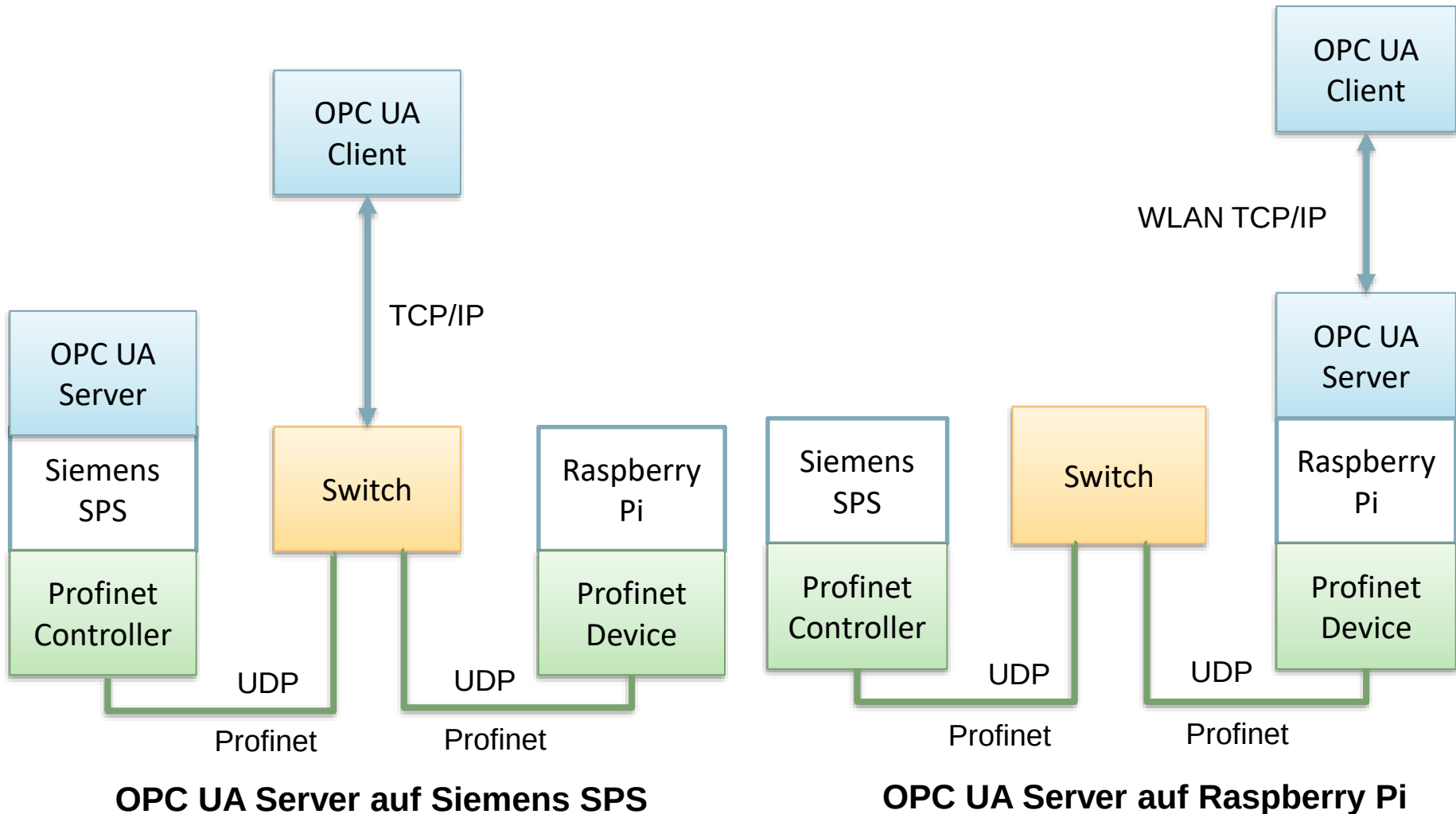


Siemens SPS als Profinet Device

Zweiter Raspberry Pi als Profinet Device

DCP - Discovery and Configuration Protocol

Siemens SPS als Profinet Controller



OPC UA - Informationsmodell

SIEMENS **SiOME - OPC UA Modeling Editor**

Information model offline + - lock unlock Show all

▼ **OPC Root**

▼ **OPC Objects**

▼ **ProfinetCompanionStandard**

▼ **Devices**

▼ **Device1**

▼ **GSDMLInfo**

▼ **IPConfig**

▼ **DefaultGateway (String)**

▼ **IPAddress (String)**

▼ **StationName (String)**

▼ **SubnetMask (String)**

▼ **Status (String)**

▶ **Server**

Attributes

Nodeid	ns=1;i=1025
NodeClass	Variable
BrowseName	1:DefaultGateway
DisplayName	DefaultGateway
Description	null
WriteMask	0
UserWriteMask	0
RolePermissions	
Value	0
DataType	String
ValueRank	Scalar
ArrayDimensions	[]
AccessLevel	3

Datenempfang über DCP

Project

- Project
 - Servers
 - open62541-based OPC UA Application
 - Documents
 - Data Access View

Address Space

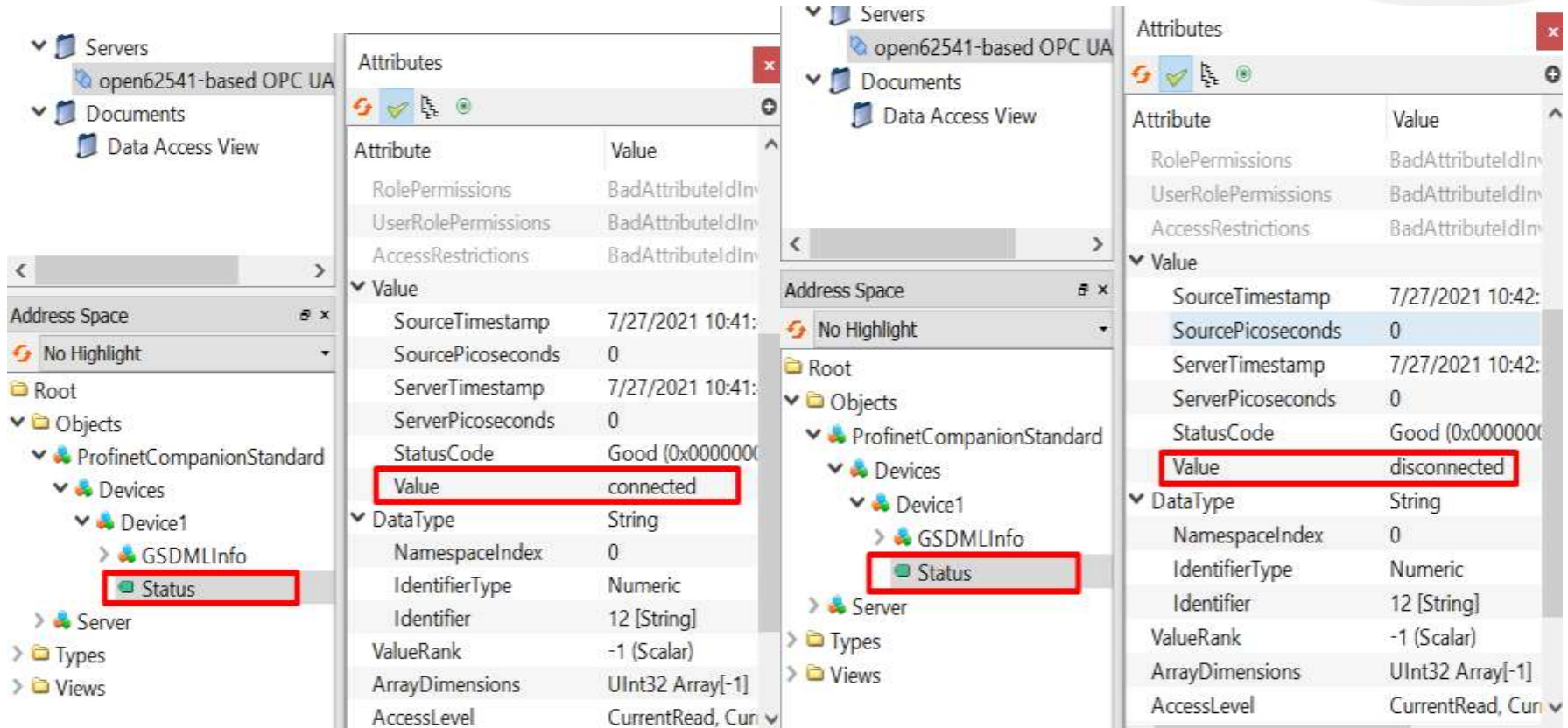
No Highlight

- Root
 - Objects
 - ProfinetCompanionStandard
 - Devices
 - Device1
 - GSDMLInfo
 - IPConfig
 - DefaultGateway
 - IPAddress
 - StationName
 - SubnetMask
 - Status

Data Access View

#	Server	Node Id	Display Name	Value	Datatype	Statuscode
1	open62541-based OPC UA Application	NS2 Numeric 1025	DefaultGateway	192.168.0.10	String	Good
2	open62541-based OPC UA Application	NS2 Numeric 1022	IPAddress	192.168.0.10	String	Good
3	open62541-based OPC UA Application	NS2 Numeric 1023	StationName	SiemensSPS	String	Good
4	open62541-based OPC UA Application	NS2 Numeric 1024	SubnetMask	255.255.255.0	String	Good

Connection-Status



The screenshot displays the SIMATIC Manager interface with three main windows showing the connection status of an OPC UA server.

Left Window (Project Explorer): Shows the project structure. The 'Status' object under 'GSDMLInfo' is highlighted with a red box.

Middle Window (Attributes): Displays the attributes of the selected 'Status' object. The 'Value' attribute is highlighted with a red box and shows the value 'connected'.

Attribute	Value
RolePermissions	BadAttributel dIn
UserRolePermissions	BadAttributel dIn
AccessRestrictions	BadAttributel dIn
Value	
SourceTimestamp	7/27/2021 10:41:
SourcePicoSeconds	0
ServerTimestamp	7/27/2021 10:41:
ServerPicoSeconds	0
StatusCode	Good (0x00000000)
Value	connected
DataType	
NamespaceIndex	0
IdentifierType	Numeric
Identifier	12 [String]
ValueRank	-1 (Scalar)
ArrayDimensions	UInt32 Array[-1]
AccessLevel	CurrentRead, Cur

Right Window (Attributes): Displays the attributes of the selected 'Status' object. The 'Value' attribute is highlighted with a red box and shows the value 'disconnected'.

Attribute	Value
RolePermissions	BadAttributel dIn
UserRolePermissions	BadAttributel dIn
AccessRestrictions	BadAttributel dIn
Value	
SourceTimestamp	7/27/2021 10:42:
SourcePicoSeconds	0
ServerTimestamp	7/27/2021 10:42:
ServerPicoSeconds	0
StatusCode	Good (0x00000000)
Value	disconnected
DataType	
NamespaceIndex	0
IdentifierType	Numeric
Identifier	12 [String]
ValueRank	-1 (Scalar)
ArrayDimensions	UInt32 Array[-1]
AccessLevel	CurrentRead, Cur

Fazit



**PROFI
NET**

OPC UA

MQTT

Fazit

- Entwicklung und Einsatz von smarten Sensoren und einem digitalen Zwilling auf Grundlage von LowCost-Hardware und OpenSource Software
- Erprobung unterschiedlicher Konzepte für Kommunikation und Steuerung

→ Alle Anforderungen konnten erfüllt werden

Bewertung des Einsatzes von Low-Cost- und Open-Source-Komponenten:

Vorteile:

- extrem hohe Flexibilität
- geringe Hardware Kosten
- große Auswahl an Tutorials und Anwendungsbeispielen im Internet

Nachteile:

- Unsicherheit, inwieweit „Community Projekte“ gepflegt werden
- „Know-How“ und Entwicklungszeit notwendig