



# CS50 Softwarehandbuch für Industrieprotokolle

Version 8.0.1, Oktober 2016

Copyright © 2016

Alle Rechte vorbehalten. Der Inhalt dieses Dokuments ist geschützt und dient ausschließlich dazu, Kunden zur Bedienung und/oder Wartung von di-soric-Geräten anzuleiten. Er darf ohne die schriftliche Zustimmung von di-soric nicht veröffentlicht, vervielfältigt oder für sonstige Zwecke genutzt werden.

In diesem Handbuch werden möglicherweise Markennamen verwendet. Wir verwenden diese Namen zum Vorteil des Markeninhabers und ohne die Absicht, gegen Markenrechte zu verstoßen.

### ***Haftungsausschluss***

Die Informationen und Angaben in diesem Handbuch können jederzeit geändert werden.

---

# Inhalt

<b>VORWORT</b>	Willkommen v Zweck dieses Handbuchs v Im Handbuch verwendete Konventionen v
<b>KAPITEL 1</b>	Industrieprotokolle aktivieren 1-1 Protokolle für den CS50-Sensor aktivieren 1-2
<b>KAPITEL 2</b>	EtherNet/IP verwenden 2-1 EtherNet/IP für den CS50-Sensor 2-2 Assembly-Schemata 2-4 Verbindungseigenschaften: Explizite Nachrichten der Klasse 3 2-14 EtherNet/IP-Steuer- und -Statussignale 2-17 Datentypen und ihre Äquivalente in SPS- und EDS/CIP-Umgebungen 2-18 SPS-Tags und serielle Befehle 2-19
<b>KAPITEL 3</b>	Allen-Bradley-AOIs (Add-on-Befehle) für den EtherNet/IP-Betrieb 3-1 Rockwell RSLogix 5000-AOIs (Add-on-Befehle) für di-soric-Geräte 3-2
<b>KAPITEL 4</b>	Allen-Bradley-SPS über EDS für den EtherNet/IP-Betrieb einrichten 4-1 AB Rockwell RSLogix 5000 v20-SPS-Integration mit EDS 4-2
<b>KAPITEL 5</b>	Allen-Bradley-SPS über das Generic Ethernet Module für den EtherNet/IP- Betrieb einrichten 5-1 Kamera in eine SPS-Umgebung integrieren 5-2
<b>KAPITEL 6</b>	EtherNet/IP-SPS-Democode 6-1 Glossar 6-2 Einrichtung der Demo 6-3 Beschreibung der SPS-Tags 6-6 Kamerabetrieb: Nutzung der EtherNet/IP-Demo bei laufender Kamera 6-16

<b>KAPITEL 7</b>	Omron-SPS für den EtherNet/IP-Betrieb einrichten 7-1 Omron-SPS einrichten 7-2
<b>KAPITEL 8</b>	PROFINET I/O verwenden 8-1 CS50-Sensor PROFINET I/O 8-2 Tabelle der Slot-Daten 8-5 Tabelle der STEP 7-SPS-Slots 8-8
<b>KAPITEL 9</b>	Demo PROFINET I/O-Objekt-Zähler mit TIA-Portal V13 und CS50-Sensor 9-1 Überblick 9-2 Fehlerbehebung 9-23
<b>KAPITEL 10</b>	Demo PROFINET I/O-Kreislokalisierung mit TIA-Portal V13 und CS50- Sensor 10-1 Überblick 10-2 Fehlerbehebung 10-24
<b>KAPITEL 11</b>	Demo PROFINET I/O-Objekt-Zähler mit STEP 7 und CS50-Sensor 11- 1 Überblick 11-2 Einrichtung der CS50- Software 11-3 Einrichtung von STEP 7 11-6
<b>KAPITEL 12</b>	Demo PROFINET I/O-Kreislokalisierung mit STEP 7 und CS50-Sensor 12-1 Überblick 12-2 Einrichtung der CS50- Software 12-3 Einrichtung von STEP 7 12-7
<b>KAPITEL 13</b>	Serielle Befehle 13-1



# Willkommen

## Zweck dieses Handbuchs

---

Dieses Handbuch enthält ausführliche Informationen zur Konfiguration und Bereitstellung von EtherNet/IP- und PROFINET I/O-basierten Anwendungen mithilfe der CS50-Software und des CS50-Sensors.

## Im Handbuch verwendete Konventionen

---

In diesem Handbuch werden folgende typografische Konventionen verwendet:

- Wichtige Informationen sind durch **Fettdruck** hervorgehoben.
- Menüpunkte und -einträge sind, sofern nicht anderweitig hervorgehoben, durch Anführungszeichen gekennzeichnet: „Run (triggered)“, „Modify“ usw. Die Menüpunkte sind in der Originalsprache Englisch angegeben; eine deutsche Entsprechung findet sich jeweils in Klammern dahinter.



---

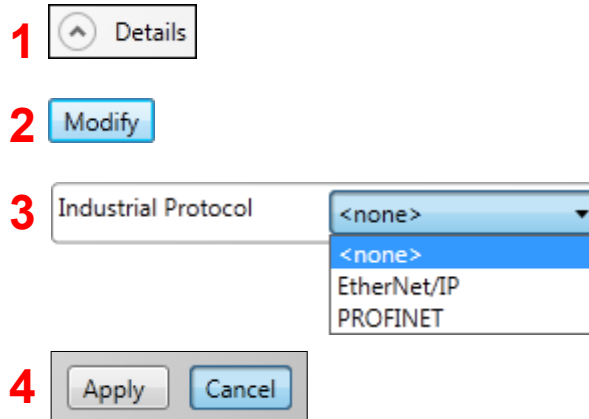
**KAPITEL 1** **Industrieprotokolle aktivieren**

In diesem Kapitel wird beschrieben, wie sich EtherNet/IP für den CS50-Sensor aktivieren und das Protokoll des CS50-Sensors von EtherNet/IP auf PROFINET I/O (oder umgekehrt) umstellen lässt.

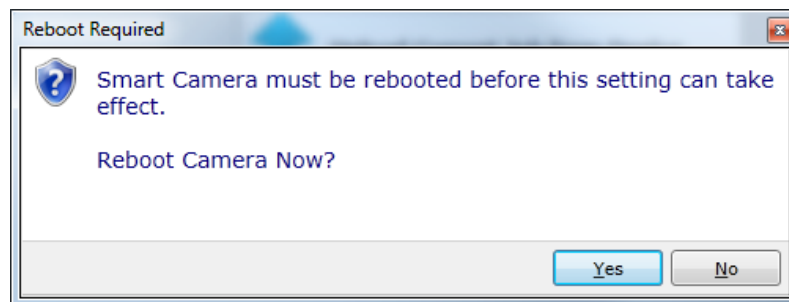
## Protokolle für den CS50-Sensor aktivieren

### CS50-Software – CS50-Sensor

Wählen Sie in der Ansicht **Connect** (Verbinden) einen CS50-Sensor aus und klicken Sie dann auf die Schaltfläche links neben **Details** (direkt unter der Abbildung des CS50-Sensors), um die Kameraeinstellungen aufzurufen. Klicken Sie in dem Dialogfenster zum Bearbeiten der Kameraeinstellungen unten auf **Modify** (Ändern). Wählen Sie aus dem Drop-down-Menü **Industrial Protocol** (Industrieprotokoll) das gewünschte Protokoll aus und klicken Sie dann unten auf **Apply** (Übernehmen).



Daraufhin öffnet sich das Dialogfenster **Reboot Required** (Neustart erforderlich). **Hinweis:** Bei jeder Änderung des Industrieprotokolls muss die Kamera neu gestartet werden, bevor die neue Einstellung wirksam wird. Wenn Sie möchten, dass die Kamera sofort automatisch neu gestartet wird, klicken Sie auf **Yes** (Ja). Während des Neustarts wird die CS50-Software deaktiviert. Wenn Sie auf **No** (Nein) klicken, wird die Änderung erst wirksam, wenn Sie die Kamera das nächste Mal manuell neu starten.



# EtherNet/IP verwenden

In diesem Kapitel finden Sie alle nötigen Informationen zur Verwendung des CS50-Sensors in einer EtherNet/IP-Umgebung.

**Hinweise:**

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für EtherNet/IP aktiviert werden. Wie man Kommunikationsprotokolle für den CS50-Sensor aktiviert oder ändert, erfahren Sie in **Kapitel 1 [Industrieprotokolle aktivieren](#)**.
- Bei CS50-Software-Jobs wird die EtherNet/IP-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert. Weitere Informationen dazu, wie man Job-Parameter und Ausgänge mit di-soric Link-Tags verknüpft, finden Sie in der *CS50-Software-Hilfe* im Abschnitt **di-soric Link**.

## EtherNet/IP für den CS50-Sensor

---

### Überblick

Die EtherNet/IP-Schnittstelle wird als „Vendor Specific (100)“ (Herstellerspezifisch) erkannt. Die Schnittstelle unterstützt den impliziten I/O-Datenaustausch der Klasse 1 sowie explizite Nachrichten der Klasse 3 für serielle Befehle, die nicht mit impliziten Nachrichten kompatibel sind.

### Erforderliche Tools

Für die Konfiguration von EtherNet/IP sind folgende Tools erforderlich:

- Die CS50-Software
- Ein EtherNet/IP-Nachrichtentool – kann eine SPS oder ein Softwaretool sein; in jedem Falle muss das Tool explizite Nachrichten senden und Verbindungen der Klasse 1 herstellen können. Ein Beispiel für ein solches Tool ist EIPScan von Pyramid Solutions.
- Ein Tool zur Terminal-Emulation oder für die serielle Kommunikation, das Verbindungen zu seriellen UART- und TCP-Sockets wie HyperTerminal oder Putty herstellen kann.

### Nutzungsbedingungen für EtherNet/IP

Die EtherNet/IP-Technologie wird von Open DeviceNet Vendor Association, Inc. (ODVA) verwaltet. Natürliche oder juristische Personen, die Produkte herstellen und verkaufen, in die die EtherNet/IP-Technologie eingebunden ist, müssen den Nutzungsbedingungen von ODVA zustimmen. Weitere Informationen dazu siehe [www.odva.org](http://www.odva.org).

## EtherNet/IP-Objektmodell

Um die meisten Daten und Dienste über eine einzige Verbindung zu übermitteln, nutzt der CS50-Sensor verknüpfte Nachrichten der Klasse 1.

### EtherNet/IP-Identität

#### Gerätetyp

Der Gerätetyp ist: 100, Vendor Specific, Machine Vision Camera.

#### Hersteller-ID

Die OVDA-Hersteller-ID von di-soric ist 1095.

#### Produktcode

Der Produktcode des CS50-Sensors lautet 7000.

#### Schnittstellenversion

Major.Minor = 1.1

#### Verbindungseigenschaften: Implizite Nachrichten

**der Klasse 1 Eingangs-Assembly-Instanz (zu**

**SPS/Client): 102Ausgangs-Assembly-Instanz (zur**

**Kamera): 114**

**Größe:** Fest, 320 Bytes in beide Richtungen

**Eingangstrigger/Triggermodus:** Zyklisch

**RPI (Requested Packet Interval):** Empfohlen wird ein Wert über 20 ms. Zulässig sind Werte im Bereich von 10 ms bis 3,2 s.

#### Eingangstyp/Verbindungstyp:

- Punkt-zu-Punkt (SPS AUS, O > T)
- Punkt-zu-Punkt (SPS EIN, T > O)

**Verbindungspriorität:** Geplant

## Assembly-Schemata

### Eingangs-Assembly

Die **Eingangs-Assembly** wird im Folgenden beschrieben.

Bytes	Name	Description
0...1	STATUS	Status register of the camera, each bit of this register represents a different state item. See <a href="#">Camera Status Register</a> for bit descriptions
2...3	ECHO	This 16 bit word value reflects back to the PLC the value that the PLC wrote to the output assembly ECHO register. The PLC can verify the output assembly has been written to the camera when this value matches the written value.
4...7	CmdCodeRslt	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdCodeRslt reflects the result of the command invoked by Control.CmdCode. See <a href="#">CmdCodeRslt</a> for definitions.
8...11	CmdRet	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdRet contains the data returned from the command invoked by Control.CmdCode. See <a href="#">CmdRet</a> for definitions.
12...13	reserved	Reserved for future use.
14...15	State	Device State register. Depending on the current state of the camera, certain STATUS and CONTROL features may or may not be operational. See <a href="#">State</a> for definitions.
16...17	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 145, the most significant bit vio point 160
18...19	reserved	Reserved for future use.
20...27	bool1-64	Each bit represents a bool value. The least significant bit of byte 20 reads the value of bool1. The most significant bit of byte 27 reads bool64.
28...47	int1-10	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 28 & 29 for the value of int1 through bytes 46 & 67 for the value of int10.
48...87	long1-10	Each group of 4 bytes represents a 32 bit signed integer value. The 40 bytes represent 10 long integers. From bytes 48-51 for the value of long1 through bytes 84-87 for the value of long10.
88...127	float1-10	Each group of 4 bytes represents a floating point value. The 40 bytes represent 10 floating point values. From bytes 88-91 for the value of float1 through bytes 124-127 for float10.
128...223	string1	These 96 bytes can store a string of up to 92, 8 bit characters, with the first 4 bytes containing the length value.
224...255	string2	Each of these 32 byte groups can store a string of up to 28, 8 bit characters, with the first 4 bytes containing the length value.
256...287	string3	
288...319	string4	



Grafische Darstellung der Eingangs-Assembly:

0	STATUS	64	long5	128		192	256
2	ECHO	66		130		194	258
4	CMD CODE RSLT	68	long6	132		196	260
6		70		134		198	262
8	CMD RET	72	long7	136		200	264
10		74		138		202	266
12	reserved	76	long8	140		204	268
14	STATE	78		142		206	270
16	VIO	80	long9	144		208	272
18	reserved	82		146		210	274
20	bool1...16	84	long10	148		212	276
22	bool17...32	86		150		214	278
24	bool33...48	88	float1	152		216	280
26	bool49...64	90		154		218	282
28	int1	92	float2	156		220	284
30	int2	94		158		222	286
32	int3	96	float3	160	string1	224	288
34	int4	98		162		226	290
36	int5	100	float4	164		228	292
38	int6	102		166		230	294
40	int7	104	float5	168		232	296
42	int8	106		170		234	298
44	int9	108	float6	172		236	300
46	int10	110		174		238	302
48	long1	112	float7	176		240	304
50		114		178		242	306
52	long2	116	float8	180		244	308
54		118		182		246	310
56	long3	120	float9	184		248	312
58		122		186		250	314
60	long4	124	float10	188		252	316
62		126		190		254	318

## Status: Kamera-Statusregister (16 Bit)

Jedes Bit dieses Registers repräsentiert einen anderen Status des Kamerabetriebs. Ein hoher Wert für 1 zeigt an, dass der Status aktiv (wahr) ist.



Bit	Name	Description
0	ONLINE	Inspections are running
1	EXP BUSY	The camera is busy capturing an image. The camera should not be triggered or the part under inspection moved during this time if illuminated.
2	ACQ BUSY	The camera is busy acquiring an image. The camera cannot be triggered while busy.
3	TRIGGER READY	The camera is ready to be triggered. This is equivalent to ONLINE == 1 and ACQ BUSY == 0.
4	ERROR	An error has occurred. Set the RESET ERROR control bit high to clear.
5	RESET COUNT ACK	This bit mirrors the RESET COUNT control bit. The PLC can be certain the reset command was received by the camera when this goes high. The PLC can then bring the RESET COUNT control signal back low.
7	EXE CMD ACK	This bit mirrors the EXE CMD control bit.
8	TRIGGER ACK	This bit mirrors the TRIGGER control bit.
9	INSP BUSY	This bit is high when inspection 1 is busy processing an image.
10	INSP STAT	This bit represents the inspection 1 status result. It is 1 if the inspection passes. It is only valid when DataValid goes high.
11	DATA VALID	This bit goes high when inspection 1 is complete. The PLC should clear this signal by setting RESET DV high once it has read results.

### CmdCodeRslt (32 Bit)

Der Wert von **CmdCodeRslt** ist nur gültig, wenn **ExeCmdAck** aktiv (1) ist, in Reaktion auf einen aktiven Status von **ExeCmd**.

CmdCodeRslt value (base 16 hex)	Meaning
0x0000_0000	Success
0x0100_0000	Fail. Possible reasons: Camera under PC control. Job cannot be changed.
0x0200_0000	Fail: No Job in slot.
0x0300_0000	Fail: Unknown cmd.

### CmdRet (32 Bit)

Der Wert von **CmdRet** ist nur gültig, wenn **ExeCmdAck** aktiv (1) ist – in Reaktion auf einen aktiven Status von **ExeCmd** – und **CmdCodeRslt** den Wert **0 (Erfolg)** hat. Die folgende Tabelle zeigt, welche CmdCodes im CmdRet-Register Werte zurückgeben.

CmdRet value (32 bit)	Associated CmdCode	Meaning
0	0x1000_0000 to 0x1300_0000 (Job Change type)	Na
1 – 255	0x1800_0000 (Query Active Job Slot)	Active Job Slot #

## Status (16 Bit)

State (Status) gibt folgende Betriebszustände der Kamera an:

State value (16 bit)	Meaning	Typical action required by the client (plc), or system operator
----------------------	---------	---

0	Offline	Perform job change or put camera online.
1	Online	Normal runtime operation: Monitor TriggerReady and DataValid signals. Trigger the camera.
2	Changing Vision Job	If camera is under pc control: Wait until State changes to Offline or Online. If plc is controlling the job change: Use ExeCmd, CmdCode, ExeCmdAck, and CmdCodeRslt to complete the operation.
3	Booting*	Wait for camera to transition to Online or Offline.
4	Empty (no Vision Job)	Load a new job from AutoVISION or Front Runner.

\* Status 3 „Booting“ (Neustart läuft): Dieser Status wird von der SPS selten erkannt.

Der Wert von „State“ ist maßgeblich dafür, welche Steuersignale (**Control**) und Statussignale (**Status**) verfügbar sind:

Control/Status Signal	State				
	0 (Offline)	1 (Online)	2 (Job Change)	3 (Booting)	4 (Empty)
Control.GO ONLINE	Y				
“.GO OFFLINE		Y			
“.RESET ERROR					
“.RESET COUNT	Y	Y			
“.EXE CMD	Y	Y	Y		Y
“.TRIGGER		Y			
“.RESET DATA VALID		Y			
Status.ONLINE	Y	Y	Y	Y	Y
“.ERROR					
“.RESET COUNT ACK	Y	Y			
“.EXE CMD ACK	Y	Y	Y		Y
“.EXP BUSY		Y			
“.ACQ BUSY		Y			
“.TRIGGER READY		Y			
“.TRIGGER ACK		Y			
“.INSP BUSY		Y			
“.INSP STAT		Y			
“.DATA VALID		Y			

Dabei gilt:

Y = Signal ist für diesen Status gültig.

Leere Tabellenzelle = Signal ist für diesen Status ungültig.

## VIO-Register-Bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v160	v159	v158	v157	v156	v155	v154	v153	v152	v151	v150	v149	v148	v147	v146	v145

## Ausgangs-Assembly

Das Schema der **Ausgangs-Assembly** wird in der folgenden Tabelle sowie in der darauffolgenden Grafik beschrieben:

Bytes	Name	Description
0...1	CONTROL	Control register of camera. Each bit of this register represents a different status item. See <a href="#">Camera Control Register</a> for bit descriptions
2...3	ECHO	This 16 bit value is reflected back to the PLC in the input assembly ECHO register. The PLC can verify the output assembly has been written to the camera when the input assembly matches this written value.
4...7	CmdCode	Specifies the process invoked in the camera when Control.ExeCmd goes active. See <a href="#">CmdCode</a> for definitions.
8...11	CmdArg	Additional argument data for the CmdCode. See <a href="#">CmdArg</a> for definition.
12...15	reserved	Reserved for future use.
16...17	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 129, the most significant bit is vio point 144
18...19	Reserved	Reserved for future use.
20...27	bool	Each bit represents a bool value. The least significant bit of byte 20 writes the value of bool101. The most significant bit of byte 27 writes bool164.
28...47	int101-110	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 28 & 29 to write the value of int101 through bytes 46 & 47 for the value of int110.
48...87	long101-110	Each group of 4 bytes represents a 32 bit signed integer value. The 40 bytes represent 10 long integers. From bytes 48-51 for the value of long101 through bytes 84-87 for the value of long110.
88...127	float101-110	Each group of 4 bytes represents a floating point value. The 40 bytes represent 10 floating point values. From bytes 88-91 for the value of float101 through bytes 124-127 for the value of float110.
128...223	string101	These 96 bytes can store a string of up to 92 bytes, with the first 4 bytes containing the length value.
224...255	string102	Each of these 32 byte groups can store a string of up to 28 bytes, with the first 4 bytes containing the length value.
256...287	string103	
288...319	string104	

Grafische Darstellung der Ausgangs-Assembly:

Byte		Byte		Byte		Byte		Byte	
0	CONTROL	64	long105	128		192		256	
2	ECHO	66		130		194		258	
4	CMD CODE	68	long106	132		196		260	
6		70		134		198		262	
8	CMD ARG	72	long107	136		200		264	
10		74		138		202		266	
12	reserved	76	long108	140		204		268	
14		78		142		206	string101	270	string103
16	VID	80	long109	144		208	(cont)	272	
18	reserved	82		146		210		274	
20	bool101_116	84	long110	148		212		276	
22	bool117_132	86		150		214		278	
24	bool133_148	88	float101	152		216		280	
26	bool149_164	90		154		218		282	
28	int101	92	float102	156		220		284	
30	int102	94		158	string101	222		286	
32	int103	96	float103	160		224		288	
34	int104	98		162		226		290	
36	int105	100	float104	164		228		292	
38	int106	102		166		230		294	
40	int107	104	float105	168		232		296	
42	int108	106		170		234		298	
44	int109	108	float106	172		236		300	
46	int110	110		174		238		302	
48	long101	112	float107	176		240	string102	304	string104
50		114		178		242		306	
52	long102	116	float108	180		244		308	
54		118		182		246		310	
56	long103	120	float109	184		248		312	
58		122		186		250		314	
60	long104	124	float110	188		252		316	
62		126		190		254		318	

## Steuerung: Kamera-Steuerregister (16 Bit)

Jedes Bit dieses Registers steuert eine bestimmte Kamerafunktion. Die entsprechende Funktion wird durch den Wechsel vom niedrigen Status **0** zum hohen Status **1** gestartet. Im Normalfall setzt die SPS den Status des Steuerbits auf **0** zurück, sobald sie erkannt hat, dass das Steuersignal von der Kamera verarbeitet wurde. Nicht verwendete Bits sollten im Status **0** verharren.



Bit	Name	Description
0	GO ONLINE	Start all inspections running
1	GO OFFLINE	Stop all inspections
4	RESET ERROR	Reset ERROR in the Status register
5	RESET COUNT	Reset all inspection counts
7	EXECMD	Execute the command specified by Control.CmdCode
8	TRIGGER	Trigger Inspection 1. The inspection must be configured for a triggered image acquisition.
11	RESET DATA VALID	Reset the Data Valid signal of the Status register

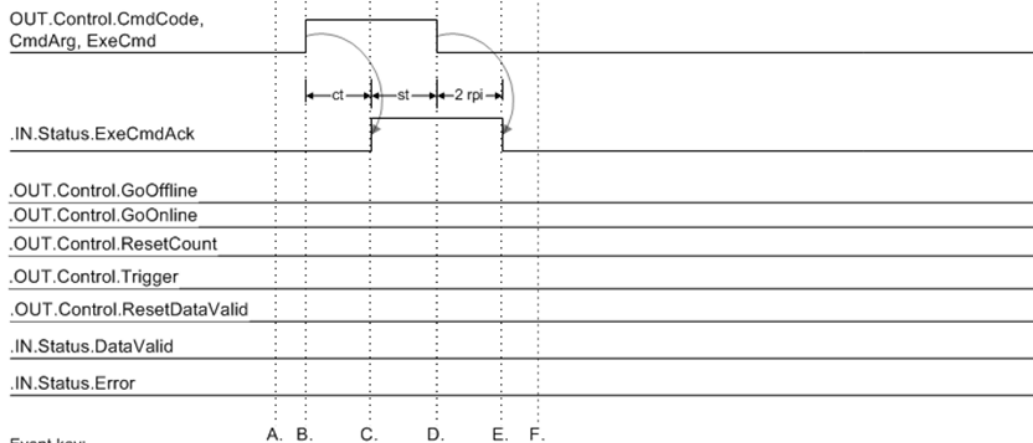
## CmdCode und CmdArg (32 Bit)

CmdCode und CmdArg definieren den Prozess, der in der Kamera ausgelöst wird, wenn **Control.ExeCmd** aktiv wird. Liste der möglichen CmdCodes und der entsprechenden CmdArg-Argumente:

CmdCode value	CmdArg	Operations performed
0x1000_0000	Job Slot (1-255)	Go Offline, Load job from specified slot
0x1100_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Go Online
0x1200_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job
0x1300_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job, and Go Online
0x1800_0000	na	Query active job slot. <a href="#">CmdRet</a> will contain the active job slot number when the operation is done.



## CmdCode und ExeCmd-Betrieb



Event key:

A. If DataValid or Error are present, clear them.

Set the following control signals idle, and keep them idle while the command is processed by the camera:

GoOffline, GoOnline, Trigger, ResetDataValid, ResetCount, ResetError.

If the command operation is a job change, populate the output tags required to configure the new job (bool, int, long, float, string).

B. Populate CmdCode and CmdArg, then activate ExeCmd.

C. Camera executes the command (may take up to a minute). While processing a Job Change command, State will be 2. Camera activates ExeCmdAck when it is done processing the command.

D. When the PLC sees an active ExeCmdAck, verify CmdCodeRslt is 0, and Error is 0. Process CmdRet if needed, then clear ExeCmd.

E. Camera clears ExeCmdAck when ExeCmd goes inactive. When ExeCmdAck goes inactive, CmdCodeRslt and CmdRet are no longer valid, and it may take a few seconds for the camera State and Online signals to settle to a final value (typically Online or Offline).

F. Camera can now be put online and triggered.

Notes:

st = PLC program scan time

ct = Command processing time in the camera. May take up to a minute for some commands.

rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.

All signals represent the state of plc tags.

## VIO-Register-Bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v144	v143	v142	v141	v140	v139	v138	v137	v136	v135	v134	v133	v132	v131	v130	v129

## Verbindungseigenschaften: Explizite Nachrichten der Klasse 3

---

Alle I/O-Assembly-Daten der Klasse 1 und zusätzlichen Daten können über explizite Nachrichten abgerufen werden. Die Eingangsdaten (Kamera an SPS/Client) belegen die Attribute **1** bis **100** der Klassen. Die Ausgangsdaten (SPS/Client an Kamera) belegen die Attribute **101** bis **200**.

### Dienst:

- Einzelnes Attribut abrufen (0xE)
- Einzelnes Attribut festlegen (0x10)

### Klassen:

- bool = 104 (0x68)
- int = 105 (0x69)
- long = 106 (0x6A)
- float = 107 (0x6B)
- string = 108 (0x6C)
- control/status (gemischte Datentypen) = 109 (0x6D)

### Instanz: 1

### Attribut:

- 1 bis 100 = In SPS/Client eingehend
- 101 bis 200 = An Kamera ausgehend

## Attribut-Schema

Bei Verwendung expliziter EtherNet/IP-Nachrichten können alle globalen Datenobjekte gelesen und geschrieben werden. Zum Auslesen der globalen Daten wird jeder Datentyp in einem eigenen Klassenobjekt mit der Instanz 1 gespeichert. Beispiel: Um **float2** auszulesen, würde folgende EtherNet/IP-Anfrage gestellt: **Dienstcode 14 (0xE), Klasse 107 (0x6B), Instanz 1, Attribut 2.**

Class 104		Class 105		Class 106		Class 107		Class 108		Class 109	
Attr#		Attr#		Attr#		Attr#		Attr#		Attr#	
1	bool1	1	int1	1	long1	1	float1	1	string1	1	CONTROL
2	bool2	2	int2	2	long2	2	float2	2	string2	2	STATUS
3	bool3	3	int3	3	long3	3	float3	3	string3	3	
4	bool4	4	int4	4	long4	4	float4	4	string4	4	
5	bool5	5	int5	5	long5	5	float5	5	string5	5	
6	bool6	6	int6	6	long6	6	float6	6	string6	6	ECHO
7	bool7	7	int7	7	long7	7	float7	7	string7	7	CMD CODE
8	bool8	8	int8	8	long8	8	float8	8	string8	8	CMD ARG
9	bool9	9	int9	9	long9	9	float9	9	string9	9	CMD CODE RSLT
10	bool10	10	int10	10	long10	10	float10	10	string10	10	CMD RET
...	...	...	...	...	...	...	...	...	...	11	STATE
...	...	...	...	...	...	...	...	...	...	...	...
199	bool199	199	int199	199	long199	199	float199	199	string199	199	
200	bool200	200	int200	200	long200	200	float200	200	string200	200	

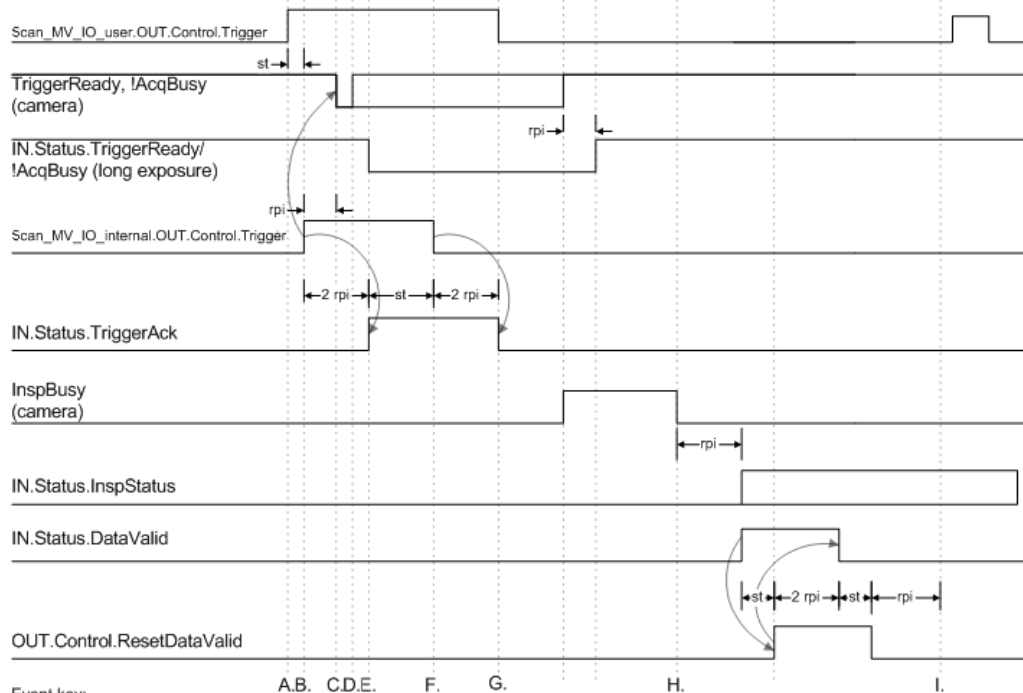
Der Wert, der durch **Einzelnes Attribut abrufen** ausgegeben wird, hängt vom Typ ab:

- **Bool:** Ein 16 Bit langes Wort wird ausgegeben, wobei **0** = falsch und **1** = wahr.
- **Ints:** Eine 16 Bit lange, vorzeichenbehaftete ganze Zahl wird ausgegeben.
- **Longs:** Eine 32 Bit lange, vorzeichenbehaftete ganze Zahl wird ausgegeben.
- **Floats:** Eine 32 Bit lange Gleitkommazahl wird ausgegeben.
- **Strings:** Eine Zeichenfolge wird zusammen mit der Anzahl der Zeichen ausgegeben. Die Größe solcher Zeichenfolgen ist auf **2048 Byte** beschränkt. Dies schließt ein 4 Byte langes „Länge“-Feld ein, gefolgt von 2044 Zeichen, die aus je 8 Bit bestehen. Wenn Zeichenfolgen explizit abgerufen werden, gilt die Größenbeschränkung in der Eingangs- bzw. Ausgangs-Assembly nicht. Beispielsweise ist **string3** in der Eingangs-Assembly auf **28 Byte** beschränkt. Ist die tatsächliche Zeichenfolge länger als 28 Byte, wird sie beim Auslesen über die Assembly gekürzt. Beim expliziten Auslesen über ein Attribut würde dieselbe Zeichenfolge hingegen nicht gekürzt.

Mithilfe der **Assembly-Klasse 109** können EtherNet/IP-spezifische Register gelesen und geschrieben werden.

Attr#	Name	Description
1	CONTROL	The control register (16 bit). See <a href="#">Camera Control Register</a> for bit definitions.
2	STATUS	The status register (16 bit). See <a href="#">Camera Status Register</a> for bit definitions.
6	ECHO	The ECHO register (16 bit) (read only if implicit write is enabled)
7	CMD CODE	The command code register (32 bit). See <a href="#">CmdCode</a> .
8	CMD ARG	The command argument register (32 bit). See <a href="#">CmdArg</a> .
9	CMD CODE RSLT	The command code result register (32 bit). See <a href="#">CmdCodeRslt</a> .
10	CMD RET	The command return value register (32 bit). See <a href="#">CmdRet</a> .
11	STATE	The device state register (16 bit). See <a href="#">State</a> for definitions.

## EtherNet/IP-Steuer- und -Statussignale



### Event key:

- A. On rising edge of system trigger, the user app activates Scan\_MV\_IO\_user.OUT.Control.Trigger to trigger the demo code.
- B. Demo code detects rising edge of Scan\_MV\_IO\_user.OUT.Control.Trigger, and if the camera is ready, sends a trigger to the camera.
- C. Camera acquisition begins (may be delayed by one rpi).
- D. If the camera's exposure time is shorter than the rpi, no change will be seen in TriggerReady and AcqBusy plc IN tags.
- E. Camera firmware acks the trigger. The demo code may not see the ack until two rpi after the trigger was sent (event B).
- F. Demo code detects TriggerAck and clears the Trigger.
- G. Demo code detect falling edge of TriggerAck and clears the user Trigger.
- H. Camera internal signal DataValid will go high when InspBusy goes low
- I. Plc logic must delay one rpi time before re-asserting ResetDataValid

### Notes:

1. The chart shows the workings of the Trigger and ResetDataValid Control signals, and the TriggerAck and DataValid Status signals.
2. st = plc program scan time
3. rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.
4. All signals represent the state of plc tags, except where noted as "(camera)". The cam signals shown are visible in the EIP interface, but the state of the plc tags and internal firmware signals will be different for at least one or two requested packet intervals (rpi).
5. The plc is running the demo code distributed with the camera. The demo code and user app use the Scan\_MV\_IO\_user tag set as the primary control, status, and data interface for the user app. All signal operations are still true even if the plc demo code is not used.
6. TriggerReady!/AcqBusy: Camera exposure times can range from less than 1 ms, up to 100 ms.

## Datentypen und ihre Äquivalente in SPS- und EDS/CIP-Umgebungen

AV	Description	RSLogix equivalent	Description	EDS/ EIP equivalents	Description
<b>Bool</b>	1 bit	BOOL	1 bit	BOOL	1 bit
				WORD	16 BOOLs
				LWORD	64 BOOLs
<b>Int</b>	16 bit signed integer	INT	16 bit signed integer	INT	16 bit signed integer
<b>Long</b>	32 bit signed integer	DINT	32 bit signed integer	DINT	32 bit signed integer
<b>Float</b>	32 bit floating point	REAL	32 bit floating point	REAL	32 bit floating point
<b>String</b>	32 bit length field followed by 8 bit ASCII characters	STRING	32 bit length field followed by 8 bit ASCII characters	DINT+ USINT[]	DINT (length) + USINT array of characters. USINT = 8 bit integer

## SPS-Tags und serielle Befehle

SPS-Tags werden nach der Datenrichtung in **IN** (eingehend) und **OUT** (ausgehend) unterschieden. IN- und OUT-Tags sind wiederum in feste Statusfelder (**Status**) und Steuerfelder (**Control**) sowie in nutzerdefinierte verknüpfte Datenfelder unterteilt. Hier ein tabellarischer Überblick über die Zuordnung serieller Befehle zu den entsprechenden SPS-Tag-Namen:

IN			OUT		
PLC tag prefix	Serial cmd prefix	Tag name	PLC tag prefix	Serial cmd prefix	Tag name
IN.Status.	eip.status.	Online (1)	OUT.Control.	eip.control.	GoOnline <sup>i</sup>
IN.Status.	eip.status.	Online (0)	OUT.Control.	eip.control.	GoOffline <sup>ii</sup>
IN.Status.	eip.status.	Error	OUT.Control.	eip.control.	ResetError
IN.Status.	eip.status.	ResetCountAck	OUT.Control.	eip.control.	ResetCount
IN.Status.	eip.status.	TriggerAck	OUT.Control.	eip.control.	Trigger
IN.Status.	eip.status.	DataValid	OUT.Control.	eip.control.	ResetDataValid
IN.Status.	eip.status.	ExeCmdAck	OUT.Control.	eip.control.	ExeCmd
IN.Status.	eip.status.	TrigReady <sup>iii</sup>	-	-	-
IN.Status.	eip.status.	AcqBusy	-	-	-
IN.Status.	eip.status.	ExpBusy	-	-	-
IN.Status.	eip.status.	InspBusy	-	-	-
IN.Status.	eip.status.	InspStat	-	-	-
IN.Status.	eip.	Echo	OUT.Control.	eip.	Echo
IN.Status.	eip.	CmdCodeRslt	OUT.Control	eip.	CmdCode
IN.Status	eip.	CmdRet	OUT.Control	eip.	CmdArg
IN.Status.	eip.	State	-	-	-
IN.vio.	io.	v[145-160]	OUT.vio.	io.	v[129-144]
IN.bool.	eip.	bool[1-100]	OUT.bool.	eip.	bool[101-200] <sup>iv</sup>
IN.int.	eip.	int[1-100]	OUT.int.	eip.	int[101-200] <sup>v</sup>
IN.long.	eip.	long[1-100]	OUT.long.	eip.	long[101-200]
IN.float.	eip.	float[1-100]	OUT.float.	eip.	float[101-200]
IN.string.	eip.	string[1-100]	OUT.string.	eip.	string[101-200]

<sup>i</sup> When GoOnline is changed from 0 to 1, Online goes to 1.

<sup>ii</sup> When GoOffline is changed from 0 to 1, Online goes to 0.

<sup>iii</sup> TrigReady, AcqBusy, ExpBusy, InspBusy, and InspStat are all IN-direction data only.

<sup>iv</sup> bool1-bool64 are mapped to PLC tags in the IN assembly. Bool101-bool164 are mapped to PLC tags in the OUT assembly. Bool members numbered 65-100 and 165-200 are accessible via Explicit Message only.

<sup>v</sup> For int, long, float, and string data:

Data members numbered 1-10 are mapped to PLC tags in the IN assembly.

Data members numbered 101-110 are mapped to PLC tags in the OUT assembly.

Data members numbered 11-100 and 111-200 are accessible via Explicit Message only.





# Allen-Bradley-AOIs (Add-on-Befehle) für den EtherNet/IP-Betrieb

In diesem Kapitel finden Sie weitere nützliche Informationen zur Verwendung des CS50-Sensors in einer EtherNet/IP-Umgebung.

## Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für EtherNet/IP aktiviert werden. Wie man Kommunikationsprotokolle für den CS50-Sensor aktiviert oder ändert, erfahren Sie in **Kapitel 1 [Industrieprotokolle aktivieren](#)**.
- Bei CS50-Software-Jobs wird die EtherNet/IP-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert. Weitere Informationen dazu, wie man Job-Parameter und Ausgänge mit di-soric Link-Tags verknüpft, finden Sie in der *CS50-Software-Hilfe* im Abschnitt **di-soric Link**.

## Rockwell RSLogix 5000-AOIs (Add-on-Befehle) für di-soric-Geräte

Die Datei mit den Add-on-Befehlen (add-on instructions, AOIs) enthält grundlegende Befehle sowie die Hauptzeile des Kontaktplans. Wenn die AOIs aufgerufen werden, können damit Daten gelesen und geschrieben werden. Die Importdatei enthält neben den AOIs selbst auch globale Tags, die Erstnutzern des RSLogix-Systems demonstrieren, wie man AOIs einsetzt. Die AOIs können mit der EDS-Datei verwendet werden.

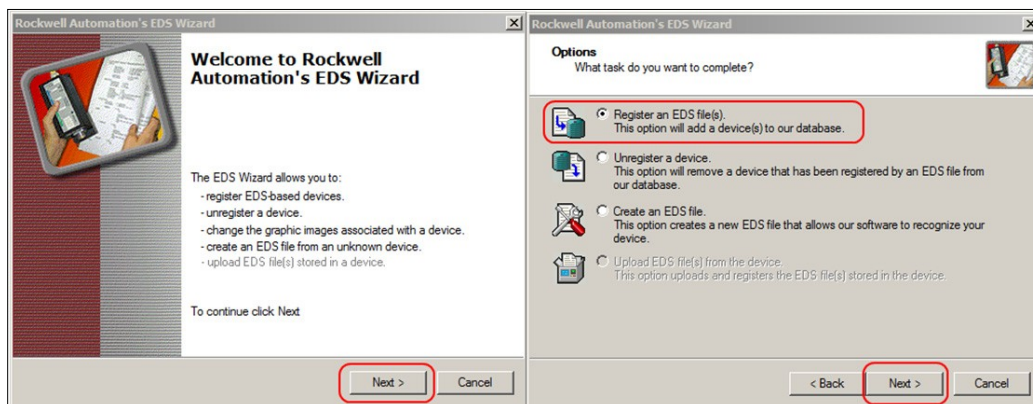
**Hinweis:** Die Beispiele in diesem Kapitel wurden mit RSLogix 5000 Version 20 erstellt.

### Schritte

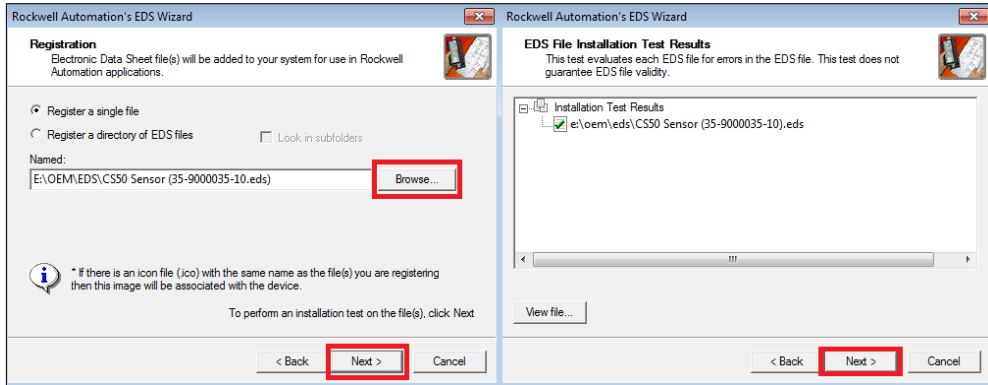
- EDS-Datei installieren
- AOI-Datei importieren
- Kommunikation testen und Daten prüfen

### EDS-Datei installieren

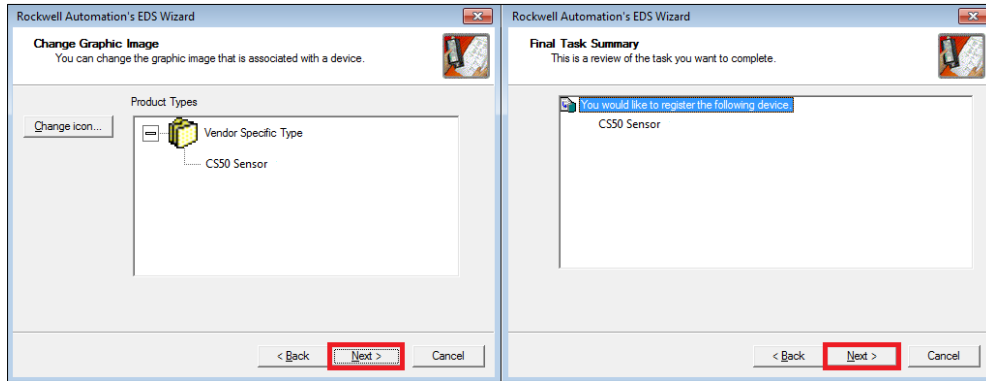
Wählen Sie in RSLogix 5000 im Hauptmenü **Tools** das **EDS Hardware Installation Tool** (Tool zur Installation der EDS-Hardware) aus. Klicken Sie auf **Next** (Weiter). Die Option **Register an EDS File(s)** (EDS-Datei[en] registrieren) muss aktiviert sein.



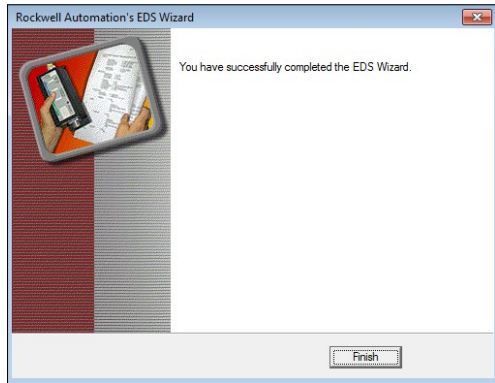
Klicken Sie auf die Schaltfläche **Browse** (Durchsuchen) und wählen Sie die EDS-Datei aus dem Verzeichnis C:\di-soric\Vscape\Firmware\eds\ aus. Klicken Sie dann auf **Next** (Weiter).



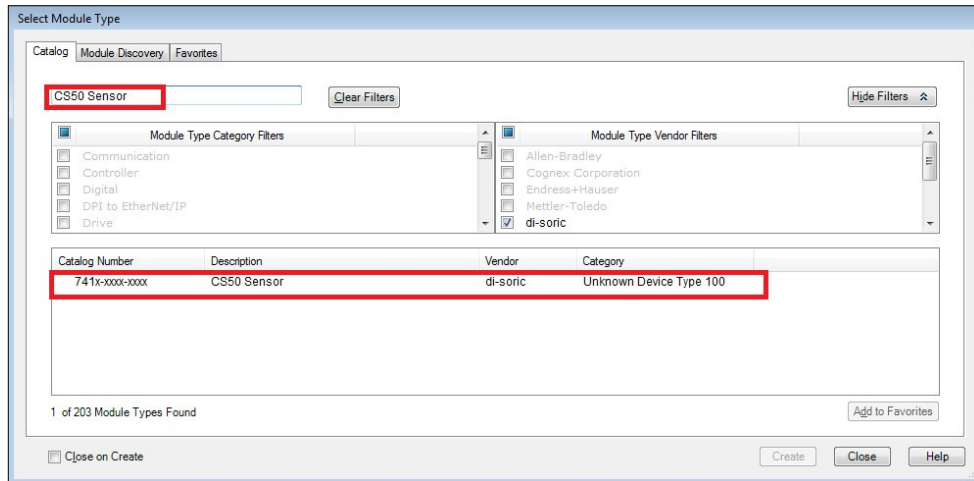
Klicken Sie auf **Next** (Weiter), um das vorgegebene Bild auszuwählen. Wenn Sie erneut auf **Next** klicken, wird eine Zusammenfassung angezeigt.



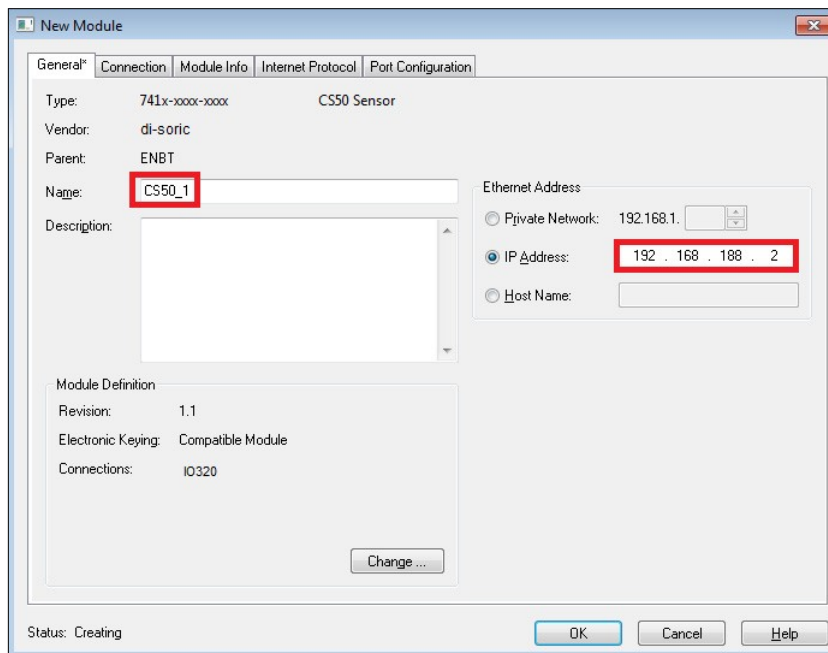
Um die EDS-Installation abzuschließen, klicken Sie auf **Finish** (Beenden).



Klicken Sie im linken Fensterbereich mit der rechten Maustaste auf den Ethernet-Knoten und wählen Sie **New Module** (Neues Modul) aus. Geben Sie in das Filterfeld die Artikelnummer ein. Daraufhin wird das Gerät angezeigt. Wählen Sie das Gerät aus der neu hinzugefügten EDS-Datei aus. Doppelklicken Sie auf das Gerät oder klicken Sie auf **Create** (Erstellen), um es dem Projekt hinzuzufügen.

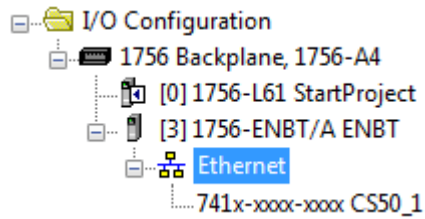


Geben Sie einen Namen und die IP-Adresse für das Gerät ein und klicken Sie dann auf **OK**.



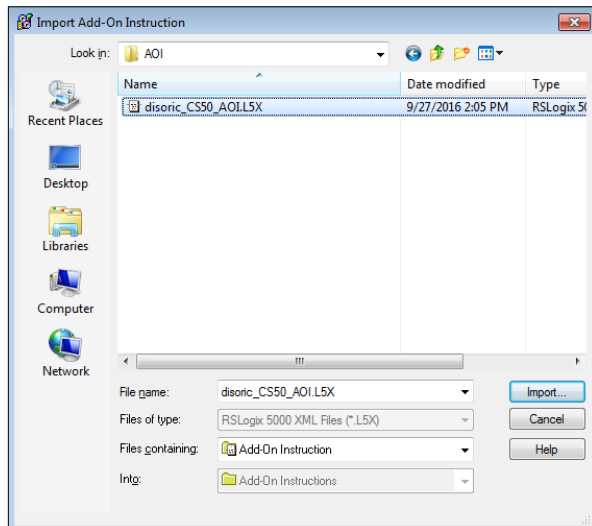
Falls das Lesegerät verschiedene Assembly-Größen aufweist, können Sie mithilfe der Schaltfläche **Change** (Ändern) ein anderes Assembly-Format auswählen.

Um fortzufahren, klicken Sie im Dialogfenster zur Auswahl des Moduls auf **Close** (Schließen). Das Gerät wurde nun dem Projekt hinzugefügt und wird in der Baumansicht unter dem Knoten **Ethernet** angezeigt.

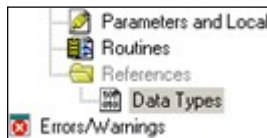


## AOI-Datei importieren

Klicken Sie im linken Fensterbereich in der Baumansicht auf den Knoten **Add-On Instructions** (Add-on-Befehle) und wählen Sie **Import Add-On Instruction** (Add-on-Befehl importieren) aus. Wählen Sie die **L5X-Datei** aus dem Verzeichnis C:\disoric\Vscape\Firmware\aoi\ aus und klicken Sie auf **Import** (Importieren).



Daraufhin öffnet sich das Dialogfenster **Import Configuration** (Importkonfiguration), in dem Sie zur Eingabe von Informationen zur AOI-Datei aufgefordert werden. Wählen Sie **Data Types** (Datentypen) aus, um sich die neuen Tags und deren Attribute anzusehen. Wenn Sie bereit zum Fortfahren sind, klicken Sie auf **OK**.



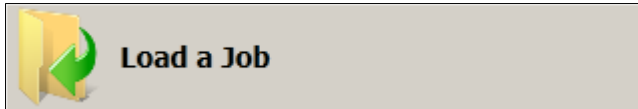
Die neuen Tags und Logiken werden nun dem Projekt hinzugefügt.

## Kommunikation testen und Daten prüfen

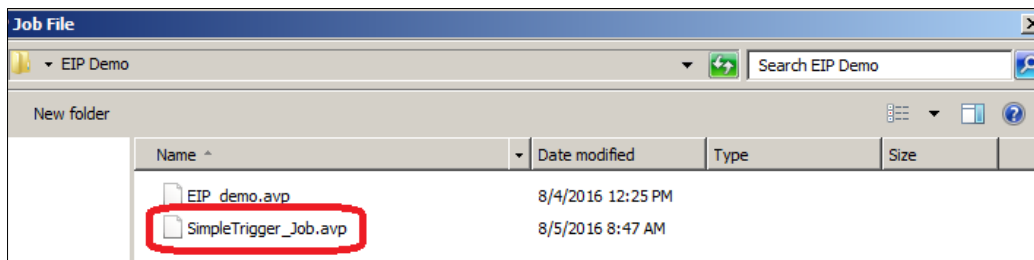
Nach der Installation des Lesegerätmoduls können Sie mit einem einfachen Kontaktplanprogramm in das Gerät ein- und aus dem Gerät ausgehende Daten testen. Die Kommunikation wird in zwei Schritten getestet: (1) Autovision-Job herunterladen; (2) Logik zum SPS-Kontaktplan hinzufügen.

Starten Sie zunächst die CS50-Software und wählen Sie die

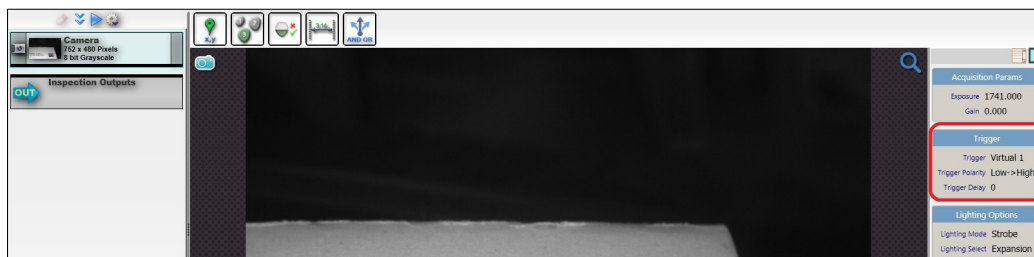
Kamera aus. Wählen Sie **Load a Job** (Job laden) aus.



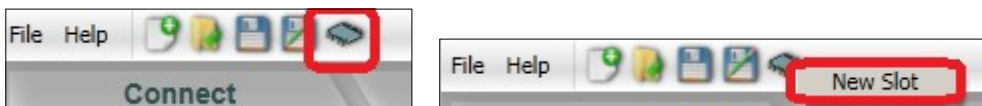
Navigieren Sie zum Verzeichnis **di-soric/Vscope/Tutorials and Samples/EIP Demo/SimpleTrigger\_Job.avp**.



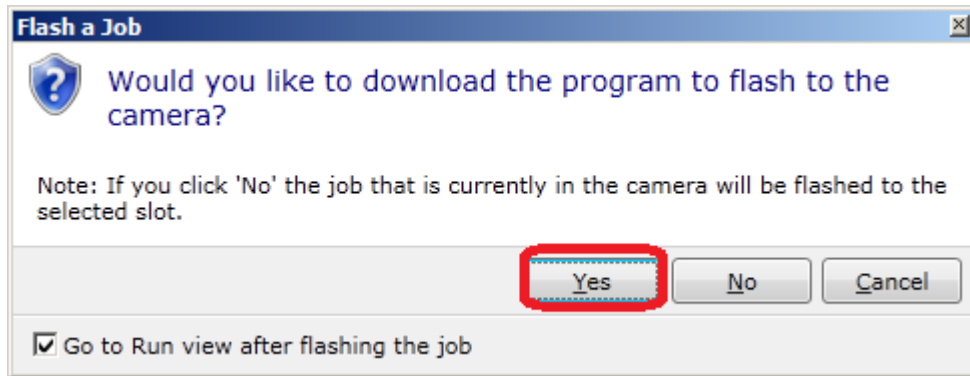
In diesem Job sind keine Tools geladen. Es handelt sich um einen einfachen Job, mit dem die SPS die Kamera auslösen kann. Rechts auf dem Bildschirm im Bereich „Trigger“ können Sie auch einen anderen Trigger festlegen.



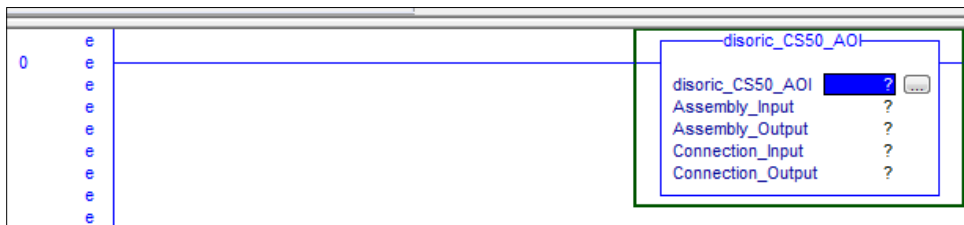
Laden Sie nun den Job auf die Kamera, indem Sie auf das Symbol **Save Job to Camera** (Job auf Kamera speichern) klicken und dann **New Slot** (Neuer Slot) auswählen.



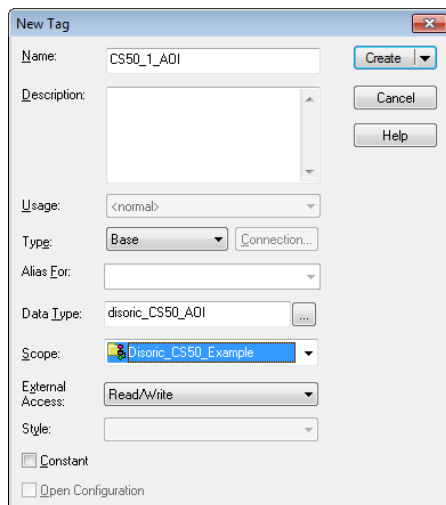
Wählen Sie **Yes** (Ja) aus. Das Programm wird nun auf den Flash-Speicher der Kamera heruntergeladen.



Als Nächstes wird die Logik der SPS hinzugefügt. Öffnen Sie dazu den Editor **MainRoutine** und erstellen Sie einen Add-on-Befehl.

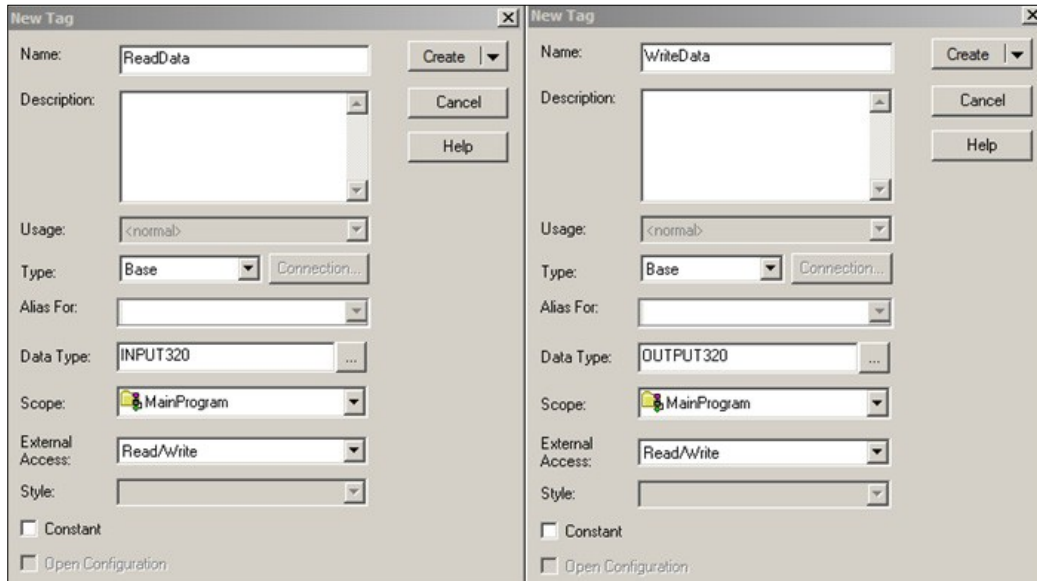


Klicken Sie mit der rechten Maustaste auf das erste Element und wählen Sie **New Tag** (Neues Tag) aus.





Klicken Sie mit der rechten Maustaste auf das zweite Element und erstellen Sie ein neues Tag für **Assembly\_Input** (Assembly\_Eingang). Als Datentyp ist standardmäßig **AOI INPUT320** festgelegt. Klicken Sie nun mit der rechten Maustaste auf das dritte Element und erstellen Sie ein neues Tag für **Assembly\_Output** (Assembly\_Ausgang). Als Datentyp ist standardmäßig **AOI OUTPUT320** festgelegt.



Doppelklicken Sie auf das vierte Element und dann im Kombinationsfeld auf den Abwärtspfeil, um **Connection\_Input** (Verbindung\_Eingang) mit den Eingangsdaten des Lesegeräts zu verknüpfen. Die richtige Verknüpfung heißt **[Name des Lesegeräts]:I.Data**.

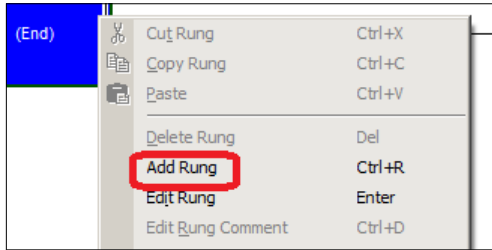
[-]	CS50_1:I	_0447:741
	CS50_1:I.ConnectionFaulted	BOOL
[+]	CS50_1:I.Data	SINT[320]

**Hinweis:** Stellen Sie keine Verbindung zu Elementen her, für die **ConnectionFaulted** (Verbindung fehlgeschlagen) angezeigt wird.

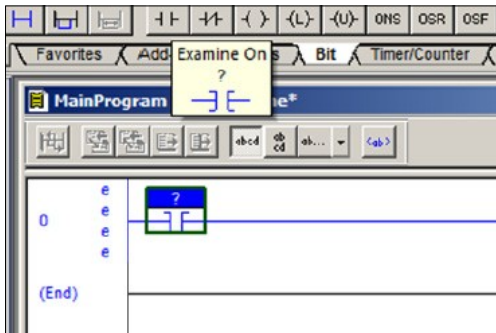
Doppelklicken Sie auf das fünfte Element und dann im Kombinationsfeld auf den Abwärtspfeil, um **Connection\_Output** (Verbindung\_Ausgang) mit den Ausgangsdaten des Lesegeräts zu verknüpfen. Die richtige Verknüpfung heißt **[Name des Lesegeräts]:O.Data**.

CS50	Show: All Tags
Name	Data Type
[+] CS50_1:I	_0447:741
[-] CS50_1:O	_0447:741
[+] CS50_1:O.Data	SINT[320]

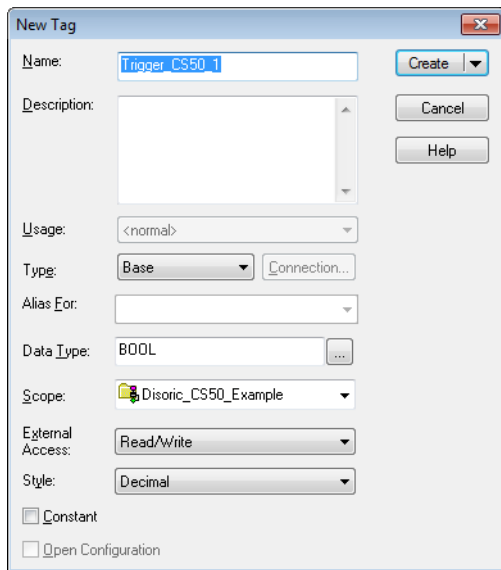
Fügen Sie eine neue Zeile hinzu, indem Sie mit der rechten Maustaste klicken und **Add Rung** (Zeile hinzufügen) auswählen.



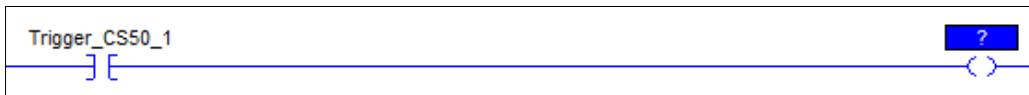
Erstellen Sie unter **Examine On** (Prüfen bei) ein Trigger-Tag.



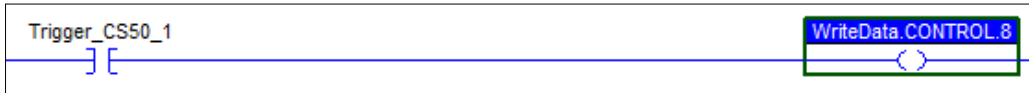
Klicken Sie mit der rechten Maustaste und erstellen Sie einen Booleschen Trigger (**Boolean Trigger**).



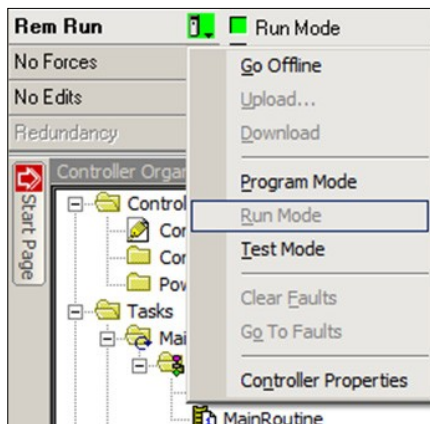
Fügen Sie am Ende der Zeile das Tag **Output Energized** (Ausgang bestromt) hinzu.



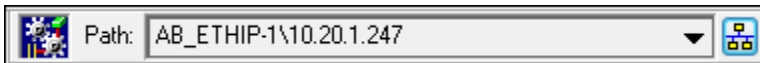
Weisen Sie das Tag **Output Energized** dem Bit **WriteData.CONTROL.8** zu.



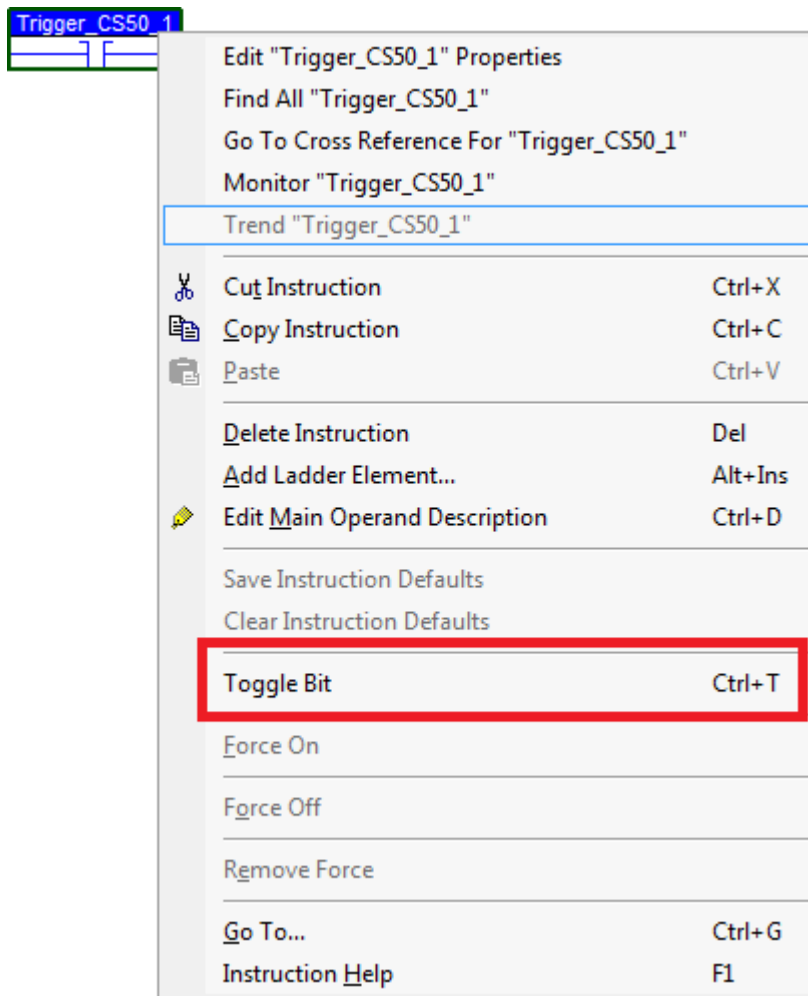
Laden Sie nun das Programm auf die SPS herunter. Wenn der Download beendet ist, schalten Sie die SPS in den Betriebsmodus (**Run Mode**).



**Hinweis:** Im Projekt muss der Pfad zur SPS angegeben sein. Ansonsten kann keine Verbindung zur SPS hergestellt werden.



Schalten Sie das Bit „Trigger“ um, indem Sie mit der rechten Maustaste klicken und die Option **Toggle Bit** (Bit umschalten) auswählen. Alternativ können Sie auch die Tastenkombination **Strg + T** verwenden.



Dadurch wechselt das Trigger-Tag und in der Folge auch das Ausgangsbit des Triggers in den Zustand „1“. Jedes Mal, wenn das Trigger-Tag zu „1“ wechselt, wird die Kamera ausgelöst. Dadurch wird bestätigt, dass eine Kommunikationsverbindung zwischen der SPS und der Kamera besteht.

# Allen-Bradley-SPS über EDS für den EtherNet/IP-Betrieb einrichten

In diesem Kapitel wird beschrieben, wie Sie mithilfe einer EDS-Datei eine Allen-Bradley-SPS für den EtherNet/IP-Betrieb einrichten.

## Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für EtherNet/IP aktiviert werden. Wie man Kommunikationsprotokolle für den CS50-Sensor aktiviert oder ändert, erfahren Sie in **Kapitel 1 [Industrieprotokolle aktivieren](#)**.
- Bei CS50-Software-Jobs wird die EtherNet/IP-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert. Weitere Informationen dazu, wie man Job-Parameter und Ausgänge mit di-soric Link-Tags verknüpft, finden Sie in der *CS50-Software-Hilfe* im Abschnitt **di-soric Link**.

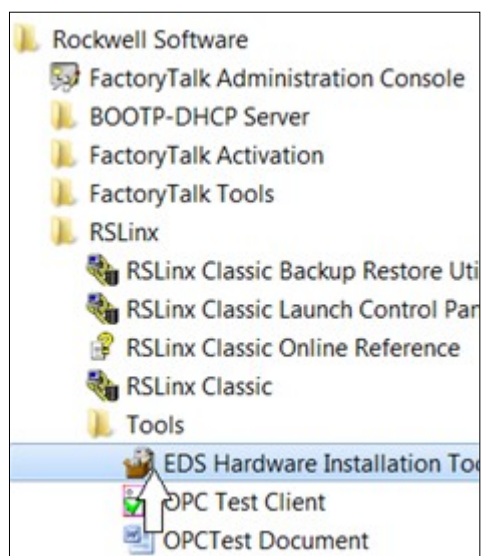
## AB Rockwell RSLogix 5000 v20-SPS-Integration mit EDS

Die Anleitung in diesem Kapitel wurde auf der Basis folgender Allen-Bradley-/Rockwell-Komponenten erstellt und getestet:

- RSLogix 5000 Version 20.00.00 (CPR 9 SR 5)
- 756-L61 ControlLogix5561 Controller, Firmware-Version 20.11
- 1756-ENBT/A EtherNet/IP-Schnittstellenkarte, Firmware-

Version 4.1 Führen Sie das **EDS Hardware Installation Tool**

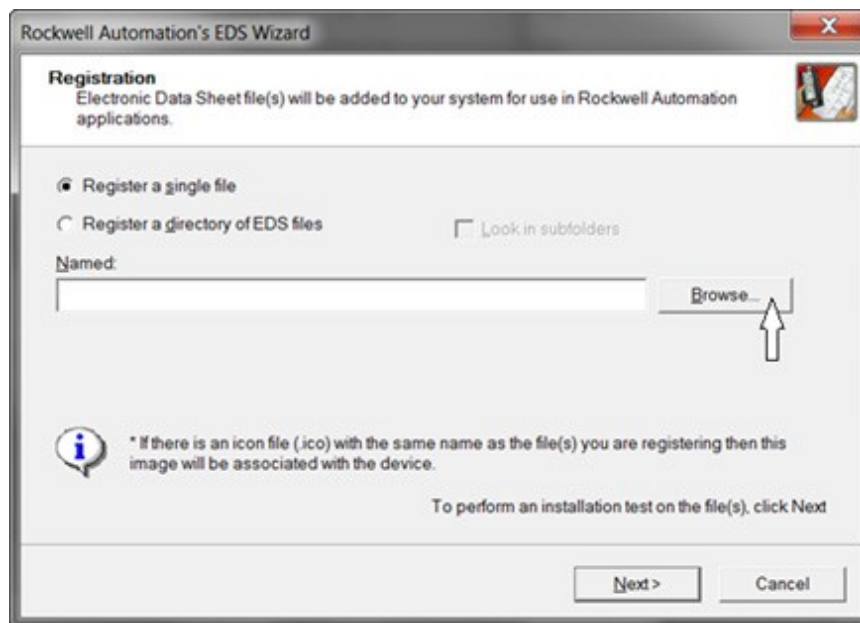
(Tool zur Installation der EDS-Hardware) von Rockwell aus.



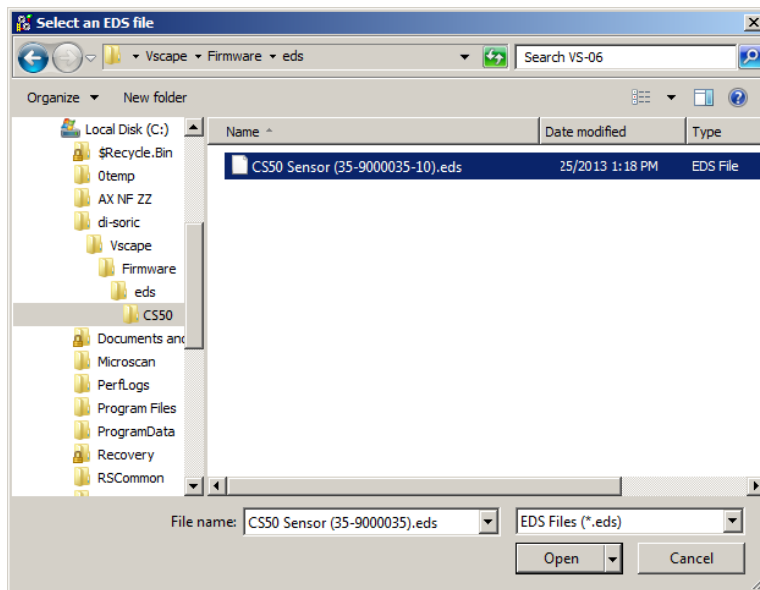
Klicken Sie auf **Add** (Hinzufügen).



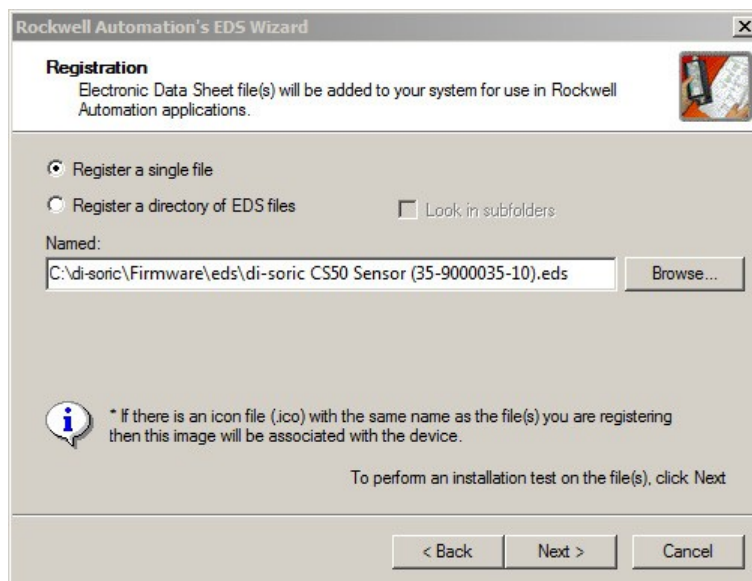
Klicken Sie auf **Browse** (Durchsuchen).



Suchen Sie die EDS-Datei des CS50-Sensors und klicken Sie dann auf **Open** (Öffnen). Als Installationspfad ist standardmäßig **C:\di-soric\Vscape\Firmware\eds\CS50** vorgegeben.

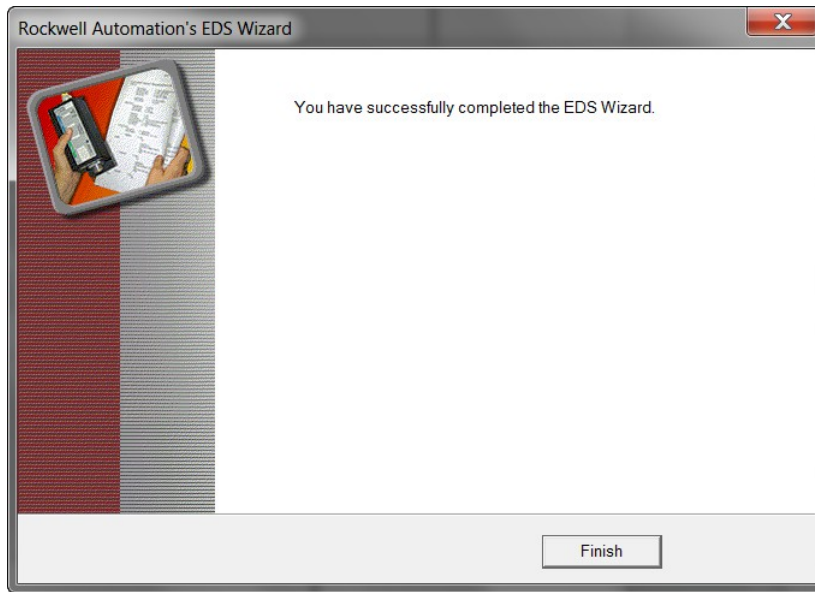


Klicken Sie auf **Next** (Weiter), bis die Schaltfläche **Finish** (Beenden) erscheint.

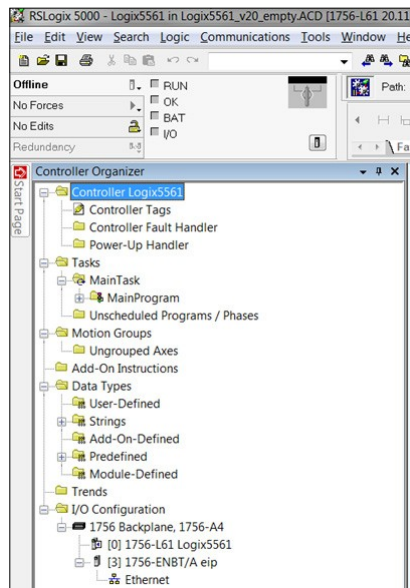




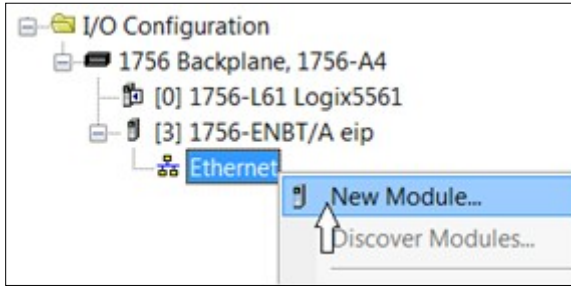
Klicken Sie auf **Finish** (Beenden).



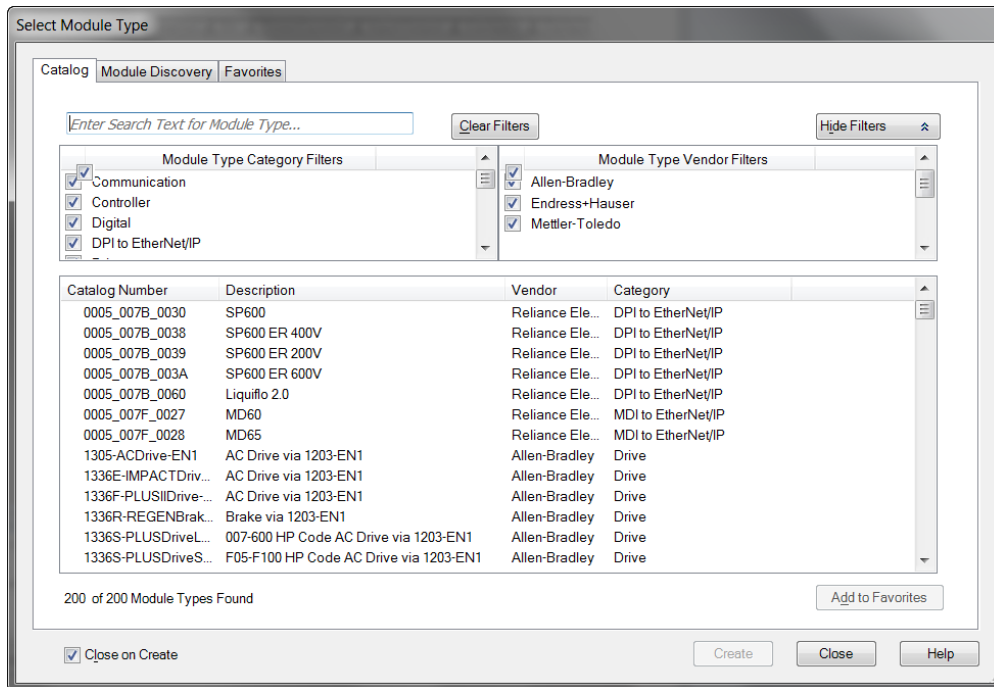
Starten Sie RSLogix 5000 v20 und erstellen Sie für das Basissystem die E/A-Konfiguration (**I/O Configuration**), einschließlich der Ethernet-Schnittstelle des Systems.



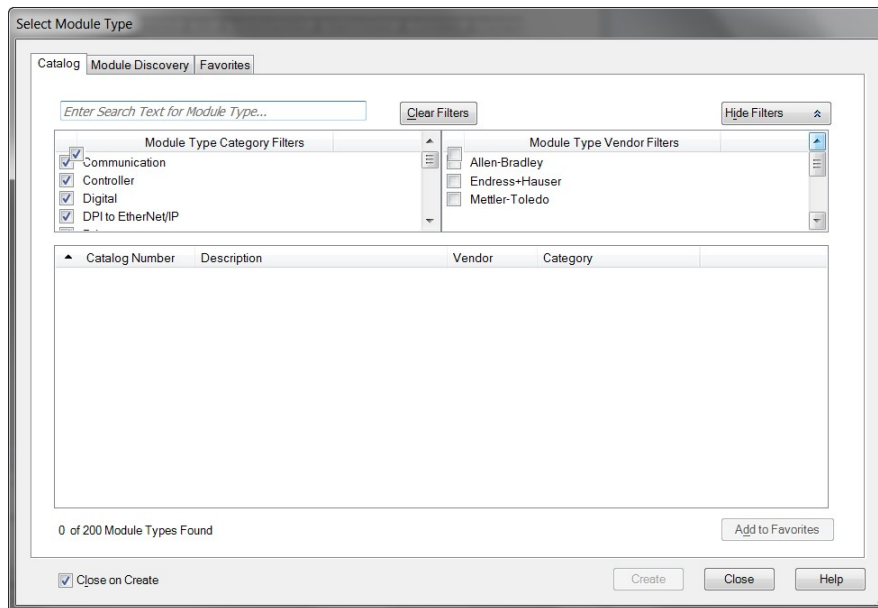
Klicken Sie mit der rechten Maustaste auf **Ethernet** und wählen Sie **New Module** (Neues Modul) aus.



Daraufhin öffnet sich das Dialogfenster **Select Module Type** (Modultyp auswählen).



Deaktivieren Sie die Filter unter **Module Type Vendor Filters**  
(Filter für herstellerspezifische Modultypen).



Scrollen Sie in der Liste „Module Type Vendor Filters“ bis zum Eintrag **di-soric** und wählen Sie diesen Eintrag aus.

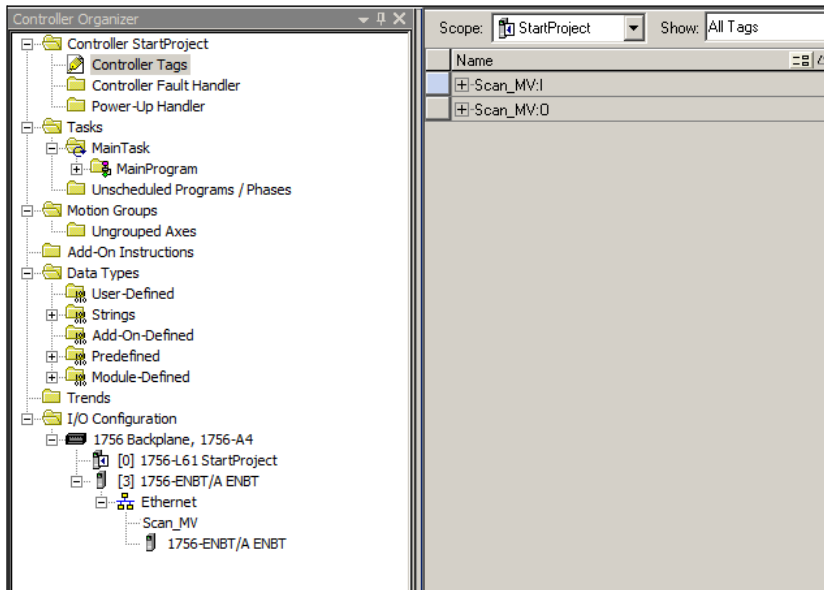
Klicken Sie auf die gewünschte Kamera und dann auf **Create** (Erstellen).

Das Dialogfenster **New Module** (Neues Modul) wird nun angezeigt. Geben Sie einen einmaligen Namen für diese Kamera und deren IP-Adresse ein.

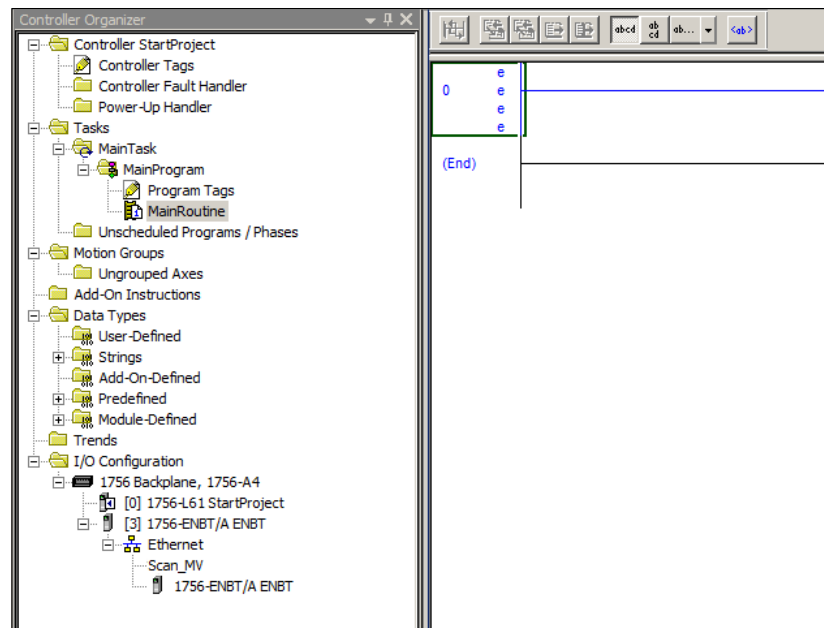
The screenshot shows the 'New Module' dialog box with the following details:

- General\* Tab:**
  - Type: CS50 Sensor (35-9000035-10).eds
  - Vendor: di-soric
  - Parent: ENBT
  - Name: Scan\_MV
  - Description: (empty text area)
  - Ethernet Address:
    - Private Network: 192.168.1. [ ]
    - IP Address: 192 . 168 . 1 . 100
    - Host Name: [ ]
- Module Definition:**
  - Revision: 1.1
  - Electronic Keying: Compatible Module
  - Connections: I0320
  - Change ... button
- Status:** Creating
- Buttons:** OK, Cancel, Help

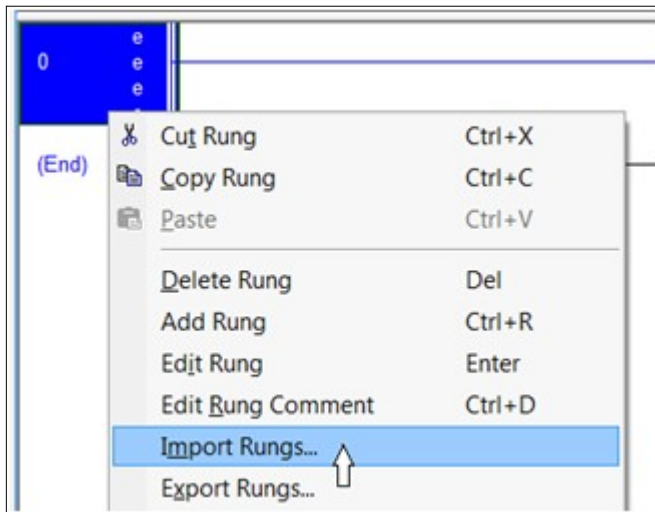
Klicken Sie auf **OK**. Vergewissern Sie sich, dass die Kamera dem Ethernet-Netzwerk hinzugefügt wurde, und öffnen Sie dann die Steuer-Tags, um zu überprüfen, ob die Tag-Sätze für **:I** und **:O** erstellt wurden.



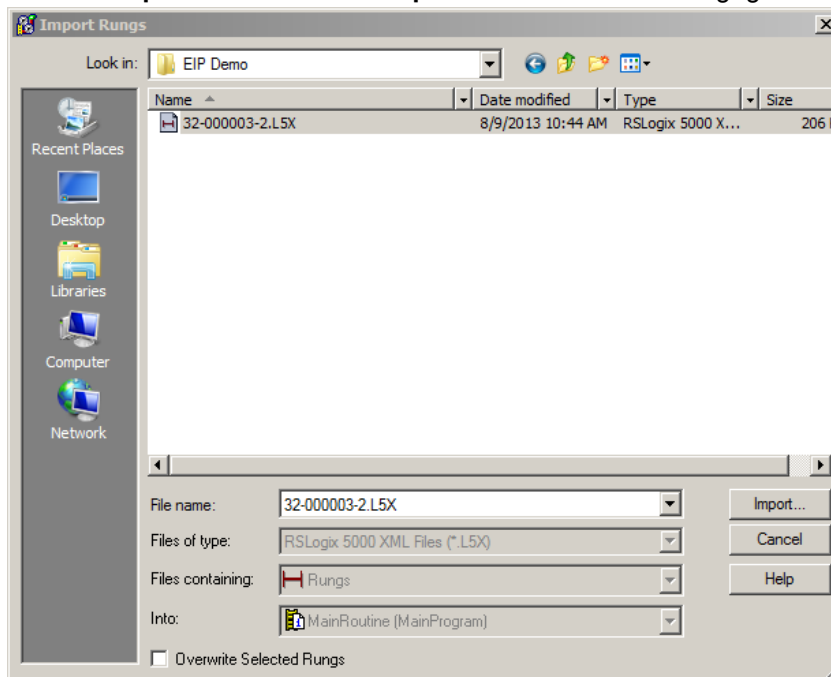
Öffnen Sie **Main Routine** (Hauptroutine).



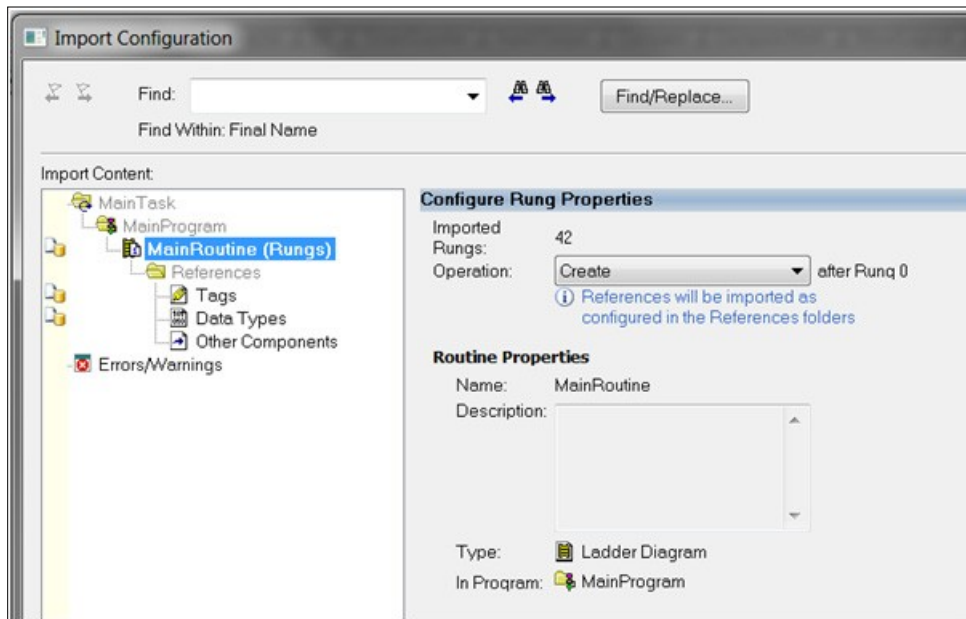
Klicken Sie mit der rechten Maustaste auf die Zeile **0** und wählen Sie dann **Import Rungs** (Zeilen importieren) aus.



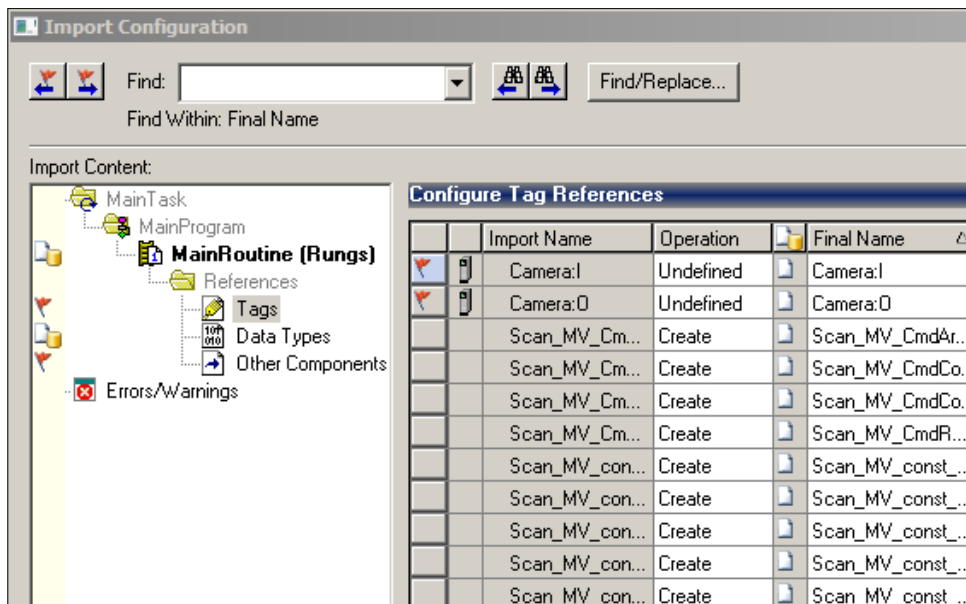
Machen Sie die Datei **32-000003-2.L5X** des CS50-Sensors ausfindig und klicken Sie dann auf **Import** (Importieren). Als Installationspfad ist standardmäßig **C:\di-  
soric\Vscape\Tutorials and Samples\CS50\EIP Demo** vorgegeben.



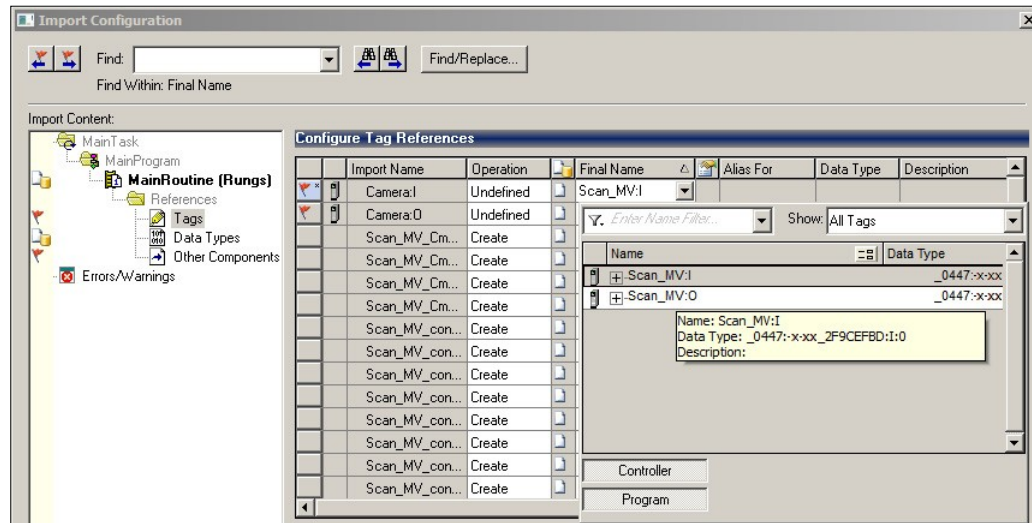
Das Dialogfenster **Import Configuration** (Importkonfiguration) öffnet sich.



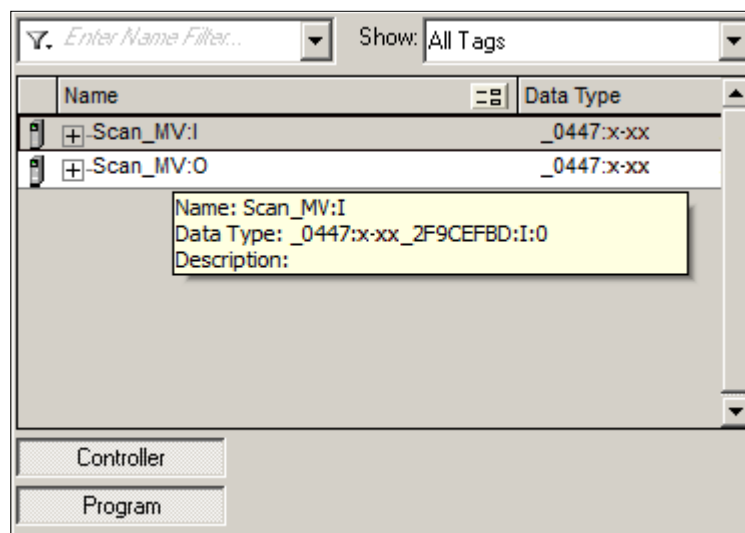
Wählen Sie **Tags** aus.



Klicken Sie in der Spalte **Final Name** (Endgültiger Name) auf **Camera:I** (Kamera:E) und anschließend auf den Abwärtspfeil rechts daneben.



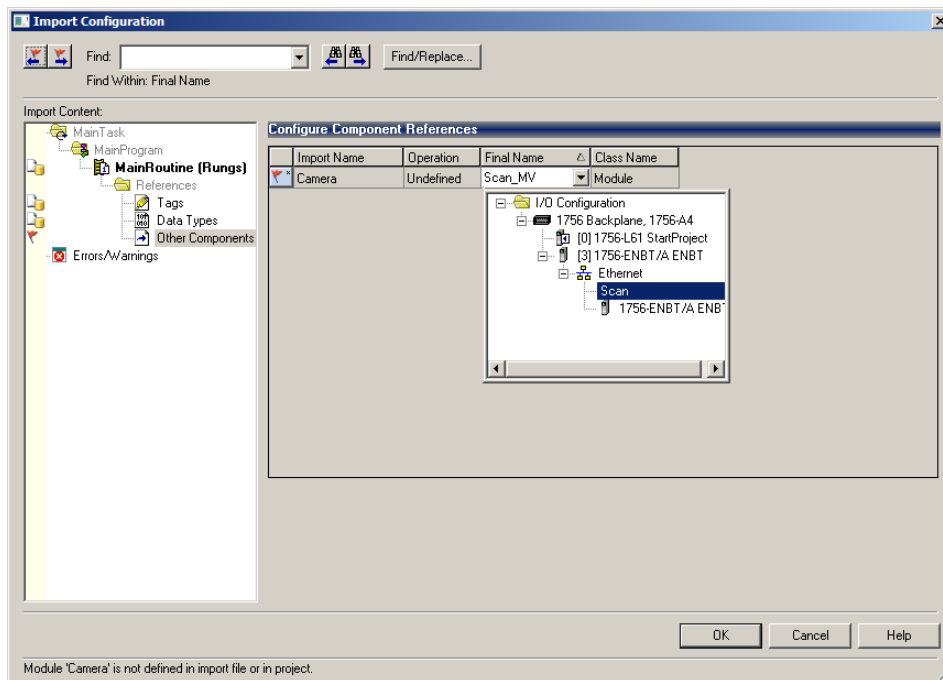
Doppelklicken Sie auf das in einem früheren Schritt zugewiesene Eingangs-Tag mit dem Kameranamen.



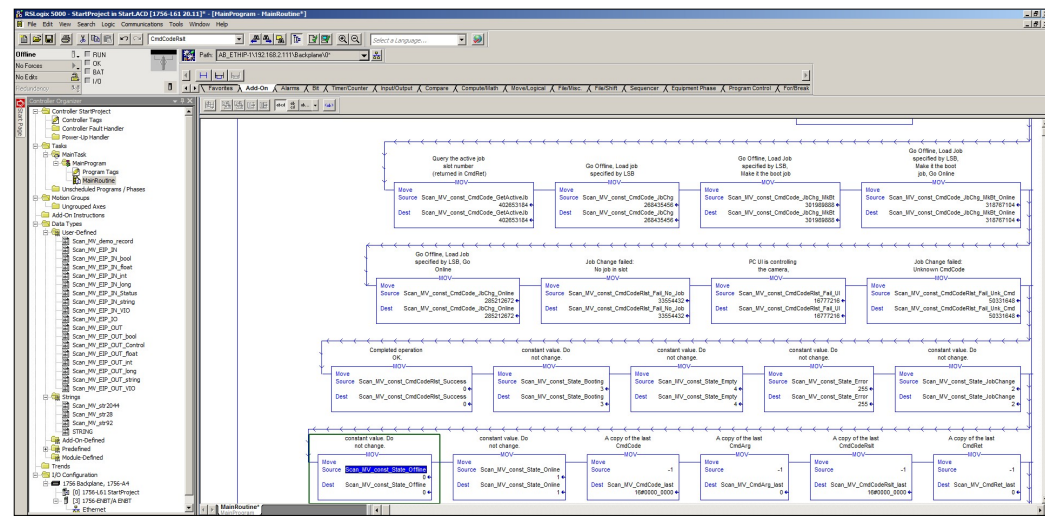




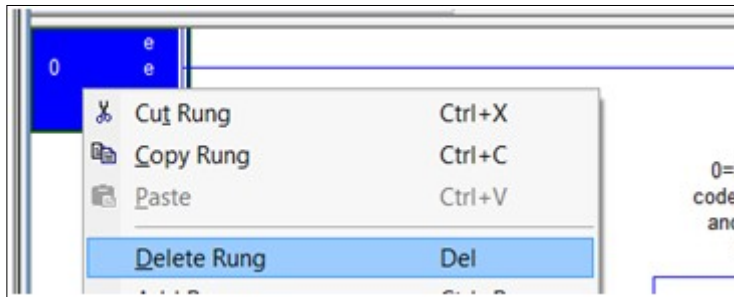
Klicken Sie in der Baumansicht auf **Other Components** (Andere Komponenten), um die Komponentenbezüge (**Component References**) festzulegen. Wählen Sie dazu in der Spalte **Final Name** (Endgültiger Name) die Kamera aus. Schließen Sie den Import mit einem Klick auf **OK** ab.



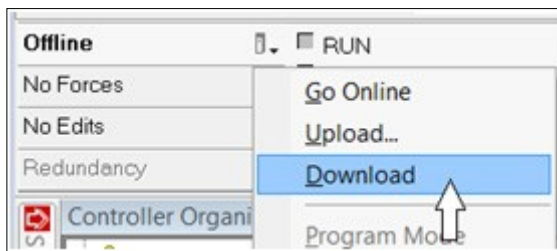
Nach dem Klick auf **OK** werden die Ansichten **Main Routine** (Hauptroutine) und **User-Defined Tags** (Nutzerdefinierte Tags) automatisch vervollständigt.



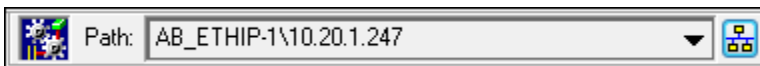
Löschen Sie alle leeren Zeilen. Prüfen Sie dazu Zeile **0**.



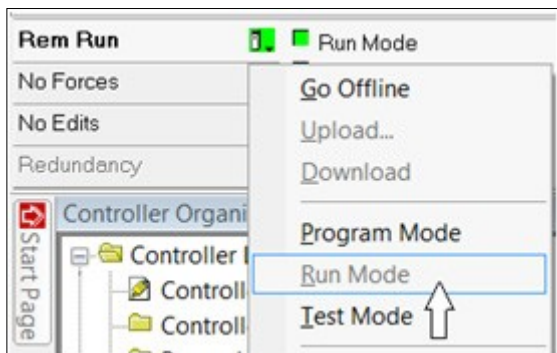
Laden Sie nun das Projekt auf die SPS herunter.



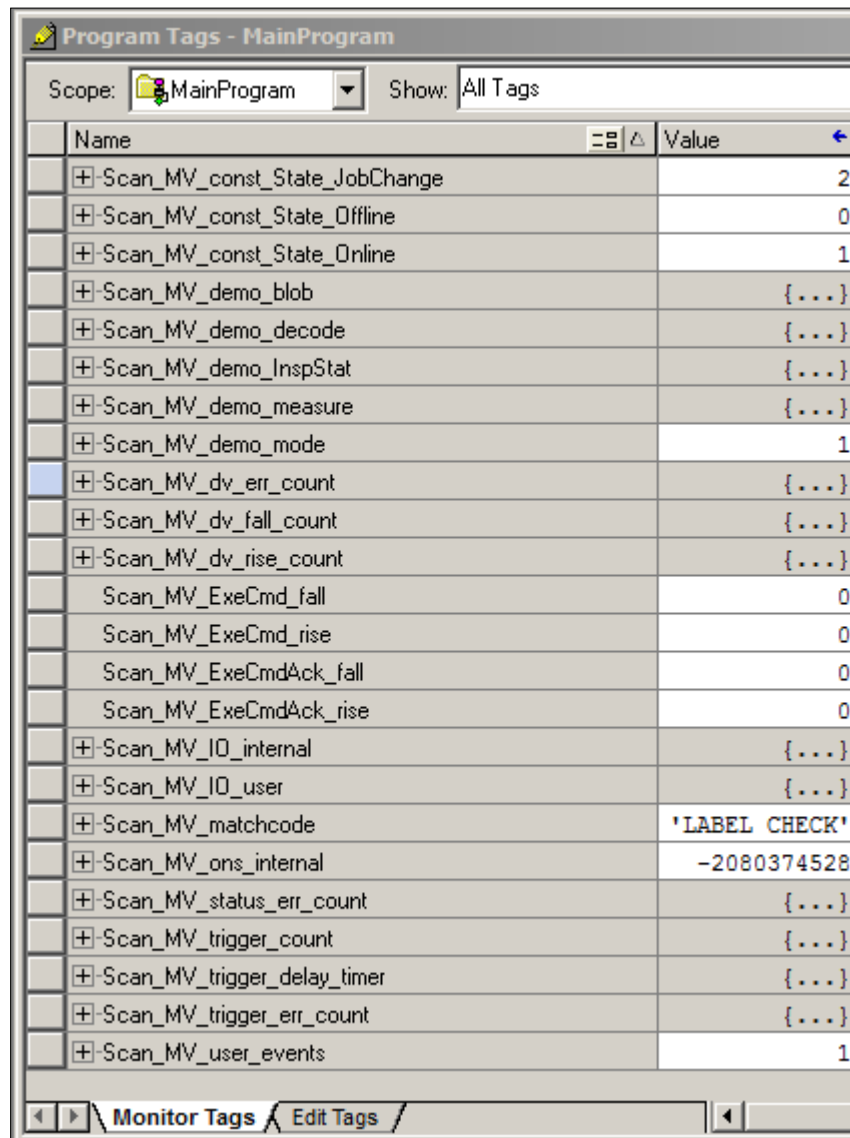
**Hinweis:** Im Projekt muss der Pfad zur SPS angegeben sein. Ansonsten kann keine Verbindung zur SPS hergestellt werden.



Schalten Sie die SPS in den Betriebsmodus (**Run Mode**).



Öffnen Sie das Fenster **Program Tags** (Programm-Tags) und wählen Sie **Monitor Tags** (Tags überwachen) aus.



The screenshot shows the 'Program Tags - MainProgram' window. At the top, there is a 'Scope' dropdown set to 'MainProgram' and a 'Show:' dropdown set to 'All Tags'. Below this is a table with two columns: 'Name' and 'Value'. The table contains 25 rows of tags. The tag 'Scan\_MV\_dv\_err\_count' is highlighted in blue. At the bottom of the window, there are two buttons: 'Monitor Tags' and 'Edit Tags'. The 'Monitor Tags' button is highlighted with a mouse cursor.

Name	Value
+ Scan_MV_const_State_JobChange	2
+ Scan_MV_const_State_Offline	0
+ Scan_MV_const_State_Online	1
+ Scan_MV_demo_blob	{...}
+ Scan_MV_demo_decode	{...}
+ Scan_MV_demo_InspStat	{...}
+ Scan_MV_demo_measure	{...}
+ Scan_MV_demo_mode	1
+ Scan_MV_dv_err_count	{...}
+ Scan_MV_dv_fall_count	{...}
+ Scan_MV_dv_rise_count	{...}
Scan_MV_ExeCmd_fall	0
Scan_MV_ExeCmd_rise	0
Scan_MV_ExeCmdAck_fall	0
Scan_MV_ExeCmdAck_rise	0
+ Scan_MV_ID_internal	{...}
+ Scan_MV_ID_user	{...}
+ Scan_MV_matchcode	'LABEL CHECK'
+ Scan_MV_ons_internal	-2080374528
+ Scan_MV_status_err_count	{...}
+ Scan_MV_trigger_count	{...}
+ Scan_MV_trigger_delay_timer	{...}
+ Scan_MV_trigger_err_count	{...}
+ Scan_MV_user_events	1

Maximieren Sie die Ansicht **Scan\_MV\_IO\_user**, sodass das **Echo** in den Strukturen **IN.Status** und **.OUT.Control** zu sehen ist.

Name	Value
Scan_MV_IO_user.IN.Status.reserved15	0
+ Scan_MV_IO_user.IN.Status.Echo	0
+ Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000
+ Scan_MV_IO_user.IN.Status.CmdRet	0
+ Scan_MV_IO_user.IN.Status.reserved96_103	0
+ Scan_MV_IO_user.IN.Status.reserved104_111	0
+ Scan_MV_IO_user.IN.Status.State	1
+ Scan_MV_IO_user.IN.Status.reserved120_127	0
+ Scan_MV_IO_user.IN.VIO	{...}
+ Scan_MV_IO_user.IN.bool	{...}
+ Scan_MV_IO_user.IN.int	{...}
+ Scan_MV_IO_user.IN.long	{...}
+ Scan_MV_IO_user.IN.float	{...}
+ Scan_MV_IO_user.IN.string	{...}
- Scan_MV_IO_user.OUT	{...}
- Scan_MV_IO_user.OUT.Control	{...}
- Scan_MV_IO_user.OUT.Control.GoOnline	0
- Scan_MV_IO_user.OUT.Control.GoOffline	0
- Scan_MV_IO_user.OUT.Control.reserved2	0
- Scan_MV_IO_user.OUT.Control.reserved3	0
- Scan_MV_IO_user.OUT.Control.ResetError	0
- Scan_MV_IO_user.OUT.Control.ResetCount	0
- Scan_MV_IO_user.OUT.Control.reserved6	0
- Scan_MV_IO_user.OUT.Control.ExeCmd	0
- Scan_MV_IO_user.OUT.Control.Trigger	0
- Scan_MV_IO_user.OUT.Control.reserved9	0
- Scan_MV_IO_user.OUT.Control.reserved10	0
- Scan_MV_IO_user.OUT.Control.ResetDataValid	0
- Scan_MV_IO_user.OUT.Control.reserved12	0
- Scan_MV_IO_user.OUT.Control.reserved13	0
- Scan_MV_IO_user.OUT.Control.reserved14	0
- Scan_MV_IO_user.OUT.Control.reserved15	0
+ Scan_MV_IO_user.OUT.Control.Echo	0
+ Scan_MV_IO_user.OUT.Control.CmdCode	16#0000_0000
+ Scan_MV_IO_user.OUT.Control.CmdArg	0

Ändern Sie **.OUT.Control.Echo** auf einen Wert ungleich null.

-Scan_MV_IO_user.OUT.Control.reserved15	0
+ Scan_MV_IO_user.OUT.Control.Echo	4321
+ Scan_MV_IO_user.OUT.Control.CmdCode	16#0000_0000

Vergewissern Sie sich, dass der Wert für **Scan\_MV\_IO\_user.IO.IN.Status.Echo** mit dem Wert für **.OUT.Control.Echo** übereinstimmt.

-Scan_MV_IO_user.IN.Status.reserved15	0
+ Scan_MV_IO_user.IN.Status.Echo	4321
+ Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000

Ist dies der Fall, können die SPS und die Kamera in beide Richtungen miteinander kommunizieren.

Der Democode erwartet, dass in die Kamera ein Demo-Vision-Job geladen wird und in der Folge die Vision-Tool-Ergebnisse in folgenden Eingangs-Tags (Kamera an SPS) erscheinen:

- .IN.bool.bool1, bool2 und bool3
- .IN.long.long1
- .IN.float.float1
- .IN.string.string1

Der Democode regelt die **Steuer-** und **Statussignale** der Kamera unabhängig davon, welcher Vision-Job geladen wurde. Genauere Informationen zum Democode und zu Vision-Jobs finden Sie im Kapitel [Allen-Bradley-SPS über das Generic Ethernet Module für den EtherNet/IP-Betrieb einrichten](#).

Zum Senden eines Triggersignals an die Kamera scrollen Sie zu **Scan\_MV\_IO\_user.Control.Trigger**.

-Scan_MV_IO_user.OUT.Control.ExeCmd	0
-Scan_MV_IO_user.OUT.Control.Trigger	0
-Scan_MV_IO_user.OUT.Control.reserved9	0

Setzen Sie den Trigger auf **1**. Dadurch löst der Democode die Kamera aus, verarbeitet die neuen Inspektionsdaten, erfasst die Ergebnisse in den **Scan\_MV\_demo\_xxxx**-Tags und löscht das Statussignal **DataValid** (DatenGültig).

Wenn die Kamera ausgelöst wird, wechselt das Steuersignal **Trigger** in den Zustand **0**. Mit fortschreitender Datenverarbeitung steigen die Werte für **Scan\_MV\_dv\_fall\_count** sowie für „pass“ (Erfolgreich) bzw. „fail“ (Fehlgeschlagen) in den **Scan\_MV\_demo\_xxxx**-Tags.

Name	Value
[-] Scan_MV_demo_blob	{...}
[+] Scan_MV_demo_blob.pass_count	{...}
[-] Scan_MV_demo_blob.fail_count	{...}
[+] Scan_MV_demo_blob.fail_count.PRE	0
[+] Scan_MV_demo_blob.fail_count.ACC	30
[-] Scan_MV_demo_blob.fail_count.CU	0
[-] Scan_MV_demo_blob.fail_count.CD	0
[-] Scan_MV_demo_blob.fail_count.DN	1
[-] Scan_MV_demo_blob.fail_count.OV	0
[-] Scan_MV_demo_blob.fail_count.UN	0
[-] Scan_MV_demo_blob.bool	0
[+] Scan_MV_demo_blob.long	6
[+] Scan_MV_demo_blob.long_max	6
[+] Scan_MV_demo_blob.long_min	4
[-] Scan_MV_demo_blob.float	0.0
[-] Scan_MV_demo_blob.float_min	0.0
[-] Scan_MV_demo_blob.float_max	0.0
[+] Scan_MV_demo_blob.string	''
[+] Scan_MV_demo_decode	{...}
[+] Scan_MV_demo_InspStat	{...}
[+] Scan_MV_demo_measure	{...}
[+] Scan_MV_demo_mode	1
[+] Scan_MV_dv_err_count	{...}
[-] Scan_MV_dv_fall_count	{...}
[+] Scan_MV_dv_fall_count.PRE	0
[+] Scan_MV_dv_fall_count.ACC	59
[-] Scan_MV_dv_fall_count.CU	0





# Allen-Bradley-SPS über das Generic Ethernet Module für den EtherNet/IP-Betrieb einrichten

In diesem Kapitel wird beschrieben, wie Sie mithilfe des Generic Ethernet Module eine Allen-Bradley-SPS für den EtherNet/IP-Betrieb einrichten.

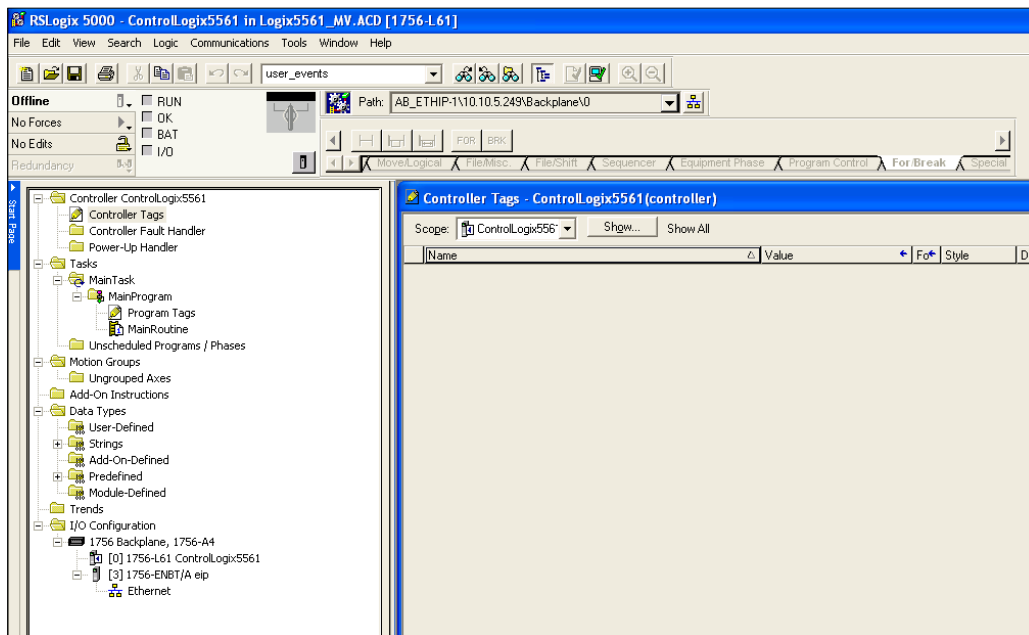
## Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für EtherNet/IP aktiviert werden. Wie man Kommunikationsprotokolle für den CS50-Sensor aktiviert oder ändert, erfahren Sie in **Kapitel 1 [Industrieprotokolle aktivieren](#)**.
- Bei CS50-Software-Jobs wird die EtherNet/IP-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert. Weitere Informationen dazu, wie man Job-Parameter und Ausgänge mit di-soric Link-Tags verknüpft, finden Sie in der *CS50-Software-Hilfe* im Abschnitt **di-soric Link**.

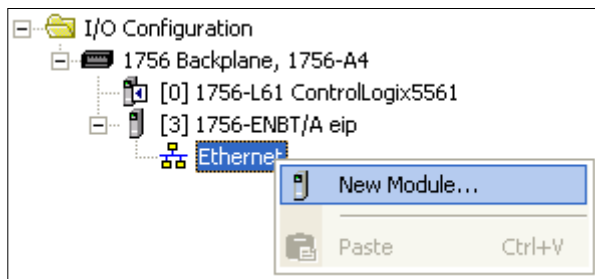
## Kamera in eine SPS-Umgebung integrieren

Im Folgenden wird vorausgesetzt, dass Sie eine Allen-Bradley-SPS mit Rockwell RSLogix 5000 Version 16 oder höher verwenden. Die Versionen 19 und 20 von RSLogix können sich geringfügig von den Screenshots unterscheiden, der Integrationsvorgang ist aber im Grunde derselbe.

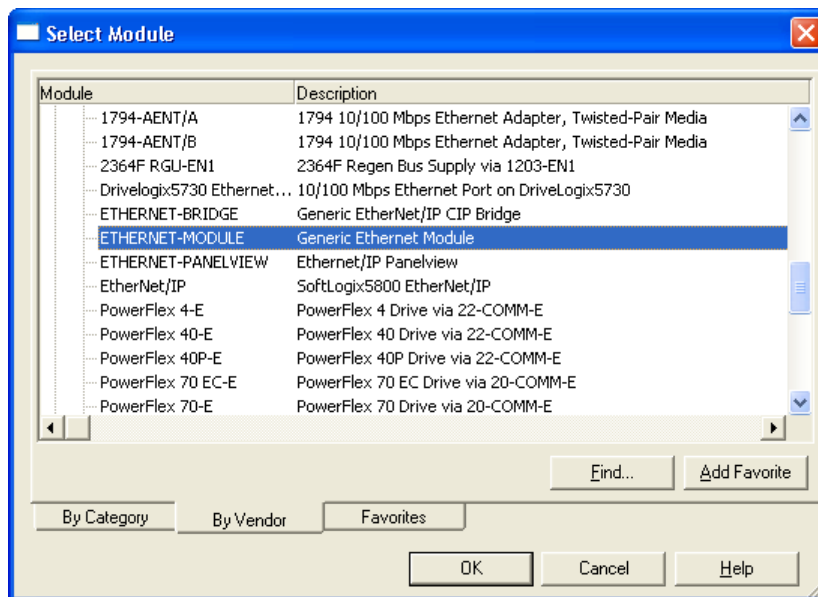
Erstellen Sie für das Basissystem die E/A-Konfiguration (I/O Configuration) einschließlich der Ethernet-Schnittstelle des Systems.



Fügen Sie die Kamera hinzu, indem Sie mit der rechten Maustaste auf die Ethernet-Schnittstelle klicken und dann **New Module** (Neues Modul) auswählen.



Wählen Sie **ETHERNET-MODULE – Generic Ethernet Module** (ETHERNET-Modul – Generic Ethernet Module) aus und klicken Sie auf **OK**.



Konfigurieren Sie folgende Felder wie angegeben:

**Name:** Legen Sie hier einen aussagekräftigen Namen für die Kamera fest.

Im Beispiel haben wir Scan\_MV verwendet.

**IP Address (IP-Adresse):** Die IP-Adresse der Kamera

**Comm Format (Kommunikationsformat):** „Data –DINT“ (Daten – DINT)

**Input (Eingang), Assembly Instance (Assembly-Instanz):** 102

**Input (Eingang), Size (Größe):** 80

**Output (Ausgang), Assembly Instance (Assembly-Instanz):** 114

**Output (Ausgang), Size (Größe):** 80

**Configuration (Konfiguration), Assembly Instance (Assembly-Instanz):** 1

**Configuration (Konfiguration), Size (Größe):** 0 (keine)

Wenn Sie alle Felder konfiguriert haben, klicken Sie auf OK.

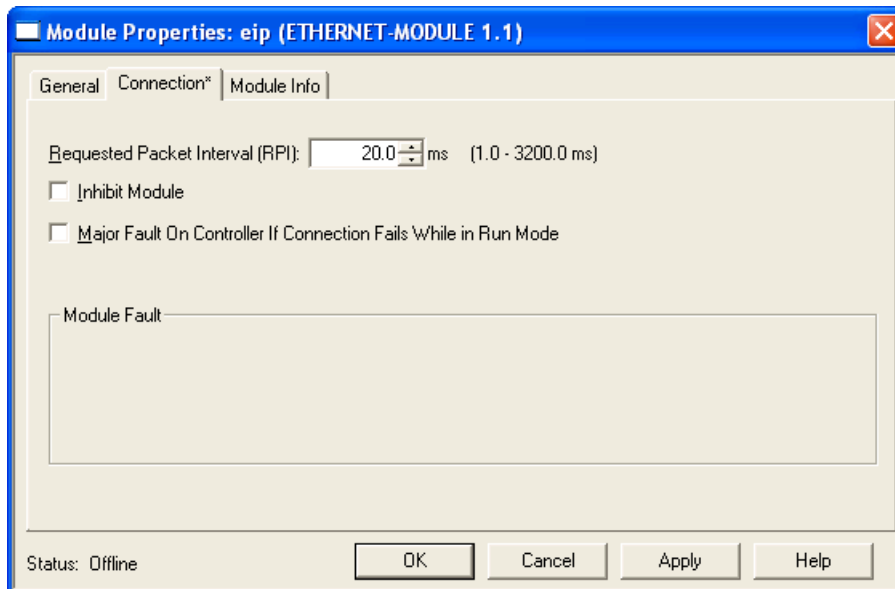
*Beispiel:*

The screenshot shows the 'New Module' dialog box with the following configuration:

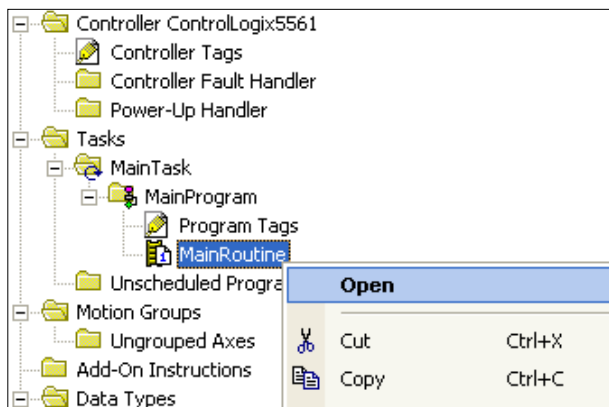
- Type: ETHERNET-MODULE Generic Ethernet Module
- Vendor: Allen-Bradley
- Parent: ENBT
- Name: Scan\_MV
- Description: (empty)
- Comm Format: Data - DINT
- Address / Host Name: IP Address (selected)
- Connection Parameters:
  - Input: Assembly Instance: 102, Size: 80 (32-bit)
  - Output: Assembly Instance: 114, Size: 80 (32-bit)
  - Configuration: Assembly Instance: 1, Size: 0 (8-bit)
  - Status Input: (empty)
  - Status Output: (empty)
- Open Module Properties
- Buttons: OK, Cancel, Help

Konfigurieren Sie den Wert für **Requested Packet Interval (RPI)** (Angefordertes Paketintervall) und klicken Sie dann auf **OK**.

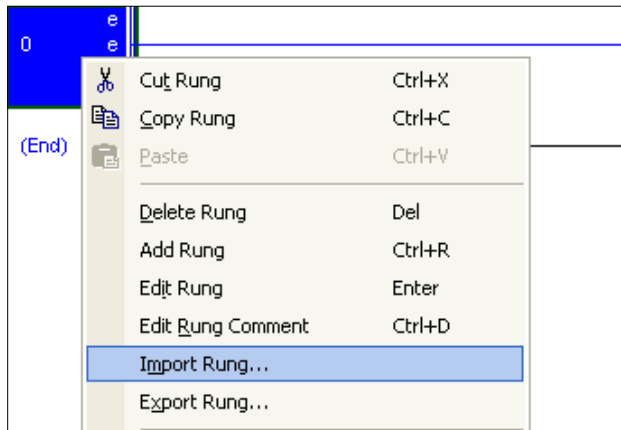
Dieser Wert muss mindestens **10 ms** betragen. Empfohlen wird ein Wert von mindestens **20 ms**.



Öffnen Sie **Main Routine** (Hauptroutine).



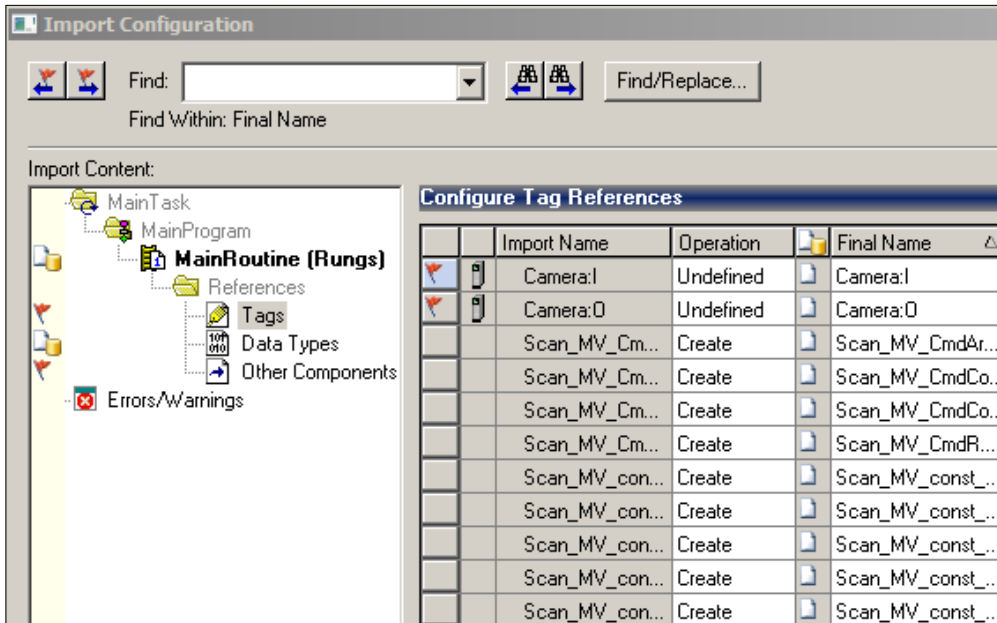
Klicken Sie mit der rechten Maustaste auf die oberste Zeile und wählen Sie **Import Rung** (Zeile importieren) aus.



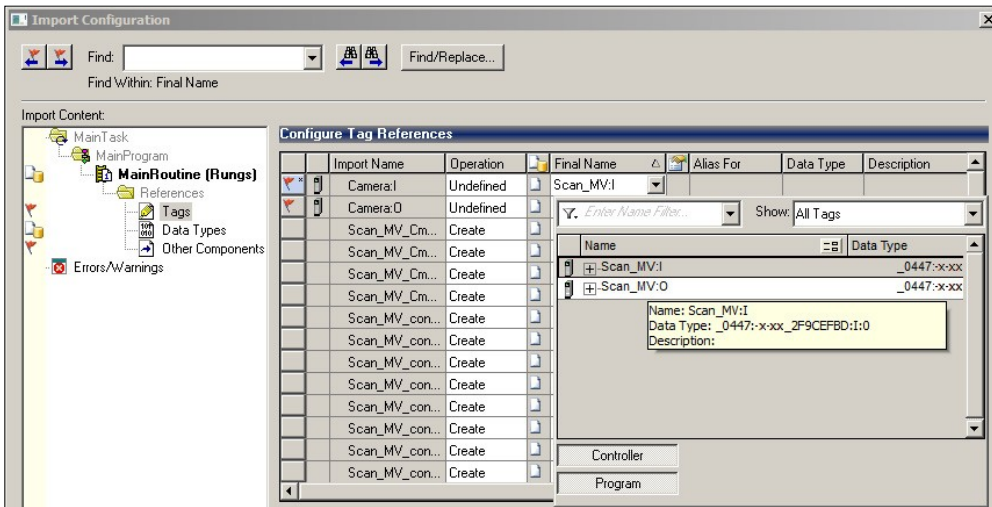
Machen Sie die Datei **32-000003-2.L5X** ausfindig und klicken Sie dann auf **Import** (Importieren).



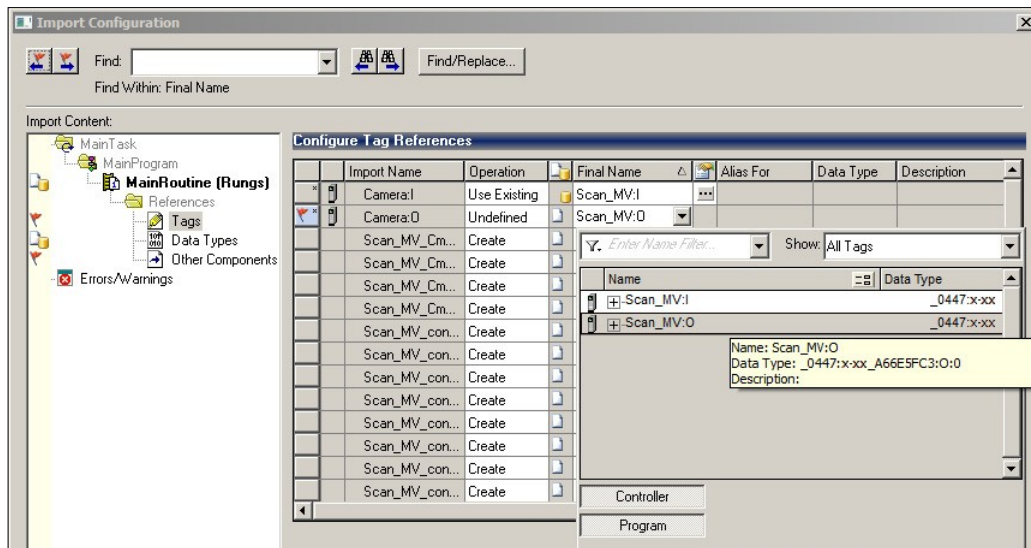
Suchen Sie im Fenster **Import Configuration** (Importkonfiguration) den **Modulnamen**, der **Generic Module** zugewiesen wurde. In diesem Falle heißt das Modul **Camera** (Kamera).



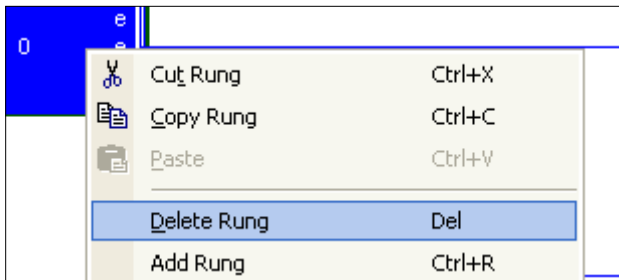
Klicken Sie auf **Camera:I** und dann auf den Abwärtspfeil. Klicken Sie anschließend auf den darunter angezeigten Eintrag **Scan\_MV:I**.



Klicken Sie auf **Camera:0** und dann auf den Abwärtspfeil. Klicken Sie anschließend auf den darunter angezeigten Eintrag **Scan\_MV:0**.



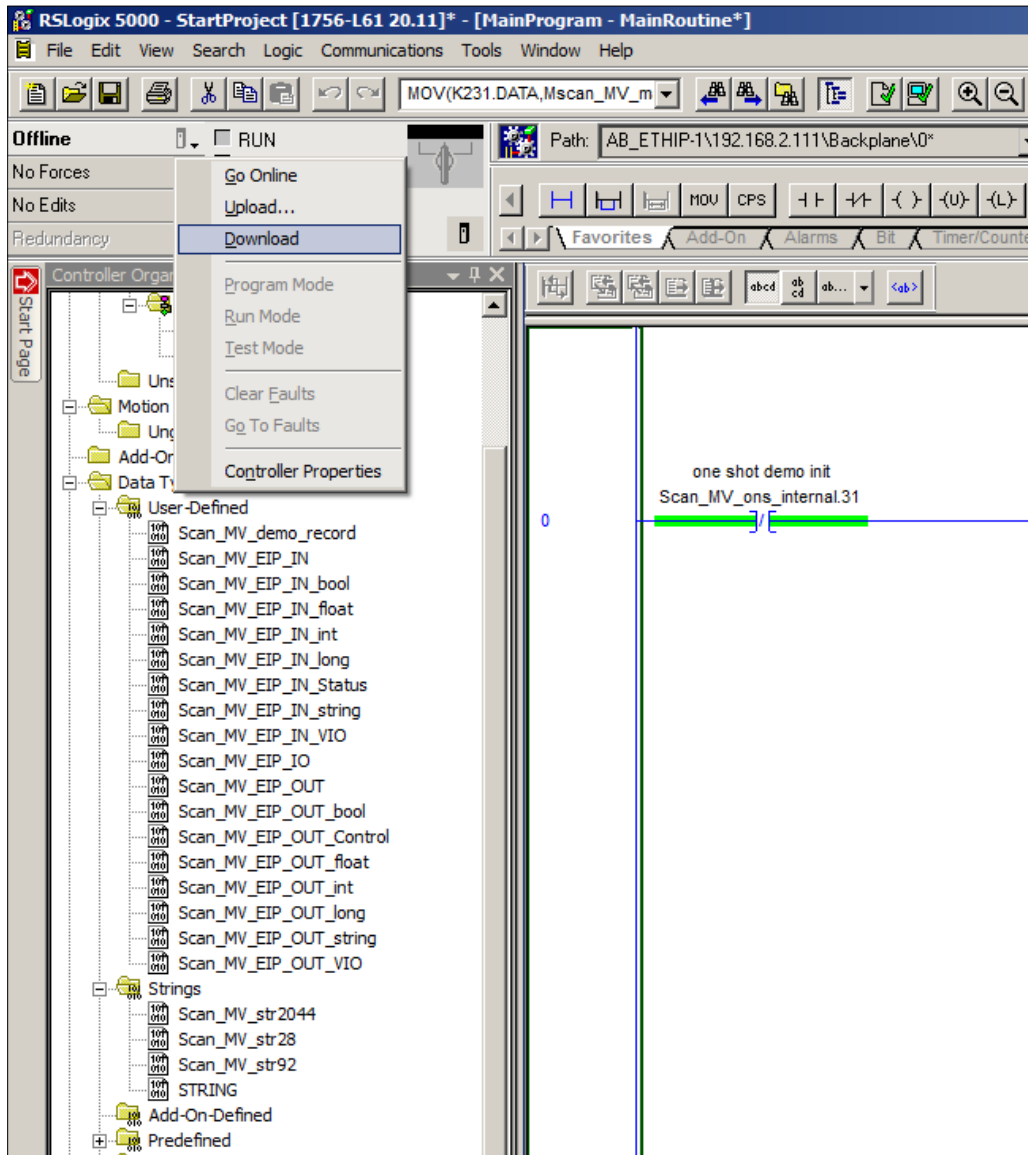
Klicken Sie in der Baumansicht auf **Other Components** (Andere Komponenten). Klicken Sie auf **OK**. Löschen Sie alle leeren Zeilen (Zeile **0** ist möglicherweise leer).



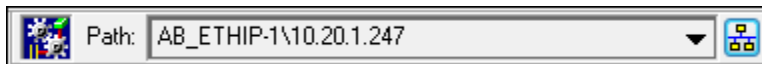
Die Tags und das Hauptprogramm sind jetzt konfiguriert. Sie können nun die Kommunikation mit der Kamera testen.



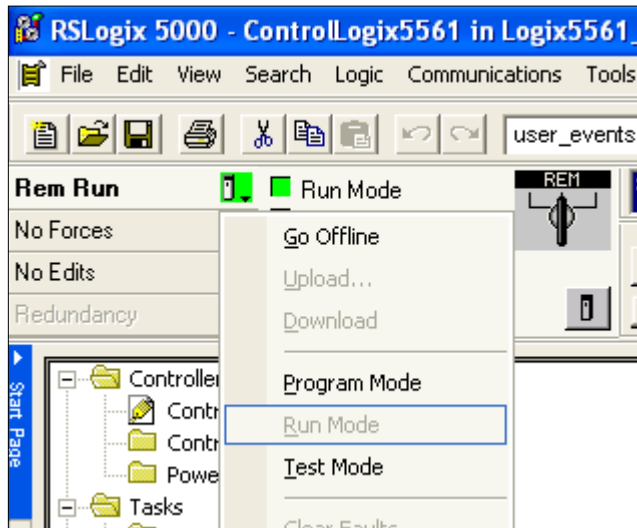
Klicken Sie auf die Steuertaste neben **Offline** und wählen Sie dann **Download** (Herunterladen) aus.



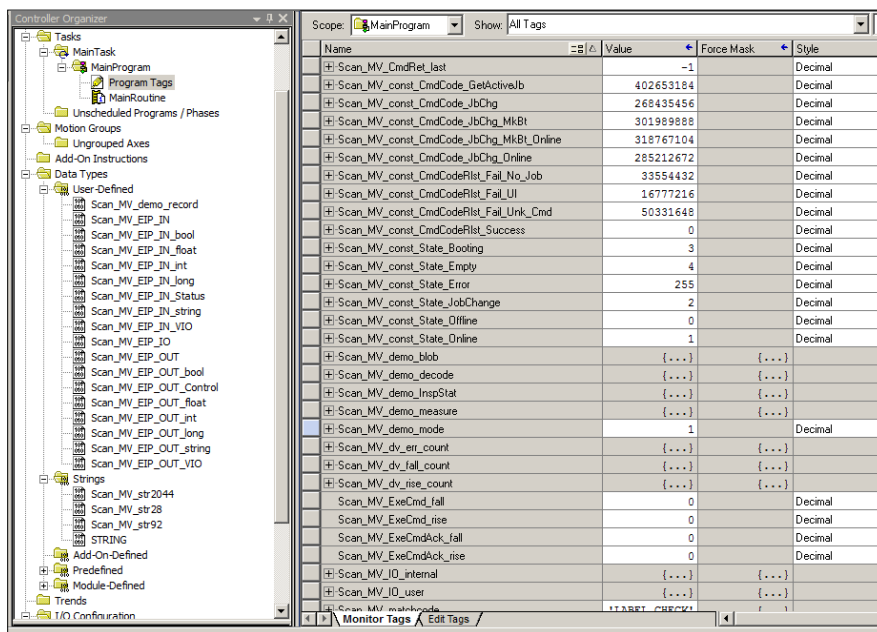
**Hinweis:** Im Projekt muss der Pfad zur SPS angegeben sein. Ansonsten kann keine Verbindung zur SPS hergestellt werden.



Wenn der Download beendet ist, stellen Sie sicher, dass sich die SPS im Betriebsmodus (**Run Mode**) befindet.



Zum Öffnen der Programm-Tags doppelklicken Sie auf **Program Tags** (Programm-Tags) und wählen Sie dann im unteren Bereich des Fensters die Registerkarte **Monitor Tags** (Tags überwachen) aus.



Maximieren Sie die Ansicht **Scan\_MV\_IO\_user**, sodass die Strukturen **.IN.Status** und **.OUT.Control** zu sehen sind. Scrollen Sie dann nach unten, bis Sie **Scan\_MV\_IO\_user.OUT.Control.Echo** sehen.

Name	Value
— Scan_MV_IO_user.IN.Status.reserved15	0
+ Scan_MV_IO_user.IN.Status.Echo	0
+ Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000
+ Scan_MV_IO_user.IN.Status.CmdRiet	0
+ Scan_MV_IO_user.IN.Status.reserved96_103	0
+ Scan_MV_IO_user.IN.Status.reserved104_111	0
+ Scan_MV_IO_user.IN.Status.State	1
+ Scan_MV_IO_user.IN.Status.reserved120_127	0
+ Scan_MV_IO_user.IN.VIO	{...}
+ Scan_MV_IO_user.IN.bool	{...}
+ Scan_MV_IO_user.IN.int	{...}
+ Scan_MV_IO_user.IN.long	{...}
+ Scan_MV_IO_user.IN.float	{...}
+ Scan_MV_IO_user.IN.string	{...}
— Scan_MV_IO_user.OUT	{...}
— Scan_MV_IO_user.OUT.Control	{...}
— Scan_MV_IO_user.OUT.Control.GoOnline	0
— Scan_MV_IO_user.OUT.Control.GoOffline	0
— Scan_MV_IO_user.OUT.Control.reserved2	0
— Scan_MV_IO_user.OUT.Control.reserved3	0
— Scan_MV_IO_user.OUT.Control.ResetError	0
— Scan_MV_IO_user.OUT.Control.ResetCount	0
— Scan_MV_IO_user.OUT.Control.reserved6	0
— Scan_MV_IO_user.OUT.Control.ExeCmd	0
— Scan_MV_IO_user.OUT.Control.Trigger	0
— Scan_MV_IO_user.OUT.Control.reserved9	0
— Scan_MV_IO_user.OUT.Control.reserved10	0
— Scan_MV_IO_user.OUT.Control.ResetDataValid	0
— Scan_MV_IO_user.OUT.Control.reserved12	0
— Scan_MV_IO_user.OUT.Control.reserved13	0
— Scan_MV_IO_user.OUT.Control.reserved14	0
— Scan_MV_IO_user.OUT.Control.reserved15	0
+ Scan_MV_IO_user.OUT.Control.Echo	0
+ Scan_MV_IO_user.OUT.Control.CmdCode	16#0000_0000
+ Scan_MV_IO_user.OUT.Control.CmdArg	0

Ändern Sie **.OUT.Control.Echo** auf einen Wert ungleich null.

Name	Value
+ Scan_MV_IO_user.IN.Status.reserved104_111	0
+ Scan_MV_IO_user.IN.Status.State	1
+ Scan_MV_IO_user.IN.Status.reserved120_127	0
+ Scan_MV_IO_user.IN.VIO	{...}
+ Scan_MV_IO_user.IN.bool	{...}
+ Scan_MV_IO_user.IN.int	{...}
+ Scan_MV_IO_user.IN.long	{...}
+ Scan_MV_IO_user.IN.float	{...}
+ Scan_MV_IO_user.IN.string	{...}
- Scan_MV_IO_user.OUT	{...}
- Scan_MV_IO_user.OUT.Control	{...}
- Scan_MV_IO_user.OUT.Control.GoOnline	0
- Scan_MV_IO_user.OUT.Control.GoOffline	0
- Scan_MV_IO_user.OUT.Control.reserved2	0
- Scan_MV_IO_user.OUT.Control.reserved3	0
- Scan_MV_IO_user.OUT.Control.ResetError	0
- Scan_MV_IO_user.OUT.Control.ResetCount	0
- Scan_MV_IO_user.OUT.Control.reserved6	0
- Scan_MV_IO_user.OUT.Control.ExeCmd	0
- Scan_MV_IO_user.OUT.Control.Trigger	0
- Scan_MV_IO_user.OUT.Control.reserved9	0
- Scan_MV_IO_user.OUT.Control.reserved10	0
- Scan_MV_IO_user.OUT.Control.ResetDataValid	0
- Scan_MV_IO_user.OUT.Control.reserved12	0
- Scan_MV_IO_user.OUT.Control.reserved13	0
- Scan_MV_IO_user.OUT.Control.reserved14	0
- Scan_MV_IO_user.OUT.Control.reserved15	0
+ Scan_MV_IO_user.OUT.Control.Echo	1234
+ Scan_MV_IO_user.OUT.Control.CmdCode	16#0000_0000
+ Scan_MV_IO_user.OUT.Control.CmdArg	0
+ Scan_MV_IO_user.OUT.Control.reserved96_127	0

Monitor Tags / Edit Tags

Scrollen Sie dann nach unten, bis Sie **Scan\_MV\_IO\_user.IO.IN.Status.Echo** sehen, und vergewissern Sie sich, dass der entsprechende Wert mit dem Wert für **.OUT.Control.Echo** übereinstimmt.

Name	Value
Scan_MV_ExeCmdAck_rise	0
+ Scan_MV_IO_internal	{...}
- Scan_MV_IO_user	{...}
- Scan_MV_IO_user.IN	{...}
- Scan_MV_IO_user.IN.Status	{...}
- Scan_MV_IO_user.IN.Status.Online	1
- Scan_MV_IO_user.IN.Status.ExpBusy	0
- Scan_MV_IO_user.IN.Status.AcqBusy	0
- Scan_MV_IO_user.IN.Status.TriggerReady	1
- Scan_MV_IO_user.IN.Status.Error	0
- Scan_MV_IO_user.IN.Status.ResetCountAck	0
- Scan_MV_IO_user.IN.Status.reserved6	0
- Scan_MV_IO_user.IN.Status.ExeCmdAck	0
- Scan_MV_IO_user.IN.Status.TriggerAck	0
- Scan_MV_IO_user.IN.Status.InspBusy	0
- Scan_MV_IO_user.IN.Status.InspStat	0
- Scan_MV_IO_user.IN.Status.DataValid	0
- Scan_MV_IO_user.IN.Status.reserved12	0
- Scan_MV_IO_user.IN.Status.reserved13	0
- Scan_MV_IO_user.IN.Status.reserved14	0
- Scan_MV_IO_user.IN.Status.reserved15	0
+ Scan_MV_IO_user.IN.Status.Echo	1234
+ Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000
+ Scan_MV_IO_user.IN.Status.CmdRet	0
+ Scan_MV_IO_user.IN.Status.reserved96_103	0
+ Scan_MV_IO_user.IN.Status.reserved104_111	0
+ Scan_MV_IO_user.IN.Status.State	1
+ Scan_MV_IO_user.IN.Status.reserved120_127	0
+ Scan_MV_IO_user.IN.VIO	{...}
+ Scan_MV_IO_user.IN.bool	{...}
+ Scan_MV_IO_user.IN.int	{...}

Ist dies der Fall, können die SPS und die Kamera in beide Richtungen miteinander kommunizieren.

Zum Senden eines Triggersignals an die Kamera scrollen Sie zu **Scan\_MV\_IO\_user.Control.Trigger**.

Name	Value
Scan_MV_IO_user.IN.int	{...}
Scan_MV_IO_user.IN.long	{...}
Scan_MV_IO_user.IN.float	{...}
Scan_MV_IO_user.IN.string	{...}
Scan_MV_IO_user.OUT	{...}
Scan_MV_IO_user.OUT.Control	{...}
Scan_MV_IO_user.OUT.Control.GoOnline	0
Scan_MV_IO_user.OUT.Control.GoOffline	0
Scan_MV_IO_user.OUT.Control.reserved2	0
Scan_MV_IO_user.OUT.Control.reserved3	0
Scan_MV_IO_user.OUT.Control.ResetError	0
Scan_MV_IO_user.OUT.Control.ResetCount	0
Scan_MV_IO_user.OUT.Control.reserved6	0
Scan_MV_IO_user.OUT.Control.ExeCmd	0
Scan_MV_IO_user.OUT.Control.Trigger	0
Scan_MV_IO_user.OUT.Control.reserved9	0

Setzen Sie den **Trigger** auf **1**. Dadurch löst der Democode die Kamera aus, verarbeitet die neuen Inspektionsdaten, erfasst die Ergebnisse in den **Scan\_MV\_demo\_xxxx**-Tags und löscht das Statussignal **DataValid** (DatenGültig). Wenn die Kamera ausgelöst wird, wechselt das Steuersignal **Trigger** in den Zustand **0**. Nachdem alle Daten verarbeitet wurden, werden der Wert **Scan\_MV\_dv\_fall\_count** sowie die Zählerstände für **pass/fail** (erfolgreich/fehlgeschlagen) in den **Scan\_MV\_demo\_xxxx**-Tags aktualisiert. Beispiel:

Name	Value
[-] Scan_MV_demo_blob	{...}
[-] Scan_MV_demo_blob.pass_count	{...}
+ Scan_MV_demo_blob.pass_count.PRE	0
+ Scan_MV_demo_blob.pass_count.ACC	1
- Scan_MV_demo_blob.pass_count.CU	0
- Scan_MV_demo_blob.pass_count.CD	0
- Scan_MV_demo_blob.pass_count.DN	1
- Scan_MV_demo_blob.pass_count.OV	0
- Scan_MV_demo_blob.pass_count.UN	0
+ Scan_MV_demo_blob.fail_count	{...}
- Scan_MV_demo_blob.bool	0
+ Scan_MV_demo_blob.long	6
+ Scan_MV_demo_blob.long_max	6
+ Scan_MV_demo_blob.long_min	4
- Scan_MV_demo_blob.float	0.0
- Scan_MV_demo_blob.float_min	0.0
- Scan_MV_demo_blob.float_max	0.0
+ Scan_MV_demo_blob.string	''
+ Scan_MV_demo_decode	{...}
+ Scan_MV_demo_InspStat	{...}
+ Scan_MV_demo_measure	{...}
+ Scan_MV_demo_mode	1
+ Scan_MV_dv_err_count	{...}
[-] Scan_MV_dv_fall_count	{...}
+ Scan_MV_dv_fall_count.PRE	0
+ Scan_MV_dv_fall_count.ACC	1
- Scan_MV_dv_fall_count.CU	0
- Scan_MV_dv_fall_count.CD	0
- Scan_MV_dv_fall_count.DN	1
- Scan_MV_dv_fall_count.OV	0
- Scan_MV_dv_fall_count.UN	0
+ Scan_MV_dv_rise_count	{...}

## Kamera parametrieren

Öffnen Sie die Tags **Scan\_MV\_IO\_user.OUT.long**, **float** und **string** und überprüfen Sie, ob sie so wie unten dargestellt konfiguriert wurden.

Name	Value	Style	Data Type
[-] Scan_MV_IO_user	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.IN	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.Control	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.VIO	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.bool	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.int	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.long	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.long.long101	4	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long102	4	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long103	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long104	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long105	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long106	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long107	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long108	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long109	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.long.long110	0	Decimal	DINT
[-] Scan_MV_IO_user.OUT.float	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.float.float101	100.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float102	200.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float103	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float104	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float105	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float106	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float107	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float108	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float109	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.float.float110	0.0	Float	REAL
[-] Scan_MV_IO_user.OUT.string	{ ... }	[	Scan_...
[-] Scan_MV_IO_user.OUT.string.string101	'LABEL CHECK'	[	Scan_...
[-] Scan_MV_IO_user.OUT.string.string102	''	[	Scan_...
[-] Scan_MV_IO_user.OUT.string.string103	''	[	Scan_...
[-] Scan_MV_IO_user.OUT.string.string104	''	[	Scan_...
[-] Scan_MV_matchcode	'LABEL CHECK'	[	Scan_...
[-] Scan_MV_ons_internal	-2080374528	Decimal	DINT
[-] Scan_MV_status_err_count	{ ... }	[	COUN...

Dadurch werden die Tools **Measure (float101 and float102)** (Messen [float101 und float102]), **Decode (string101)** (Dekodieren [string101]) und **Count Blob (long101 and long102)** (Objekt-Zähler [long101 und long102]) genauso konfiguriert wie mit der Funktion „Try Out“ (Ausprobieren) in der CS50-Software.

Achten Sie auf die Spalte **Description** (Beschreibung). Sie weist darauf hin, welche Funktion die einzelnen verknüpften Tags im Vision-Job übernehmen.



## Kamera auslösen

Zum Senden eines Triggersignals an die Kamera scrollen Sie zu **Scan\_MV\_IO\_user.Control.Trigger**.

Name	Value
[-] Scan_MV_IO_user	{...}
[+] Scan_MV_IO_user.IN	{...}
[-] Scan_MV_IO_user.OUT	{...}
[-] Scan_MV_IO_user.OUT.Control	{...}
[-] Scan_MV_IO_user.OUT.Control.GoOnline	0
[-] Scan_MV_IO_user.OUT.Control.GoOffline	0
[-] Scan_MV_IO_user.OUT.Control.reserved2	0
[-] Scan_MV_IO_user.OUT.Control.reserved3	0
[-] Scan_MV_IO_user.OUT.Control.ResetError	0
[-] Scan_MV_IO_user.OUT.Control.ResetCount	0
[-] Scan_MV_IO_user.OUT.Control.reserved6	0
[-] Scan_MV_IO_user.OUT.Control.ExeCmd	0
[-] Scan_MV_IO_user.OUT.Control.Trigger	0
[-] Scan_MV_IO_user.OUT.Control.reserved9	0
[-] Scan_MV_IO_user.OUT.Control.reserved10	0
[-] Scan_MV_IO_user.OUT.Control.ResetDataValid	0
[-] Scan_MV_IO_user.OUT.Control.reserved12	0

Setzen Sie den **Trigger** auf **1**. Nachdem der Trigger wieder zu **0** gewechselt hat, kann die Kamera erneut ausgelöst werden.

Ist die Kamera mit der CS50-Software verbunden, wird bei jeder Auslösung ein neues Inspektionsergebnis angezeigt. Beachten Sie, dass der Vision-Job mit vordefinierten Bildern erstellt wurde, um vorhersehbare **Erfolge** und **Fehlschläge** zu gewährleisten. Die Kameraleuchten leuchten nicht auf, wenn die Kamera ausgelöst wird.

Die Inspektionsergebnisse können in den **IN**-Tags (EIN) der SPS sowie in der CS50-Software eingesehen werden. Öffnen Sie das Fenster des RSLogix-Tags, sodass **Scan\_MV\_IO\_user.IN.Status** und **bool** zu sehen sind.

An dieser Stelle wird exemplarisch eine **erfolgreiche** Inspektion dargestellt, bei der alle folgenden Tags den Wert **1** haben:

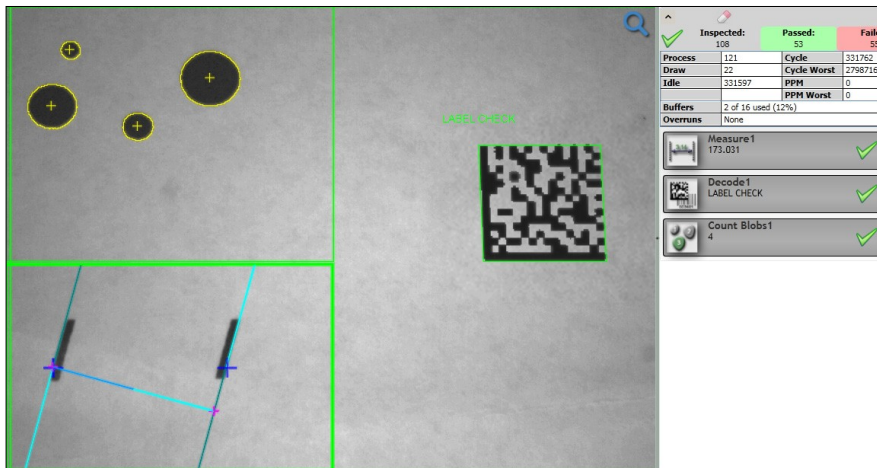
- IN.Status.InspStat
- IN.bool.bool1 (Messstatus)
- IN.bool.bool2 (Dekodier- und Matchcode-Status)
- IN.bool.bool3 (Objekt-Zähler-Status)

Name	Value	Style	Data Type	Description
[-] Scan_MV_IO_user	{...}	[	Scan_...	user's device tags when M
[-] Scan_MV_IO_user.IN	{...}	[	Scan_...	user's device tags when M
[-] Scan_MV_IO_user.IN.Status	{...}	[	Scan_...	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.Offline	1	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.ExpBusy	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.AcqBusy	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.TriggerReady	1	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.Error	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.ResetCountAck	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.reserved6	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.ExeCmdAck	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.TriggerAck	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.InspBusy	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.InspStat	1	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.DataValid	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.reserved12	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.reserved13	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.reserved14	0	Decimal	BOOL	user's device tags when M
[-] Scan_MV_IO_user.IN.Status.reserved15	0	Decimal	BOOL	user's device tags when M
[+] Scan_MV_IO_user.IN.Status.Echo	0	Decimal	INT	user's device tags when M
[+] Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000	Hex	DINT	user's device tags when M
[+] Scan_MV_IO_user.IN.Status.CmdRet	0	Decimal	DINT	user's device tags when M
[+] Scan_MV_IO_user.IN.Status.reserved96_103	0	Decimal	SINT	user's device tags when M
[+] Scan_MV_IO_user.IN.Status.reserved104_111	0	Decimal	SINT	user's device tags when M
[+] Scan_MV_IO_user.IN.Status.State	1	Decimal	SINT	user's device tags when M
[+] Scan_MV_IO_user.IN.Status.reserved120_127	0	Decimal	SINT	user's device tags when M
[+] Scan_MV_IO_user.IN.VIO	{...}	[	Scan_...	user's device tags when M
[-] Scan_MV_IO_user.IN.bool	{...}	[	Scan_...	user's device tags when M
[-] Scan_MV_IO_user.IN.bool.bool1	1	Decimal	BOOL	measure status
[-] Scan_MV_IO_user.IN.bool.bool2	1	Decimal	BOOL	decode+matchcode status
[-] Scan_MV_IO_user.IN.bool.bool3	1	Decimal	BOOL	blob count status

Wenn Sie nach unten bis zu den Werten **IN.long**, **float** und **string** scrollen, sehen Sie die genauen Ergebnisse der Vision-Tools.

Name	Value	Style	Data Type	Description
+ Scan_MV_IO_user.IN.VIO	{...}		Scan_...	user's device tag
+ Scan_MV_IO_user.IN.bool	{...}		Scan_...	user's device tag
+ Scan_MV_IO_user.IN.int	{...}		Scan_...	user's device tag
- Scan_MV_IO_user.IN.long	{...}		Scan_...	user's device tag
+ Scan_MV_IO_user.IN.long.long1	4	Decimal	DINT	Blob count
+ Scan_MV_IO_user.IN.long.long2	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long3	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long4	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long5	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long6	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long7	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long8	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long9	0	Decimal	DINT	user's device tag
+ Scan_MV_IO_user.IN.long.long10	0	Decimal	DINT	user's device tag
- Scan_MV_IO_user.IN.float	{...}		Scan_...	user's device tag
- Scan_MV_IO_user.IN.float.float1	173.0306	Float	REAL	Measure value
- Scan_MV_IO_user.IN.float.float2	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float3	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float4	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float5	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float6	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float7	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float8	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float9	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.float.float10	0.0	Float	REAL	user's device tag
- Scan_MV_IO_user.IN.string	{...}		Scan_...	user's device tag
+ Scan_MV_IO_user.IN.string.string1	'LABEL CHECK'		Scan_...	Decode text

Dies entspricht dem Inspektionsergebnis der CS50-Software.



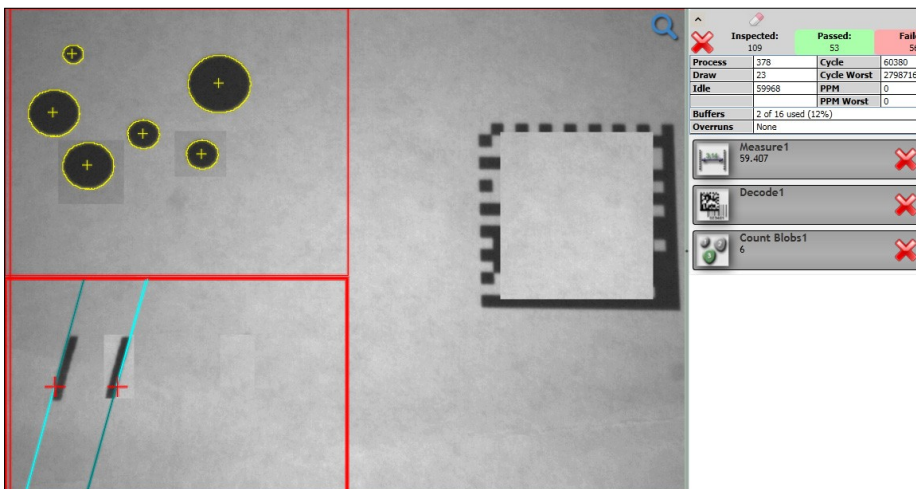
Im Beispiel ist eine **fehlgeschlagene** Inspektion zu sehen, bei der jedes Tool einen Fehler meldet.

Name	Value	Style	Data Type	Description
Scan_MV_IO_user.IN	{...}			Scan... user's device tags
Scan_MV_IO_user.IN.Status	{...}			Scan... user's device tags
Scan_MV_IO_user.IN.Status.Online	1	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.ExpBusy	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.AcqBusy	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.TriggerReady	1	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.Error	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.ResetCountAck	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved6	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.ExeCmdAck	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.TriggerAck	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.InspBusy	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.InspStat	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.DataValid	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved12	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved13	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved14	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved15	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.Echo	0	Decimal	INT	user's device tags
Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000	Hex	DINT	user's device tags
Scan_MV_IO_user.IN.Status.CmdRet	0	Decimal	DINT	user's device tags
Scan_MV_IO_user.IN.Status.reserved96_103	0	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.Status.reserved104_111	0	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.Status.State	1	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.Status.reserved120_127	0	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.VIO	{...}			Scan... user's device tags
Scan_MV_IO_user.IN.bool	{...}			Scan... user's device tags
Scan_MV_IO_user.IN.bool.bool1	0	Decimal	BOOL	measure status
Scan_MV_IO_user.IN.bool.bool2	0	Decimal	BOOL	decode+matchcod
Scan_MV_IO_user.IN.bool.bool3	0	Decimal	BOOL	blob count status

Die exakten Daten der fehlgeschlagenen Inspektion sehen in diesem Fall so aus:

Name	Value	Style	Data Type	Description
[-] Scan_MV_ID_user.IN.long	{ ... }	[	Scan_...	user's device tags
[-] Scan_MV_ID_user.IN.long.long1	6	Decimal	DINT	Blob count
[-] Scan_MV_ID_user.IN.long.long2	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long3	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long4	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long5	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long6	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long7	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long8	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long9	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.long.long10	0	Decimal	DINT	user's device tags
[-] Scan_MV_ID_user.IN.float	{ ... }	[	Scan_...	user's device tags
[-] Scan_MV_ID_user.IN.float.float1	59.406998	Float	REAL	Measure value
[-] Scan_MV_ID_user.IN.float.float2	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float3	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float4	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float5	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float6	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float7	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float8	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float9	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.float.float10	0.0	Float	REAL	user's device tags
[-] Scan_MV_ID_user.IN.string	{ ... }	[	Scan_...	user's device tags
[-] Scan_MV_ID_user.IN.string.string1	' '	[	Scan_...	Decode text

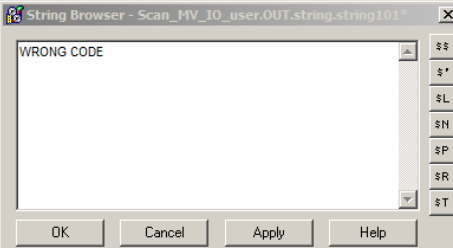
Dies entspricht dem Inspektionsergebnis der CS50-Software.



## Kamera erneut parametrieren

Die Tools **Measure** (Messen) und **Count Blob** (Objekt-Zähler) können von der SPS so parametrieren werden, dass sie immer erfolgreiche Ergebnisse liefern. Dagegen kann das Tool **Decode** (Dekodieren) so parametrieren werden, dass das Ergebnis immer ein Fehlschlag ist, entweder aufgrund einer nicht erfolgten Dekodierung oder einer Nichtübereinstimmung der **abzugleichenden Zeichenfolgen**. Scrollen Sie im Tag-Fenster bis zu den Werten **OUT.long**, **float** und **string** und ändern Sie sie dann wie unten dargestellt.

Name	Value	Style	Data Typ	Description	Constant
Scan_MV_IO_user.OUT.VID	{...}	[	Scan...	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.bool	{...}	[	Scan...	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.int	{...}	[	Scan...	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long	{...}	[	Scan...	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long101	4	Decimal	DINT	Blob count must be equal to or higher than this to pass	
Scan_MV_IO_user.OUT.long.long102	6	Decimal	DINT	Blob count must be equal to or lower than this to pass	
Scan_MV_IO_user.OUT.long.long103	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long104	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long105	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long106	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long107	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long108	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long109	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.long.long110	0	Decimal	DINT	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.float	{...}	[	Scan...	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.float.float101	50.0	Float	REAL	Measure value must be higher than this to pass	
Scan_MV_IO_user.OUT.float.float102	200.0	Float	REAL	Measure value must be lower than this to pass	
Scan_MV_IO_user.OUT.float.float103	0.0	Float	REAL	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.float.float104	0.0	Float	REAL	user's device tags when MV_demo_mode is 1 or 2.	
Scan_MV_IO_user.OUT.float.float105	0.0	Float	REAL		
Scan_MV_IO_user.OUT.float.float106	0.0	Float	REAL		
Scan_MV_IO_user.OUT.float.float107	0.0	Float	REAL		
Scan_MV_IO_user.OUT.float.float108	0.0	Float	REAL		
Scan_MV_IO_user.OUT.float.float109	0.0	Float	REAL		
Scan_MV_IO_user.OUT.float.float110	0.0	Float	REAL		
Scan_MV_IO_user.OUT.string	{...}	[	Scan...		
Scan_MV_IO_user.OUT.string.string101	*LABEL CHECK*	[	Scan...		
Scan_MV_IO_user.OUT.string.string102	*\$t\$00\$00\$...	[	Scan...		
Scan_MV_IO_user.OUT.string.string103	**	[	Scan...		
Scan_MV_IO_user.OUT.string.string104	**	[	Scan...		
Scan_MV_matchcode	*LABEL CHECK*	[	Scan...		
Scan_MV_ons_internal	-2080374528	Decimal	DINT	0 Error(s)	10 INS 10 of 92



## Kamera erneut auslösen

Wird die Kamera zweimal ausgelöst, bleiben die Statusergebnisse für beide Auslöseereignisse unverändert:

### bool2 (Dekodier- und Matchcode-Status) = 0

Ursache: Als Dekodier- und Matchcode-Status ergibt sich immer ein Fehlschlag, da entweder der Matchcode in einen falschen Code geändert wurde oder keine Dekodierung erfolgt ist.

### bool1 (Messstatus) und bool3 (Objekt-Zähler-Status) = 1

Ursache: Die inspizierten Werte liegen nun innerhalb des Toleranzbereichs.

### InspStat = 0

Ursache: Das Dekodiertool meldet einen Fehlschlag, dementsprechend ist das gesamte Inspektionsergebnis fehlgeschlagen.

### SPS-Tags:

Name	Value	Style	Data Type	Description
Scan_MV_IO_user.IN	{ ... }		Scan_...	user's device tags
Scan_MV_IO_user.IN.Status	{ ... }		Scan_...	user's device tags
Scan_MV_IO_user.IN.Status.Online	1	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.ExpBusy	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.AcqBusy	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.TriggerReady	1	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.Error	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.ResetCountAck	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved6	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.ExeCmdAck	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.TriggerAck	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.InspBusy	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.InspStat	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.DataValid	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved12	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved13	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved14	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.reserved15	0	Decimal	BOOL	user's device tags
Scan_MV_IO_user.IN.Status.Echo	0	Decimal	INT	user's device tags
Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000	Hex	DINT	user's device tags
Scan_MV_IO_user.IN.Status.CmdRet	0	Decimal	DINT	user's device tags
Scan_MV_IO_user.IN.Status.reserved96_103	0	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.Status.reserved104_111	0	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.Status.State	1	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.Status.reserved120_127	0	Decimal	SINT	user's device tags
Scan_MV_IO_user.IN.VID	{ ... }		Scan_...	user's device tags
Scan_MV_IO_user.IN.bool	{ ... }		Scan_...	user's device tags
Scan_MV_IO_user.IN.bool.bool1	1	Decimal	BOOL	measure status
Scan_MV_IO_user.IN.bool.bool2	0	Decimal	BOOL	decode+matchcode
Scan_MV_IO_user.IN.bool.bool3	1	Decimal	BOOL	blob count status

Damit ist die EtherNet/IP-Demo abgeschlossen.





# EtherNet/IP-SPS-Democode

Dieses Kapitel beschreibt die Anwendung des di-soric-SPS-Democodes in einem Vision-Job mit Testobjekt.

Die EtherNet/IP-Demo-Dateien befinden sich im Installationsordner der CS50-Software: **C:\di-soric\Vscape\Tutorials and Samples\CS50\EIP demo**. Öffnen Sie die Datei **EIP\_demo.avp** in der CS50-Software und laden Sie sie auf die Kamera herunter.

Importieren Sie während der SPS-Integration die Datei „32-000003-2.L5X“, um die Demo-Tags und den Kontaktplan für die Kamera zu erstellen.

## Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für EtherNet/IP aktiviert werden. Wie man Kommunikationsprotokolle für den CS50-Sensor aktiviert oder ändert, erfahren Sie in **Kapitel 1 [Industrieprotokolle aktivieren](#)**.
- Bei CS50-Software-Jobs wird die EtherNet/IP-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert. Weitere Informationen dazu, wie man Job-Parameter und Ausgänge mit di-soric Link-Tags verknüpft, finden Sie in der *CS50-Software-Hilfe* im Abschnitt **di-soric Link**.

## Glossar

---

Folgende Begriffe werden in der Beschreibung des SPS-Demoprogramms von di-soric verwendet.

### Kamera

Die in dieser Anwendung eingesetzte di-soric-Kamera mit EtherNet/IP-Kommunikationsschnittstelle.

### Nutzeranwendung

Der vom Endnutzer oder Systemintegrator geschriebene SPS-Programmcode.

### Democode

Der von di-soric bereitgestellte SPS-Programmcode, der in den Kontaktplan-Bereich der SPS importiert werden kann. Er umfasst den größten Teil der Steuerungs- und Statusverwaltung des Geräts.

Der Democode erwartet, dass in der Kamera ein Demo-Vision-Job geladen wird, funktioniert aber auch, wenn kein solcher Job geladen ist.

### Aktivieren / Auf „1“ setzen

Ein einzelnes Steuerbit oder ein anderes Boolesches Bit mit dem Wert „1“ beschreiben.

### Aktiv

Ein Steuer-, Status-, Boolescher oder SPS-Programm-„Kontakt“ ist aktiv, wenn er den Wert „1“ hat.

### Leer

Ein Steuer-, Status-, Boolescher oder SPS-Programm-„Kontakt“ ist leer, wenn er den Wert „0“ hat.

### Einmaliger Vorgang

Ein SPS-Tag-Schreibvorgang, der nur ein einziges Mal durchgeführt wird, im Regelfall in Reaktion auf ein Ereignis. Nach einem einmaligen Vorgang beschreibt das SPS-Programm denselben Tag kein weiteres Mal, solange kein neues Ereignis auftritt.

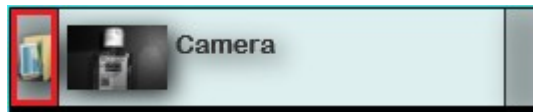
---

## Einrichtung der Demo

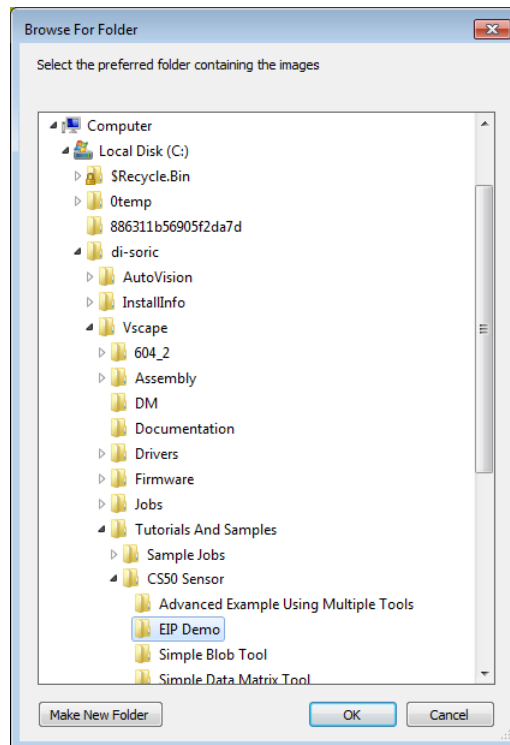
### Einrichtung des Vision-Jobs

Die EtherNet/IP-Demo-Dateien befinden sich im Installationsordner der CS50-Software. Der Standardpfad dieses Ordners ist **C:\di-soric\Vscape\Tutorials and Samples\CS50\EIP demo**.

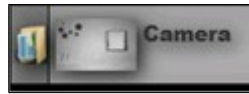
1. Öffnen Sie die Datei **EIP\_demo.avp** mit der CS50-Software.
2. Um vordefinierte Bilder zu verwenden, klicken Sie auf das unten dargestellte Symbol.



3. Suchen Sie den Ordner **EIP demo** (EIP-Demo), wählen Sie ihn aus und klicken Sie auf **OK**.



Nach der Auswahl des Ordners **EIP demo** werden die Bilder heruntergeladen und das Kamerasymbol wird durch einen Ordner



ersetzt.

4. Bevor der Job an die Kamera gesendet wird, können Sie im Modus **Edit** (Bearbeiten) mithilfe der Option **Try Out** (Ausprobieren) eine



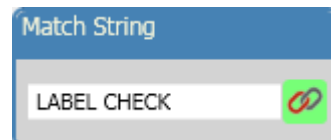
Vorschau aufrufen.

Zuvor müssen jedoch die Parameter **Measure** (Messen), **Decode** (Dekodieren) und **Count Tool** (Zähler) konfiguriert werden. Die Tool-Parameter werden nach dem Download des Jobs von der SPS übermittelt.

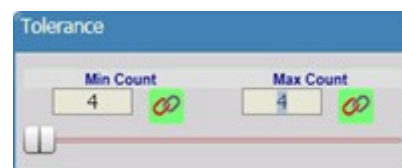
#### Messtoleranz:



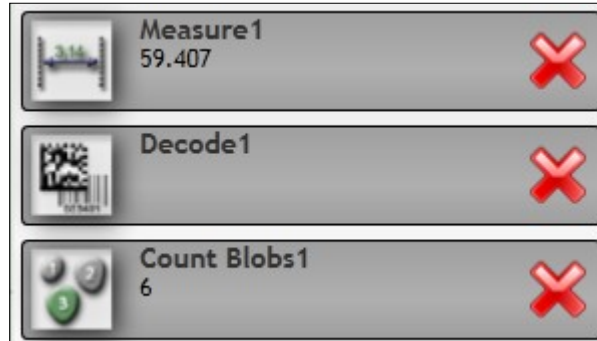
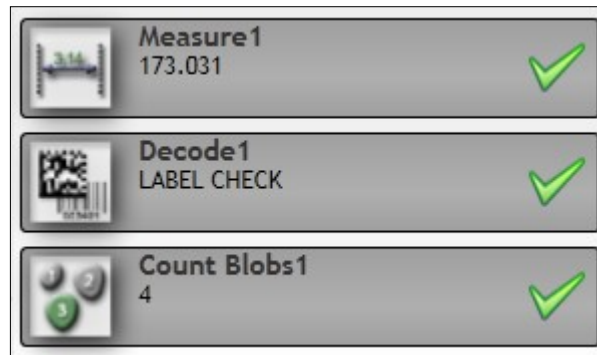
#### Dekodierung der Match-Zeichenfolge:



#### Zählertoleranz:



Wenn Sie die Tool-Parameter wie oben dargestellt konfiguriert und die Option „Try Out“ (Ausprobieren) ausgewählt haben, sehen Sie folgende Ergebnisse:

**Fehlgeschlagen:****Erfolgreich:**

5. Laden Sie den Job auf die Kamera herunter.
6. Fügen Sie die Kamera und den Democode der SPS-Umgebung hinzu (siehe nächstes Kapitel).

## Einrichtung des SPS Democodes

Importieren Sie während der SPS-Integration die Datei **32-000003-2.L5X** aus dem Ordner **EIP demo** (EIP-Demo). Dadurch werden die Demo-Tags und der Kontaktplan für die Kamera angelegt (siehe die Kapitel zur Einrichtung der Allen-Bradley-SPS).

## Beschreibung der SPS-Tags

---

### Scan\_MV\_demo\_mode

#### Zweck

Dieses Tag dient lediglich zu Demonstrationszwecken. Es modifiziert die Funktionsweise des Democodes und gibt dem Erstenutzer die Möglichkeit, das Gerät direkt zu steuern, ohne Unterstützung durch den Democode. Außerdem kann der Democode mithilfe dieses Tags die Steuer- und Statussignale uneingeschränkt verwalten.

Das Demo-Modus-Tag kann drei verschiedene Werte annehmen und damit den Democode in drei verschiedene Modi versetzen:

- Nur E/A-Datenaustausch
- Automatische Gerätesteuerung und Verwaltung von Status und Demodaten
- Automatische Auslösung des Geräts nach 1 Sekunde Leerlaufzeit

#### Nutzeranwendungsmethode

In der Nutzeranwendung kann der Demo-Modus einen von drei Werten annehmen, die den Betriebsmodus des Democodes definieren.

#### 0 = Nur E/A-Datenaustausch

In diesem Modus hat der Nutzer direkten Zugriff auf die **Scan\_MV\_IO\_internal**-Tags. Der Democode tauscht dann lediglich Daten mit der Kamera aus, steuert aber das Gerät nicht und reagiert auch nicht auf Geräteereignisse.

**1 = Automatische Gerätesteuerung und Reaktion auf Geräteereignisse.** Dieser Modus ist für den Democode standardmäßig voreingestellt. In diesem Modus kann die Nutzeranwendung auf das **Scan\_MV\_IO\_user**-Tag zugreifen und so die Kamera steuern und überwachen. Die Nutzeranwendung benötigt dazu keinen Zugriff auf die **Scan\_MV\_IO\_internal**-Tags.

Der Nutzer aktiviert in diesem Modus die Steuerbefehle in **Scan\_MV\_IO\_user.OUT.Control** (Trigger, ResetCount, GoOnline, GoOffline, ResetError, ExeCmd); der Rest wird vom Democode automatisch verwaltet.

#### 2 = Automatische Auslösung

In diesem Modus werden die **Steuer-** und **Statussignale** vollständig vom Democode verwaltet. Dies entspricht der Einstellung des Modus auf den Wert **1**. Außerdem wird der Trigger der Kamera nach 1 Sekunde Leerlaufzeit automatisch aktiviert. Der Timer für den Trigger wird in **Scan\_MV\_trigger\_delay\_timer** definiert.

### Verwendung des Democodes

Der Ausführungsumfang des Democodes variiert je nach dem gewählten Modus.

- Im **Modus 0** werden nur die Zeilen für den E/A-Datenaustausch ausgeführt. Alle anderen werden ignoriert.
- Im **Modus 1** werden nur die Zeilen für die automatische Auslösung ignoriert. Dieser Modus ist für den Democode standardmäßig voreingestellt.
- Im **Modus 2** werden alle Zeilen ausgeführt.

## Scan\_MV\_IO\_user

### Zweck

E/A-Daten für die Kamera, auf die der Nutzer zugreifen kann. Die Nutzeranwendung liest und schreibt diese E/A-Tags, während der Democode die eigentliche On-the-Wire-Steuerung der Kamera übernimmt.

### Nutzeranwendungsmethode

Ein Steuerbefehl wird aktiviert, indem sein Wert auf **1** gesetzt wird.

Wenn der Steuerwert leer ist (d. h. der Democode den Steuerbefehl auf einen Bit-Wert/Booleschen Wert von **0** gesetzt hat), erkennt die Nutzeranwendung, dass der Steuervorgang abgeschlossen ist. Aktivieren Sie einen Steuerbefehl nur dann, wenn dessen Wert leer ist.

Die Nutzeranwendung sollte die Steuerbefehle mittels einmaliger Schreibvorgänge aktivieren, aber einen Steuerbefehl nicht dauerhaft aktiv halten. Verharrt ein Steuerbefehl dauerhaft im aktiven Zustand, kann der Democode der Nutzeranwendung nicht durch Leeren des Steuerwertes mitteilen, dass der Steuervorgang abgeschlossen ist.

Ist ein Steuerwert leer (**0**), ist die Kamera für eine erneute Aktivierung des Steuerbefehls bereit. Nähere Informationen dazu finden Sie in den folgenden Abschnitten **Besondere Hinweise zu Steuerbefehlen** und **Besondere Hinweise zu Status**.

### Verwendung des Democodes

Der Democode wartet, bis die Nutzeranwendung einen Steuerbefehl aktiviert. Sobald ein Steuerbefehl aktiviert wurde, übernimmt der Democode das gesamte Handshaking und bestätigt, dass dieser Steuerbefehl nun von der Kamera ausgeführt wird. Wenn der Vorgang abgeschlossen ist, setzt der Democode den Wert des Steuerbefehls wieder auf **0**.

## Besondere Hinweise zu Steuerbefehlen

### GoOnline und GoOffline

Die beiden Steuerbefehle GoOnline und GoOffline regeln, ob die Kamera online oder offline ist. Es kann jeweils nur einer dieser beiden Befehle aktiv geschaltet (d. h. von 0 auf 1 gesetzt) werden oder aktiv sein.

### ResetCount

Nachdem die Nutzeranwendung den Steuerbefehl ResetCount aktiviert hat, leert der Democode den entsprechenden Wert wieder, sobald der Vorgang abgeschlossen ist. Der nächste Inspektionsausgangswert ist 1 (wie angezeigt wird, wenn die CS50-Software im Betriebsmodus mit der Kamera verbunden ist).

### Trigger

Lösen Sie die Kamera nur dann aus, wenn der Status TriggerReady aktiv ist. Wird der Trigger aktiviert, obwohl dieser Status inaktiv ist, erhöht der Democode den Zähler Scan\_MV\_trigger\_err\_count und leert sofort den Wert des Steuerbefehls des Triggers, ohne die Kamera tatsächlich auszulösen. Nachdem die Nutzeranwendung den Trigger aktiviert hat, wird der Wert des Triggers vom Democode geleert, sofern die Kamera das Triggerereignis akzeptiert hat.

Lösen Sie die Kamera erst dann wieder aus, wenn der Befehl **DataValid** im Register **Status** aktiv ist, alle **Inspektionsdaten** verarbeitet wurden und der Wert für „DataValid“ über den Steuerbefehl **ResetDataValid Control** geleert wurde.

### ResetDataValid

Wenn die Nutzeranwendung erkennt, dass der Befehl „DataValid“ aktiv geschaltet wurde, müsste sie die Inspektionsdaten verarbeiten und dann durch Aktivieren des Befehls ResetDataValid den Wert für „DataValid“ leeren.

Weitere Informationen dazu finden Sie im Abschnitt [DataValid](#).



**ResetError**

Um den Status „Error“ (Fehler) zu leeren,  
aktivieren Sie den Befehl **ResetError**.

**ExeCmd, CmdCode, CmdArg**

Mit diesen Steuerbefehlen können Jobs geändert und der aktive Job-Slot abgefragt werden. Der Democode enthält Tags mit vordefinierten Werten für **CmdCode** und **CmdCodeRslt**:

Tag	Meaning
Scan_MV_const_CmdCode_GetActiveJb	Query the active job slot number (returned in CmdRet)
Scan_MV_const_CmdCode_JbChg	Go Offline, Load job specified by LSB
Scan_MV_const_CmdCode_JbChg_MkBt	Go Offline, Load Job specified by LSB, Make it the boot job
Scan_MV_const_CmdCode_JbChg_MkBt_Online	Go Offline, Load Job specified by LSB, Make it the boot job, Go Online
Scan_MV_const_CmdCode_JbChg_Online	Go Offline, Load Job specified by LSB, Go Online
Scan_MV_const_CmdCodeRslt_Fail_No_Job	Job Change failed: No job in slot
Scan_MV_const_CmdCodeRslt_Fail_UI	PC UI is controlling the camera,
Scan_MV_const_CmdCodeRslt_Fail_Unk_Cmd	Job Change failed: Unknown CmdCode
Scan_MV_const_CmdCodeRslt_Success	Completed operation OK.

Die Steuerbefehle **ExeCmd**, **CmdCode** und **CmdArg** werden in Kombination mit diesen Statussignalen eingesetzt:

Control signal	Status signal
<b>ExeCmd</b>	<b>ExeCmdAck</b>
<b>CmdCode</b>	<b>CmdCodeRslt</b>
<b>CmdArg</b>	<b>CmdRet</b>

Der Democode erfasst das Endergebnis des Befehlsvorgangs, indem er **CmdCode**, **CmdArg**, **CmdCodeRslt** und **CmdRet** in folgende Tags kopiert:

Source Control/Status tag	Final result tag
Scan_MV_IO_user.OUT.Control.ExeCmd	Scan_MV_CmdCode_last
Scan_MV_IO_user.OUT.Control.CmdArg	Scan_MV_CmdArg_last
Scan_MV_IO_internal.IN.Status.CmdCodeRslt	Scan_MV_CmdCodeRslt_last
Scan_MV_IO_internal.IN.Status.CmdRet	Scan_MV_CmdRet_last

Der Democode automatisiert den Befehlsvorgang, wenn **Scan\_MV\_demo\_mode** den (beim Programmstart standardmäßig voreingestellten) Wert **1** hat. Der Democode funktioniert hier ähnlich wie bei den Steuerbefehlen „Trigger“ und „DataValid“. Der SPS-Integrator kann den Befehlsvorgang initiieren, indem er auf die **Scan\_MV\_IO\_user**-Tags des Democodes für die Steuer- und Statussignale zugreift. Solange ein Befehl aktiv ausgeführt wird, werden alle Steuersignale mit Ausnahme des Echos vom Democode deaktiviert. Bis zum Abschluss des Befehlsvorgangs können keine weiteren Steuerbefehle aktiviert werden. Um während eines Befehlsvorgangs zu überprüfen, ob die Kamera noch funktioniert, kann der Wert **Control.Echo** erhöht werden; **Status.Echo** wird dann entsprechend aktualisiert.

Falls Befehle vom Democode automatisch ausgeführt werden, ist der SPS-Integrator für folgende Schritte zuständig:

1. Deaktivieren Sie alle Steuerbefehle und leeren Sie die Statussignale für „DataValid“ und „Error“. Dies ist die beste Methode, um sicherzustellen, dass die SPS vom Auslösen und Verarbeiten der Inspektion zur Befehlsausführung übergeht.
2. Falls ein Änderungsbefehl ausgeführt werden soll, füllen Sie die Ausgangs-Tags aus, die zur Konfiguration des neuen Jobs nötig sind (**bool, int, long, float, string**).
3. Geben Sie die erforderlichen Werte für **CmdCode** (siehe **Scan\_MV\_const\_CmdCode\_xxxx**-Tags) und **CmdArg** ein und aktivieren Sie dann **ExeCmd**.
4. Warten Sie, bis „ExeCmd“ inaktiv ist (bei jedem typischen Vorgang im Demo-Modus **1**). Beachten Sie, dass Job-Änderungen bis zu einer Minute dauern können. Solange ein Befehl zur Job-Änderung ausgeführt wird, hat das Tag **Status.State** den Wert **2**.
5. Sobald „ExeCmd“ wieder inaktiv ist, überprüfen Sie Folgendes:  
**Scan\_MV\_CmdCodeRslt\_last** muss den Wert **0** haben (Erfolg).  
**Scan\_MV\_CmdRest\_last** muss ggf. die vom Befehl zurückgegebenen Daten enthalten.  
**Status.State** muss in den Status **0** (Offline) oder **1** (Online) gewechselt sein.  
**ExeCmdAck** muss inaktiv (**0**) sein.  
**Status.Error** muss inaktiv (**0**) sein.
6. Schalten Sie die Kamera wieder online (falls nötig) und fahren Sie mit dem normalen Betrieb fort.

## Besondere Hinweise zu Status

### Online

Die Kamera kann nur ausgelöst werden oder Inspektionsdaten generieren, Status **Online** aktiv ist. Näheres dazu können Sie im Abschnitt zum Steuerbefehl **GoOnline** nachlesen. **TriggerReady** Versuchen Sie niemals, die Kamera auszulösen, wenn der Status

**TriggerReady** inaktiv ist. Weitere Informationen dazu finden Sie im Abschnitt **Trigger**.

### TriggerAck und ResetCountAck

Mit diesem Status beendet der Democode die entsprechenden Vorgänge.

### DataValid

Wenn der Status **DataValid** aktiv geschaltet wird, müsste die Nutzeranwendung die Inspektionsdaten verarbeiten und anschließend „DataValid“ mithilfe des Steuerbefehls **ResetDataValid** leeren. In den Modi **1** und **2** wird dies als Demonstration für die Nutzeranwendung vom Democode übernommen.

Wenn „DataValid“ für die Kamera aktiviert wird, aber ein vorheriges DataValid-Ereignis von der Nutzeranwendung noch nicht geleert wurde, überschreibt der Democode **Scan\_MV\_IO\_user** nicht mit neuen Inspektionsdaten. Stattdessen wird dann der Zähler **Scan\_MV\_dv\_err\_count** erhöht. Die neuen Inspektionsdaten bleiben dann in den **Scan\_MV\_internal**-Tags und sind damit praktisch verloren.

## Scan\_MV\_trigger\_count

Dieser Zählerstand wird vom Democode erhöht, wenn über die EtherNet/IP-Schnittstelle ein neues Triggersignal an die Kamera gesendet (d. h. der Steuerwert „Trigger“ aktiviert) wird.

## Scan\_MV\_trigger\_err\_count

Dieser Zählerstand wird vom Democode erhöht, wenn die Nutzeranwendung versucht, die Kamera auszulösen, obwohl der Status „TriggerReady“ inaktiv ist.

### Scan\_MV\_dv\_err\_count

Dieser Zählerstand wird vom Democode erhöht, wenn von der Kamera neue Inspektionsdaten eingehen, aber die Nutzeranwendung den vorherigen DataValid-Wert noch nicht geleert hat.

### Scan\_MV\_status\_err\_count

Dieser Zählerstand wird vom Democode erhöht, wenn der Status „Error“ (Fehler) aktiviert wird.

### Scan\_MV\_demo\_blob, Scan\_MV\_demo\_decode, Scan\_MV\_demo\_InspStat, Scan\_MV\_demo\_measure

#### Zweck

In diesen Tags werden Zählerstände sowie Mindest- und Höchstwerte verschiedener Datenelemente von „EIP IN“ erfasst. Der Democode erwartet, dass in die Kamera ein Demo-Vision-Job geladen wird und dass sich im Sichtbereich der Kamera ein Demo-Objekt befindet. Er funktioniert aber auch, wenn kein solcher Job geladen ist.

Die Datensätze sind dann allerdings nicht gültig.

Der Demo-Vision-Job umfasst folgende Datenelemente, die mit bestimmten Job-Tools verknüpft sind:

#### IN

**Bool1** = Messstatus (Erfolg/Fehlschlag)

**Bool2** = Dekodier- und Matchcode-Status (Erfolg/Fehlschlag)

**Bool3** = Objekt-Zähler-Status (Erfolg/Fehlschlag)

**Long1** = Objekt-Zähler

**Float1** =

Messwert**String1** =

Dekodierter Text **OUT**

**Long101** = Untere Toleranzgrenze für den Objekt-Zähler

**Long102** = Obere Toleranzgrenze für den Objekt-Zähler

**Float101** = Untere Toleranzgrenze

für den Messwert**Float102** = Obere

Toleranzgrenze für den

Messwert**String101** = Matchcode

In jedem Tag-Satz werden pro Vision-Job-Tool-Ergebnis, das im Inspektionsbericht enthalten ist, folgende Daten erfasst:

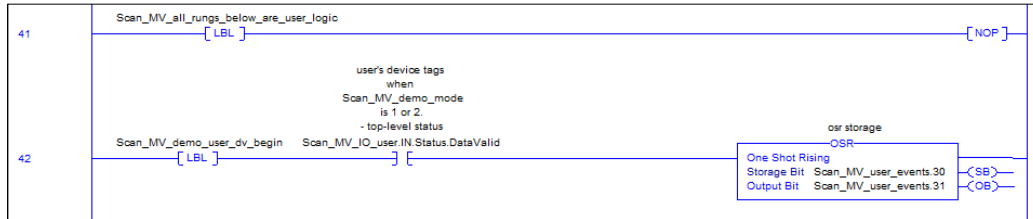
Tool Result	Record tag	EIP IN tag
Measurement Status	Scan_MV_demo_measure. bool = last status(pass/fail) pass_count = count of passes fail_count = count of fails	Scan_MV_user_IN.bool.bool1
Measurement	Scan_MV_demo_measure. float = last value float_max = max value recorded float_min = minimum value recorded	Scan_MV_user_IN.float.float1
Decode Text Status (matchcode)	Scan_MV_demo_decode bool = last status(pass/fail) pass_count = count of passes fail_count = count of fails	Scan_MV_user_IN.bool.bool2
Decode Text	Scan_MV_demo_decode string = text of the last barcode decode attempt (null if noread)	Scan_MV_user_IO.string.string1
Blob Status	Scan_MV_demo_blob bool = last status(pass/fail) pass_count = count of passes fail_count = count of fails	Scan_MV_user_IN.bool.bool3
Blob Count	Scan_MV_demo_blob long = last value long_max = max value recorded long_min = minimum value recorded	Scan_MV_user_IN.long.long1

### Nutzeranwendungsmethode

Die Nutzeranwendung kann die Verwendung dieser Tags durch den Democode erfassen. Die entsprechenden Daten stehen dann für die Entwicklung weiterer Anwendungsprogramme zur Verfügung.

Im laufenden Betrieb kann die Nutzeranwendung die Datenelemente von „OUT“ ändern und die Änderung des Toolstatus nach einem neuen Triggerereignis überwachen.

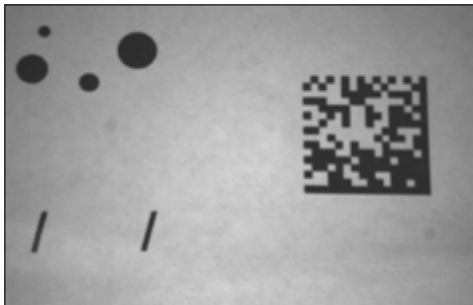
Im Einzelnen nimmt der SPS-Integrator Programmänderungen normalerweise beginnend bei folgenden Zeilen vor:

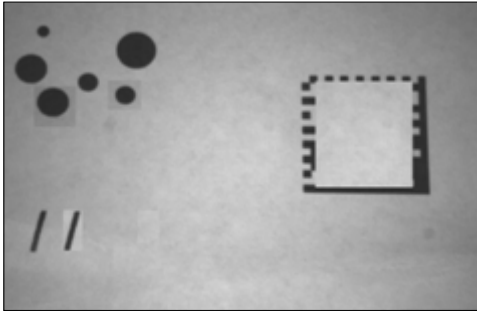


### Demo-Objekt

Für den Demo-Vision-Job werden vordefinierte Bilder verwendet. Die Kamera muss nicht auf ein bestimmtes Objekt ausgerichtet werden. Falls gewünscht, lässt sich der Job so abändern, dass der Bildsensor der Kamera aktiviert wird. In diesem Fall sollten die Demo-Objekte mit einer Breite von ca. **2,5 Zoll (63 mm)** und einer Höhe von ca. **1,6 Zoll (40 mm)** zentriert auf weißes Papier gedruckt werden. Das Papier muss größer als der Sichtbereich der Kamera sein. Rechts neben den Demo-Objekten sollte der DataMatrix-Code abgebildet sein:

### Bild für erfolgreiche Lesung



**Bild für fehlgeschlagene Lesung****Verwendung des Democodes**

Im Nutzeranwendungsbeispiel überwacht der Democode den Status „DataValid“. Wird der Status aktiviert, verarbeitet die Nutzeranwendung die E/A-Daten des Nutzers, aktualisiert die jeweiligen Demo-Datensätze mit den Ergebnissen und leert dann mit dem Befehl **ResetDataValid** den Status „DataValid“.

**Scan\_MV\_IO\_internal, Scan\_MV\_ons\_internal****Zweck**

Damit verwaltet der Democode die Kamera.

**Nutzeranwendungsmethode**

Keine. Die Nutzeranwendung darf nicht versuchen, diese Tags zu lesen oder zu überschreiben.

**Verwendung des Democodes**

Anhand dieser Tags abstrahiert der Democode die On-the-Wire-Steuerung der Kamera von der Nutzeranwendung.

## **Kamerabetrieb: Nutzung der EtherNet/IP-Demo bei laufender Kamera**

An dieser Stelle wird vorausgesetzt, dass Sie den Demo-Vision-Job auf die Kamera geladen haben und dass die SPS den EtherNet/IP-Democode verwendet und mit der Kamera Daten austauscht. Die SPS kann die Kamera nun über EtherNet/IP parametrieren, auslösen und überwachen.



# Omron-SPS für den EtherNet/IP-Betrieb einrichten

Dieses Kapitel enthält eine Anleitung für die Einrichtung einer Omron-SPS für den EtherNet/IP-Betrieb mithilfe eines CS50-Sensors und der Software CX-One.

## Hinweise:

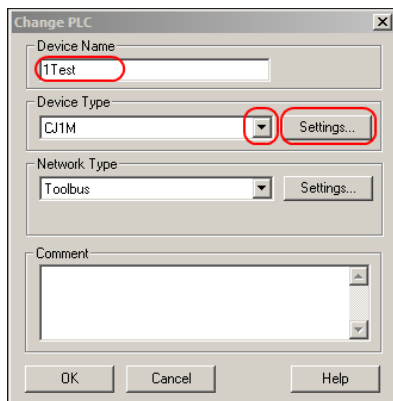
- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für EtherNet/IP aktiviert werden. Wie man Kommunikationsprotokolle für den CS50-Sensor aktiviert oder ändert, erfahren Sie in **Kapitel 1** [Industrieprotokolle aktivieren](#).
- Bei CS50-Software-Jobs wird die EtherNet/IP-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert. Weitere Informationen dazu, wie man Job-Parameter und Ausgänge mit di-soric Link-Tags verknüpft, finden Sie in der *CS50-Software-Hilfe* im Abschnitt **di-soric Link**.

## Omron-SPS einrichten

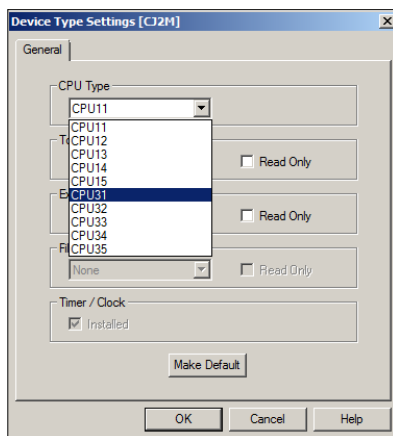
Die Anleitung in diesem Kapitel wurde auf der Basis der folgenden Omron-Software und -Hardware erstellt:

- CX-Programmer Version 9.43
- Network Configurator Version 3.55
- SPS CJ2M CPU31

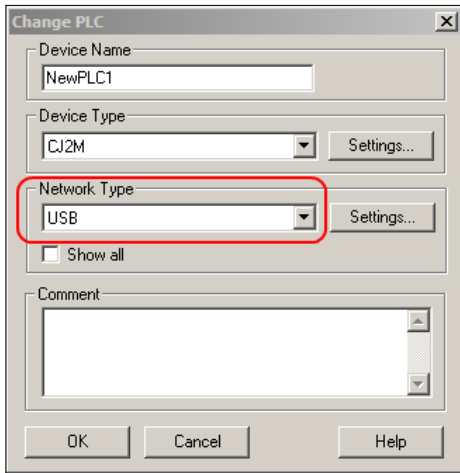
Starten Sie die Anwendung CX-Programmer und wählen Sie **Datei > Neu** aus. Geben Sie unter **Gerätename** den gewünschten Namen ein. Wählen Sie den **Gerätetyp** aus und klicken Sie dann rechts neben dem Menü „Gerätetyp“ auf **Einstellungen**.



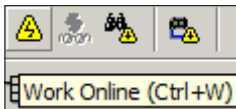
Daraufhin öffnet sich das Dialogfenster **Gerätetyp-Einstellungen**. Wählen Sie darin den richtigen **CPU-Typ** aus und klicken Sie auf **OK**.



Wählen Sie aus dem Menü **Netzwerktyp** die Option **USB** aus und klicken Sie auf **OK**.

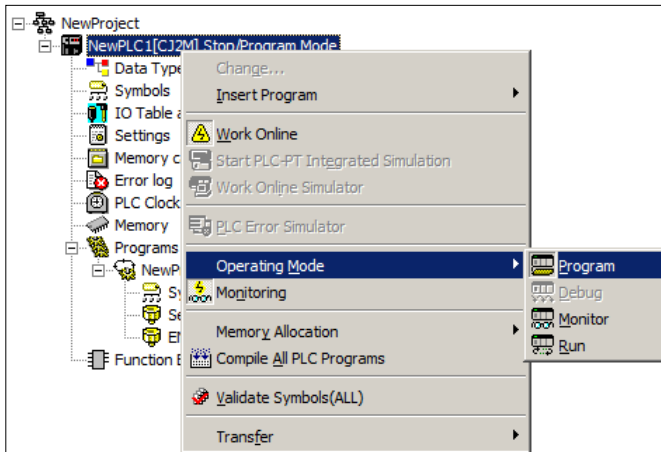


Stellen Sie eine USB-Verbindung zur SPS her. Wählen Sie den Menüeintrag **SPS > Online arbeiten** aus oder klicken Sie in der Symbolleiste auf das „Online“-Symbol. Klicken Sie in der entsprechenden Aufforderung auf **Ja**, um den Verbindungsvorgang abzuschließen.

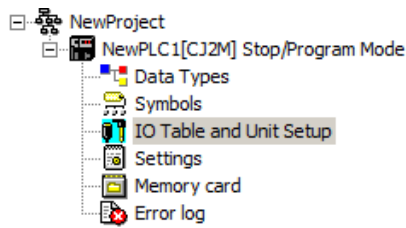


Sobald die Verbindung hergestellt wurde, ändert sich die Hintergrundfarbe des rechten Bereichs in Grau und die Onlinesymbole in der Leiste bleiben ausgewählt.

Für die folgenden Schritte muss sich die SPS im **Programm-Modus** befinden. Klicken Sie dazu in der Baumansicht im linken Bereich mit der rechten Maustaste auf den SPS-Knoten und wählen Sie **Betriebsart > Programm** aus.

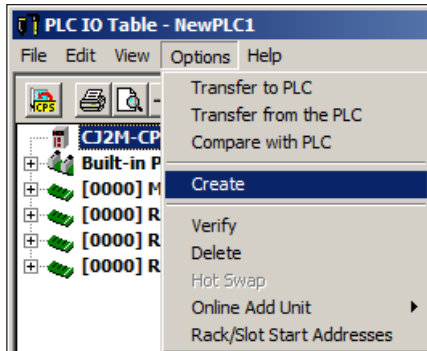


Um die E/A-Tabelle zu registrieren, doppelklicken Sie auf **E/A-Tabelle und Baugruppen-**

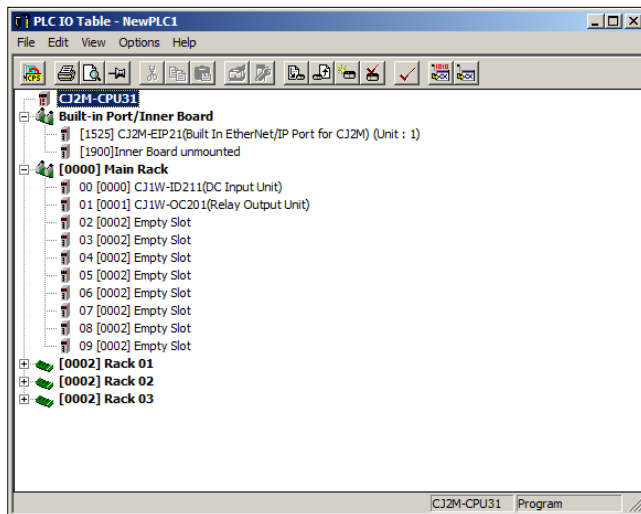


**Setup.**

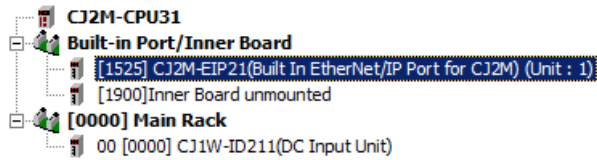
Wählen Sie **Optionen > Erstellen** aus.



Klicken Sie auf **Ja**, wenn Sie beim Erstellen der E/A-Tabelle dazu aufgefordert werden. Klicken Sie in der Aufforderung zum Initialisieren der CPU-Bus-Einstellungen erneut auf **Ja**. Klicken Sie in der Aufforderung zur Datenübertragung auf **Übertragen**. Klicken Sie im Fenster mit den Ergebnissen auf **OK**. Die E/A-Tabelle wird nun mit den aktuellen SPS-Hardwareeinstellungen aktualisiert.

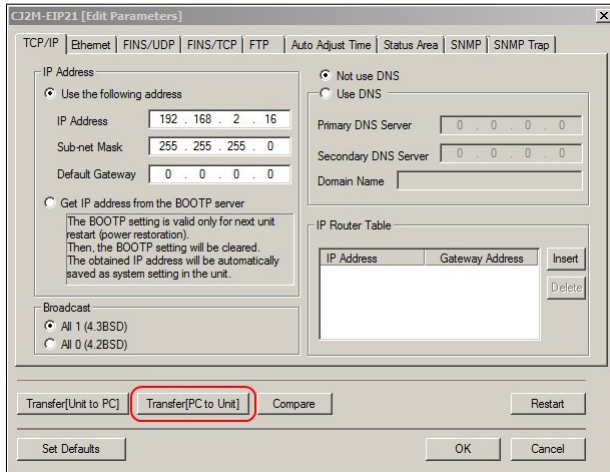


Um die EtherNet/IP-Elemente und deren Mapping zu bearbeiten, doppelklicken Sie auf den



EtherNet/IP-Knoten.

Geben Sie die gewünschten IP-Einstellungen ein und klicken Sie dann auf **Übertragung [P Baugruppe]**.



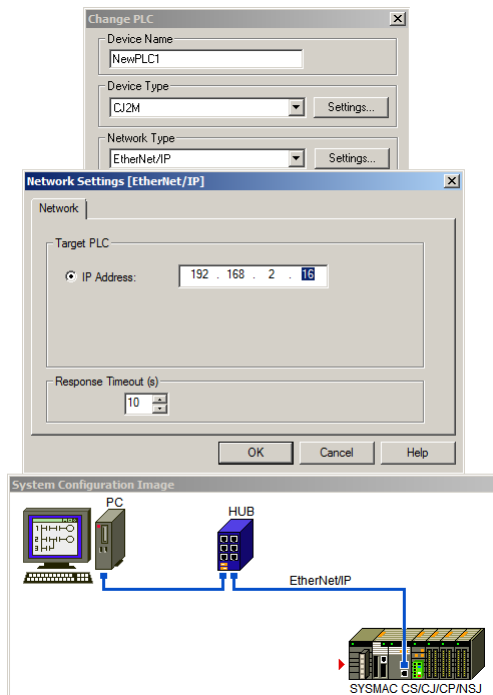
Wenn Sie zur Übertragung aufgefordert werden, klicken Sie auf **Ja**. Sobald die Übertragung abgeschlossen ist, klicken Sie auf **Schließen**. Wenn Sie zum Neustarten der Baugruppe aufgefordert werden, klicken Sie auf **Ja**. Klicken Sie nach dem Neustart in der Aufforderung auf **OK**. Schließen Sie das Dialogfenster zur E/A-Bearbeitung.

Schalten Sie die Baugruppe aus und stellen Sie die Drehschalter auf das letzte Oktett der oben genannten neuen IP-Adresse. Schalten Sie dann die Baugruppe wieder ein.



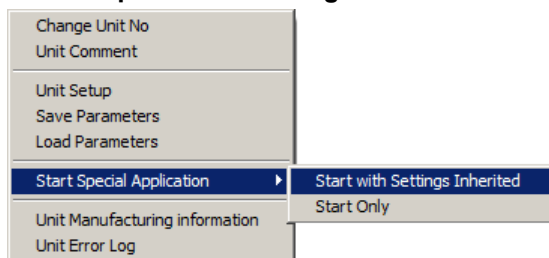
Stellen Sie CX-Programmer auf offline, indem Sie **SPS > Offline arbeiten** auswählen oder auf das Onlinesymbol in der Leiste klicken.

Doppelklicken Sie in CX-Programmer auf den SPS-Knoten. Wählen Sie im Menü „Netzwerktyp“ die Option **EtherNet/IP** aus. Klicken Sie rechts neben „Netzwerktyp“ auf **Einstellungen** und geben Sie die neue IP-Adresse der SPS ein. Klicken Sie auf die **OK-**



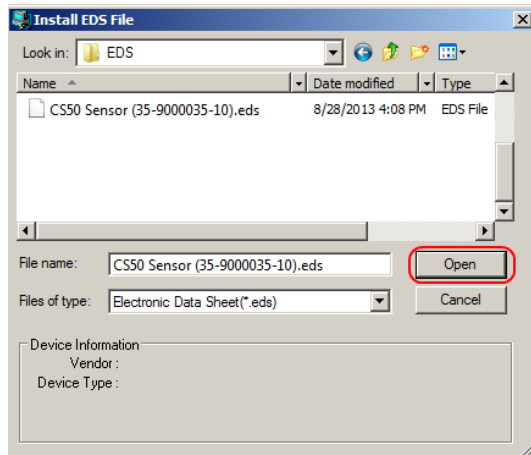
Schaltflächen, um die Dialogfenster zu schließen.

Stellen Sie CX-Programmer wieder auf online. Sie werden nun dazu aufgefordert, eine Verbindung über EtherNet/IP herzustellen. Klicken Sie auf **Ja**, um den Verbindungsvorgang abzuschließen. Doppelklicken Sie auf den Knoten **E/A-Tabelle und Baugruppen-Setup**. Maximieren Sie den Knoten **Integrierter Port/Innere Karte**. Klicken Sie auf **Spezialanwendung starten > Start mit übernommenen Einstellungen**.



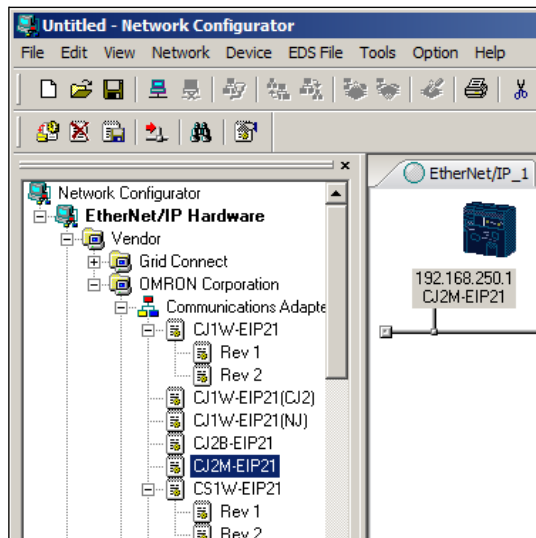
Wählen Sie **Netzwerk-Konfigurator** aus und klicken Sie auf **OK**. Wählen Sie den Port **TCP:2** aus und klicken Sie auf **OK**. Klicken Sie dann für die Verbindung **EtherNet/IP\_1** erneut auf **OK**.

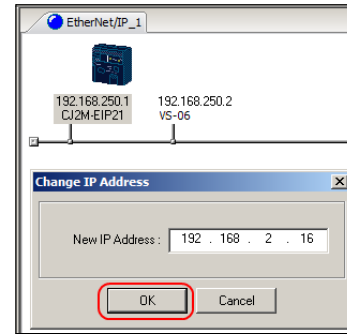
Um die EDS-Datei zu installieren, wählen Sie **EDS-Datei > Installieren** aus. Navigieren Sie zum EDS-Ordner **C:\di-soric\Vscape\Firmware\eds\CS50**. Wählen Sie die richtige Datei aus und klicken Sie dann auf **Öffnen**, um die Datei zu laden. Alle anderen EDS-



Dateien können von [www.di-soric.com](http://www.di-soric.com) heruntergeladen werden.

Maximieren Sie die Baumansicht links, um die **OMRON Corporation**-Dateien zu öffnen. Suchen Sie in diesem Fall nach der Datei **CJ2M-EIP21** und ziehen Sie sie mit der Maus in die Zeile im rechten Bereich.

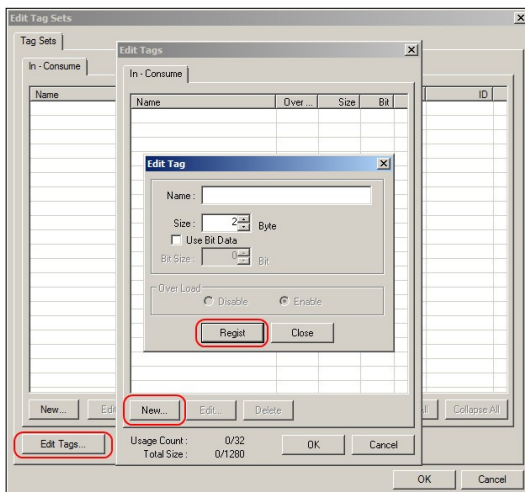




Maximieren Sie den Knoten **Hersteller-Sammlung** und ziehen Sie die Kamera, die mit der SPS verbunden ist, von dort ebenfalls in die Zeile im rechten Bereich.

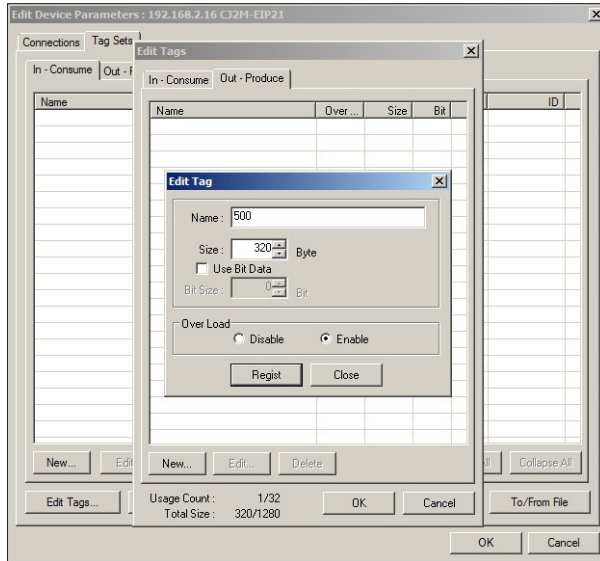
Klicken Sie mit der rechten Maustaste auf die neu hinzugefügten Symbole und wählen Sie **IP-Adresse ändern** aus. Geben Sie die IP-Adresse für die SPS und die damit verbundene Kamera bzw. das damit verbundene Lesegerät ein und klicken Sie auf **OK**.

Zum Bearbeiten der Geräteparameter doppelklicken Sie auf das SPS-Symbol. Nun können Sie die EtherNet/IP-Assembly-Daten mit dem internen Speicher der SPS verknüpfen und darin abbilden. Wählen Sie die Registerkarte **Tag-Sätze** aus. Wählen Sie oben die Registerkarte **Ein – Verbrauch** aus. Klicken Sie unten auf **Tags bearbeiten**. Klicken Sie dann auf **Neu**, um einen neuen Tag zu erstellen/bearbeiten. In diesem Beispiel haben wir den Tag für den peripheren Speicher, der mit den Eingangsdaten verknüpft ist, **300** genannt. Wählen Sie für die gesamte Eingangs-Assembly die Größe **320 Byte** aus. Klicken Sie auf **Registrieren** und anschließend auf **Schließen**, um fortzufahren.

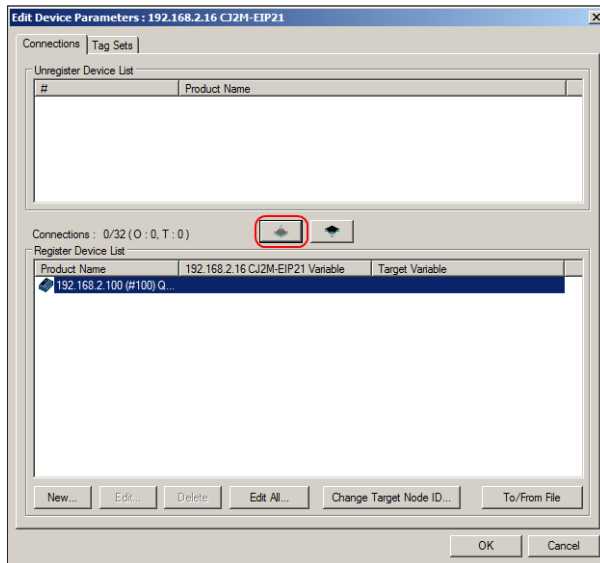




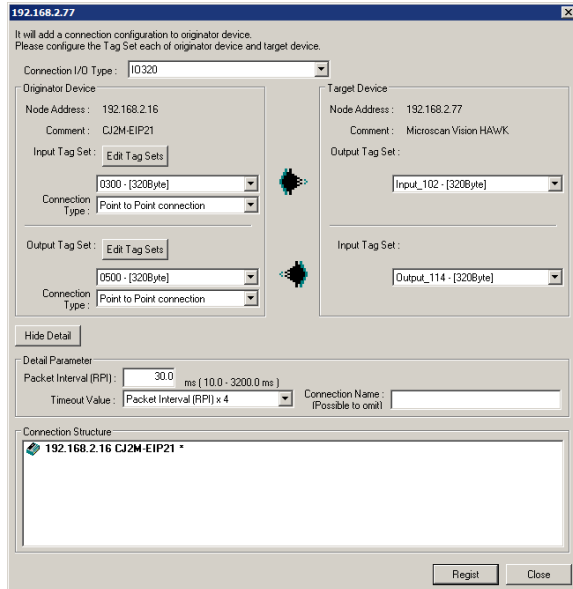
Wählen Sie die Registerkarte **Aus – Produktion** aus und klicken Sie dann auf **Neu**. Für die Ausgangs-Assembly müssen Sie die Daten mit **320 Byte** im peripheren Speicher unter der Adresse **500** abbilden. Klicken Sie auf **Registrieren** und anschließend auf **Schließen**. Klicken Sie im Dialogfenster **Tags bearbeiten** auf **OK**. Wenn Sie zum Registrieren neuer Tags aufgefordert werden, klicken Sie auf **Ja**.



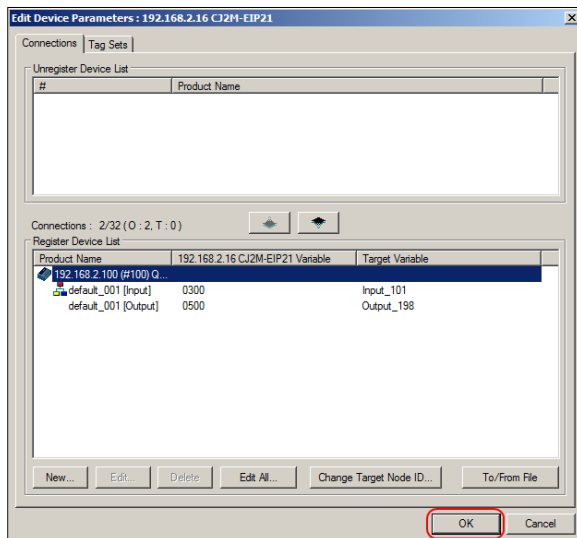
Klicken Sie im Dialogfenster **Geräteparameter bearbeiten** auf die Registerkarte **Verbindungen**. Um das Gerät zu registrieren, klicken Sie in der Mitte auf die Download-Schaltfläche.



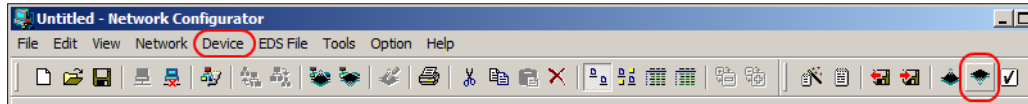
Doppelklicken Sie im unteren Bereich in der Liste registrierter Geräte auf die SPS. Dadurch wird das Dialogfenster zum Verknüpfen geöffnet. Falls mehrere Verbindungstypen vorhanden sind, können diese aus dem Menü **Verbindungs-E/A-Typ** ausgewählt werden. Wählen Sie im Abschnitt **Quellgerät** den **Eingangst-Tag** und dann den **Ausgangst-Tag** aus. Passen Sie ggf. den RPI-Wert an. Wenn Sie fertig sind, klicken Sie auf **Registrieren** und anschließend auf **Schließen**.



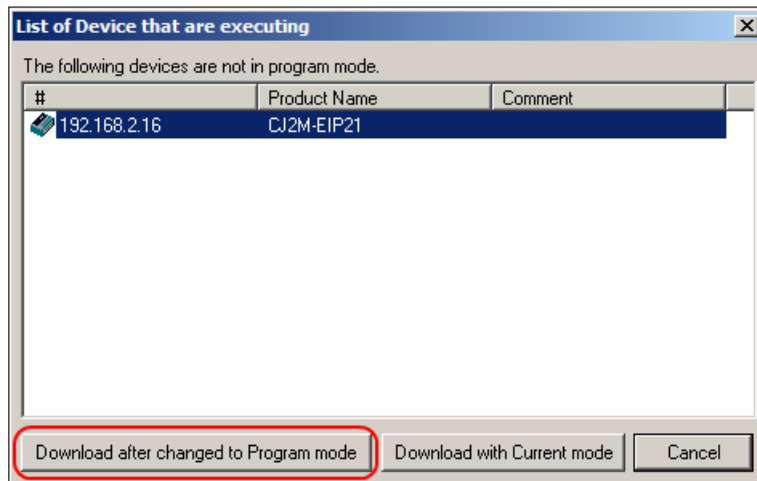
Das registrierte Gerät zeigt nun die verknüpften Tags an. Klicken Sie auf **OK**, um fortzufahren.



Laden Sie die neuen Tags und Verknüpfungen auf die SPS herunter, indem Sie **Gerät > Parameter > Herunterladen** auswählen oder in der Leiste auf das Download-Symbol klicken. Wenn Sie zum Herunterladen aufgefordert werden, klicken Sie auf **Ja**.



Wenn das Dialogfenster **Liste der ausgeführten Geräte** erscheint, wählen Sie die SPS aus und klicken Sie auf **Nach Wechsel zum Programm-Modus herunterladen**. Wenn Sie gefragt werden, ob der Status ausgegeben werden soll, klicken Sie auf **Ja**.



Wählen Sie im Netzwerkkonfigurator **Netzwerk > Verbindung überprüfen** aus und vergewissern Sie sich, dass keine Verbindungsfehler auftreten.



# PROFINET I/O verwenden

In diesem Kapitel finden Sie alle nötigen Informationen zur Verwendung des CS50-Sensors in einer PROFINET I/O-Umgebung.

## Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für PROFINET I/O aktiviert werden. Wie man Kommunikationsprotokolle für den CS50-Sensor aktiviert oder ändert, erfahren Sie in **Kapitel 1 [Industrieprotokolle aktivieren](#)**.
- Bei CS50-Software-Jobs wird die PROFINET I/O-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert. Weitere Informationen dazu, wie man Job-Parameter und Ausgänge mit di-soric Link-Tags verknüpft, finden Sie in der *CS50-Software-Hilfe* im Abschnitt **di-soric Link**.

**Wichtig:** Mit PROFINET I/O können die IP-Adresse und die Subnetzmaske des CS50-Sensors von der SPS automatisch zugewiesen werden. Dazu darf dem CS50-Sensor jedoch beim Neustart so lange keine IP-Adresse zugewiesen werden, bis die SPS in den Betriebsmodus wechselt. In der Zwischenzeit wird der CS50-Sensor im Netzwerk der CS50-Software nicht angezeigt.

## CS50-Sensor PROFINET I/O

### PROFINET I/O-Identität

#### Hersteller-ID

Die Hersteller-ID von di-soric ist

#### 0x221. Geräte-ID

Die Geräte-ID ist **0x7000**.

### Verbindungseigenschaften: Zyklische RT-Nachrichten

SPS-Eingänge sind an ungeraden, SPS-Ausgänge an geraden Slot-Nummern zu erkennen.

Daten dürfen unabhängig von der Richtung höchstens 518 Byte groß sein. Die Datengröße lässt sich verringern, indem nicht verwendete Slots entfernt werden.

Aktualisierungsintervall des Zyklus: 16 ms

### Beschreibung der Slots/Subslots

Slot	Dir	Bytes	Name	Description
1	In	2	STATUS	Status register of the camera, each bit of this register represents a different state item. See <a href="#">Camera Status Register</a> for bit descriptions
3	In	2	ECHO	This 16 bit word value reflects back to the PLC the value that the PLC wrote to the output assembly ECHO register. The PLC can verify the output assembly has been written to the camera when this value matches the written value.
5	In	4	CmdCodeRslt	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdCodeRslt reflects the result of the command invoked by Control.CmdCode. See <a href="#">CmdCodeRslt</a> for definitions.
7	In	4	CmdRet	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdRet contains the data returned from the command invoked by Control.CmdCode. See <a href="#">CmdRet</a> for definitions.
9	In	2	State	Device State register. Depending on the current state of the camera, certain STATUS and CONTROL features may or may not be operational. See <a href="#">State</a> for

				definitions.
2	Out	2	CONTROL	Control register of camera. Each bit of this register represents a different status item. See <a href="#">Camera Control Register</a> for bit descriptions
4	Out	2	ECHO	This 16 bit value is reflected back to the PLC in the input assembly ECHO register. The PLC can verify the output assembly has been written to the camera when the input assembly matches this written value.
6	Out	4	CmdCode	Specifies the process invoked in the camera when Control.ExeCmd goes active. See <a href="#">CmdCode</a> for definitions.
8	Out	4	CmdArg	Additional argument data for the CmdCode. See <a href="#">CmdArg</a> for definition.
11	In	2	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 145, the most significant bit vio point 160
10	Out	2	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 129, the most significant bit is vio point 144
13	In	8	bool1-64	Each bit represents a bool value. The least significant bit of byte 0 reads the value of bool1. The most significant bit of byte 7 reads bool64.
12	Out	8	Bool101-164	Each bit represents a bool value. The least significant bit of byte 0 writes the value of bool101. The most significant bit of byte 7 writes bool164.
15	In	20	int1-10	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 0-1 for the value of int1 through bytes 18-19 for the value of int10.

14	Out	20	int101-110	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 0-1 to write the value of int101 through bytes 18-19 for the value of int110.
17	In	64	long1-16	Each group of 4 bytes represents a 32 bit signed integer value. The 64 bytes represent 16 long integers. From bytes 0-3 for the value of long1 through bytes 60-63 for the value of long16.
16	Out	64	long101-116	Each group of 4 bytes represents a 32 bit signed integer value. The 64 bytes represent 16 long integers. From bytes 0-3 for the value of long101 through bytes 60-63 for the value of long116.
19	In	96	float1-24	Each group of 4 bytes represents a 32 bit signed integer value. The 96 bytes represent 24 long integers. From byte offsets 0-1 for the value of float1 through byte offsets 92-95 for the value of float24.
18	Out	96	float101-124	Each group of 4 bytes represents a 32 bit signed integer value. The 96 bytes represent 24 long integers. From bytes 0-3 for the value of float101 through bytes 92-95 for the value of float124.
21	In	96	string1	These 96 bytes can store a string of up to 94, 8 bit characters, with the first 2 bytes containing the storage length and string length values.
20	Out	96	string101	These 96 bytes can store a string of up to 94, 8 bit characters, with the first 2 bytes containing the storage length and string length values.
23	In	96	string2-string7	6 consecutive strings, each of 32 bytes can store a string of up to 30, 8 bit characters, with the first 2 bytes of each string group containing the storage length and string length values.
22	Out	96	string102-string107	6 consecutive strings, each of 32 bytes can store a string of up to 30, 8 bit characters, with the first 2 bytes of each string group containing the storage length and string length values.



## Tabelle der Slot-Daten

PLC Input			PLC Output		
Slot	Byte Offset	Data	Slot	Byte Offset	Data
1	0	STATUS	2	0	CONTROL
3	0	Echo In	4	0	Echo Out
5	0	CMD CODE RSLT	6	0	CMD CODE
7	0	CMD RET	8	0	CMD ARG
9	0	STATE			
11	0	VIO 145.. 160	10	0	VIO 129.. 144
13	0	bool 1.. 16	12	0	bool 101.. 116
	2	bool 17.. 32		2	bool 117.. 132
	4	bool 33.. 48		4	bool 133.. 148
	6	bool 49.. 64		6	bool 149.. 164
15	0	int 1	14	0	int 101
	2	int 2		2	int 102
	4	int 3		4	int 103
	6	int 4		6	int 104
	8	int 5		8	int 105
	10	int 6		10	int 106
	12	int 7		12	int 107
	14	int 8		14	int 108
	16	int 9		16	int 109
	18	int 10		18	int 110
17	0	long 1	16	0	long 101
	4	long 2		4	long 102
	8	long 3		8	long 103
	12	long 4		12	long 104
	16	long 5		16	long 105
	20	long 6		20	long 106
	24	long 7		24	long 107
	28	long 8		28	long 108
	32	long 9		32	long 109
	36	long 10		36	long 110
	40	long 11		40	long 111
	44	long 12		44	long 112
	48	long 13		48	long 113
	52	long 14		52	long 114
	56	long 15		56	long 115
	60	long 16		60	long 116

**PLC Input**



























Slot	Byte Offset	Data
19	0	float 1
	4	float 2
	8	float 3
	12	float 4
	16	float 5
	20	float 6
	24	float 7
	28	float 8
	32	float 9
	36	float 10
	40	float 11
	44	float 12
	48	float 13
	52	float 14
	56	float 15
	60	float 16
	64	float 17
	68	float 18
	72	float 19
	76	float 20
	80	float 21
	84	float 22
	88	float 23
	92	float 24

**PLC Output**

Slot	Byte Offset	Data
18	0	float 101
	4	float 102
	8	float 103
	12	float 104
	16	float 105
	20	float 106
	24	float 107
	28	float 108
	32	float 109
	36	float 110
	40	float 111
	44	float 112
	48	float 113
	52	float 114
	56	float 115
	60	float 116
	64	float 117
	68	float 118
	72	float 119
	76	float 120
	80	float 121
	84	float 122
	88	float 123
	92	float 124

PLC Input			PLC Output		
Slot	Byte Offset	Data	Slot	Byte Offset	Data
21	0	94	20	0	94
	1	<str 1 len>		1	<str 101 len>
	2	string 1		2	string 101
	95			95	
23	0	30	22	0	30
	1	<str 2 len>		1	<str 102 len>
	2	string 2		2	string 102
	31			31	
	32	30		32	30
	33	<str 3 len>		33	<str 103 len>
	34	string 3		34	string 103
	63			63	
	64	30		64	30
	65	<str 4 len>		65	<str 104 len>
	66	string 4		66	string 104
	95			95	
	96	30		96	30
	97	<str 5 len>		97	<str 105 len>
	98	string 5		98	string 105
	127			127	
	160	30		160	30
	161	<str 6 len>		161	<str 106 len>
162	string 6	162	string 106		
191		191			

## Tabelle der STEP 7-SPS-Slots

Slot	Module	Order number	I address	Q address	Diagnostic address:
0	 PS 3075				2041*
X1	 PS 3075				2040*
X1A	 PS 3075				2039*
1	 PS 3075		10...11		
2	 PS 3075			0...1	
3	 PS 3075		540...541		
4	 PS 3075			264...265	
5	 PS 3075		532...535		
6	 PS 3075			260...263	
7	 PS 3075		536...539		
8	 PS 3075			256...259	
9	 PS 3075		542		
10	 PS 3075			11...12	
11	 PS 3075		12...13		
12	 PS 3075			3...10	
13	 PS 3075		2...9		
14	 PS 3075			362...381	
15	 PS 3075		352...371		
16	 PS 3075			382...445	
17	 PS 3075		372...435		
18	 PS 3075			266...361	
19	 PS 3075		256...351		
20	 PS 3075			446...541	
21	 PS 3075		436...531		
22	 PS 3075			542...733	
23	 PS 3075		543...734		

## Status: Kamera-Statusregister (16 Bit)

Jedes Bit dieses Registers repräsentiert ein Signal, das den Betriebsstatus der Kamera wiedergibt. Ein hoher Wert für **1** zeigt an, dass das Signal **aktiv** (wahr) ist.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DATA VALID	INSP STAT	INSP BUSY	TRIGGER ACK	EXE CMD ACK		RESET COUNT ACK	ERROR	TRIGGER READY	ACQ BUSY	EXP BUSY	ONLINE
← Inspection 1 →								← All Inspections →							
Bit	Name	Description													
0	ONLINE	Inspections are running													
1	EXP BUSY	The camera is busy capturing an image. The camera should not be triggered or the part under inspection moved during this time if illuminated.													
2	ACQBUSY	The camera is busy acquiring an image. The camera cannot be triggered while busy.													
3	TRIGGER READY	The camera is ready to be triggered. This is equivalent to $ONLINE == 1$ and $ACQBUSY == 0$ .													
4	ERROR	An error has occurred. Set the RESET ERROR control bit high to clear.													
5	RESET COUNT ACK	This bit mirrors the RESET COUNT control bit. The PLC can be certain the reset command was received by the camera when this goes high. The PLC can then bring the RESET COUNT control signal back low.													
7	EXECMD ACK	This bit mirrors the EXE CMD control bit.													
8	TRIGGER ACK	This bit mirrors the TRIGGER control bit.													
9	INSP BUSY	This bit is high when inspection1 is busy processing an image.													
10	INSP STAT	This bit represents the inspection 1 status result. It is 1 if the inspection passes. It is only valid when DataValid goes high.													
11	DATA VALID	This bit goes high when inspection1 is complete. The PLC should clear this signal by setting RESET DV high once it has read results.													

## CmdCodeRslt (32 Bit)

Der Wert von CmdCodeRslt ist nur gültig, wenn **ExeCmdAck** aktiv (1) ist, in Reaktion auf einen aktiven Status von **ExeCmd**.

CmdCodeRslt value (base 16 hex)	Meaning
0x0000_0000	Success
0x0100_0000	Fail.  Possible reasons: Camera under PC control. Job cannot be changed.
0x0200_0000	Fail: NoJob in slot.
0x0300_0000	Fail: Unknown cmd.

## CmdRet (32 Bit)

Der Wert von **CmdRet** ist nur gültig, wenn **ExeCmdAck** aktiv (1) ist – in Reaktion auf einen aktiven Status von **ExeCmd** – und **CmdCodeRslt** den Wert 0 (Erfolg) hat. Die folgende Tabelle zeigt, welche **CmdCodes** im **CmdRet**-Register Werte zurückgeben.

CmdRet value (32 bit)	Associated CmdCode	Meaning
0	0x1000_0000 to 0x1300_0000 (Job Change type)	Na
1 – 255	0x1800_0000 (Query Active Job Slot)	Active Job Slot #

## Status (16 Bit)

**State** (Status) gibt folgende Betriebszustände der Kamera an:

State value	Meaning	Typical action required by the client (PLC), or system operator
<b>(16 bit)</b>		
0	Offline	Perform job change or put camera online.
1	Online	Normal runtime operation: Monitor TriggerReady and DataValid signals. Trigger the camera.
2	Changing Vision Job	<p>If camera is under pc control: Wait until State changes to Offline or Online.</p> <p>If PLC is controlling the job change: Use ExeCmd, CmdCode, ExeCmdAck, and CmdCodeRslt to complete the operation.</p>
3	Booting*	Wait for camera to transition to Online or Offline.
4	Empty (no Vision Job)	Load a new job from AutoVISION or Front Runner.

\*Booting (3) State: This will rarely be seen by the PLC.

Der Wert von „State“ ist maßgeblich dafür, welche Steuersignale (Control) und

Control/Status Signal	State				
	0 (Offline)	1 (Online)	2 (Job Change)	3 (Booting)	4 (Empty)
Control.GO ONLINE	Y				
“.GO OFFLINE		Y			
“.RESET ERROR					
“.RESET COUNT	Y	Y			
“.EXEC CMD	Y	Y	Y		Y
“.TRIGGER		Y			
“.RESET DATA VALID		Y			
Status.ONLINE	Y	Y	Y	Y	Y
“.ERROR					
“.RESET COUNT ACK	Y	Y			
“.EXEC CMD ACK	Y	Y	Y		Y
“.EXP BUSY		Y			
“.ACQ BUSY		Y			
“.TRIGGER READY		Y			
“.TRIGGER ACK		Y			
“.INSP BUSY		Y			
“.INSP STAT		Y			
“.DATA VALID		Y			

Where:

Y = Signal is valid for this State

Empty cell = Signal is not valid for this State

Statussignale (Status) verfügbar sind:



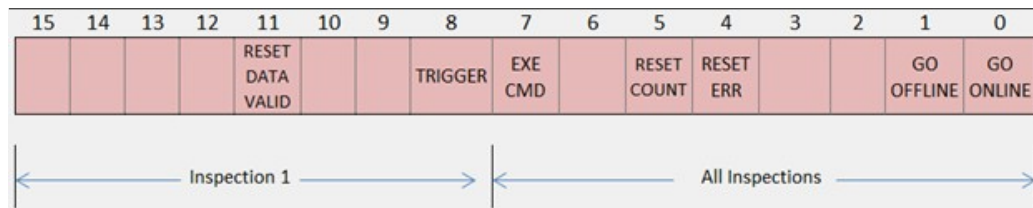
## VIO-Ausgangs-Register-Bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v144	v143	v142	v141	v140	v139	v138	v137	v136	v135	v134	v133	v132	v131	v130	v129

## Steuerung: Kamera-Steuerregister (16 Bit)

Jedes Bit dieses Registers steuert eine bestimmte Kamerafunktion. Die entsprechende Funktion wird durch den Wechsel vom niedrigen Status **0** zum hohen Status **1** gestartet. Im Normalfall setzt die SPS den Status des Steuerbits auf **0** zurück, sobald sie erkannt hat, dass das Steuersignal von der Kamera verarbeitet wurde.

Nicht verwendete Bits sollten im Status **0** verharren.



Bit	Name	Description
0	GO ONLINE	Start all inspections running
1	GO OFFLINE	Stop all inspections
4	RESET ERROR	Reset ERROR in the Status register
5	RESET COUNT	Reset all inspection counts
7	EXE CMD	Execute the command specified by Control.CmdCode
8	TRIGGER	Trigger Inspection 1. The inspection must be configured for a triggered image acquisition.
11	RESET DATA VALID	Reset the Data Valid signal of the Status register

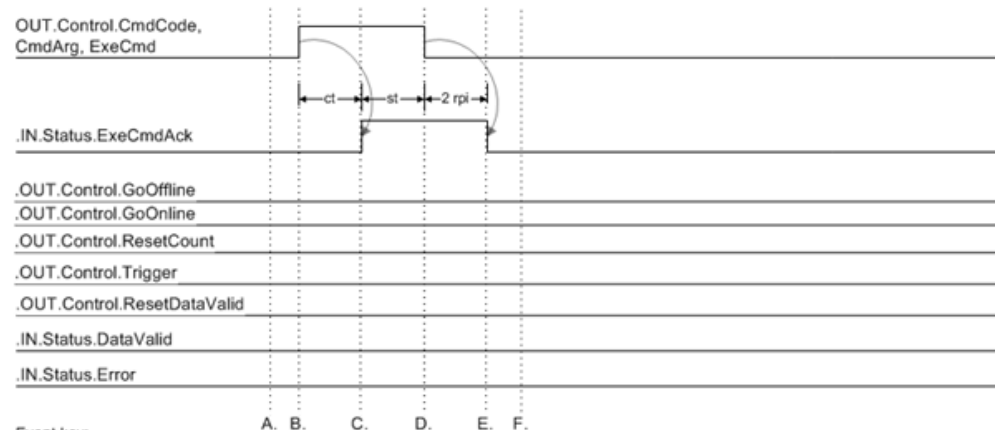
## CmdCode und CmdArg (32 Bit)

CmdCode und CmdArg definieren den Prozess, der in der Kamera ausgelöst wird, wenn **Control.ExeCmd** aktiv wird.

### Liste der möglichen CmdCodes und der entsprechenden CmdArg-Argumente

CmdCode value	CmdArg	Operations performed
0x1000_0000	Job Slot (1-255)	Go Offline, Load job from specified slot
0x1100_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Go Online
0x1200_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job
0x1300_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job, and Go Online
0x1800_0000	na	Query active job slot. <u>CmdRet</u> will contain the active job slot number when the operation is done.

### CmdCode und ExeCmd-Betrieb



Event key:

A. If DataValid or Error are present, clear them.

Set the following control signals idle, and keep them idle while the command is processed by the camera:

GoOffline, GoOnline, Trigger, ResetDataValid, ResetCount, ResetError.

If the command operation is a job change, populate the output tags required to configure the new job (bool, int, long, float, string).

B. Populate CmdCode and CmdArg, then activate ExeCmd.

C. Camera executes the command (may take up to a minute). While processing a Job Change command, State will be 2. Camera activates ExeCmdAck when it is done processing the command.

D. When the PLC sees an active ExeCmdAck, verify CmdCodeRslt is 0, and Error is 0. Process CmdRet if needed, then clear ExeCmd.

E. Camera clears ExeCmdAck when ExeCmd goes inactive. When ExeCmdAck goes inactive, CmdCodeRslt and CmdRet are no longer valid, and it may take a few seconds for the camera State and Online signals to settle to a final value (typically Online or Offline).

F. Camera can now be put online and triggered.

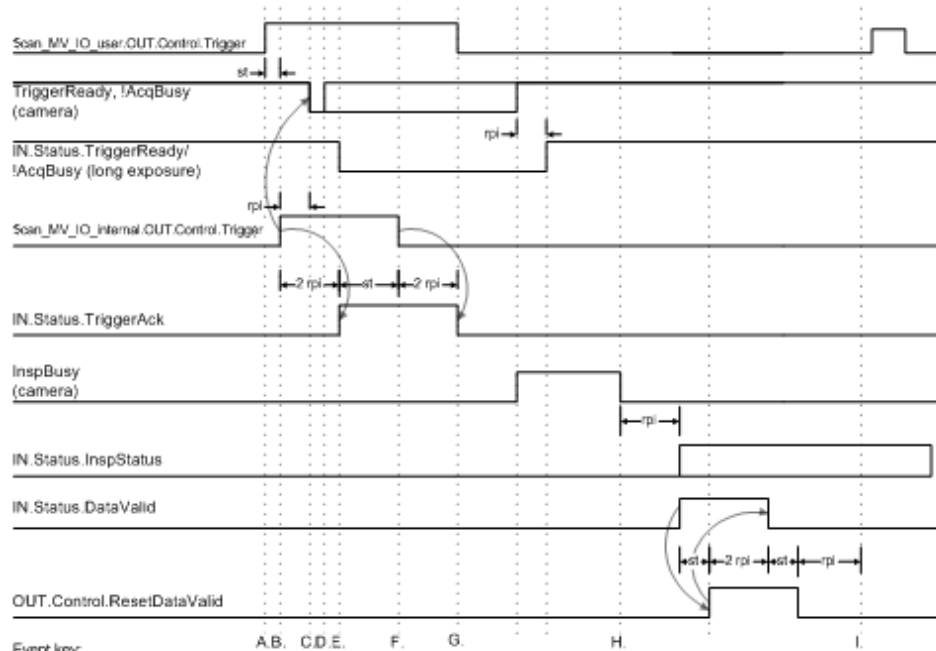
Notes:

st = PLC program scan time

ct = Command processing time in the camera. May take up to a minute for some commands.

rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.

All signals represent the state of plc tags.

**PROFINET I/O-Steuer- und -Statussignale****Event key:**

- A. On rising edge of system trigger, the user app activates Scan\_MV\_IO\_user.OUT.Control.Trigger to trigger the demo code.
- B. Demo code detects rising edge of Scan\_MV\_IO\_user.OUT.Control.Trigger, and if the camera is ready, sends a trigger to the camera.
- C. Camera acquisition begins (may be delayed by one rpi).
- D. If the camera's exposure time is shorter than the rpi, no change will be seen in TriggerReady and AcqBusy plc IN tags.
- E. Camera firmware acks the trigger. The demo code may not see the ack until two rpi after the trigger was sent (event B).
- F. Demo code detects TriggerAck and clears the Trigger.
- G. Demo code detect falling edge of TriggerAck and clears the user Trigger.
- H. Camera internal signal DataValid will go high when InspBusy goes low
- I. Plc logic must delay one rpi time before re-asserting ResetDataValid

**Notes:**

1. The chart shows the workings of the Trigger and ResetDataValid Control signals, and the TriggerAck and DataValid Status signals.
2. st = plc program scan time
3. rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.
4. All signals represent the state of plc tags, except where noted as "(camera)". The cam signals shown are visible in the EIP interface, but the state of the plc tags and internal firmware signals will be different for at least one or two requested packet intervals (rpi).
5. The plc is running the demo code distributed with the camera. The demo code and user app use the Scan\_MV\_IO\_user tag set as the primary control, status, and data interface for the user app. All signal operations are still true even if the plc demo code is not used.
6. TriggerReady/!AcqBusy: Camera exposure times can range from less than 1 ms, up to 100 ms.



# Demo PROFINET I/O-Objekt-Zähler mit TIA-Portal V13 und CS50-Sensor

Dieses Kapitel beschreibt die Anwendung des di-soric-SPS-Democodes in einem Vision-Job mit Testobjekt. Die PROFINET I/O-Demodateien befinden sich im Installationsordner der CS50-Software: **C:\di-soric\Tutorials and Samples\CS50 Sensor\PROFINET demo\TIAPortal\_Demos\BlobDemo**. Öffnen Sie die Datei **Blob\_demo.avp** in der CS50-Software und laden Sie sie auf die Kamera herunter.

#### Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für PROFINET I/O aktiviert werden. Informationen zum Aktivieren und Umschalten von Kommunikationsprotokollen finden Sie in **Kapitel 1 Industrieprotokolle aktivieren**.
- Bei CS50-Software-Jobs wird die PROFINET I/O-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert.

**Wichtig:** Mit PROFINET I/O können die IP-Adresse und die Subnetzmaske der Kamera von der SPS automatisch zugewiesen werden. Dazu darf dem CS50-Sensor jedoch beim Neustart so lange keine IP-Adresse zugewiesen werden, bis die SPS in den Betriebsmodus wechselt. In der Zwischenzeit wird der CS50-Sensor im Netzwerk der CS50-Software nicht angezeigt.

## Überblick

In dieser Demonstration erfahren Sie, wie man einen gespeicherten Job auf die Kamera lädt und über PROFINET I/O eine Verbindung zu einer Siemens ET200SP-CPU-SPS herstellt. Außerdem zeigen wir Ihnen anhand einiger Beispiele, wie man Programme ausführt, die über eine Schnittstelle mit der Kamera verbunden sind. Während der Prüfung der PROFINET I/O-Kapazitäten werden folgende Schritte ausgeführt:

- **Start mit der CS50-Software**

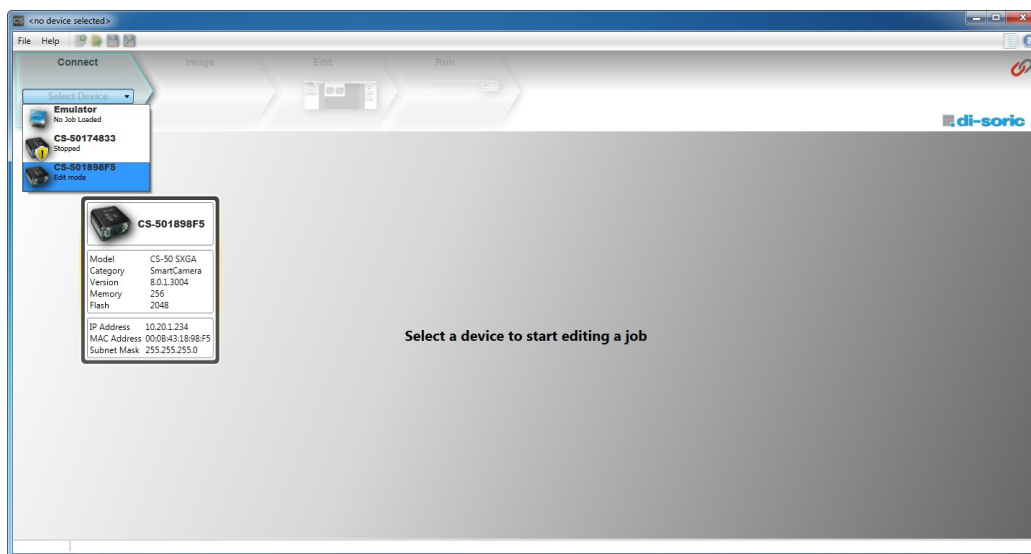
Öffnen Sie einen Demo-Vision-Job in der CS50-Software. Bevor die Kamera mit der SPS verbunden wird, können Sie mithilfe der Option „Try Out“ (Ausprobieren) eine Vorschau aufrufen.

- **Vorbereitung der SPS**

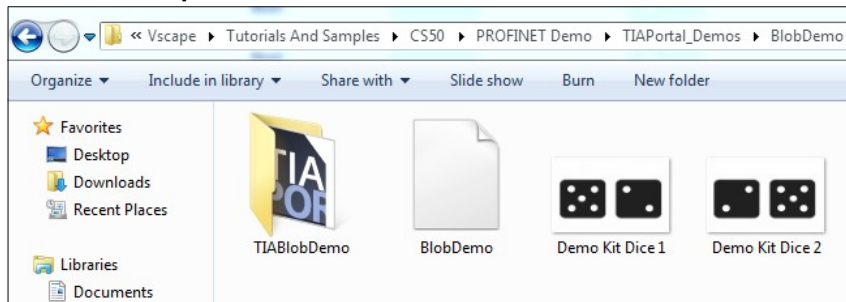
Integrieren Sie die Kamera in die SPS-Umgebung mit der Software TIA Portal und der GSD-Datei.

- **Start der Kamera**

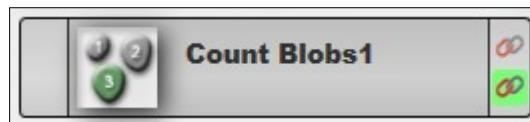
Lösen Sie die Kamera, während sie online ist, mit der SPS aus und beobachten Sie die Veränderungen im Inspektionsstatus, während die SPS die Parameter des Vision-Jobs neu konfiguriert. Vor dem Start der CS50-Software sollten Sie sicherstellen, dass entweder die Kamera mit der SPS verbunden ist oder sich SPS und Kamera beide im gleichen Netzwerk befinden. Vergewissern Sie sich, dass der PC, die SPS und die Kamera dieselbe Netzwerkkategorie und die entsprechenden Subnetzadressen aufweisen. Starten Sie die CS50-Software und wählen Sie die Kamera aus.



Klicken Sie in der Ansicht „Image“ (Bild) auf **Load a Job** (Job laden). Suchen Sie dann den Ordner **C:\di-soric\Vscape\Tutorials And Samples\CS50 Sensor\PROFINET demo\TIAPortal\_Demos\BlobDemo**. Wählen Sie die Datei **BlobDemo.avp** aus.



Der Demo-Job enthält ein Tool: **Count Blobs** (Objekte zählen).

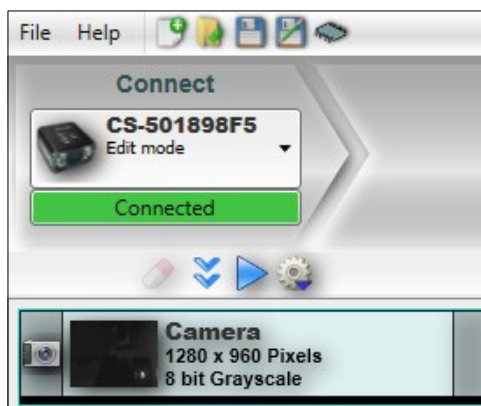


Das mit der PROFINET I/O-Struktur verknüpfte Objekt-Tool-Daten-Element ist hier zu sehen:

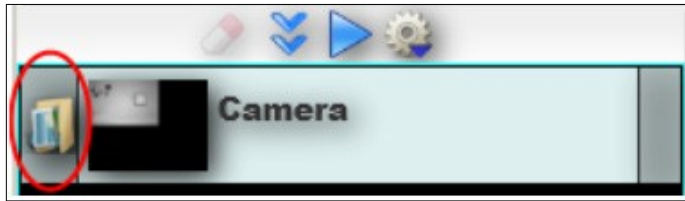
Tool Result	PLC IN
Number of parts	"UserData".NumberOfParts (%DB2.DBW0)

Diese Daten werden in zyklischen Intervallen zwischen Kamera und SPS ausgetauscht.

Sobald der Job geladen ist, verknüpfen Sie die vorgeschichteten Bilder auf dem lokalen PC. Klicken Sie dazu auf der Schaltfläche **Camera** (Kamera) auf das Symbol ganz links, um ein Bild auszuwählen und zu laden.



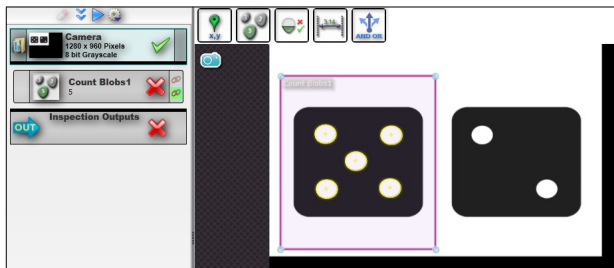
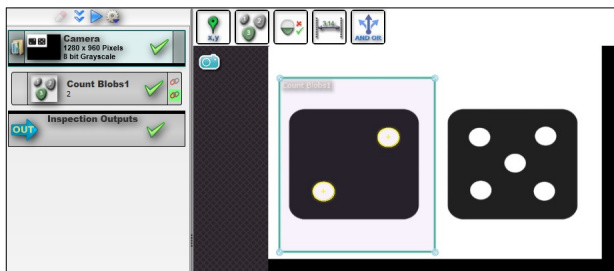
Daraufhin öffnet sich ein Fenster, in dem Sie eine Datei auswählen können. Navigieren Sie zu demselben Ordner, in dem der Demo-Job (PROFINET I/O Demo) geladen wurde. Sobald die Bilder gefunden wurden, ändert sich das Symbol: Anstelle der Kamera erscheint nun ein Ordner.



Klicken Sie auf das Symbol **Try Job Once** (Job einmal testen), um den gesamten Job mit dem geladenen Bild durchlaufen zu lassen.



Die Datei enthält zwei Bilder. Eines davon führt bei der Inspektion zu einem Erfolg, das andere zu einem Fehlschlag.





Klicken Sie nun oben in der Symbolleiste auf **Run** (Ausführen). Dadurch wird der Job auf die Kamera heruntergeladen. Der Job kann nun ausgeführt und getestet werden. Damit die Demo mit Job-Änderung ausgeführt werden kann, muss dieser Job allerdings in **Slot 1** geladen werden. Klicken Sie oben in der Symbolleiste wieder auf die Schaltfläche zum Öffnen der Ansicht **Edit** (Bearbeiten).

Klicken Sie auf das Slot-Symbol und wählen Sie „Slot 1“ oder **New Slot** (Neuer Slot) aus, falls derzeit keine Slots belegt sind.



Der Job und die Bilder werden nun im Flash-Speicher der Kamera gespeichert.

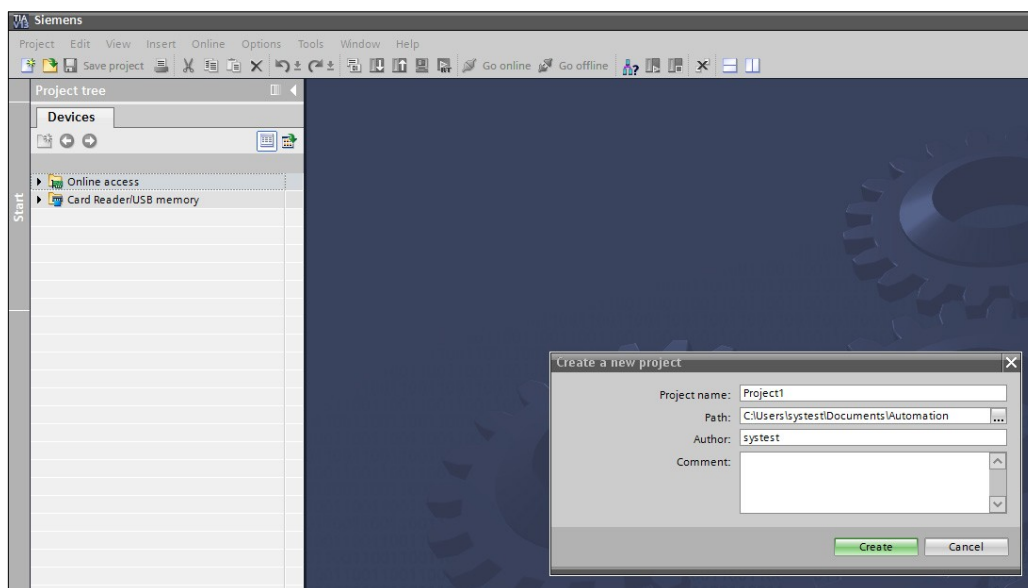
## Einrichtung von TIA Portal Version 13

Die Anleitung in diesem Kapitel wurde auf der Basis der Software Siemens TIA Portal (Version 13) und einer ET200SP-CPU (Katalognummer 6ES7 510-1DJ01-0AB0, Hardwareversion 2, Firmwareversion 1.8.2) erstellt.

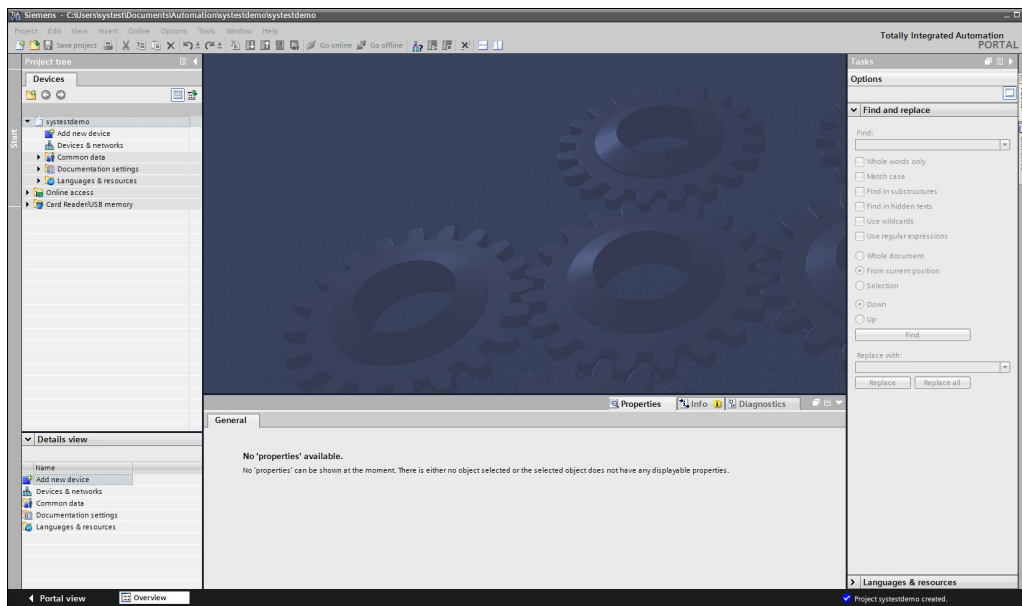
Starten Sie das TIA Portal vom Desktop aus.



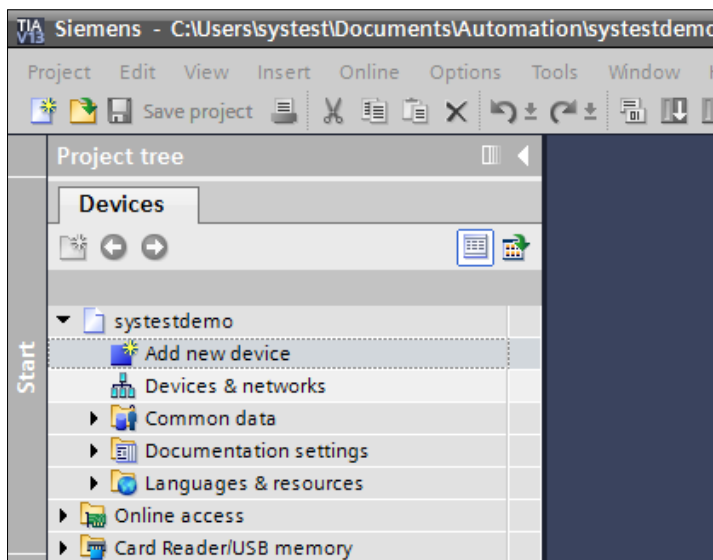
Erstellen Sie über das Menü **Projekt > Neu** ein neues Projekt. Der Standardname lautet „Project1“, wie unten dargestellt. Geben Sie die nötigen Informationen an und klicken Sie dann auf **Erstellen**.



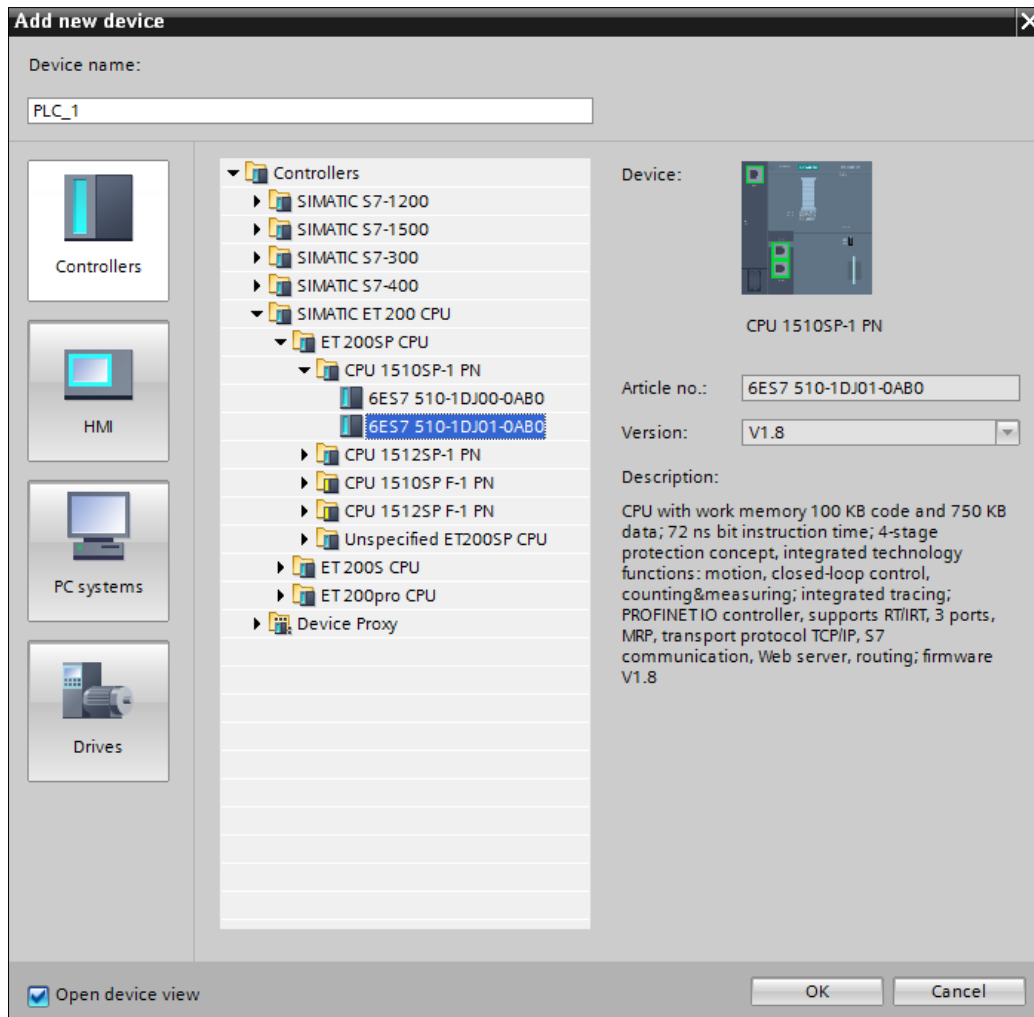
Dieser Screenshot zeigt den Hauptbildschirm, der sich öffnet, wenn Sie ein Projekt öffnen.



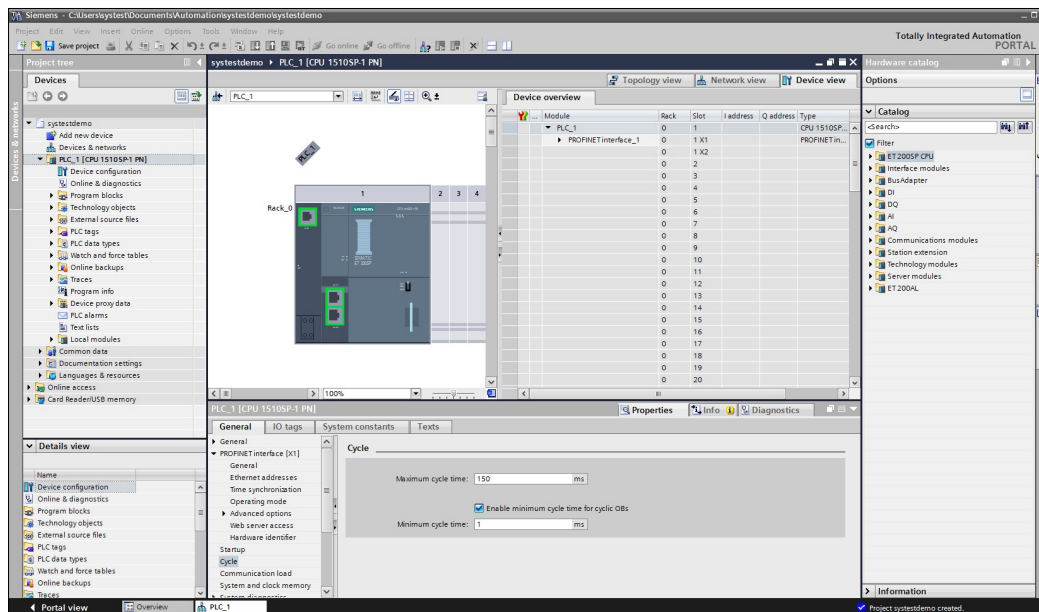
Doppelklicken Sie auf **Neues Gerät hinzufügen**.



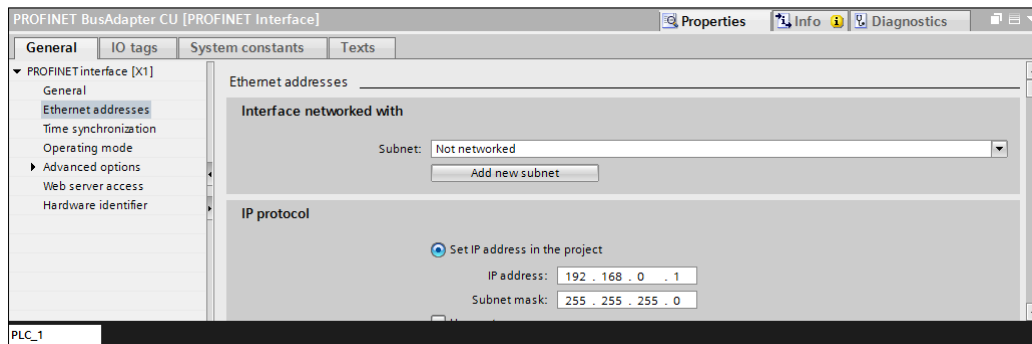
Öffnen Sie den Pfeil **SIMATIC ET200 CPU** und dann den Pfeil **ET200SP CPU**. Öffnen Sie **CPU 1510SP-1 PN** und klicken Sie anschließend auf **6ES7 510-1DJ01-0AB0**. Daraufhin öffnet sich der unten abgebildete Bildschirm. Klicken Sie auf **OK**.



Sie sehen nun folgenden Bildschirm.

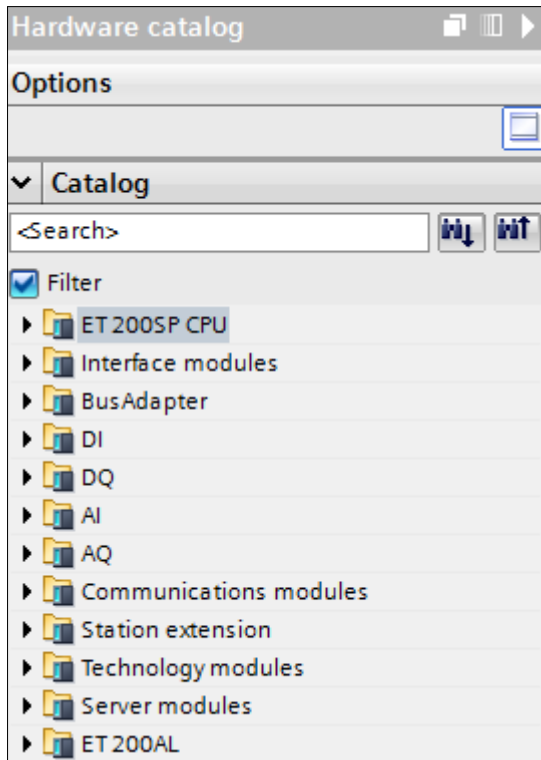


Klicken Sie im SPS-Bild auf das untere grüne Kästchen. Daraufhin sehen Sie im unteren Bereich Folgendes:



Klicken Sie auf **Neues Subnetz hinzufügen**. Im Feld „Subnetz“ wird nun „PN/IE\_1“ angezeigt. Geben Sie nun in das Feld „IP-Adresse“ die IP-Adresse ein, die für dieses Gerät verwendet werden soll. Denken Sie daran, regelmäßig auf **Projekt speichern** zu klicken, damit Ihre Änderungen nicht verloren gehen.

Auf der rechten Seite des Bildschirms sehen Sie nun dies:

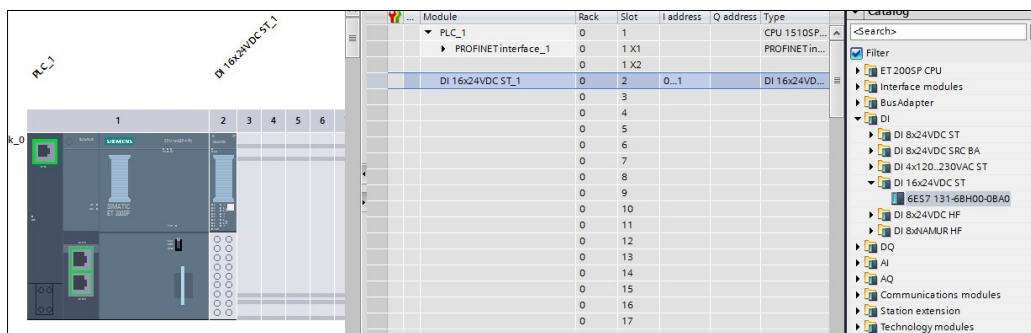


Unser Testgerät umfasst folgende Module:

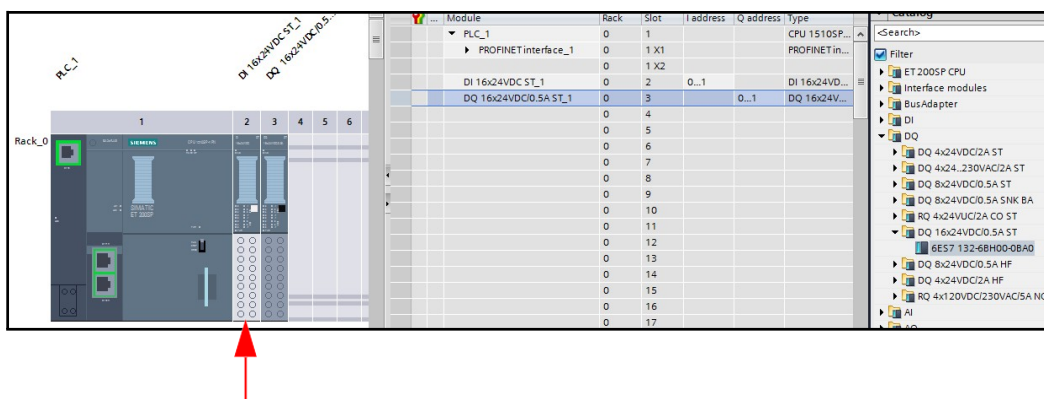
1. DI 16x24VDC ST, Artikelnummer: 6ES7 131-6BH00-0BA0
2. DQ 16x24VDC ST, Artikelnummer: 6ES7 132-6BH00-0BA0

Um diese Module hinzuzufügen, klicken Sie zunächst auf den Pfeil neben **DI** und öffnen Sie dann über die entsprechenden Pfeile die Untermenüs, bis Sie die Artikelnummer von Gerät 1 gefunden haben. Klicken Sie darauf und halten Sie die linke Maustaste gedrückt. Ziehen Sie die Nummer nun nach links in Slot 2.

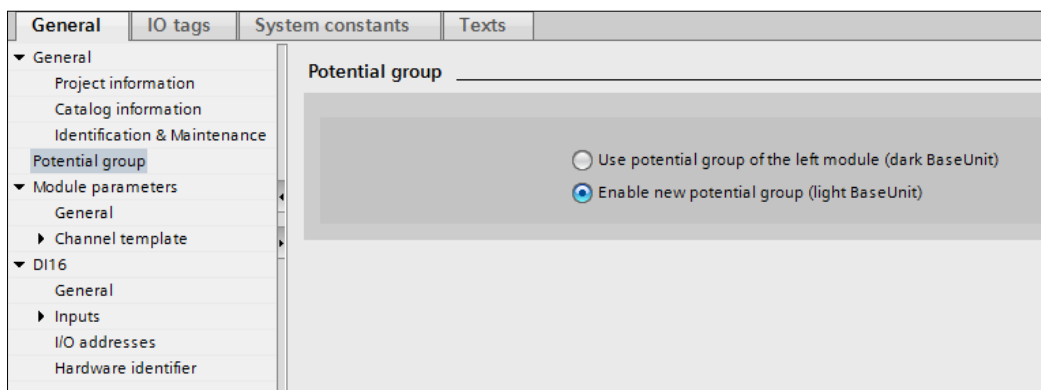
Sie müssten nun Folgendes sehen:



Wiederholen Sie diese Schritte für Gerät 2, indem Sie auf den Pfeil neben **DQ** klicken und das entsprechende Untermenü öffnen. Sie müssten nun Folgendes sehen:

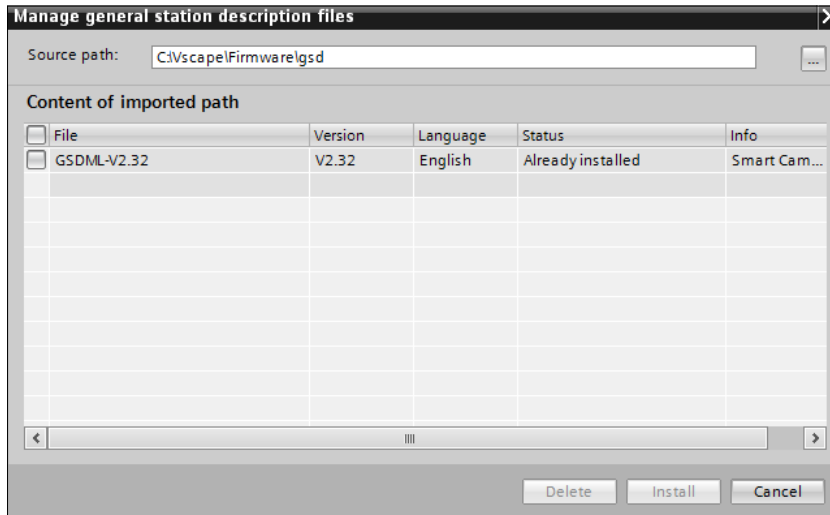


Klicken Sie auf das SPS-Bild in Slot 2. Sie sehen nun folgenden Bildschirm:



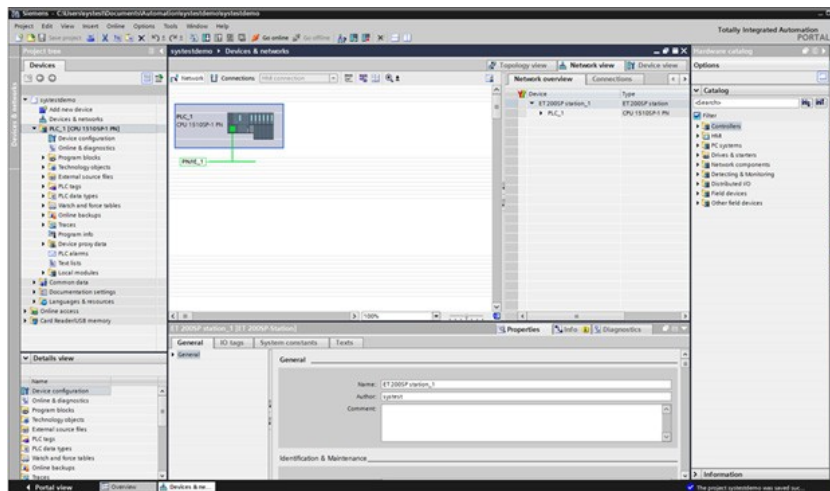
Vergewissern Sie sich, dass die Option **Neue Potenzialgruppe ermöglichen (Helle BaseUnit)** ausgewählt ist. Wiederholen Sie diese Schritte für Slot 3. Speichern Sie das Projekt.

Klicken Sie als Nächstes auf **Extras > Gerätebeschreibungsdateien (GSD) verwalten**.



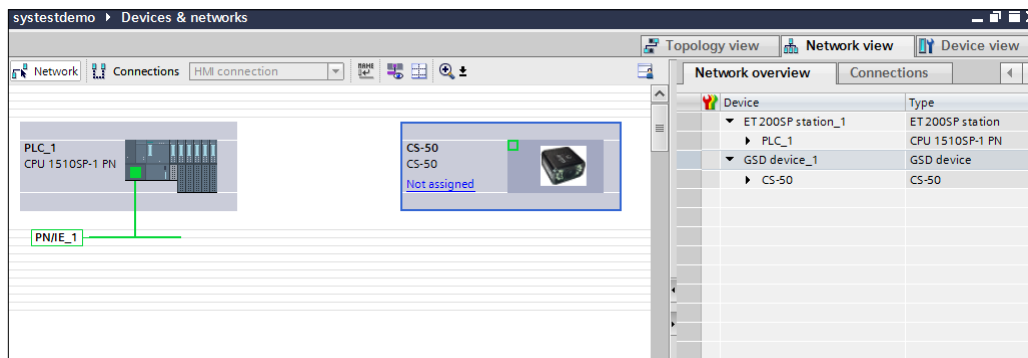
Navigieren Sie zu folgendem Dateipfad: **C:\di-soric\Vscape\Firmware\gsd\CS50 Sensor**. Wählen Sie die entsprechende GSD-Datei aus, indem Sie das Kästchen neben dem Namen aktivieren, und klicken Sie dann auf **Installieren**.

Klicken Sie auf die Registerkarte **Netzwerkansicht**:

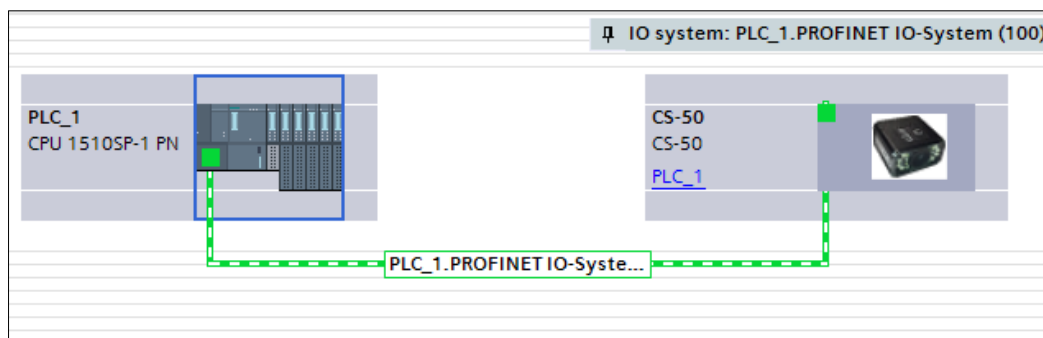




Klicken Sie im rechten Bereich auf **Hardware-Katalog**. Navigieren Sie mithilfe der Pfeile zu folgendem Bereich: **Andere Feldgeräte** > **PROFINET IO** > **General (Allgemein)** > **di-soric** > **Smart Camera (Smartkamera)** > **CS50 Sensor**. Klicken Sie auf den CS50-Sensor und halten Sie die linke Maustaste gedrückt. Ziehen Sie den Sensor nun nach links in den Bereich „Geräte & Netzwerke“. Sie sehen nun Folgendes:



Klicken Sie im Bild des CS50-Sensors auf den Link **Nicht zugeordnet** und dann auf **PLC\_1.PROFINET IO-System**. Dadurch wird der CS50-Sensor mit dem PROFINET-Netzwerk verbunden.



Doppelklicken Sie auf das Bild des CS50-Sensors. Im unteren Bereich sehen Sie nun Folgendes:

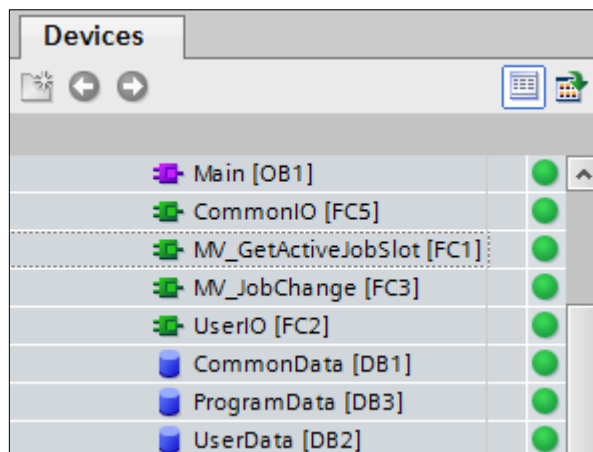
The screenshot shows the configuration window for a CS50 sensor in TIA Portal. The interface has a left-hand navigation tree and a main configuration area. The navigation tree is expanded to show the 'PROFINET interface [X1]' section, with 'General' selected. The main configuration area is titled 'General' and contains the following fields:

- Name: CS50
- Author: systest
- Comment: (empty text area)
- Rack: 0
- Slot: 0

Geben Sie in das Feld „Name:“ den Namen des Geräts ein. Klicken Sie dann links auf „Ethernet-Adressen“ und geben Sie die IP-Adresse der Kamera ein. Dadurch kann die SPS die IP-Adresse der Kamera festlegen.

## Beispielprogramm importieren

Öffnen Sie das Beispielprogramm im TIA Portal über das Menü **Projekt > Öffnen**. Klicken Sie auf **Durchsuchen** und suchen Sie das Programm **BlobDemo.ap13**. Es befindet sich im Ordner **C:\di-soric\Vscape\Tutorials and Samples\CS50 Sensor\PROFINET Demo\TIAPortal\_Demos\BlobDemo**. Öffnen Sie auf der linken Seite auf der Registerkarte **Geräte** den Pfeil neben „PLC\_1[CPU 1510SP-1 PN]“. Klicken Sie auf den Pfeil neben „Programmblöcke“ und wählen Sie dann alle Blöcke wie unten dargestellt aus.

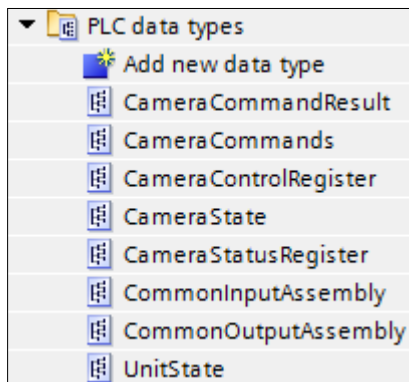


Klicken Sie mit der rechten Maustaste und wählen Sie „Kopieren“ aus. Klappen Sie nun für Ihr neues Projekt die gleichen Pfeile wie oben beschrieben aus. Klicken Sie dann mit der rechten Maustaste auf „Programmblöcke“ und dann auf „Einfügen“. Denken Sie daran, Ihre Änderungen zu speichern! Es kann sein, dass Sie zuvor im neuen Projekt den Programmbaustein „Main [OB1]“ löschen müssen.

Klicken Sie nun im linken Fensterabschnitt des Projekts „BlobDemo“ auf „SPS-Datentypen“. Klicken Sie dann mit der rechten Maustaste, wählen Sie „Kopieren“ aus und fügen Sie die kopierten Inhalte in die SPS-Datentypen des neuen Projekts ein.

Klicken Sie links im Projekt „BlobDemo“ auf „Beobachtungs- und Forcetabellen“. Klicken Sie mit der rechten Maustaste auf „Beobachtungstabelle\_1“ und wählen Sie dann „Kopieren“ aus. Fügen Sie die Tabelle in „Beobachtungs- und Forcetabellen“ des neuen Projekts ein. Denken Sie daran, das Projekt zu speichern!

Die Daten im neuen Projekt müssten nun so aussehen:



## Mapping von IO überprüfen

Je nachdem, in welcher Reihenfolge die SPS Elemente laden kann, müssen möglicherweise die E/A-Adressen im Block geändert werden, damit sie mit denen in der Geräteansicht unter „Geräteübersicht“ übereinstimmen.

Doppelklicken Sie auf das Bild des CS50-Sensors. Daraufhin wird rechts die Geräteübersicht angezeigt.

Der folgende Screenshot dient lediglich der Veranschaulichung.  
sDie Geräteübersicht kann bei Ihnen anders aussehen.

Device overview						
Module	Rack	Slot	I address	Q address	Type	
CS501898f5	0	0			CS50	
Interface	0	0 X1			CS50	
Status_1	0	1	2...3		Status	
Control_1	0	2		2...3	Control	
Echo In_1	0	3	256...257		Echo In	
Echo Out_1	0	4		256...257	Echo Out	
Cmd Code Rslt_1	0	5	258...261		Cmd Code Rslt	
Cmd Code_1	0	6		258...261	Cmd Code	
Cmd Ret_1	0	7	262...265		Cmd Ret	
Cmd Arg_1	0	8		262...265	Cmd Arg	
State_1	0	9	266		State	
	0	10				
	0	11				
	0	12				
	0	13				
	0	14				
	0	15				
	0	16				

Öffnen Sie ganz rechts im „Hardware-Katalog“ den Abschnitt **Modul**. Sie sehen nun Folgendes:

Module
Boolean In
Boolean Out
Float In
Float Out
Int In
Int Out
Long In
Long Out
Long String In
Long String Out
Short String In
Short String Out
VIO In
VIO Out

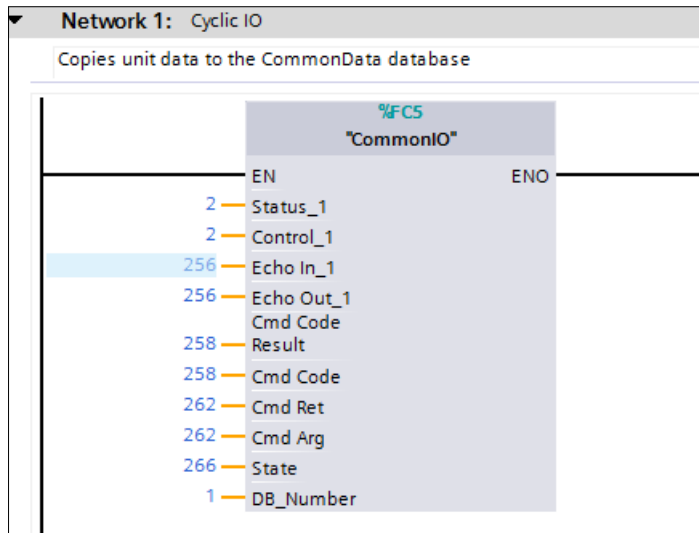
Ziehen Sie „Int In“ mit der Maus in die Geräteübersicht. Wenn Sie auf „Int In“ klicken, zeigt TIA den entsprechenden Ziel-Slot an. „Int In“ kann nur in den angezeigten Slot gezogen werden; alles andere wird von TIA unterbunden.

Ihre Geräteübersicht müsste nun folgendermaßen aussehen. Die hier dargestellten Adressen sind lediglich exemplarisch und stimmen nicht mit den Adressen in Ihrer Übersicht überein.

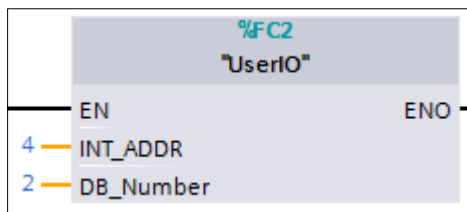
Device overview							
	...	Module	Rack	Slot	I address	Q address	Type
		▼ CS501898f5	0	0			CS50
		▶ Interface	0	0 X1			CS50
		Status_1	0	1	2...3		Status
		Control_1	0	2		2...3	Control
		Echo In_1	0	3	256...257		Echo In
		Echo Out_1	0	4		256...257	Echo Out
		Cmd Code Rslt_1	0	5	258...261		Cmd Code Rslt
		Cmd Code_1	0	6		258...261	Cmd Code
		Cmd Ret_1	0	7	262...265		Cmd Ret
		Cmd Arg_1	0	8		262...265	Cmd Arg
		State_1	0	9	266		State
			0	10			
			0	11			
			0	12			
			0	13			
			0	14			
		Int In_1	0	15	4...23		Int In
			0	16			

Als Nächstes müssen Sie die SPS-Programmblöcke an die folgende Ansicht anpassen. Klappen Sie dazu links unter „Geräte“ den Pfeil neben „Programmblöcke“ aus und doppelklicken Sie auf **OB1 [OB1]**. Dadurch wird in der Mitte ein neuer Bereich geöffnet. Scrollen Sie in diesem Bereich nach unten zu Netzwerk 1.

Gleichen Sie hier die Zahlen an die in der Geräteübersicht angezeigten Werte an (siehe Beispiel unten). Lassen Sie den Wert „DB\_Nummer“ unverändert.



Gehen Sie für Netzwerk 4 genauso vor. Verwenden Sie dazu die Adresse für „Int In\_1“ aus der Geräteübersicht und fügen Sie sie unten bei „INT\_ADDR“ ein. Lassen Sie den Wert „DB\_Nummer“ unverändert.



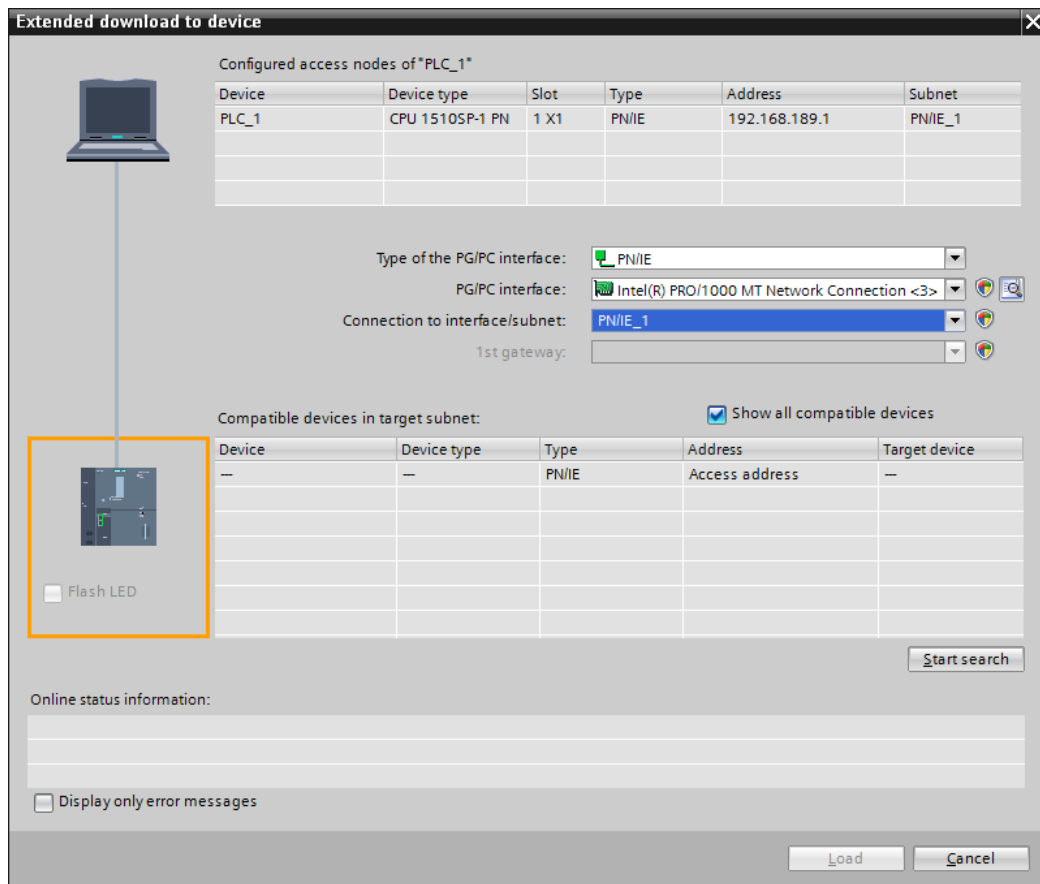
Denken Sie daran, das Projekt zu speichern! Klicken Sie in der obersten Ebene auf **PLC\_1 [CPU 1510SP-1 PN]** und dann auf das Symbol zum Kompilieren.



Es dürfen KEINE Kompilierfehler auftreten. Sofern keine Fehler aufgetreten sind, klicken Sie auf das Download-Symbol.

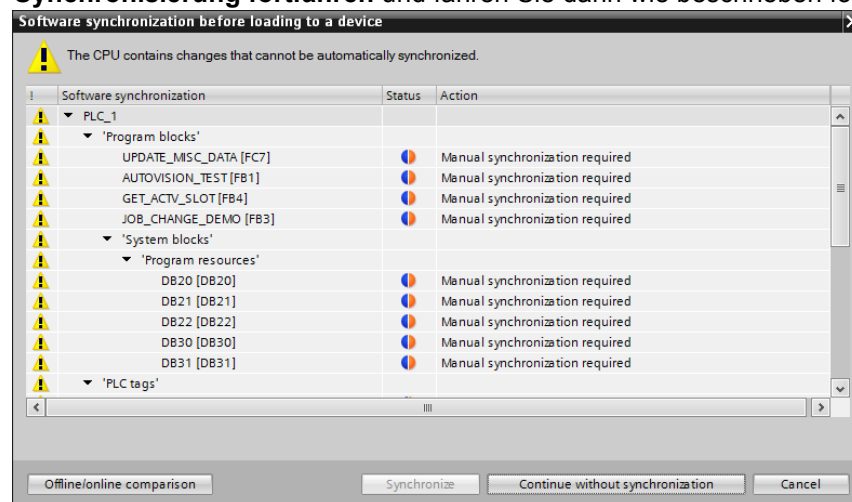


Sie sehen nun folgenden Bildschirm:

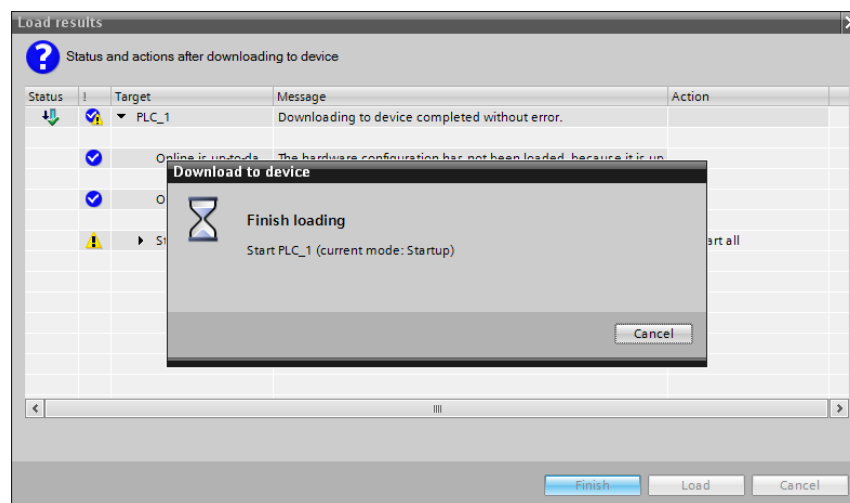




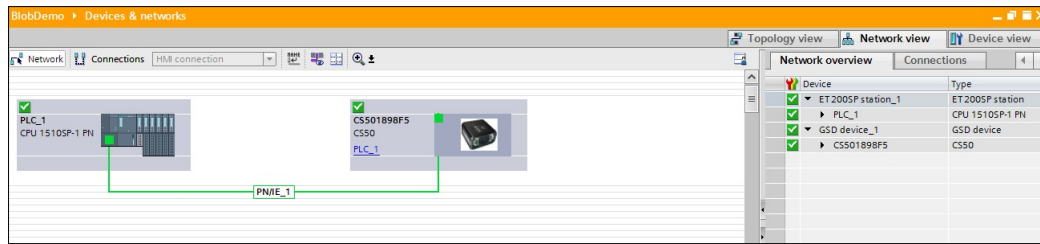
Wählen Sie Ihre PC-Netzwerkschnittstelle aus und ändern Sie die Verbindung zu Schnittstelle/Subnetz in **PN/IE\_1**. Klicken Sie auf **Suche starten**. Es müsste nun „SPS\_1“ angezeigt werden, zusammen mit der zugewiesenen Adresse. Klicken Sie auf **Laden**. Falls ein ähnlicher Bildschirm wie unten angezeigt wird, führen Sie für die SPS einen Speicher-Reset (MRES) durch und wiederholen Sie den Ladevorgang noch einmal. Sollte das Problem dann immer noch auftreten, klicken Sie auf **Ohne Synchronisierung fortfahren** und fahren Sie dann wie beschrieben fort.



Wenn das Herunterladen erfolgreich war, sehen Sie folgenden Bildschirm:



Alle LEDs der SPS sollten **GRÜN** sein. Klicken Sie auf **Online gehen**. Sie müssten nun



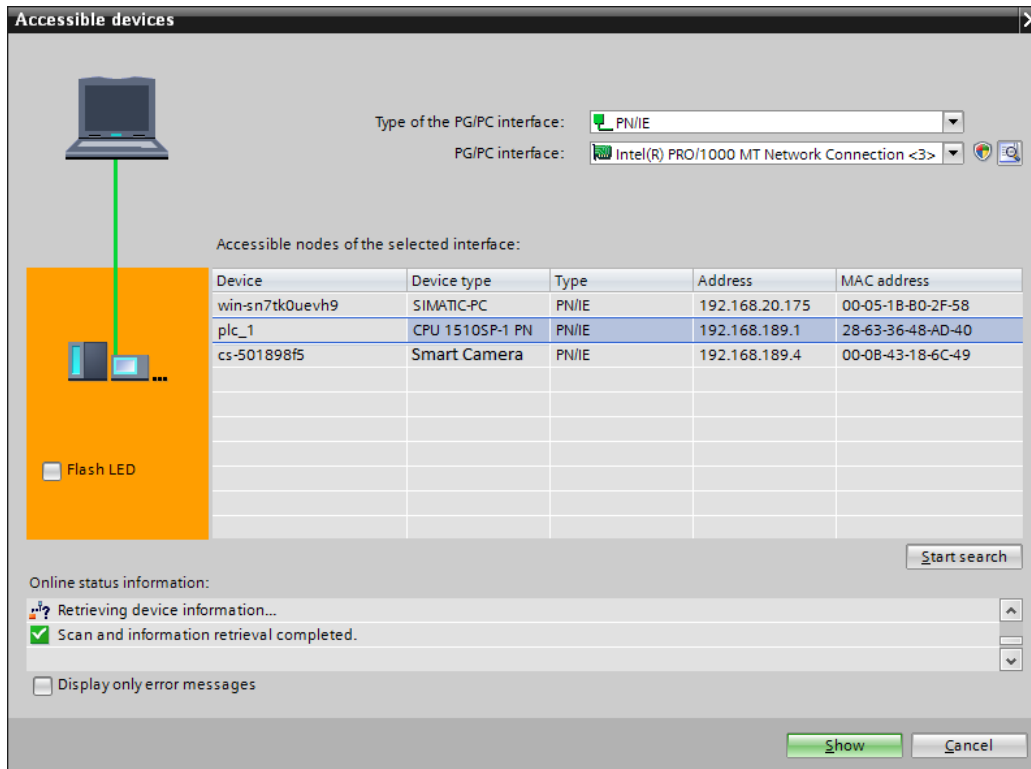
Folgendes sehen:

## Fehlerbehebung

Falls nicht alle LEDs grün sind, gehen Sie offline und klicken Sie auf das Symbol für die verfügbaren Geräte.





Klicken Sie auf **Suche starten**. Die SPS und der CS50-Sensor müssten nun angezeigt werden.



Falls der Name des CS50-Sensors falsch ist, wird keine Verbindung hergestellt. Der Standardname lautet „cs50“ plus die letzten drei Zeichenpaare der MAC-Adresse, z. B. „cs50186c49“. Kehren Sie zu den allgemeinen Einstellungen des CS50-Sensors in der Geräteübersicht zurück und ändern Sie ggf. den Namen. Wiederholen Sie das Kompilieren und Herunterladen. Versuchen Sie dann noch einmal, eine Verbindung herzustellen. Dies ist die häufigste Ursache für Fehler beim Herstellen einer Verbindung.

## Demo ausführen

Klicken Sie im linken Fensterbereich auf den Pfeil neben „Beobachtungs- und Forcetabellen“. Doppelklicken Sie auf „Beobachtungstabelle\_1“. Die Beobachtungstabelle wird daraufhin im zentralen Bereich des Fensters geöffnet. Klicken Sie auf das Symbol zum Überwachen der Variablen.

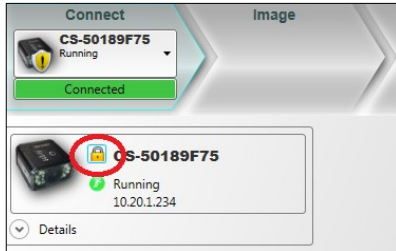
i	Name	Address	Display format	Monitor v...	Modify value	
1	*ProgramData*.ChangeJob		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
2	*ProgramData*.New_JobSlot		DEC+/-	2		<input type="checkbox"/>
3	*ProgramData*.Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
4	*ProgramData*.Current_JobSlot		DEC+/-	1		<input type="checkbox"/>
5	*ProgramData*.Trigger 		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
6	*ProgramData*.TotalTriggerCount		DEC+/-	1		<input type="checkbox"/>
7	*ProgramData*.Pass		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
8	*ProgramData*.TotalPassedCount		DEC+/-	0		<input type="checkbox"/>
9	*ProgramData*.Fail		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
10	*ProgramData*.TotalFailedCount		DEC+/-	1		<input type="checkbox"/>
11	*ProgramData*.TotalInspCount		DEC+/-	1		<input type="checkbox"/>
12	*ProgramData*.ResetCounters		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
13	*UserData*.NumberOfParts	%DB2.DBW0	Hex	16#0000		<input type="checkbox"/>
14	*ProgramData*.ResetError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
15	*ProgramData*.CommandResult.Fail		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
16	*ProgramData*.CommandResult.No Job In Slot		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
17	*ProgramData*.CommandResult.UnknownCmd		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
18	*ProgramData*.CommandResult.Success		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>

Hier können nun folgende Vorgänge getestet werden:

Change Job	Switch jobs from one slot to another
Get Current Job Slot	Displays the slot of the currently running job
Trigger an Inspection	Causes the camera to cycle through its inspection.
Reset Counters	Resets the TotalTriggerCount, TotalPassedCount, TotalFailedCount, and TotalInspCount to 0, and also sends a command to the camera to reset its internal counters.
Reset an Error condition	If an error occurs during run time, this will reset it, so demo may be continued.

## Jobs ändern und aktuellen Job-Slot abrufen

Denken Sie daran, wie oben beschrieben einen Job in einen Slot des Kameraspeichers zu laden. Damit dieser Test korrekt durchgeführt werden kann, laden Sie einen weiteren Job in **Slot 2** der Kamera. Laden Sie beispielsweise den Job **Circle\_LocatorDemo**. Geben Sie die Gerätesteuerung frei, indem Sie auf die Registerkarte **Verbinden** klicken und dann, wie im Bild unten dargestellt, das Schlosssymbol zu **geschlossen** ändern. Klicken Sie anschließend erneut auf die Registerkarte **Ausführen**, damit der neue Job richtig ausgeführt wird.



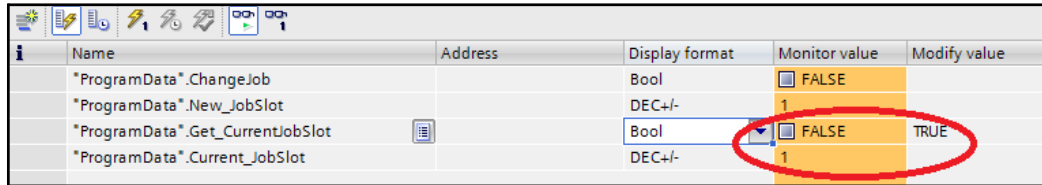
Navigieren Sie dann zur **Beobachtungstabelle\_1** und klicken Sie dort auf die Schaltfläche mit der Brille und dem Symbol zum Abspielen. Damit beginnt nun die Datenüberwachung.

Name	Address	Display format	Monitor value	Modify value
"ProgramData".ChangeJob		Bool	<input type="checkbox"/> FALSE	
"ProgramData".New_JobSlot		DEC+-	1	
"ProgramData".Get_CurrentJobslot		Bool	<input type="checkbox"/> FALSE	
"ProgramData".Current_JobSlot		DEC+-	2	
"ProgramData".Trigger		Bool	<input type="checkbox"/> FALSE	
"ProgramData".TotalTriggerCount		DEC+-	7	
"ProgramData".Pass		Bool	<input type="checkbox"/> FALSE	
"ProgramData".TotalPassedCount		DEC+-	4	
"ProgramData".Fail		Bool	<input checked="" type="checkbox"/> TRUE	
"ProgramData".TotalFailedCount		DEC+-	3	
"ProgramData".TotalMissCount		DEC+-	7	
"ProgramData".ResetCounters		Bool	<input type="checkbox"/> FALSE	
"UserData".NumberOfParts	%DB2.DBWD	Hex	16#0000	
"ProgramData".ResetError		Bool	<input type="checkbox"/> FALSE	
"ProgramData".CommandResult.Fail		Bool	<input type="checkbox"/> FALSE	
"ProgramData".CommandResult.No Job In Slot"		Bool	<input type="checkbox"/> FALSE	
"ProgramData".CommandResult.UnknownCmd		Bool	<input type="checkbox"/> FALSE	
"ProgramData".CommandResult.Success		Bool	<input type="checkbox"/> FALSE	

Rufen Sie die „Beobachtungstabelle\_1“-Elemente **ChangeJob** (Job ändern) und **New\_JobSlot** (Neuer Job-Slot) auf. Klicken Sie dann neben **New\_JobSlot** in die Spalte **Modify value** (Wert bearbeiten), geben Sie den Wert „2“ ein und drücken Sie die **Eingabetaste**. Klicken Sie in der Zeile „ChangeJob“ auf den Wert in der Spalte „Modify value“ und geben Sie **TRUE** (WAHR) ein. Klicken Sie oben in der Symbolleiste auf die Schaltfläche mit dem Blitzsymbol und der Zahl **1**. Dadurch wird der Job-Slot als Slot 2 festgelegt. Klicken Sie dann erneut auf das Blitzsymbol.

Name	Address	Display format	Monitor value	Modify value	Comment
"ProgramData".ChangeJob		Bool	<input type="checkbox"/> FALSE	<input checked="" type="checkbox"/> TRUE	
"ProgramData".New_JobSlot		DEC+-	1	2	
"ProgramData".Get_CurrentJobslot		Bool	<input type="checkbox"/> FALSE		
"ProgramData".Current_JobSlot		DEC+-	2		
"ProgramData".Trigger		Bool	<input type="checkbox"/> FALSE		
"ProgramData".TotalTriggerCount		DEC+-	0		
"ProgramData".Pass		Bool	<input type="checkbox"/> FALSE		

Statt des aktuellen Jobs wird daraufhin der Job in Slot 2 geöffnet. Um sicherzustellen, dass dadurch alle Änderungen gelöscht werden, die Sie in der Spalte „Modify value“ (Wert bearbeiten) vorgenommen haben, klicken Sie in der Zeile **Get\_CurrentJobSlot** (Aktuellen Job-Slot abrufen) auf den Wert in der Spalte „Modify value“. Geben Sie **TRUE** (WAHR) ein und klicken Sie dann erneut auf das Blitzsymbol.

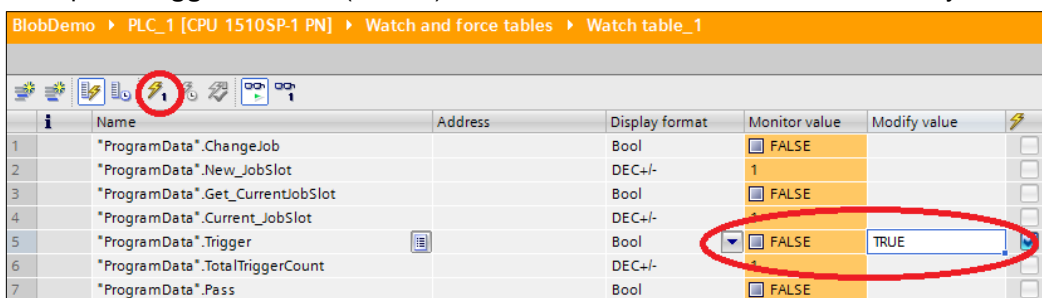


Name	Address	Display format	Monitor value	Modify value
*ProgramData*.ChangeJob		Bool	<input type="checkbox"/> FALSE	
*ProgramData*.New_JobSlot		DEC+/-	1	
*ProgramData*.Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE	TRUE
*ProgramData*.Current_JobSlot		DEC+/-	1	

Der aktuelle Job erscheint nun rechts neben **Current\_JobSlot** (Aktueller Job-Slot). Sie können diese Vorgehensweise wiederholen, indem Sie wieder zu Slot 1 wechseln und überprüfen, ob als aktueller Job-Slot tatsächlich **1** angezeigt wird.

### Inspektion auslösen und Zählerstände zurücksetzen

Löschen Sie alle Änderungen, die Sie in der Spalte „Modify value“ (Wert bearbeiten) vorgenommen haben. Falls Sie die Inspektion mitverfolgen möchten, vergewissern Sie sich, dass die CS50-Software aktiv und mit dem Gerät verbunden ist. Überprüfen Sie auch, ob der Job ausgeführt wird. Falls Sie die CS50-Software nicht ausführen möchten, können Sie die Trigger anhand der LEDs am Gerät überwachen. Ändern Sie den Wert in der Spalte **Trigger** in **TRUE** (WAHR) und klicken Sie dann erneut auf das Blitzsymbol.



BlobDemo ▶ PLC\_1 [CPU 1510SP-1 PN] ▶ Watch and force tables ▶ Watch table\_1

Name	Address	Display format	Monitor value	Modify value
1 *ProgramData*.ChangeJob		Bool	<input type="checkbox"/> FALSE	
2 *ProgramData*.New_JobSlot		DEC+/-	1	
3 *ProgramData*.Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE	
4 *ProgramData*.Current_JobSlot		DEC+/-	1	
5 *ProgramData*.Trigger		Bool	<input type="checkbox"/> FALSE	TRUE
6 *ProgramData*.TotalTriggerCount		DEC+/-	1	
7 *ProgramData*.Pass		Bool	<input type="checkbox"/> FALSE	

Der Wert für **NumberOfParts** (Anzahl der Teile) ändert sich nun in **2** oder **5**. Die Booleschen Werte für „Pass“ (Erfolg) und „Fail“ (Fehlschlag) signalisieren, ob die Inspektion erfolgreich war oder nicht. Außerdem werden die entsprechenden Zählerstände erhöht. Jedes Mal, wenn Sie auf die Schaltfläche mit dem Blitzsymbol klicken, wird eine Inspektion gestartet und damit zum nächsten Bild gewechselt. Die Inspektionsergebnisse sind abwechselnd Erfolge und Fehlschläge.

Um die Zählerstände zurückzusetzen, löschen Sie alle Einstellungen in der Spalte „Modify value“ (Wert bearbeiten) und ändern Sie dann die entsprechende Einstellung in der Zeile **ResetCounters** (Zähler zurücksetzen) zu **TRUE** (WAHR). Klicken Sie anschließend wieder auf das Blitzsymbol. Alle im Bild unten blau markierten Zählerstände werden nun auf **0** zurückgesetzt.

Name	Address	Display format	Monitor value	Modify value	
*ProgramData*.ChangeJob		Bool	FALSE		
*ProgramData*.New_JobSlot		DEC+/-	1		
*ProgramData*.Get_CurrentJobSlot		Bool	FALSE		
*ProgramData*.Current_JobSlot		DEC+/-	1		
*ProgramData*.Trigger		Bool	FALSE		
*ProgramData*.TotalTriggerCount		DEC+/-	0		
*ProgramData*.Pass		Bool	TRUE		
*ProgramData*.TotalPassedCount		DEC+/-	0		
*ProgramData*.Fail		Bool	FALSE		
*ProgramData*.TotalFailedCount		DEC+/-	0		
*ProgramData*.TotalInspCount		DEC+/-	0		
*ProgramData*.ResetCounters		Bool	FALSE	TRUE	
*UserData*.NumberOfParts	%DB2.DBW0	Hex	16#0002		
*ProgramData*.ResetError		Bool	FALSE		

Wenn Sie diese Einstellung erneut löschen, den Wert für „Trigger“ auf „TRUE“ (WAHR) setzen und noch einmal auf das Blitzsymbol klicken, starten sowohl der SPS-Zähler als auch der Zähler der CS50-Software bei **1**.

### Fehler zurücksetzen

Sollte einer der unten dargestellten „TRUE“-Werte einen Fehler zurückgeben, können Sie diesen Fehler beheben, indem Sie in der Zeile **ResetError** (Fehler zurücksetzen) „TRUE“ (WAHR) eingeben und dann wieder auf die Schaltfläche mit dem Blitzsymbol klicken. Setzen Sie den Wert dann auf **FALSE** (FALSCH) zurück und klicken Sie erneut auf die Schaltfläche. Sämtliche Fehler in den unten dargestellten Werten müssten nun beseitigt werden.

*ProgramData*.CommandResult.Fail		Bool	FALSE
*ProgramData*.CommandResult.*No Job In Slot*		Bool	FALSE
*ProgramData*.CommandResult.UnknownCmd		Bool	FALSE
*ProgramData*.CommandResult.Success		Bool	FALSE





# Demo PROFINET I/O- Kreislokalisierung mit TIA- Portal V13 und CS50- Sensor

Dieses Kapitel beschreibt die Anwendung des di-soric-SPS-Democodes in einem Vision-Job mit Testobjekt. Die PROFINET I/O-Demodateien befinden sich im Installationsordner der CS50-Software: **C:\di-soric\Tutorials and Samples\CS50 Sensor\PROFINET demo\TIAPortal\_Demos\ Circle\_LocateDemo**. Öffnen Sie die Datei **Circle\_LocatorDemo.avp** in der CS50-Software und laden Sie sie auf die Kamera herunter.

#### Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für PROFINET I/O aktiviert werden. Informationen zum Aktivieren und Umschalten von Kommunikationsprotokollen finden Sie in **Kapitel 1 Industrieprotokolle aktivieren**.
- Bei CS50-Software-Jobs wird die PROFINET I/O-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert.

**Wichtig:** Mit PROFINET I/O können die IP-Adresse und die Subnetzmaske der Kamera von der SPS automatisch zugewiesen werden. Dazu darf dem CS50-Sensor jedoch beim Neustart so lange keine IP-Adresse zugewiesen werden, bis die SPS in den Betriebsmodus wechselt. In der Zwischenzeit wird der CS50-Sensor im Netzwerk der CS50-Software nicht angezeigt.

## Überblick

In dieser Demonstration erfahren Sie, wie man einen gespeicherten Job auf die Kamera lädt und über PROFINET I/O eine Verbindung zu einer Siemens ET200SP-CPU-SPS herstellt. Außerdem zeigen wir Ihnen anhand einiger Beispiele, wie man Programme ausführt, die über eine Schnittstelle mit der Kamera verbunden sind. Während der Prüfung der PROFINET I/O-Kapazitäten werden folgende Schritte ausgeführt:

- **Start mit der CS50-Software**

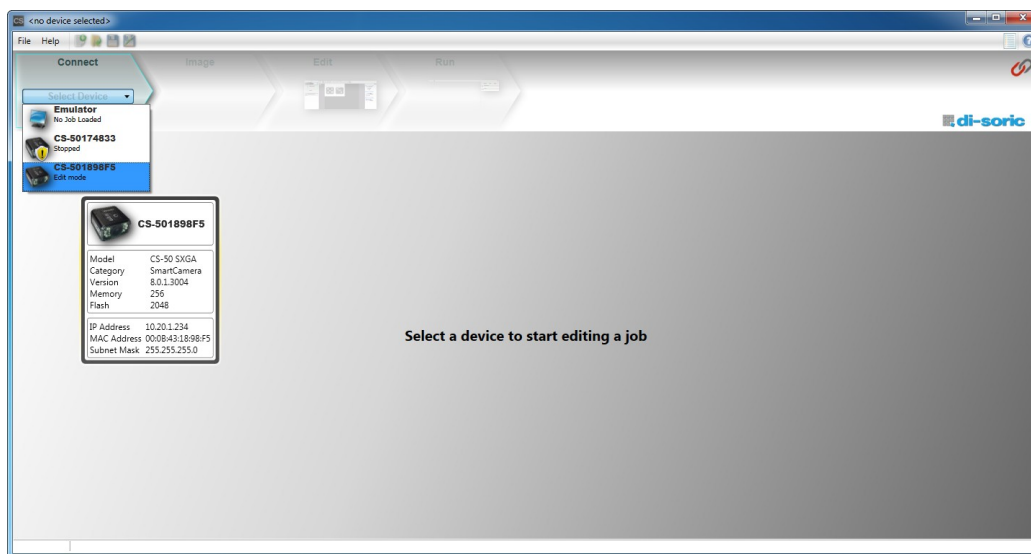
Öffnen Sie einen Demo-Vision-Job in der CS50-Software. Bevor die Kamera mit der SPS verbunden wird, können Sie mithilfe der Option „Try Out“ (Ausprobieren) eine Vorschau aufrufen.

- **Vorbereitung der SPS**

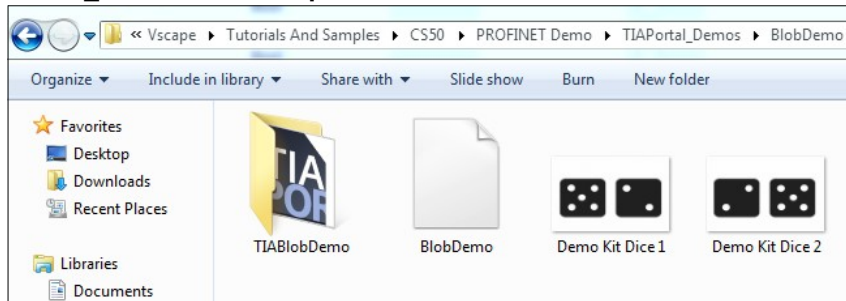
Integrieren Sie die Kamera in die SPS-Umgebung mit der Software TIA Portal und der GSD-Datei.

- **Start der Kamera**

Lösen Sie die Kamera, während sie online ist, mit der SPS aus und beobachten Sie die Veränderungen im Inspektionsstatus, während die SPS die Parameter des Vision-Jobs neu konfiguriert. Vor dem Start der CS50-Software sollten Sie sicherstellen, dass entweder die Kamera mit der SPS verbunden ist oder sich SPS und Kamera beide im gleichen Netzwerk befinden. Vergewissern Sie sich, dass der PC, die SPS und die Kamera dieselbe Netzwerkkategorie und die entsprechenden Subnetzadressen aufweisen. Starten Sie die CS50-Software und wählen Sie die Kamera aus.



Klicken Sie in der Ansicht „Image“ (Bild) auf **Load a Job** (Job laden). Suchen Sie dann den Ordner **C:\di-soric\Vscape\Tutorials And Samples\CS50 Sensor\PROFINET demo\TIAPortal\_Demos\Circle\_LocatorDemo**. Wählen Sie die Datei **Circle\_LocatorDemo.avp** aus.



Der Demo-Job enthält zwei Tools: **Locate Shape** (Form lokalisieren) und **Circle** (Kreis).

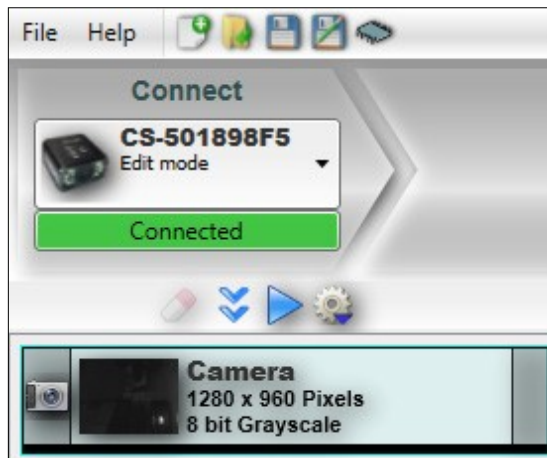


Hier eine Übersicht der PROFINET I/O-Struktur:

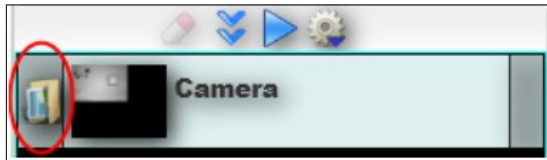
Tool Result	PLC IN
Instance Point1	"UserData".Instance1Point1(%DB2.DBD2)
Instance Point2	"UserData".Instance1Point2(%DB2.DBD6)
Instance Point3	"UserData".InstancePoint3(%DB2.DBD10)
Status	"UserData".Status[0](%DB2.DBX0.0)
RadiumDatum	"UserData".RadiusDatum(%DB2.DBD14)

Diese Daten werden in zyklischen Intervallen zwischen Kamera und SPS ausgetauscht.

Sobald der Job geladen ist, verknüpfen Sie die vorgeschichteten Bilder auf dem lokalen PC. Klicken Sie dazu auf der Schaltfläche **Camera** (Kamera) auf das Symbol ganz links, um ein Bild auszuwählen und zu laden.



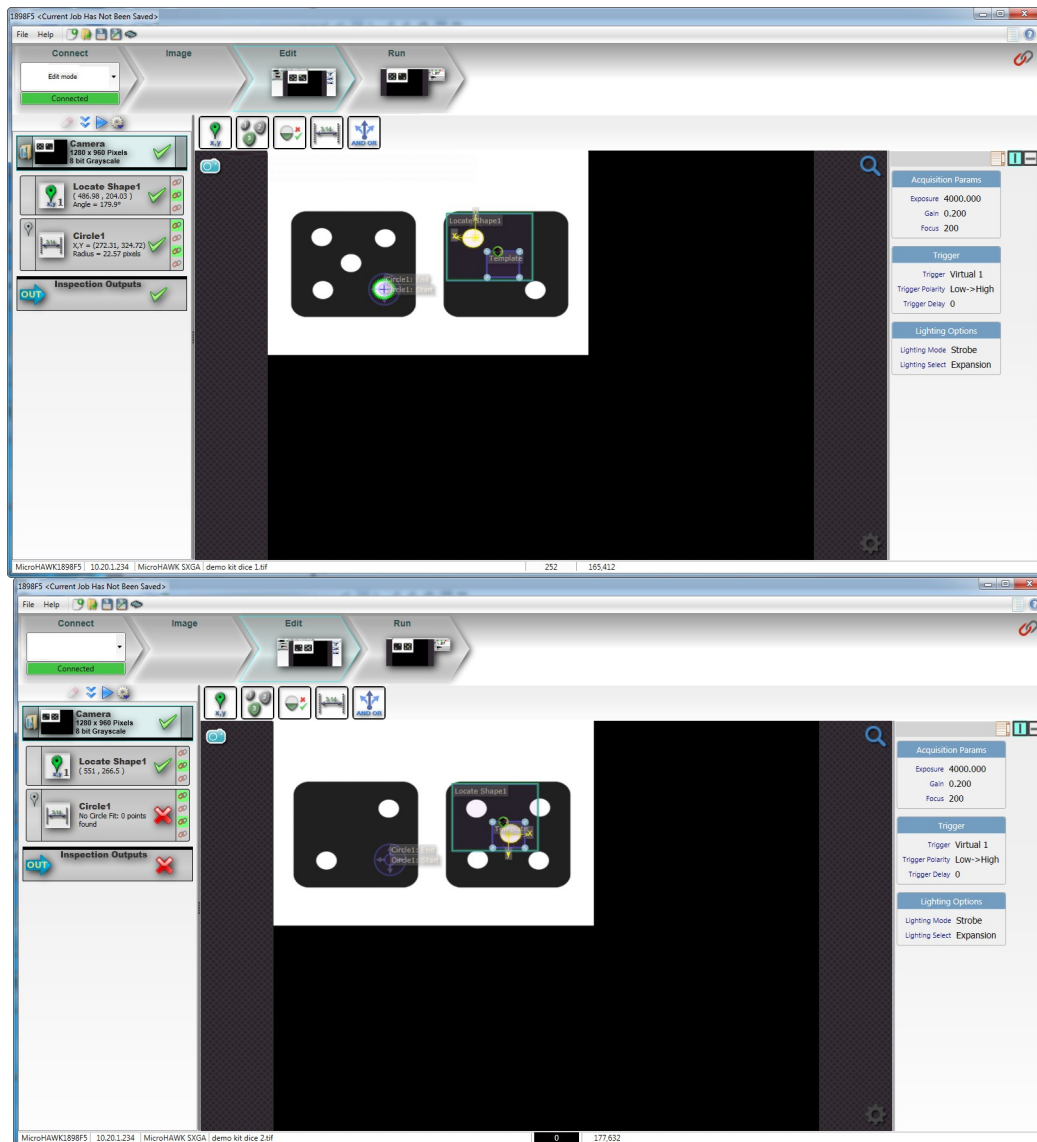
Daraufhin öffnet sich ein Fenster, in dem Sie eine Datei auswählen können. Navigieren Sie zu demselben Ordner, in dem der Demo-Job (PROFINET I/O Demo) geladen wurde. Sobald die Bilder gefunden wurden, ändert sich das Symbol: Anstelle der Kamera erscheint nun ein Ordner.



Klicken Sie auf das Symbol **Try Job Once** (Job einmal testen), um den gesamten Job mit dem geladenen Bild durchlaufen zu lassen.



Die Datei enthält zwei Bilder. Eines davon führt bei der Inspektion zu einem Erfolg, das andere zu einem Fehlschlag.



Klicken Sie nun oben in der Symbolleiste auf **Run** (Ausführen). Dadurch wird der Job auf die Kamera heruntergeladen. Der Job kann nun ausgeführt und getestet werden. Damit die Demo mit Job-Änderung ausgeführt werden kann, muss dieser Job allerdings in **Slot 1** geladen werden. Klicken Sie oben in der Symbolleiste wieder auf die Schaltfläche zum Öffnen der Ansicht **Edit** (Bearbeiten).

Klicken Sie auf das Slot-Symbol und wählen Sie „Slot 1“ oder **New Slot** (Neuer Slot) aus, falls derzeit keine Slots belegt sind.



Der Job und die Bilder werden nun im Flash-Speicher der Kamera gespeichert.

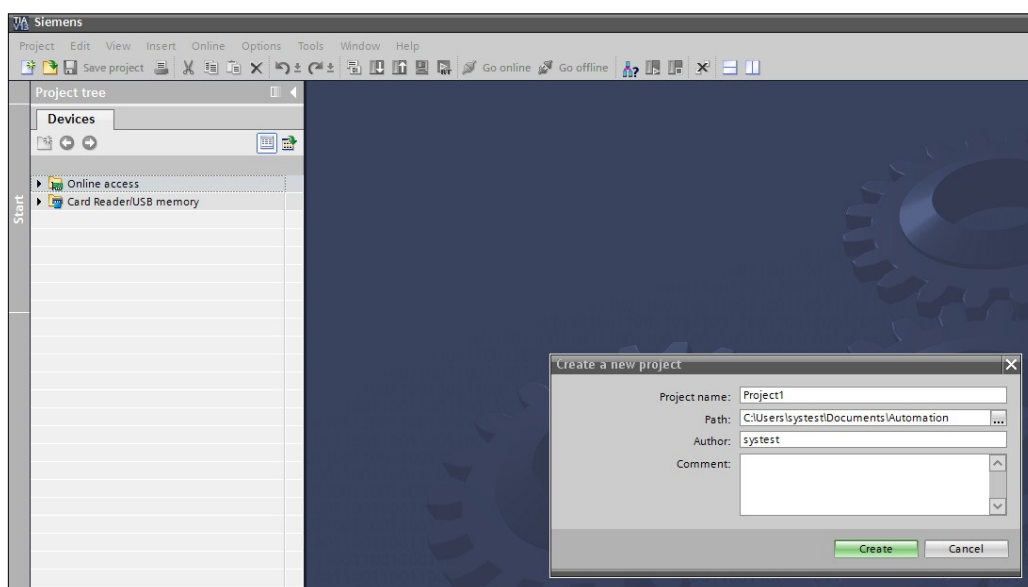
## Einrichtung von TIA Portal Version 13

Die Anleitung in diesem Kapitel wurde auf der Basis der Software Siemens TIA Portal (Version 13) und einer ET200SP-CPU (Katalognummer 6ES7 510-1DJ01-0AB0, Hardwareversion 2, Firmwareversion 1.8.2) erstellt. In diesem Beispiel wird die Vorgehensweise für den CS50-Sensor beschrieben. Die Vorgehensweise für andere Geräte ist jedoch ähnlich.

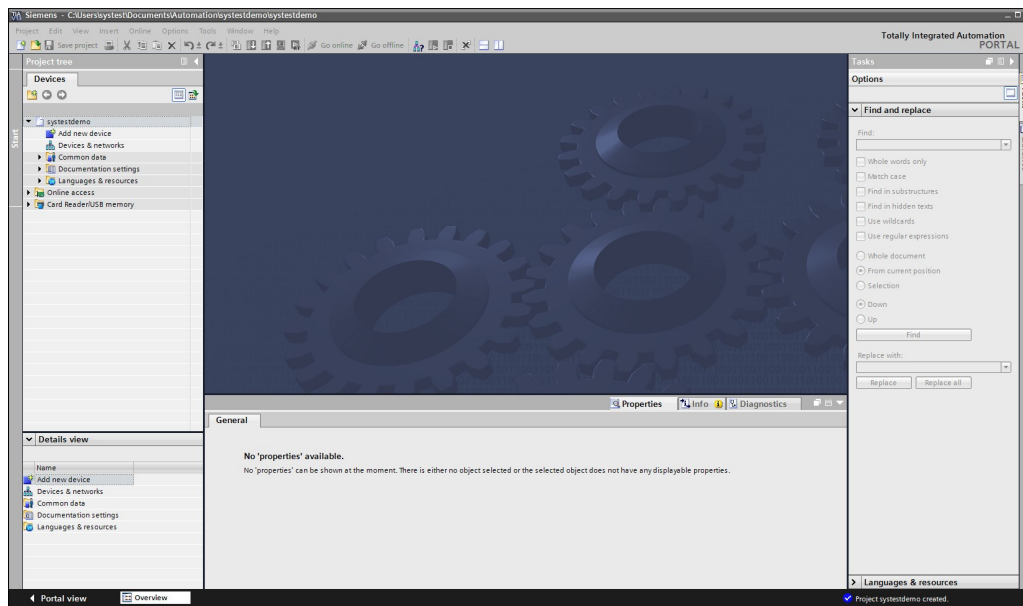
Starten Sie das TIA Portal vom Desktop aus.



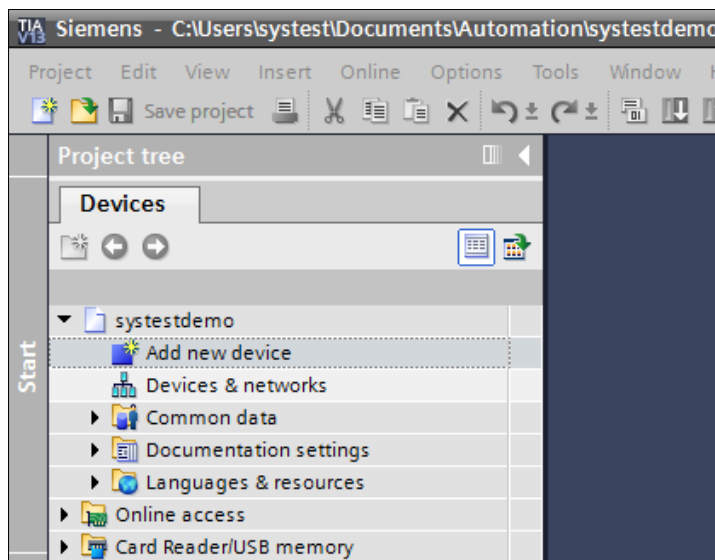
Erstellen Sie über das Menü **Projekt > Neu** ein neues Projekt. Der Standardname lautet „Project1“, wie unten dargestellt. Geben Sie die nötigen Informationen an und klicken Sie dann auf **Erstellen**.



Dieser Screenshot zeigt den Hauptbildschirm, der sich öffnet, wenn Sie ein Projekt öffnen.

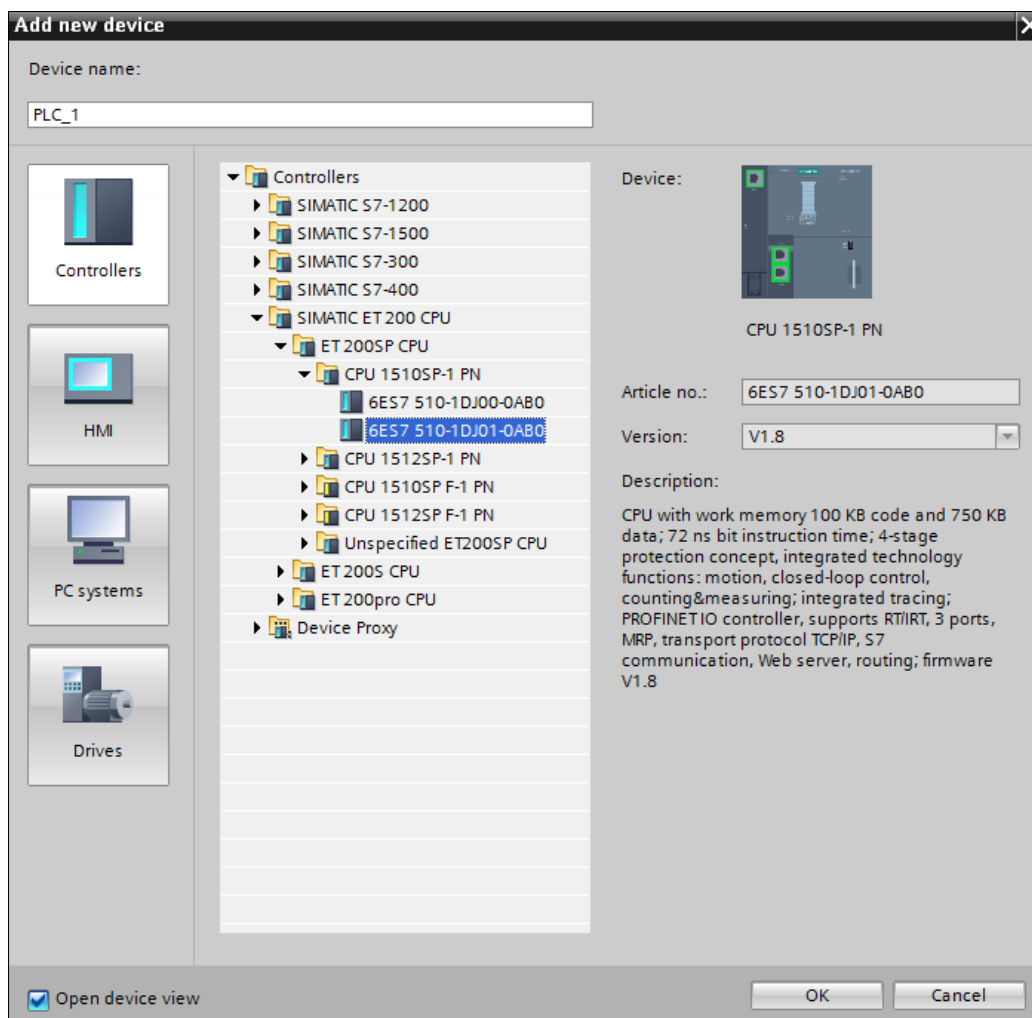


Doppelklicken Sie auf **Neues Gerät hinzufügen**.

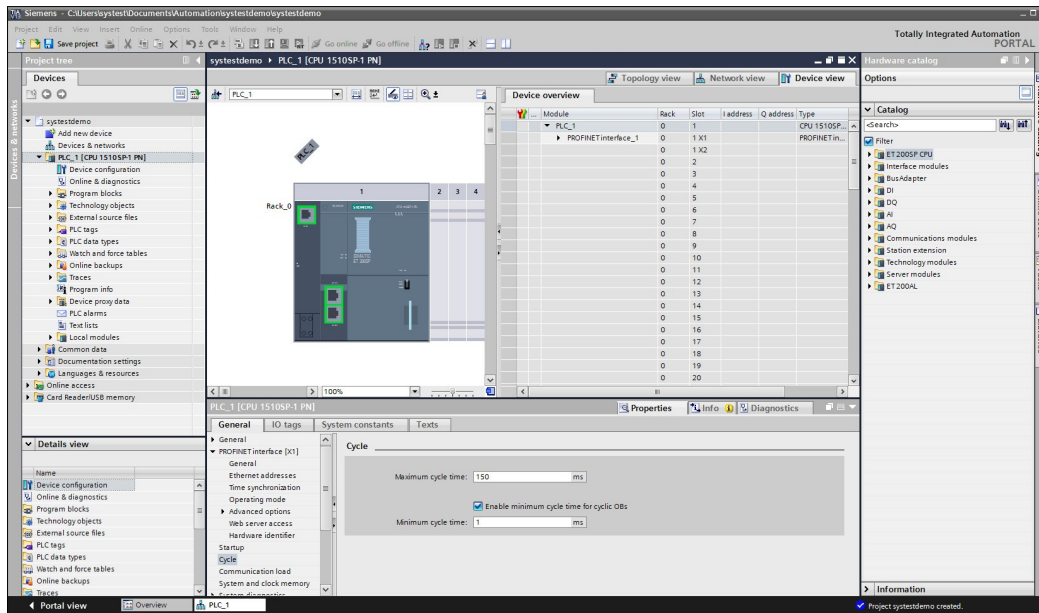




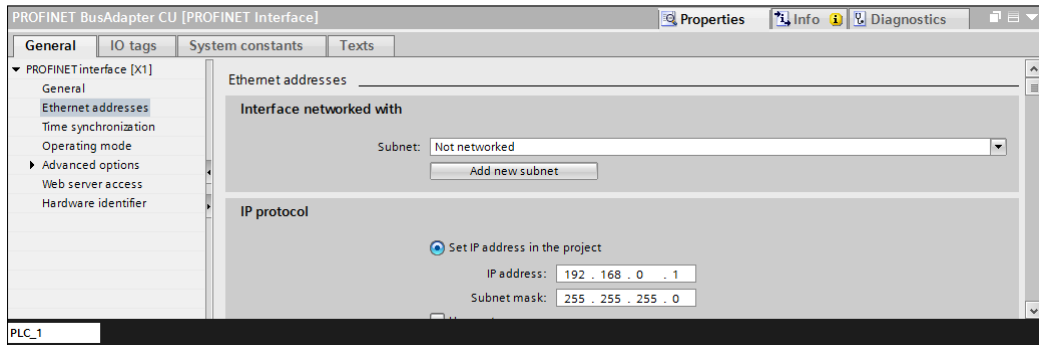
Öffnen Sie den Pfeil **SIMATIC ET200 CPU** und dann den Pfeil **ET200SP CPU**. Öffnen Sie **CPU 1510SP-1 PN** und klicken Sie anschließend auf **6ES7 510-1DJ01-0AB0**. Daraufhin öffnet sich der unten abgebildete Bildschirm. Klicken Sie auf **OK**.



Sie sehen nun folgenden Bildschirm.

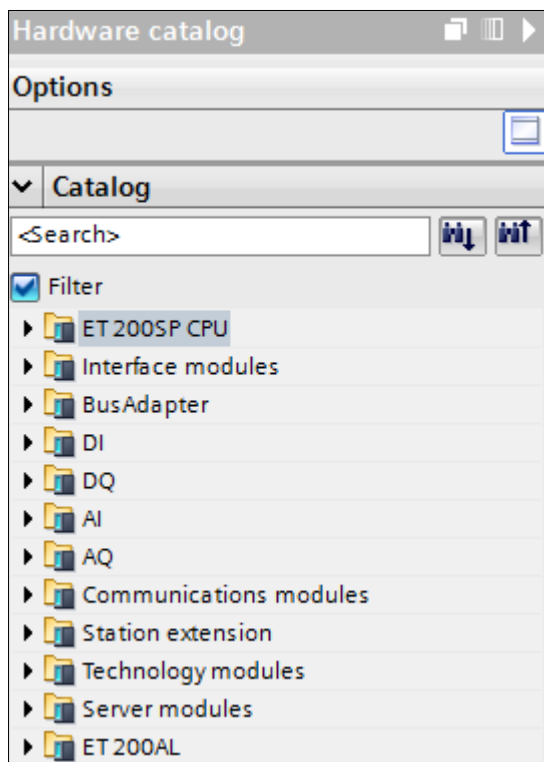


Klicken Sie im SPS-Bild auf das untere grüne Kästchen. Daraufhin sehen Sie im unteren Bereich Folgendes:



Klicken Sie auf **Neues Subnetz hinzufügen**. Im Feld „Subnetz“ wird nun „PN/IE\_1“ angezeigt. Geben Sie nun in das Feld „IP-Adresse“ die IP-Adresse ein, die für dieses Gerät verwendet werden soll. Denken Sie daran, regelmäßig auf **Projekt speichern** zu klicken, damit Ihre Änderungen nicht verloren gehen.

Auf der rechten Seite des Bildschirms sehen Sie nun dies:

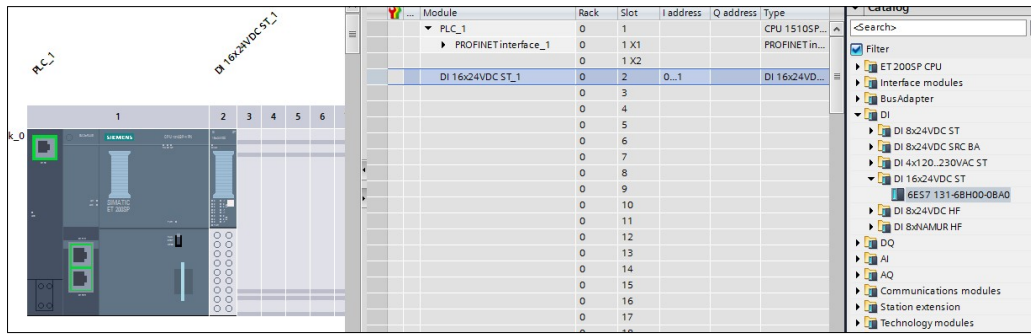


Unser Testgerät umfasst folgende Module:

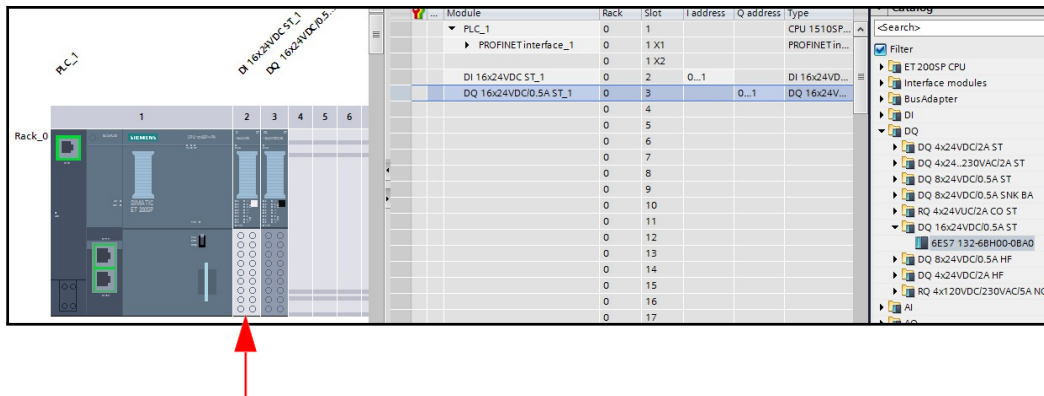
1. DI 16x24VDC ST, Artikelnummer: 6ES7 131-6BH00-0BA0
2. DQ 16x24VDC ST, Artikelnummer: 6ES7 132-6BH00-0BA0

Um diese Module hinzuzufügen, klicken Sie zunächst auf den Pfeil neben **DI** und öffnen Sie dann über die entsprechenden Pfeile die Untermenüs, bis Sie die Artikelnummer von Gerät 1 gefunden haben. Klicken Sie darauf und halten Sie die linke Maustaste gedrückt. Ziehen Sie die Nummer nun nach links in Slot 2.

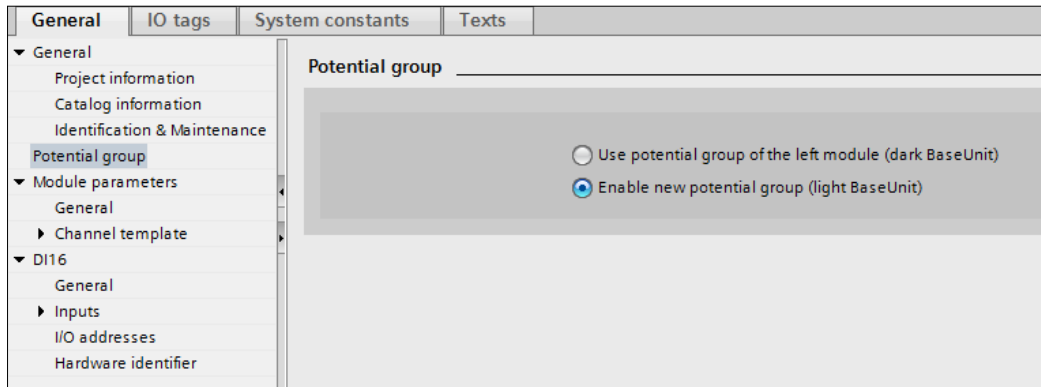
Sie müssten nun Folgendes sehen:



Wiederholen Sie diese Schritte für Gerät 2, indem Sie auf den Pfeil neben **DQ** klicken und das entsprechende Untermenü öffnen. Sie müssten nun Folgendes sehen:

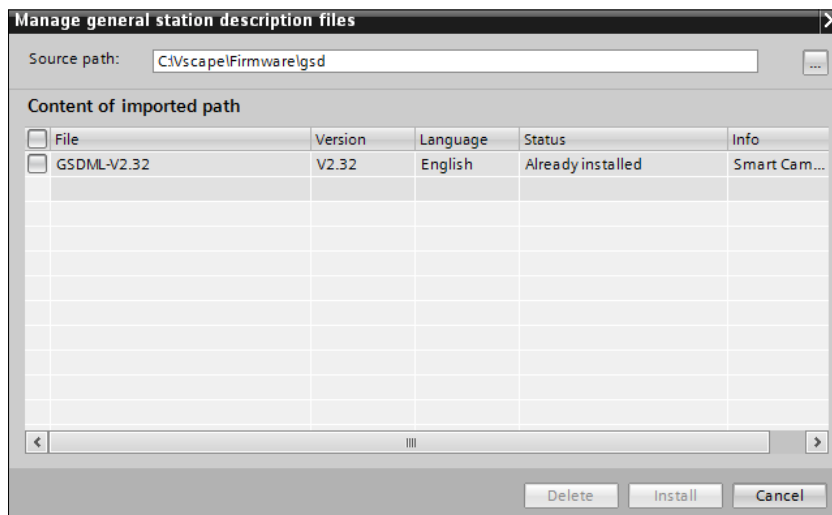


Klicken Sie auf das SPS-Bild in Slot 2. Sie sehen nun folgenden Bildschirm:



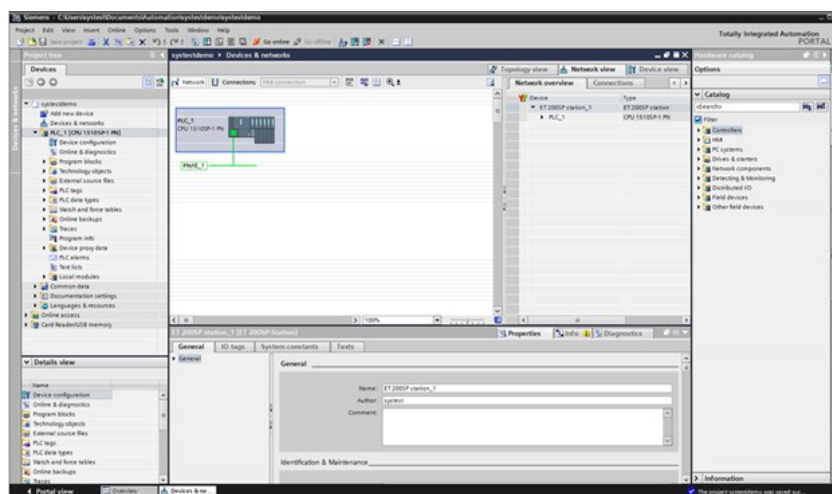
Vergewissern Sie sich, dass die Option **Neue Potenzialgruppe ermöglichen (Helle BaseUnit)** ausgewählt ist. Wiederholen Sie diese Schritte für Slot 3. Speichern Sie das Projekt.

Klicken Sie als Nächstes auf **Extras > Gerätebeschreibungsdateien (GSD) verwalten**.

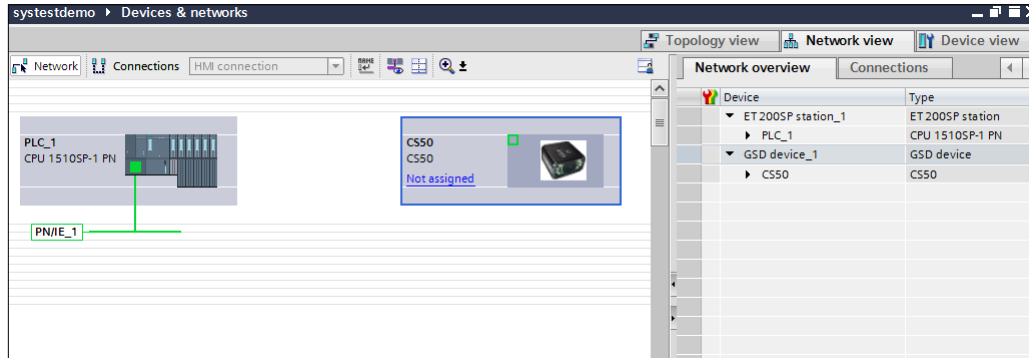


Navigieren Sie zu folgendem Dateipfad: **C:\di-soric\Vscapel\Firmware\gdsd\CS50 Sensor**. Wählen Sie die entsprechende GSD-Datei aus, indem Sie das Kästchen neben dem Namen aktivieren, und klicken Sie dann auf **Installieren**.

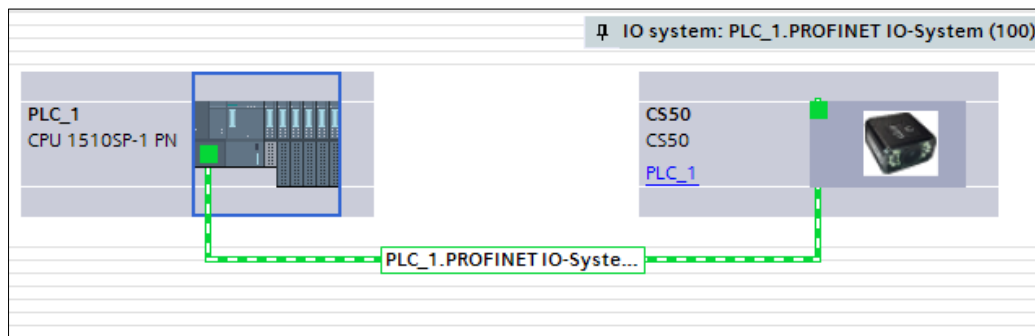
Klicken Sie auf die Registerkarte **Netzwerkansicht**:



Klicken Sie im rechten Bereich auf **Hardware-Katalog**. Navigieren Sie mithilfe der Pfeile zu folgendem Bereich: **Andere Feldgeräte > PROFINET IO > General (Allgemein) > di-soric > Smart Camera (Smartkamera) > CS50 Sensor**. Klicken Sie auf den CS50-Sensor und halten Sie die linke Maustaste gedrückt. Ziehen Sie den Sensor nun nach links in den Bereich „Geräte & Netzwerke“. Sie sehen nun Folgendes:



Klicken Sie im Bild des CS50-Sensors auf den Link **Nicht zugeordnet** und dann auf **PLC\_1.PROFINET IO-System**. Dadurch wird der CS50-Sensor mit dem PROFINET-Netzwerk verbunden.



Doppelklicken Sie auf das Bild des CS50-Sensors. Im unteren Bereich sehen Sie nun Folgendes:

The screenshot shows a software interface for configuring a CS50 sensor. The interface has a top navigation bar with tabs: 'General', 'IO tags', 'System constants', and 'Texts'. The 'General' tab is selected. On the left, there is a tree view under 'PROFINET interface [X1]' with sub-items: 'General', 'Ethernet addresses', 'Advanced options', 'Hardware identifier', 'Identification & Maintenance', and 'Hardware identifier'. The main area is titled 'General' and contains the following fields:

- Name: CS50
- Author: systest
- Comment: (empty text area)
- Rack: 0
- Slot: 0

Geben Sie in das Feld „Name:“ den Namen des Geräts ein. Klicken Sie dann links auf „Ethernet-Adressen“ und geben Sie die IP-Adresse der Kamera ein. Dadurch kann die SPS die IP-Adresse der Kamera festlegen.

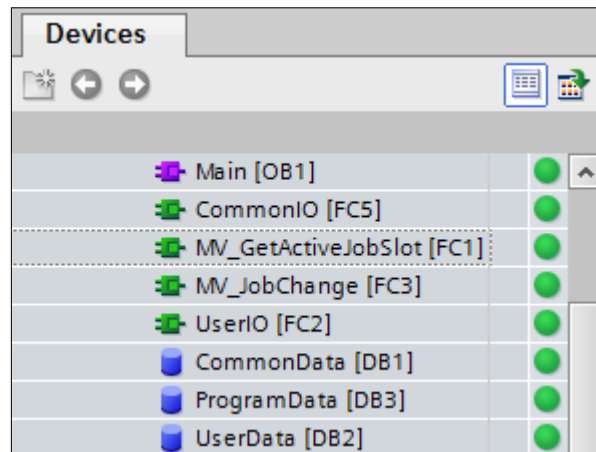
## Beispielprogramm importieren

Öffnen Sie das Beispielprogramm im TIA Portal über das Menü **Projekt > Öffnen**.

Klicken Sie auf **Durchsuchen** und suchen Sie das Programm

**Circle\_LocatorDemo.ap13**. Es befindet sich im Ordner **C:\di-soric\Vscape\Tutorials and Samples\CS50 Sensor\PROFINET**

**Demo\TIAPortal\_Demos\Circle\_LocatorDemo\TIACircle\_LocatorDemo**. Öffnen Sie auf der linken Seite auf der Registerkarte **Geräte** den Pfeil neben „PLC\_1[CPU 1510SP-1 PN]“. Klicken Sie auf den Pfeil neben „Programmblöcke“ und wählen Sie dann alle Blöcke wie unten dargestellt aus.



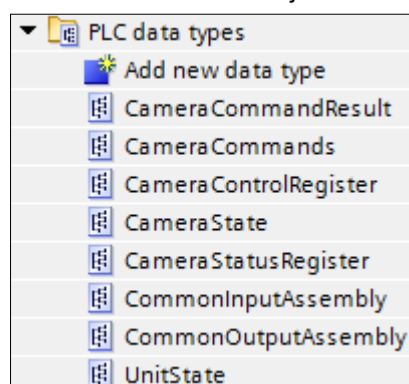


Klicken Sie mit der rechten Maustaste und wählen Sie „Kopieren“ aus. Klappen Sie nun für Ihr neues Projekt die gleichen Pfeile wie oben beschrieben aus. Klicken Sie dann mit der rechten Maustaste auf „Programmblöcke“ und dann auf „Einfügen“. Denken Sie daran, Ihre Änderungen zu speichern! Es kann sein, dass Sie zuvor im neuen Projekt den Programmbaustein „Main [OB1]“ löschen müssen.

Klicken Sie nun im linken Fensterabschnitt des Projekts „Circle\_LocatorDemo“ auf „SPS-Datentypen“. Klicken Sie dann mit der rechten Maustaste, wählen Sie „Kopieren“ aus und fügen Sie die kopierten Inhalte in die SPS-Datentypen des neuen Projekts ein.

Klicken Sie links im Projekt „Circle\_LocatorDemo“ auf „Beobachtungs- und Forcetabellen“. Klicken Sie mit der rechten Maustaste auf „Beobachtungstabelle\_1“ und wählen Sie dann „Kopieren“ aus. Fügen Sie die Tabelle in „Beobachtungs- und Forcetabellen“ des neuen Projekts ein. Denken Sie daran, das Projekt zu speichern!

Die Daten im neuen Projekt müssten nun so aussehen:



### Mapping von IO überprüfen

Je nachdem, in welcher Reihenfolge die SPS Elemente laden kann, müssen möglicherweise die E/A-Adressen im Block geändert werden, damit sie mit denen in der Geräteansicht unter „Geräteübersicht“ übereinstimmen.

Doppelklicken Sie auf das Bild des CS50-Sensors. Daraufhin wird rechts die Geräteübersicht angezeigt.

Der folgende Screenshot dient lediglich der Veranschaulichung. Die Geräteübersicht kann bei Ihnen anders aussehen.

Device overview						
Module	Rack	Slot	I address	Q address	Type	
CS501898f5	0	0			CS50	
Interface	0	0 X1			CS50	
Status_1	0	1	2...3		Status	
Control_1	0	2		2...3	Control	
Echo In_1	0	3	256...257		Echo In	
Echo Out_1	0	4		256...257	Echo Out	
Cmd Code Rslt_1	0	5	258...261		Cmd Code Rslt	
Cmd Code_1	0	6		258...261	Cmd Code	
Cmd Ret_1	0	7	262...265		Cmd Ret	
Cmd Arg_1	0	8		262...265	Cmd Arg	
State_1	0	9	266		State	
	0	10				
	0	11				
	0	12				
	0	13				
	0	14				
	0	15				
	0	16				

Öffnen Sie ganz rechts im „Hardware-Katalog“ den Abschnitt **Modul**. Sie sehen nun Folgendes:

- Module
  - Boolean In
  - Boolean Out
  - Float In
  - Float Out
  - Int In
  - Int Out
  - Long In
  - Long Out
  - Long String In
  - Long String Out
  - Short String In
  - Short String Out
  - VIO In
  - VIO Out

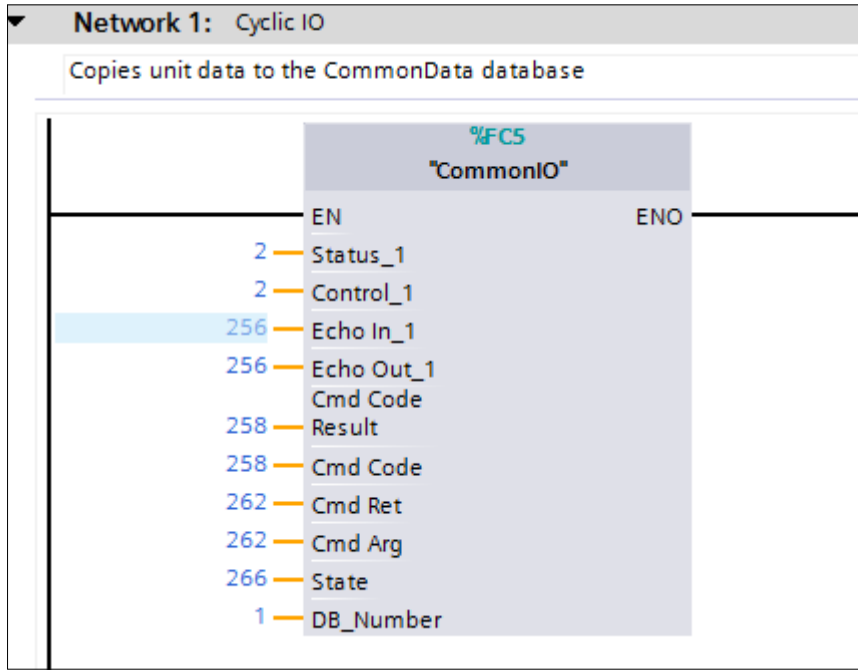
Ziehen Sie nun „Boolean In\_1“ und „Float In\_1“ manuell mit der Maus in die Geräteübersicht. Wenn Sie auf „Boolean In\_1“ klicken, zeigt TIA den entsprechenden Ziel-Slot an. Wiederholen Sie diesen Schritt für „Float In\_1“. „Int In“ kann nur in den angezeigten Slot gezogen werden; alles andere wird von TIA unterbunden.

Ihre Geräteübersicht müsste nun folgendermaßen aussehen. Die hier dargestellten Adressen sind lediglich exemplarisch und stimmen nicht mit den Adressen in Ihrer Übersicht überein.

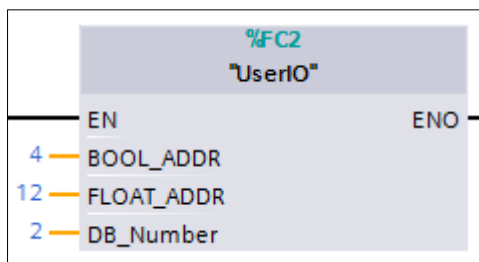
Device overview							
	...	Module	Rack	Slot	I address	Q address	Type
		▼ CS501898f5	0	0			CS50
		▶ Interface	0	0 X1			CS50
		Status_1	0	1	2...3		Status
		Control_1	0	2		2...3	Control
		Echo In_1	0	3	256...257		Echo In
		Echo Out_1	0	4		256...257	Echo Out
		Cmd Code Rslt_1	0	5	258...261		Cmd Code Rslt
		Cmd Code_1	0	6		258...261	Cmd Code
		Cmd Ret_1	0	7	262...265		Cmd Ret
		Cmd Arg_1	0	8		262...265	Cmd Arg
		State_1	0	9	266		State
			0	10			
			0	11			
			0	12			
			0	13			
			0	14			
		Int In_1	0	15	4...23		Int In
			0	16			

Als Nächstes müssen Sie die SPS-Programmblöcke an die folgende Ansicht anpassen. Klappen Sie dazu links unter „Geräte“ den Pfeil neben „Programmblöcke“ aus und doppelklicken Sie auf „OB1 [OB1]“. Dadurch wird in der Mitte ein neuer Bereich geöffnet. Scrollen Sie in diesem Bereich nach unten zu Netzwerk 1.

Gleichen Sie hier die Zahlen an die in der Geräteübersicht angezeigten Werte an (siehe Beispiel unten). Lassen Sie den Wert „DB\_Nummer“ unverändert.



Gehen Sie für Netzwerk 4 genauso vor. Verwenden Sie dazu die Adressen für „Boolean In\_1“ und „Float In\_1“ aus der Geräteübersicht und fügen Sie sie unten bei „BOOL\_ADDR“ bzw. „FLOAT\_ADDR“ ein. Lassen Sie den Wert „DB\_Nummer“ unverändert.



Denken Sie daran, das Projekt zu speichern! Klicken Sie in der obersten Ebene auf **PLC\_1 [CPU 1510SP-1 PN]** und dann auf das Symbol zum Kompilieren.



Es dürfen KEINE Kompilierfehler auftreten. Sofern keine Fehler aufgetreten sind, klicken Sie auf das Download-Symbol.



Sie sehen nun folgenden Bildschirm:

**Extended download to device**

Configured access nodes of "PLC_1"					
Device	Device type	Slot	Type	Address	Subnet
PLC_1	CPU 1510SP-1 PN	1 X1	PN/IE	192.168.189.1	PN/IE_1

Type of the PG/PC interface: PN/IE

PG/PC interface: Intel(R) PRO/1000 MT Network Connection <3>

Connection to interface/subnet: PN/IE\_1

1st gateway:

Compatible devices in target subnet:  Show all compatible devices

Device	Device type	Type	Address	Target device
—	—	PN/IE	Access address	—

Flash LED

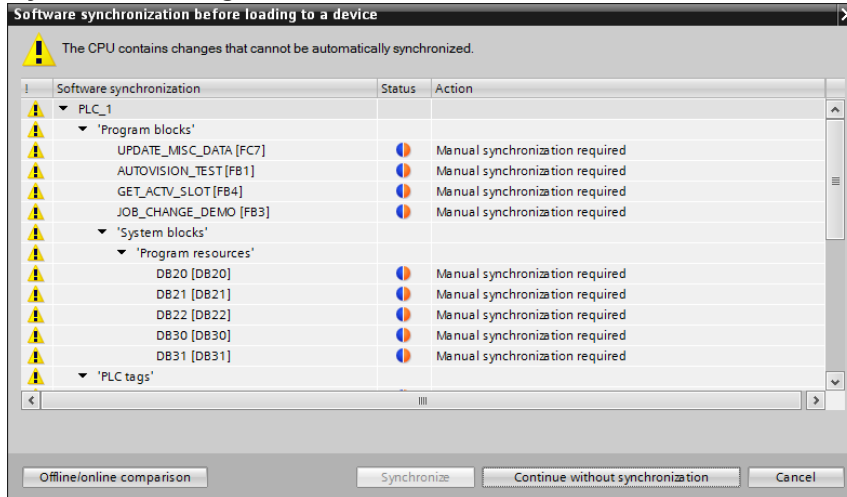
Start search

Online status information:

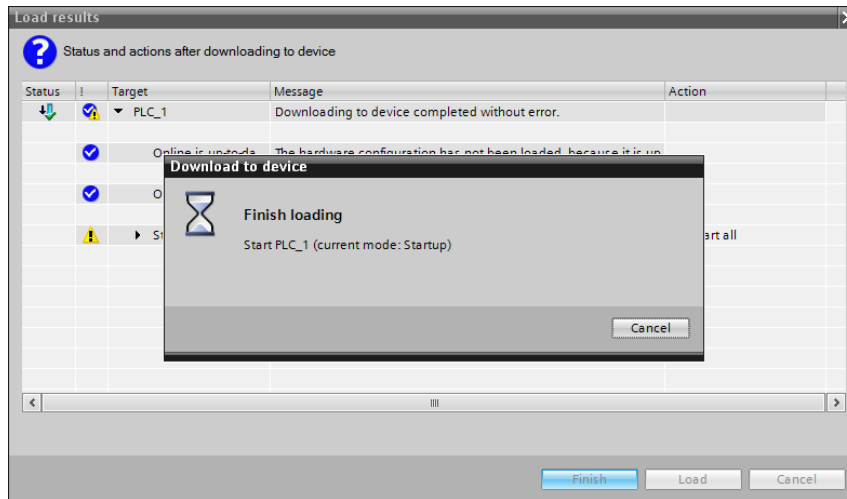
Display only error messages

Load
Cancel

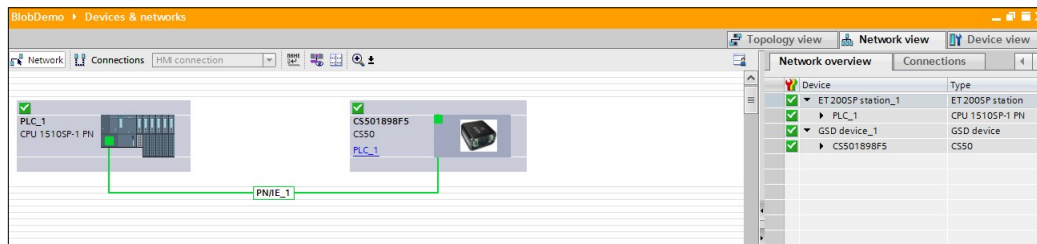
Wählen Sie Ihre PC-Netzwerkschnittstelle aus und ändern Sie die Verbindung zu Schnittstelle/Subnetz in **PN/IE\_1**. Klicken Sie auf **Suche starten**. Es müsste nun „SPS\_1“ angezeigt werden, zusammen mit der zugewiesenen Adresse. Klicken Sie auf **Laden**. Falls ein ähnlicher Bildschirm wie unten angezeigt wird, führen Sie für die SPS einen Speicher-Reset (MRES) durch und wiederholen Sie den Ladevorgang noch einmal. Sollte das Problem dann immer noch auftreten, klicken Sie auf **Ohne Synchronisierung fortfahren** und fahren Sie dann wie beschrieben fort.



Wenn das Herunterladen erfolgreich war, sehen Sie folgenden Bildschirm:



Alle LEDs der SPS sollten **GRÜN** sein. Klicken Sie auf **Online gehen**.  
Sie müssten nun Folgendes sehen:



## Fehlerbehebung

Falls nicht alle LEDs grün sind, gehen Sie offline und klicken Sie auf das Symbol für die verfügbaren Geräte.



Klicken Sie auf **Suche starten**. Die SPS und der CS50-Sensor müssten nun angezeigt werden.

Accessible devices

Type of the PG/PC interface:

PG/PC interface:

Accessible nodes of the selected interface:

Device	Device type	Type	Address	MAC address
win-sn7tk0uevh9	SIMATIC-PC	PN/IE	192.168.20.175	00-05-1B-80-2F-58
plc_1	CPU 1510SP-1 PN	PN/IE	192.168.189.1	28-63-36-48-AD-40
cs501898f5	Smart Camera	PN/IE	192.168.189.4	00-0B-43-18-6C-49

Flash LED

Online status information:

- Retrieving device information...
- Scan and information retrieval completed.
- Display only error messages

Falls der Name des CS50-Sensors falsch ist, wird keine Verbindung hergestellt. Der Standardname lautet „cs50“ plus die letzten drei Zeichenpaare der MAC-Adresse, z. B. „cs50186c49“. Kehren Sie zu den allgemeinen Einstellungen des CS50-Sensors in der Geräteübersicht zurück und ändern Sie ggf. den Namen. Wiederholen Sie das Kompilieren und Herunterladen. Versuchen Sie dann noch einmal, eine Verbindung herzustellen. Dies ist die häufigste Ursache für Fehler beim Herstellen einer Verbindung.



## Demo ausführen

Klicken Sie im linken Fensterbereich auf den Pfeil neben „Beobachtungs- und Forcetabellen“. Doppelklicken Sie auf „Beobachtungstabelle\_1“. Die Beobachtungstabelle wird daraufhin im zentralen Bereich des Fensters geöffnet. Klicken Sie auf das Symbol zum Überwachen der Variablen.

	Name	Address	Display format	Monitor v...	Modify value	
1	*ProgramData*.ChangeJob		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
2	*ProgramData*.New_JobSlot		DEC+/-	0		<input type="checkbox"/>
3	*ProgramData*.Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
4	*ProgramData*.Current_JobSlot		DEC+/-	2		<input type="checkbox"/>
5	*ProgramData*.Trigger		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
6	*ProgramData*.TotalTriggerCount		DEC+/-	0		<input type="checkbox"/>
7	*ProgramData*.Pass		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
8	*ProgramData*.TotalPassedCount		DEC+/-	0		<input type="checkbox"/>
9	*ProgramData*.Fail		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
10	*ProgramData*.TotalFailedCount		DEC+/-	0		<input type="checkbox"/>
11	*ProgramData*.TotalInspCount		DEC+/-	0		<input type="checkbox"/>
12	*ProgramData*.ResetCounters		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
13	*ProgramData*.ResetError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
14	*ProgramData*.CommandResult.Fail		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
15	*ProgramData*.CommandResult.No Job In Slot*		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
16	*ProgramData*.CommandResult.UnknownCmd		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
17	*ProgramData*.CommandResult.Success		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
18						<input type="checkbox"/>
19	*UserData*.Instance1Point1	%DB2.DB2	Floating-point nu...	470.7997		<input type="checkbox"/>
20	*UserData*.Instance1Point2	%DB2.DB6	Floating-point nu...	208.1492		<input type="checkbox"/>
21	*UserData*.Instance1Point3	%DB2.DB10	Floating-point nu...	-0.04094...		<input type="checkbox"/>
22	*UserData*.Status[0]	%DB2.DBX0.0	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
23	*UserData*.RadiusDatum	%DB2.DB14	Floating-point nu...	22.56597		<input type="checkbox"/>
24		<Add new>				<input type="checkbox"/>

Hier können nun folgende Vorgänge getestet werden:

Change Job	Switch jobs from one slot to another
Get Current Job Slot	Displays the slot of the currently running job
Trigger an Inspection	Causes the camera to cycle through its inspection.
Reset Counters	Resets the TotalTriggerCount, TotalPassedCount, TotalFailedCount, and TotalInspCount to 0, and also sends a command to the camera to reset its internal counters.
Reset an Error condition	If an error occurs during run time, this will reset it, so demo may be continued.

### Jobs ändern und aktuellen Job-Slot abrufen

Denken Sie daran, wie oben beschrieben einen Job in einen Slot des Kameraspeichers zu laden. Damit dieser Test korrekt durchgeführt werden kann, laden Sie einen weiteren Job in **Slot 2** der Kamera. Laden Sie beispielsweise den Job **BlobDemo**. Geben Sie die Gerätesteuerung frei, indem Sie auf die Registerkarte **Verbinden** klicken und dann, wie im Bild unten dargestellt, das Schlosssymbol zu **geschlossen** ändern. Klicken Sie anschließend erneut auf die Registerkarte **Ausführen**, damit der neue Job richtig ausgeführt wird.



Navigieren Sie dann zur **Beobachtungstabelle\_1** und klicken Sie dort auf die Schaltfläche mit der Brille und dem Symbol zum Abspielen. Damit beginnt nun die Datenüberwachung.

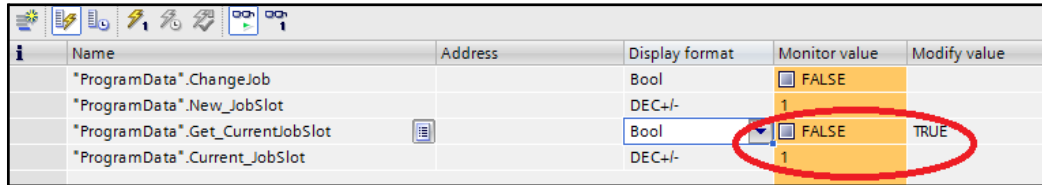
#	Name	Address	Display format	Monitor value	Modify value
1	"ProgramData".ChangeJob		Bool	<input type="checkbox"/> FALSE	
2	"ProgramData".New_JobSlot		DEC+/-	1	
3	"ProgramData".Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE	
4	"ProgramData".Current_JobSlot		DEC+/-	2	
5	"ProgramData".Trigger		Bool	<input type="checkbox"/> FALSE	
6	"ProgramData".TotalTriggerCount		DEC+/-	7	
7	"ProgramData".Pass		Bool	<input type="checkbox"/> FALSE	
8	"ProgramData".TotalPassedCount		DEC+/-	4	
9	"ProgramData".Fail		Bool	<input checked="" type="checkbox"/> TRUE	
10	"ProgramData".TotalFailedCount		DEC+/-	3	
11	"ProgramData".TotalInspCount		DEC+/-	7	
12	"ProgramData".ResetCounters		Bool	<input type="checkbox"/> FALSE	
13	"UserData".NumberOfParts	%DB2.DBW0	Hex	16#0000	
14	"ProgramData".ResetError		Bool	<input type="checkbox"/> FALSE	
15	"ProgramData".CommandResult.Fail		Bool	<input type="checkbox"/> FALSE	
16	"ProgramData".CommandResult.No Job In Slot		Bool	<input type="checkbox"/> FALSE	
17	"ProgramData".CommandResult.UnknownCmd		Bool	<input type="checkbox"/> FALSE	
18	"ProgramData".CommandResult.Success		Bool	<input type="checkbox"/> FALSE	
19					

Rufen Sie die „Beobachtungstabelle\_1“-Elemente **ChangeJob** (Job ändern) und **New\_JobSlot** (Neuer Job-Slot) auf. Klicken Sie dann neben **New\_JobSlot** in die Spalte **Modify value** (Wert bearbeiten), geben Sie den Wert „2“ ein und drücken Sie die **Eingabetaste**. Klicken Sie in der Zeile „ChangeJob“ auf den Wert in der Spalte „Modify value“ und geben Sie **TRUE** (WAHR) ein. Klicken Sie oben in der Symbolleiste auf die

#	Name	Address	Display format	Monitor value	Modify value	Comment
1	"ProgramData".ChangeJob		Bool	<input type="checkbox"/> FALSE	TRUE	
2	"ProgramData".New_JobSlot		DEC+/-	1	2	
3	"ProgramData".Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE		
4	"ProgramData".Current_JobSlot		DEC+/-	2		
5	"ProgramData".Trigger		Bool	<input type="checkbox"/> FALSE		
6	"ProgramData".TotalTriggerCount		DEC+/-	0		
7	"ProgramData".Pass		Bool	<input type="checkbox"/> FALSE		

Schaltfläche mit dem Blitzsymbol und der Zahl 1. Dadurch wird der Job-Slot als Slot 2 festgelegt. Klicken Sie dann erneut auf das Blitzsymbol.

Statt des aktuellen Jobs wird daraufhin der Job in Slot 2 geöffnet. Um sicherzustellen, dass dadurch alle Änderungen gelöscht werden, die Sie in der Spalte „Modify value“ (Wert bearbeiten) vorgenommen haben, klicken Sie in der Zeile **Get\_CurrentJobSlot** (Aktuellen Job-Slot abrufen) auf den Wert in der Spalte „Modify value“. Geben Sie **TRUE** (WAHR) ein und klicken Sie dann erneut auf das Blitzsymbol.

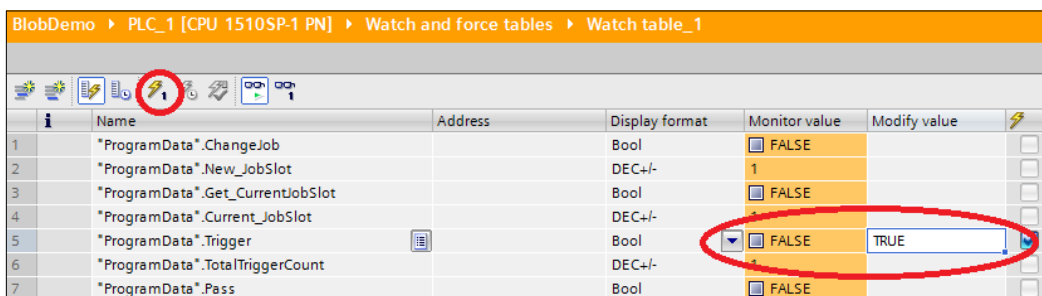


Name	Address	Display format	Monitor value	Modify value
*ProgramData*.ChangeJob		Bool	<input type="checkbox"/> FALSE	
*ProgramData*.New_JobSlot		DEC+/-	1	
*ProgramData*.Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE	TRUE
*ProgramData*.Current_JobSlot		DEC+/-	1	

Der aktuelle Job erscheint nun rechts neben **Current\_JobSlot** (Aktueller Job-Slot). Sie können diese Vorgehensweise wiederholen, indem Sie wieder zu Slot 1 wechseln und überprüfen, ob als aktueller Job-Slot tatsächlich 1 angezeigt wird.

### Inspektion auslösen und Zählerstände zurücksetzen

Löschen Sie alle Änderungen, die Sie in der Spalte „Modify value“ (Wert bearbeiten) vorgenommen haben. Falls Sie die Inspektion mitverfolgen möchten, vergewissern Sie sich, dass die CS50-Software aktiv und mit dem Gerät verbunden ist. Überprüfen Sie auch, ob der Job ausgeführt wird. Falls Sie die CS50-Software nicht ausführen möchten, können Sie die Trigger anhand der LEDs am Gerät überwachen. Ändern Sie den Wert in der Spalte **Trigger** in **TRUE** (WAHR) und klicken Sie dann erneut auf das Blitzsymbol.



BlobDemo > PLC\_1 [CPU 1510SP-1 PN] > Watch and force tables > Watch table\_1

Name	Address	Display format	Monitor value	Modify value	⚡
1 *ProgramData*.ChangeJob		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
2 *ProgramData*.New_JobSlot		DEC+/-	1		<input type="checkbox"/>
3 *ProgramData*.Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
4 *ProgramData*.Current_JobSlot		DEC+/-	1		<input type="checkbox"/>
5 *ProgramData*.Trigger		Bool	<input type="checkbox"/> FALSE	TRUE	<input type="checkbox"/>
6 *ProgramData*.TotalTriggerCount		DEC+/-	1		<input type="checkbox"/>
7 *ProgramData*.Pass		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>

Die Werte für **InstancePoints** (Instanzpunkte), **Status** und **RadiusDatum** (Radius-Bezugspunkt) ändern sich nun wie im Bild unten dargestellt. Die Booleschen Werte für „Pass“ (Erfolg) und „Fail“ (Fehlschlag) signalisieren, ob die Inspektion erfolgreich war oder nicht. Außerdem werden die entsprechenden Zählerstände erhöht. Jedes Mal, wenn Sie auf die Schaltfläche mit dem Blitzsymbol klicken, wird eine Inspektion gestartet und damit zum nächsten Bild gewechselt. Die Inspektionsergebnisse sind abwechselnd Erfolge und Fehlschläge.

Um die Zählerstände zurückzusetzen, löschen Sie alle Einstellungen in der Spalte „Modify value“ (Wert bearbeiten) und ändern Sie dann die entsprechende Einstellung in der Zeile **ResetCounters** (Zähler zurücksetzen) zu **TRUE** (WAHR). Klicken Sie anschließend wieder auf das Blitzsymbol. Alle im Bild unten blau markierten Zählerstände werden nun auf **0** zurückgesetzt.

BlobDemo ▶ PLC\_1 [CPU 1510SP-1 PN] ▶ Watch and force tables ▶ Watch table\_1

	Name	Address	Display format	Monitor value	Modify value	
1	*ProgramData*.ChangeJob		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
2	*ProgramData*.New_JobSlot		DEC+/-	1		<input type="checkbox"/>
3	*ProgramData*.Get_CurrentJobSlot		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
4	*ProgramData*.Current_JobSlot		DEC+/-	1		<input type="checkbox"/>
5	*ProgramData*.Trigger		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
6	*ProgramData*.TotalTriggerCount		DEC+/-	0		<input type="checkbox"/>
7	*ProgramData*.Pass		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
8	*ProgramData*.TotalPassedCount		DEC+/-	0		<input type="checkbox"/>
9	*ProgramData*.Fail		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
10	*ProgramData*.TotalFailedCount		DEC+/-	0		<input type="checkbox"/>
11	*ProgramData*.TotalInspCount		DEC+/-	0		<input type="checkbox"/>
12	*ProgramData*.ResetCounters		Bool	<input type="checkbox"/> FALSE	<input type="text" value="TRUE"/>	<input type="checkbox"/>
13	*UserData*.NumberOfParts	%DB2.DBW0	Hex	16#0002		<input type="checkbox"/>
14	*ProgramData*.ResetError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>

Wenn Sie diese Einstellung erneut löschen, den Wert für „Trigger“ auf „TRUE“ (WAHR) setzen und noch einmal auf das Blitzsymbol klicken, starten sowohl der SPS-Zähler als auch der Zähler der CS50-Software bei **1**.

### Fehler zurücksetzen

Sollte einer der unten dargestellten „TRUE“-Werte einen Fehler zurückgeben, können Sie diesen Fehler beheben, indem Sie in der Zeile **ResetError** (Fehler zurücksetzen) „TRUE“ (WAHR) eingeben und dann wieder auf die Schaltfläche mit dem Blitzsymbol klicken. Setzen Sie den Wert dann auf **FALSE** (FALSCH) zurück und klicken Sie erneut auf die Schaltfläche. Sämtliche Fehler in den unten dargestellten Werten müssten nun beseitigt werden.

*ProgramData*.CommandResult.Fail		Bool	<input type="checkbox"/> FALSE
*ProgramData*.CommandResult.*No Job In Slot*		Bool	<input type="checkbox"/> FALSE
*ProgramData*.CommandResult.UnknownCmd		Bool	<input type="checkbox"/> FALSE
*ProgramData*.CommandResult.Success		Bool	<input type="checkbox"/> FALSE

# Demo PROFINET I/O-Objekt-Zähler mit STEP 7 und CS50-Sensor

Dieses Kapitel beschreibt die Anwendung des di-soric-SPS-Democodes in einem Vision-Job mit Testobjekt.

Die PROFINET I/O-Demodateien befinden sich im Installationsordner der CS50-Software: **C:\di-soric\Vscope\Tutorials and Samples\CS50 Sensor\PROFINET Demo\Step7\_Demos\BlobDemo**. Öffnen Sie die Datei **BlobDemo.avp** und laden Sie sie auf die Kamera herunter.

#### Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für PROFINET I/O aktiviert werden. Informationen zum Aktivieren und Umschalten von Kommunikationsprotokollen finden Sie in **Kapitel 1 [Industrieprotokolle aktivieren](#)**.
- Bei CS50-Software-Jobs wird die PROFINET I/O-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert.

**Wichtig:** Mit PROFINET I/O können die IP-Adresse und die Subnetzmaske der Kamera von der SPS automatisch zugewiesen werden. Dazu darf dem CS50-Sensor jedoch beim Neustart so lange keine IP-Adresse zugewiesen werden, bis die SPS in den Betriebsmodus wechselt. In der Zwischenzeit wird der CS50-Sensor im Netzwerk der CS50-Software nicht angezeigt.

## Überblick

---

In dieser Demonstration erfahren Sie, wie man einen gespeicherten Job auf die Kamera lädt und über **PROFINET I/O** eine Verbindung zu einer **Siemens S7-SPS** herstellt. Außerdem zeigen wir Ihnen anhand einiger Beispiele, wie man Programme ausführt, die über eine Schnittstelle mit der Kamera verbunden sind.

Während der Prüfung der PROFINET I/O-Kapazitäten werden folgende Schritte ausgeführt:

- **Start mit der CS50-Software**

Öffnen Sie einen Demo-Vision-Job in der CS50-Software. Bevor die Kamera mit der SPS verbunden wird, können Sie mithilfe der Option „Try Out“ (Ausprobieren) eine Vorschau aufrufen.

- **Vorbereitung der SPS**

Integrieren Sie die Kamera in die SPS-Umgebung mit der Software STEP 7 und der GSD-Datei.

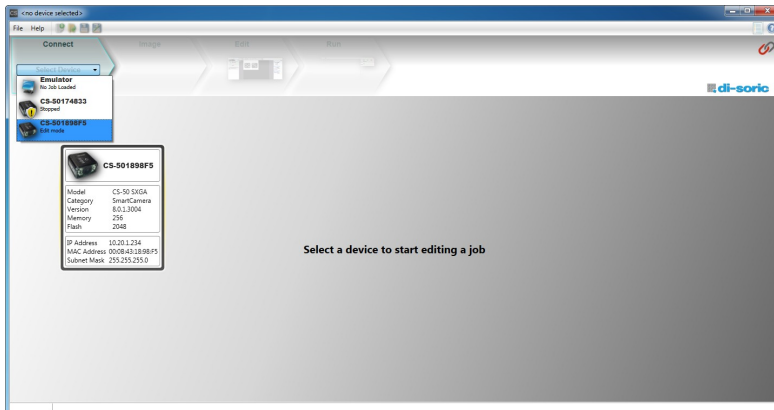
- **Start der Kamera**

Lösen Sie die Kamera, während sie online ist, mit der SPS aus und beobachten Sie die Veränderungen im Inspektionsstatus, während die SPS die Parameter des Vision-Jobs neu konfiguriert.

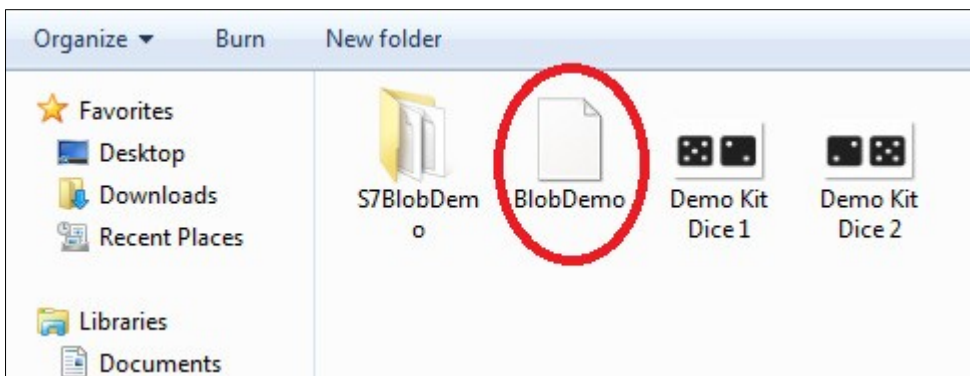
## Einrichtung der CS50-Software

Vor dem Start der CS50-Software sollten Sie sicherstellen, dass entweder die Kamera mit der SPS verbunden ist oder sich SPS und Kamera beide im gleichen Netzwerk befinden. Vergewissern Sie sich, dass der PC, die SPS und die Kamera dieselbe Netzwerkkategorie und die entsprechenden Subnetzadressen aufweisen.

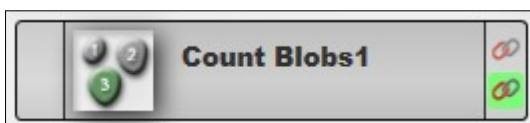
Starten Sie die CS50-Software und wählen Sie die Kamera aus.



Klicken Sie in der Ansicht „Image“ (Bild) auf **Load a Job** (Job laden). Suchen Sie dann den Ordner **C:\di-soric\VscapelTutorials And Samples\CS50 Sensor\PROFINET DemoStep7\_Demos\BlobDemo**. Wählen Sie die Datei **BlobDemo.avp** aus.



Der Demo-Job enthält ein Tool: **Count Blobs** (Objekte zählen).

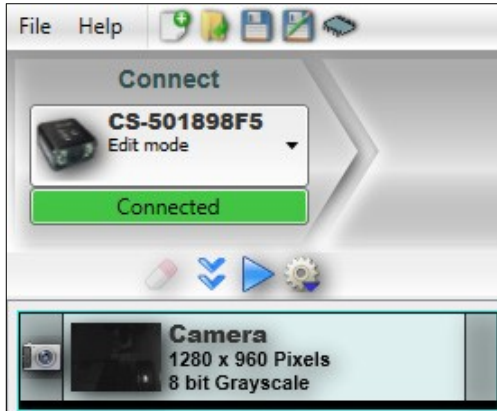


Das mit der PROFINET I/O-Struktur verknüpfte Objekt-Tool Daten-Element ist hier zu sehen:

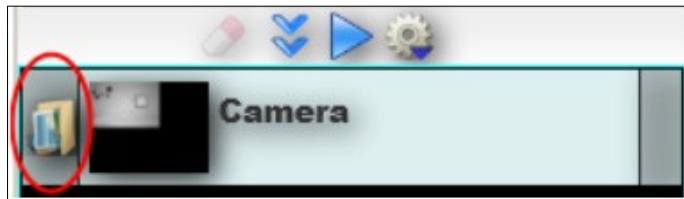
Tool Result	PLC IN
Number of parts	"UserData".NumberOfParts (%DB2.DBW0)

Diese Daten werden in zyklischen Intervallen zwischen Kamera und SPS ausgetauscht.

Sobald der Job geladen ist, verknüpfen Sie die vorgeschichteten Bilder auf dem lokalen PC. Klicken Sie dazu auf der Schaltfläche **Camera** (Kamera) auf das Symbol ganz links, um ein Bild auszuwählen und zu laden.



Daraufhin öffnet sich ein Fenster, in dem Sie eine Datei auswählen können. Navigieren Sie zu demselben Ordner, in dem der Demo-Job (PROFINET I/O Demo) geladen wurde. Sobald die Bilder gefunden wurden, ändert sich das Symbol: Anstelle der Kamera erscheint nun ein Ordner.

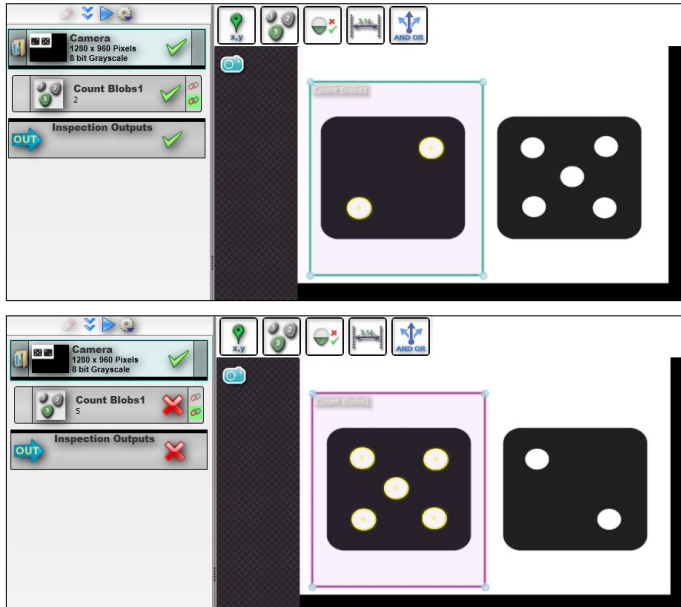


Klicken Sie auf das Symbol **Try Job Once** (Job einmal testen), um den gesamten Job mit dem geladenen Bild durchlaufen zu lassen.





Die Datei enthält zwei Bilder. Eines davon führt bei der Inspektion zu einem Erfolg, das andere zu einem Fehlschlag.



Klicken Sie nun oben in der Symbolleiste auf **Run** (Ausführen). Dadurch wird der Job auf die Kamera heruntergeladen. Der Job kann nun ausgeführt und getestet werden. Damit die Demo mit Job-Änderung ausgeführt werden kann, muss dieser Job allerdings in **Slot 1** geladen werden. Klicken Sie oben in der Symbolleiste wieder auf die Schaltfläche zum Öffnen der Ansicht **Edit** (Bearbeiten).

Klicken Sie auf das Slot-Symbol und wählen Sie „Slot 1“ oder **New Slot** (Neuer Slot) aus, falls derzeit keine Slots belegt sind.



Der Job und die Bilder werden nun im Flash-Speicher der Kamera gespeichert.

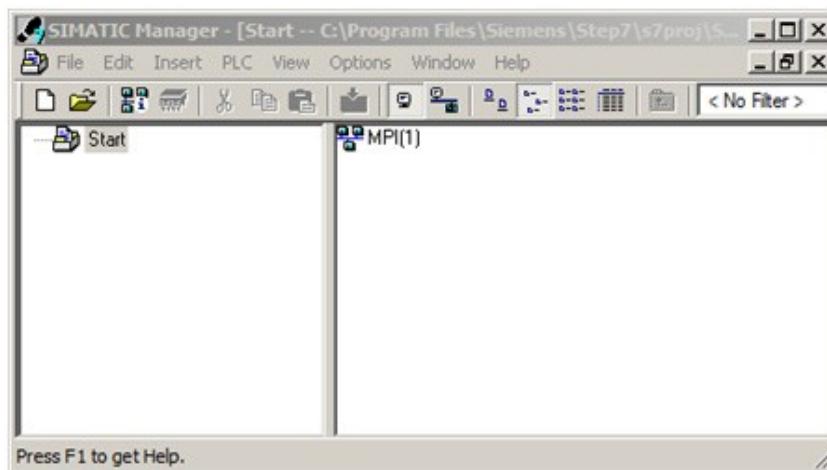
## Einrichtung von STEP 7

Die Anleitung in diesem Kapitel wurde auf der Basis der Software Siemens STEP 7 (Version 5.5 mit SP2) und einer ET200S-SPS (Katalognummer 6ES7 151-8AB01-0AB0, CPU-Version 3.2) erstellt. Getestet wurde mit einer 315-2 PN/DP-SPS (Katalognummer 6ES7-315-2EH13-0AB0, CPU-Version 2.6).

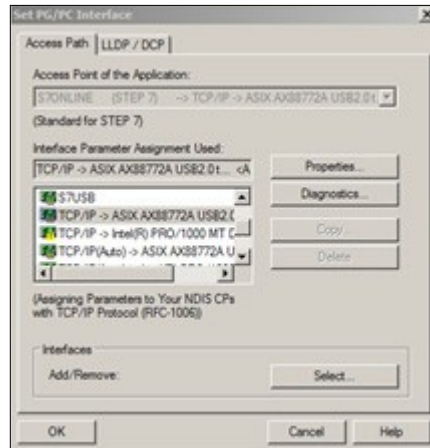
Starten Sie den SIMATIC-Manager vom Desktop aus.



Erstellen Sie über das Menü **Projekt > Neu** ein neues Projekt. Legen Sie einen Speicherort für das Projekt fest, geben Sie einen Projektnamen ein und klicken Sie auf **OK**. Im Beispiel unten lautet der Projektname **Start**. Sobald das Projekt erstellt wurde, sehen Sie das unten abgebildete Fenster. Dabei handelt es sich um das Hauptfenster für das SPS-Programm und die Hardware-Einstellungen.

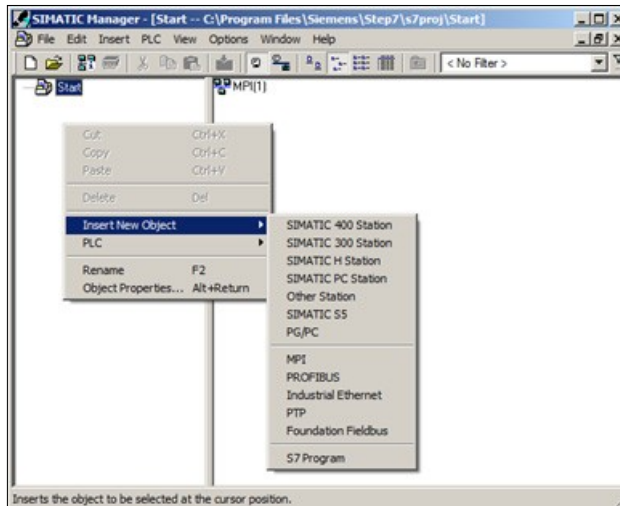


Falls der PC mehrere Netzwerkkarten hat, achten Sie darauf, für die SIMATIC-Software die richtige Karte auszuwählen. Wählen Sie im Menü „Extras“ und dann „PG/PC-Schnittstelle einstellen“ aus. Dadurch wird das Dialogfenster für die PG/PC-Schnittstelle mit einer Liste aller vorhandenen Netzwerkkarten geöffnet. Wählen Sie auf der

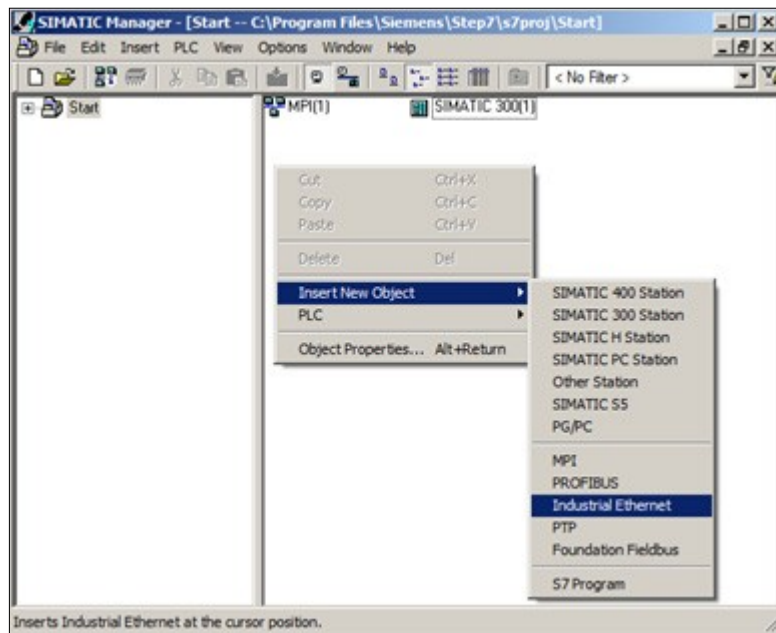


Registerkarte „Zugriffsweg“ die NIC-Karte aus, die den Namensbestandteil „TCP/IP >“ enthält.

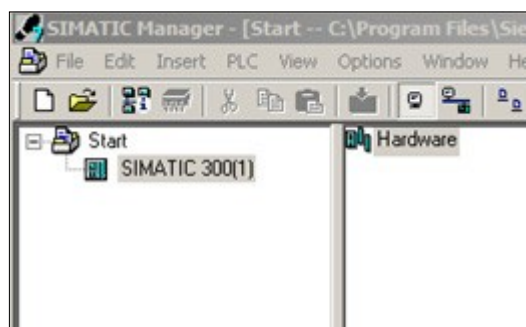
Klicken Sie mit der rechten Maustaste in das Dialogfenster des SIMATIC-Managers und wählen Sie „Neues Objekt einfügen“ aus. Hier können Sie nun die Basisstation festlegen. Wenn Sie beispielsweise eine ET200S konfigurieren möchten, wählen Sie die Station SIMATIC 300 aus, da diese auf der CPU-Serie 300 basiert.



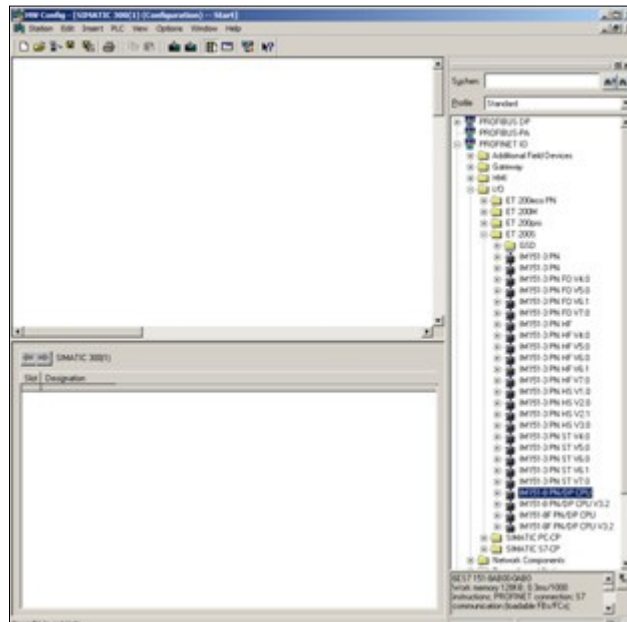
Nachdem Sie die gewünschte Station hinzugefügt haben, klicken Sie mit der rechten Maustaste und wählen Sie **Industrial Ethernet** aus.



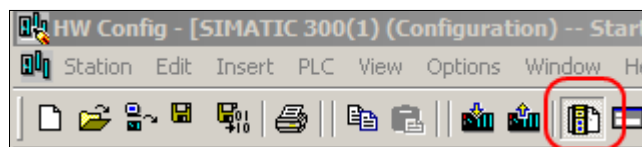
Maximieren Sie links in der Baumansicht den Knoten „Start“ und klicken Sie auf die Station. Auf der rechten Seite erscheint nun ein Hardware-Symbol.



Doppelklicken Sie auf **Hardware**, um das Dialogfenster **HW-Konfig** aufzurufen.



Vergewissern Sie sich, dass in der Symbolleiste oben das Symbol **Katalog** ausgewählt ist. Dadurch wird im rechten Fensterbereich eine Baumansicht mit allen vorhandenen Hardwaregeräten eingeblendet.

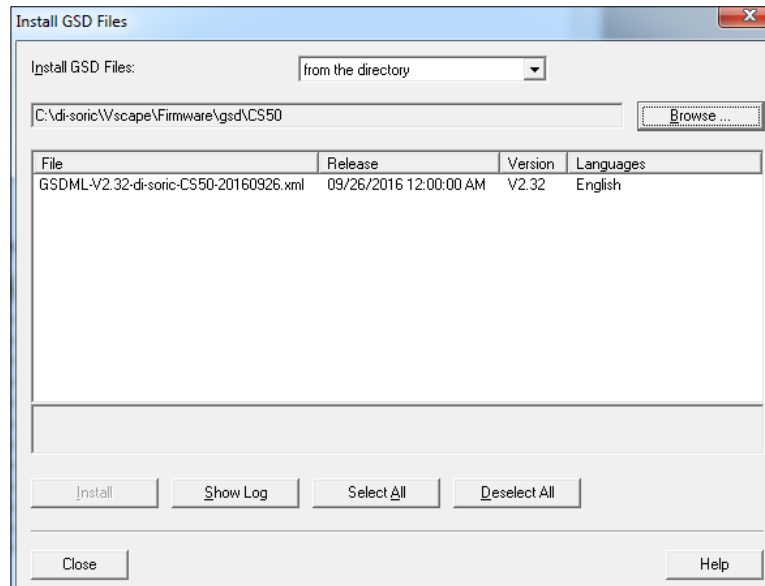


Falls die neueste GSDML-Datei noch nicht importiert wurde, folgen Sie der unten stehenden Anleitung zum Importieren.

## GSDML

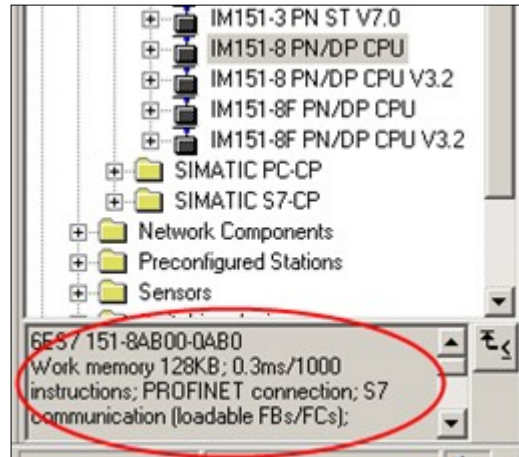
- Installieren Sie die GSD-Datei über das Menü **Extras > GSD-Datei installieren**.
- Klicken Sie auf die Schaltfläche „Durchsuchen“ und wählen Sie den Ordner **di-soric\Vscape\Firmware\gsd\CS50 Sensor** aus, in dem sich die GSDML-Datei befindet.

- Im mittleren Bereich des Bildschirms werden alle vorhandenen GSDML-Dateien angezeigt.



- Wählen Sie die gewünschte Datei aus und klicken Sie auf **Installieren**.
- Wenn Sie fertig sind, schließen Sie das Dialogfenster.
- Die Kamera müsste nun im Menü **Weitere Feldgeräte > Allgemein** unter „PROFINET I/O“ erscheinen.

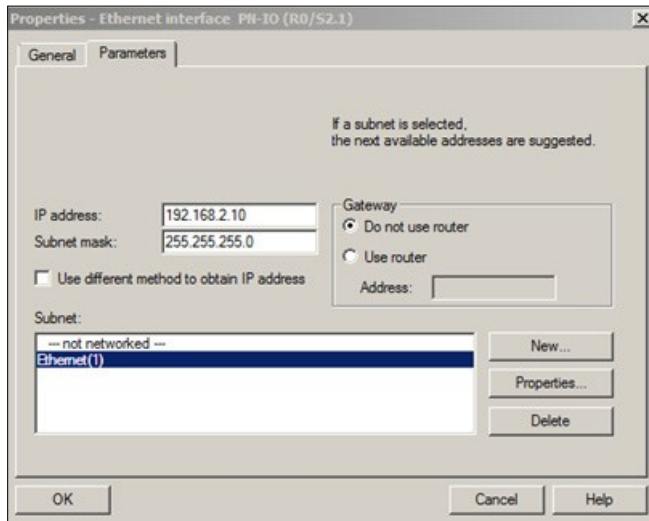
Fügen Sie die CPU aus der Katalogansicht hinzu, indem Sie sie herüberziehen oder darauf doppelklicken. Katalog- und Versionsnummer müssen exakt mit denen der SPS übereinstimmen. Die Katalognummer wird unten in der Ansicht eingeblendet.



Bei einigen CPU-Modulen muss zuvor ein allgemeines Rack hinzugefügt werden. Falls Ihre CPU ein Rack voraussetzt, werden Sie zum Hinzufügen eines Racks aufgefordert, bevor Sie das CPU-Modul einfügen können.

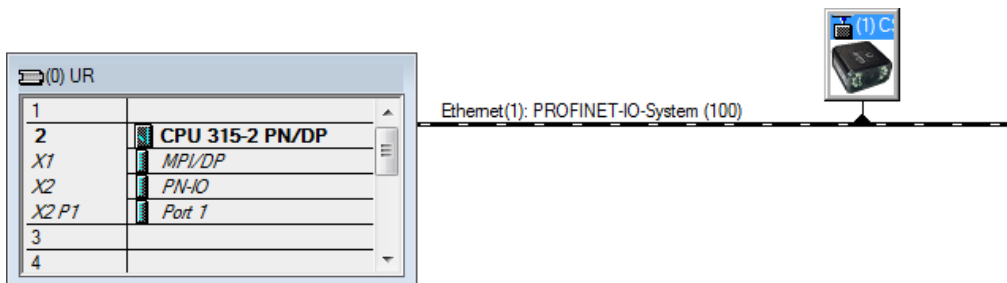
Sobald die CPU hinzugefügt wurde, erscheint ein Pop-up-Fenster, in dem Sie aufgefordert werden, die Eigenschaften bezüglich der IP-Informationen anzugeben.

Wählen Sie aus dem Listenfeld unten die Option **Ethernet(1)** aus und geben Sie die IP-Adresse der CPU ein.



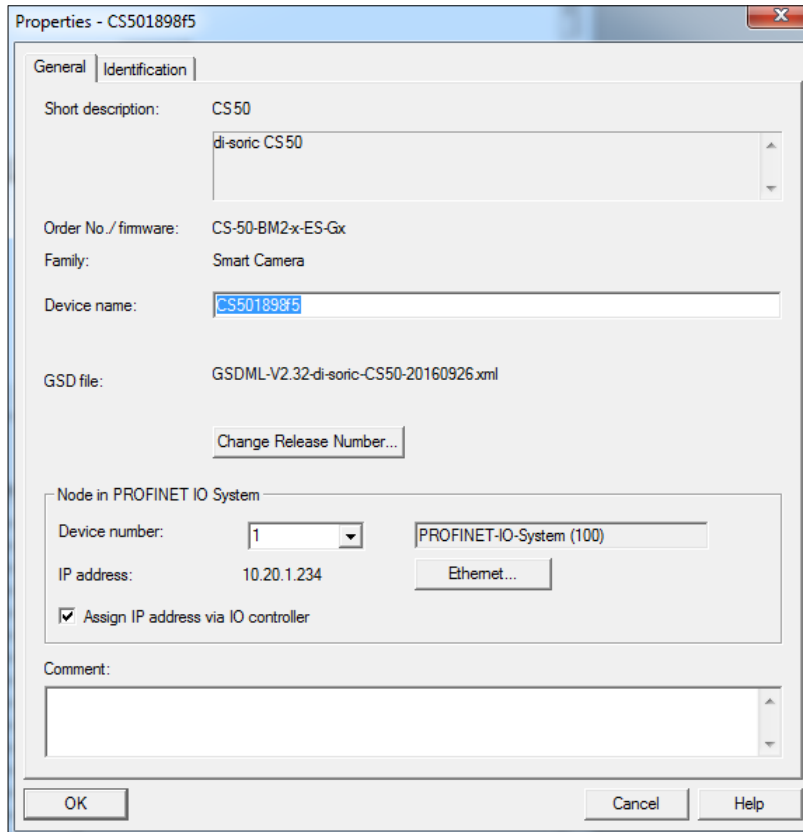
Nachdem Sie die CPU hinzugefügt haben, können Sie ggf. zusätzliche Erweiterungsmodule hinzufügen. Klicken Sie dazu im linken Abschnitt des Fensters auf die CPU. Daraufhin erscheint im unteren Abschnitt eine Liste aller vorhandenen Ports und Slots.

Fügen Sie die Kamera unter **PROFINET I/O > Weitere Feldgeräte > Allgemein > disoric** hinzu. Sie können die Kamera hinzufügen, indem Sie sie auf **Ethernet(1): PROFINET-I/O System (100)** ziehen oder indem Sie zunächst die Zeile **Ethernet(1)** auswählen und dann auf die Kamera doppelklicken. Sobald Sie die Kamera hinzugefügt haben, erscheint im Konfigurationsfenster ein entsprechendes Symbol.





Doppelklicken Sie auf das Symbol für die hinzugefügte Kamera, um ein Dialogfenster mit den Kameraeigenschaften zu öffnen. Im Feld „Gerätename“ können Sie entweder den vorhandenen Namen der Kamera beibehalten oder einen neuen, eindeutigen Namen eingeben.



Falls Sie einen neuen Namen eingegeben haben, muss das Gerät manuell aktualisiert werden. Wie das funktioniert, erfahren Sie im Abschnitt „Kameranamen aktualisieren“.

Klicken Sie auf das Kamerasymbol. Daraufhin öffnet sich das unten abgebildete Fenster, in dem Sie die Daten-Slot-Adresse zuordnen können. Achten Sie genau auf die richtigen Adresswerte, da diese Werte in der Demo-Anwendung benötigt werden. Angesichts einer unbegrenzten Zahl möglicher Kombinationen aus Modulen und Slot-Konfigurationen sind die Adresswerte bei jeder Einrichtung anders.

The screenshot shows the SIMATIC 300 configuration window. A hardware rack is displayed with the following modules:

- Slot 1: MPI/DP
- Slot 2: CPU 315-2 PN/DP
- Slot 3: PN-IO
- Slot 4: Port 1
- Slot 5: CS50-BM2-x-E
- Slot 6: Interf.
- Slot 7: Port 1
- Slot 8: Status
- Slot 9: Contr.
- Slot 10: Echo

Below the rack, a table provides detailed information for the modules:

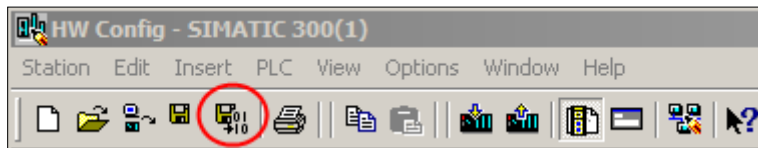
Slot	M.	Order number	I address	Q address	Diagnostic address:	Comment
0	CS50	CS50-BM2-x-E			2043*	
X1	Interf.				2042**	
X1 A	Port 1				2041**	
1	Status		0..1			
2	Contr.			0..1		
3	Echo		264..265			
4	Echo			264..265		
5	Cmd		256..259			
6	Cmd			260..263		
7	Cmd		260..263			
8	Cmd			256..259		
9	Status		266			
10						

In dieser Demo werden die **Int In**-Daten verwendet. Klicken Sie zum Auswählen rechts unter dem PROFINET I/O CS50-Sensor auf „Int In“. Dadurch wird der entsprechende Slot hervorgehoben. Ziehen Sie die Auswahl in diesen Slot. Die Tabelle müsste dann folgendermaßen aussehen:

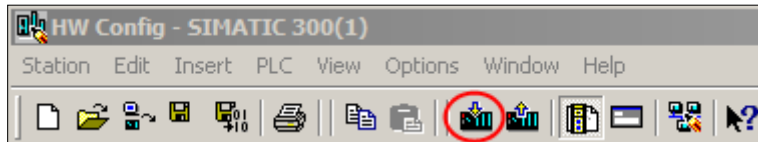
The screenshot shows the SIMATIC Manager configuration window for a SIMATIC 300 system. A hardware rack is visible with slots 1 through 4. Slot 2 contains a CPU 315-2 PN/DP. A PROFINET IO System (100) is connected to the rack. Below the rack, a table for the CS501898f5 sensor is displayed.

Slot	M..	Order number	I address	Q address	Diagnostic address:	Comment
0	CS50	CS50-BM2-x-E			2043*	
X1	Interf.				2042*	
X1 A	Port 1				2041*	
1	Status		0..1			
2	Control			0..1		
3	Echo		264...265			
4	Echo			264...265		
5	Cmd A		256...259			
6	Cmd B			260...263		
7	Cmd A		260...263			
8	Cmd B			256...259		
9	Status		266			
10						
11						
12						
13						
14						
15	Int In		267...286			
16						

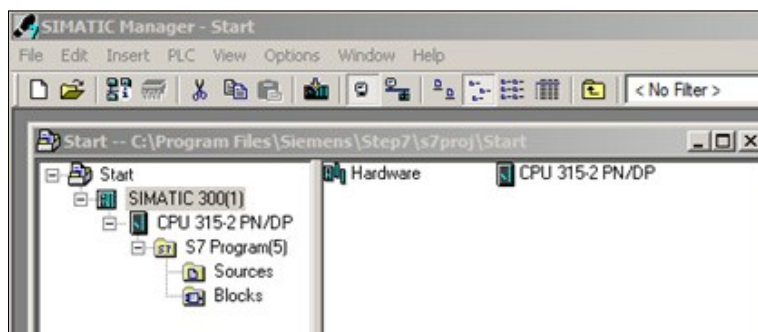
Nach Abschluss der Hardwarekonfiguration ist der nächste Schritt das Kompilieren und Herunterladen. Klicken Sie dazu in der Symbolleiste auf das Symbol zum Kompilieren und Speichern. Falls Ihnen bei der Konfiguration ein Fehler unterlaufen sein sollte, sehen Sie an dieser Stelle eine Warnmeldung.



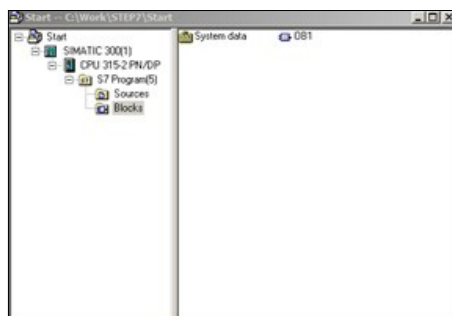
Klicken Sie nach dem Kompilieren in der Symbolleiste auf das Download-Symbol, damit die Daten an die SPS übermittelt werden.



Schließen oder minimieren Sie an dieser Stelle das Dialogfenster **HW-Konfig** und öffnen Sie das Fenster **SIMATIC-Manager**. Die CPU müsste nun neben dem Hardware-Symbol sowie in der Baumansicht links erscheinen. Klappen Sie in der Baumansicht den CPU-Knoten und dessen Unterknoten auf.



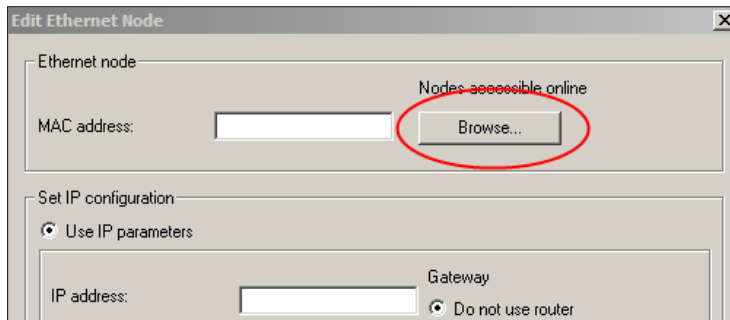
Um sich die Programmobjekte anzusehen, klicken Sie auf den Knoten „Blocks“ (Blöcke).



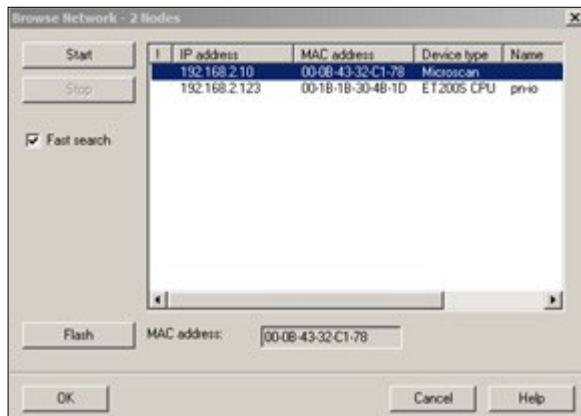
## Kameranamen aktualisieren

Im Dialogfenster „HW-Konfig“ öffnen Sie das Menü **SPS > Ethernet > Ethernet-Knoten bearbeiten** und wählen dort **Ethernet-Knoten bearbeiten** aus. Im Dialogfenster „SIMATIC-Manager“ öffnen Sie das Menü **SPS > Ethernet-Knoten bearbeiten** und wählen dort **Ethernet-Knoten bearbeiten** aus.

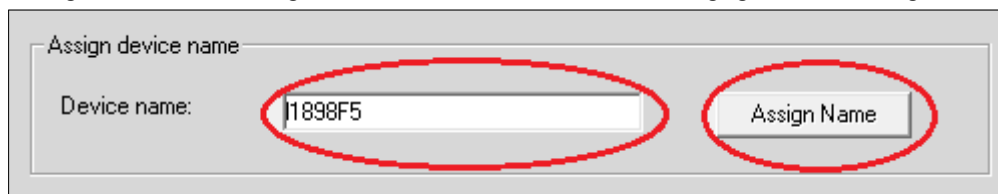
Um das Dialogfenster zur Auswahl des Knotens zu öffnen, klicken Sie auf **Durchsuchen**.



Wählen Sie die Kamera aus und klicken Sie auf **OK**.



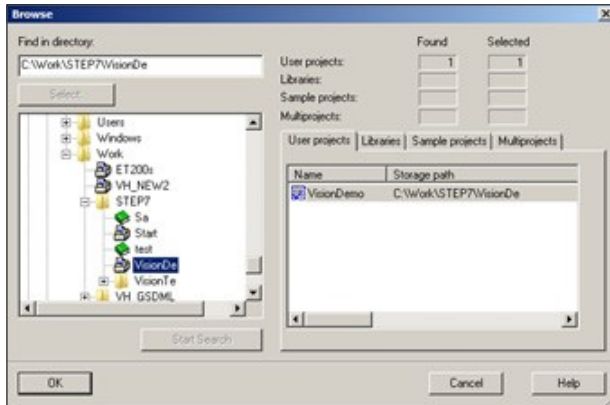
Falls der **Gerätename** nicht mit dem Namen übereinstimmt, den Sie zuvor im Dialogfenster mit den Eigenschaften des CS50-Sensors eingegeben haben, geben Sie



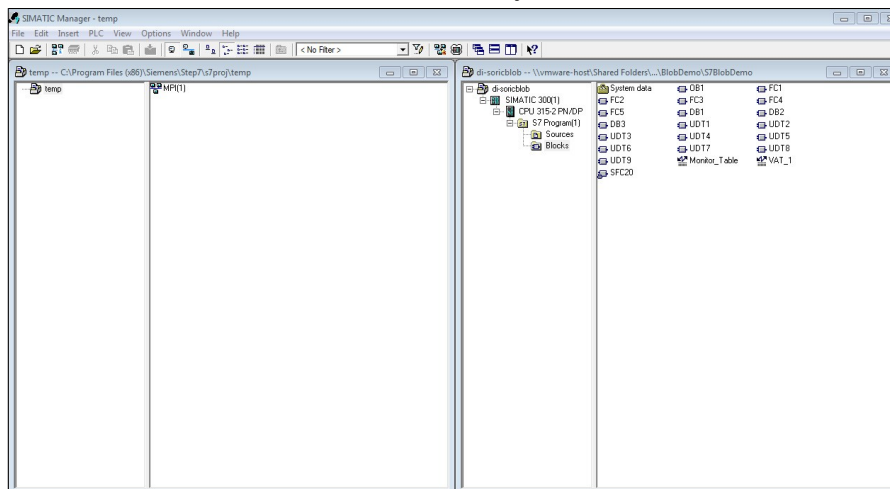
den richtigen Gerätenamen ein. Klicken Sie dann auf **Name zuweisen**.

## Beispielprogramm importieren

Öffnen Sie das Beispielprogramm im SIMATIC-Manager über das Menü **Datei > Öffnen**. Klicken Sie auf **Durchsuchen** und suchen Sie das Programm BlobDemo. Es befindet sich im Ordner `\\di-soric\Vscape\Tutorials and Samples\CS50 Sensor\PROFINET Demo\Step7_Demos\BlobDemo\S7BlobDemo\BlobDemo.s7p`.

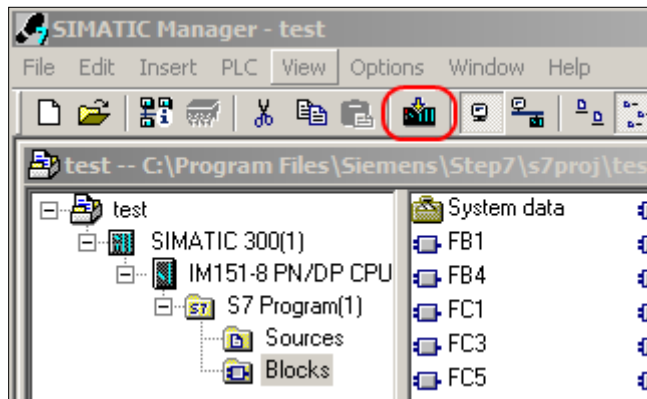


Teilen Sie den Bildschirm wie unten abgebildet, sodass auf einer Seite das Beispielprojekt und auf der anderen Seite das aktuelle Projekt zu sehen ist.

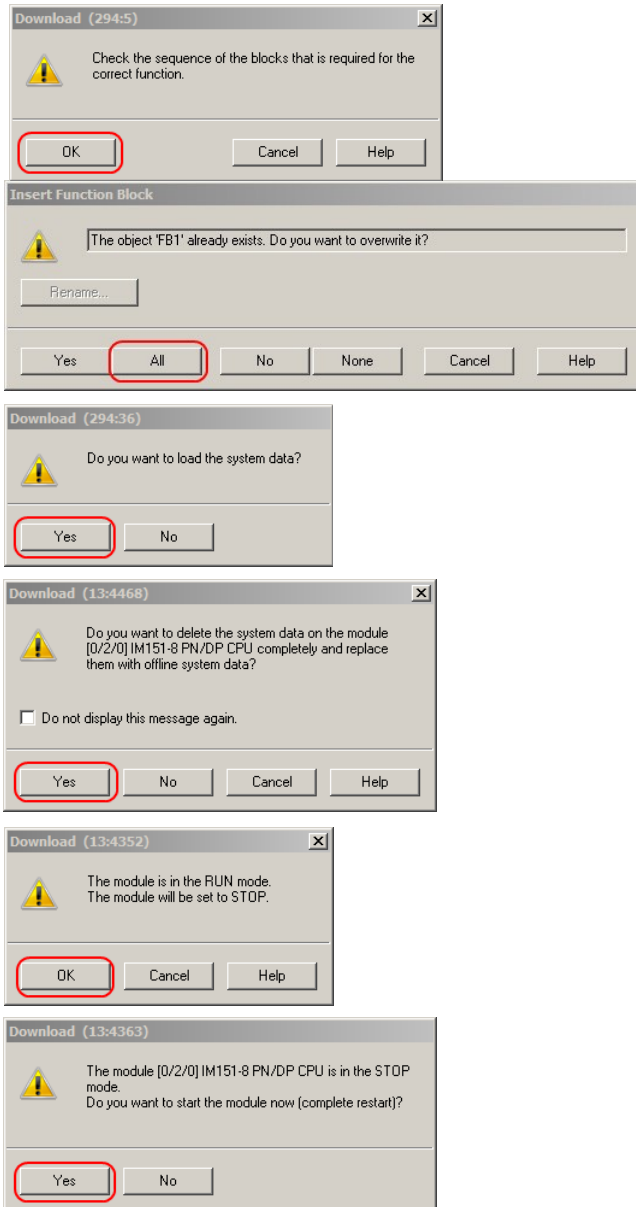


Wählen Sie die Objekte – außer dem Datenobjekt **System** – aus und kopieren Sie sie dann, indem Sie entweder mit der rechten Maustaste klicken und „Kopieren“ auswählen oder die Tastenkombination **Strg + C** verwenden. Fügen Sie die Objekte mit **Strg + V** oder Rechtsklick und der Option „Einfügen“ in das aktuelle Projekt ein. Wenn angezeigt wird, dass **OB1** bereits vorhanden ist, klicken Sie auf **Ja**, um dieses Objekt zu überschreiben. Sie haben nun das Beispielprogramm in das aktuelle Projekt importiert.

Wählen Sie im linken Bereich das Symbol **Blöcke** aus. Klicken Sie im Anschluss daran oben in der Symbolleiste auf das Download-Symbol. Dadurch werden die soeben kopierten Funktionen und Systemdaten auf die SPS heruntergeladen. Als Nächstes müssen Sie die Adressen aus der vorherigen Hardwareinstallation aktualisieren. Doppelklicken Sie dazu auf den Block **OB1**. Der **LAD/STL/FBD**-Editor wird daraufhin geöffnet. OB1 ist die Hauptroutine des SPS-Programms.



Scrollen Sie nach unten bis zum **Netzwerk 1**. Hier werden die Daten von der Kamera der lokalen Datenstruktur zugeordnet. Zur Übertragung der Eingangsdaten von der Kamera in die Programmstruktur dient die Funktion **FC1**. Klicken Sie auf die entsprechenden Nummern, um die Adressen der Hardware wie abgebildet anzugleichen. Klicken Sie in den sich öffnenden Pop-up-Fenstern auf folgende Schaltflächen:





**Hinweis:** Die Ansicht zur Adressenzuordnung des CS50-Sensormoduls, die von der Ansicht zur Hardwarekonfiguration abhängig ist, sieht so aus:

	Slot	Module	Order number	I address	Q address	Diagnost
	0	CS50189815	CS50-BM2-x-E			2043*
	X1	Interface				2042*
	X1	Port 1				2041*
0	1	Status		0..1		
	2	Control			0..1	
264	3	Echo In		264..265		
	4	Echo Out			264..265	
CMDCODERS	5	Cmd Code Ret		256..259		
256	6	Cmd Code			260..263	
LT	7	Cmd Ret		260..263		
260	8	Cmd Arg			256..259	
266	9	State		266		
0	10					
	11					
	12					
264	13					
	14					
260	15	Int In		267..286		
256	16					

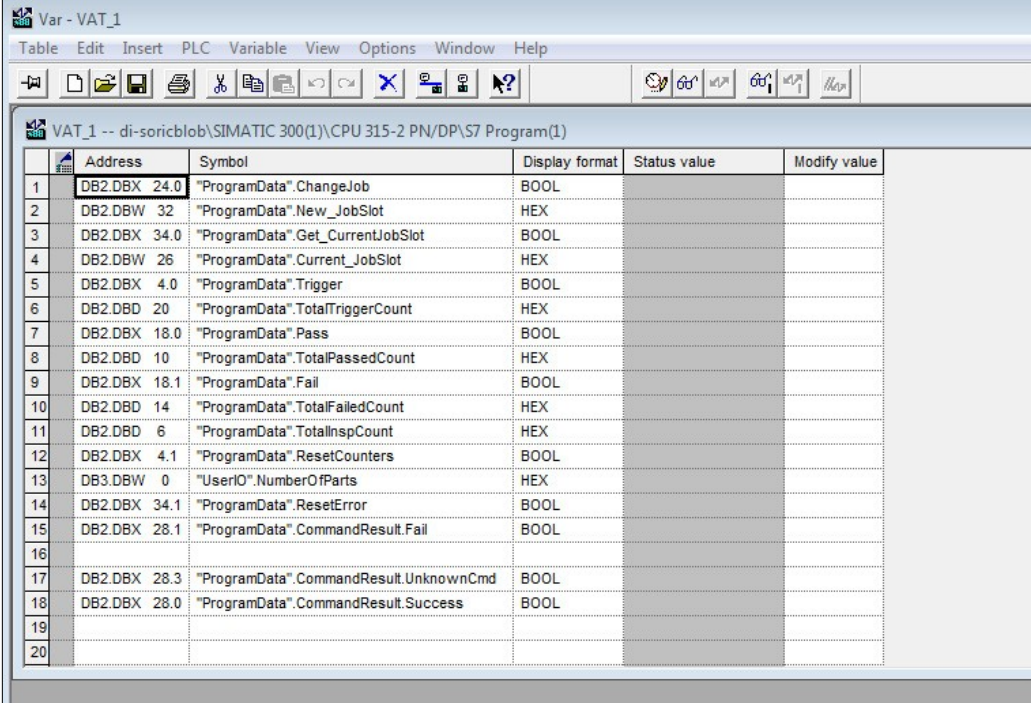
Damit alle Adressen übereinstimmen, müssen die Adressen auf der rechten Seite in **Netzwerk 1** kopiert werden. Beispiel: In der Hardwarekonfiguration wird für den Status die Adresse „0..1“ angezeigt. Kopieren Sie **0** und fügen Sie es unter „Netzwerk 1“ in den Wert für **STATUS** ein.

Zur korrekten Zuordnung der Ausgangsdaten wiederholen Sie diese Schritte für **Netzwerk 6**. Beachten Sie, dass alle Eingangsadressen in der Spalte **I-Adresse** und alle Ausgangsadressen in der Spalte **Q-Adresse** stehen. Speichern Sie die Daten auf dem PC und laden Sie sie dann auf die SPS herunter.

Führen Sie einen Master-Reset durch und schalten Sie die SPS in den Betriebsmodus. Vergewissern Sie sich, dass alle LEDs eine **korrekte Funktion** signalisieren.

## Demo ausführen

Doppelklicken Sie im Dialogfenster **SIMATIC-Manager** auf das Symbol **VAT\_1**. Daraufhin wird ein Fenster mit der Variablen-tabelle für die Datentyp-Demo geöffnet. Maximieren Sie das darin angezeigte Dialogfenster, um den Sichtbereich zu vergrößern.



	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL		
2	DB2.DBW 32	"ProgramData".New_JobSlot	HEX		
3	DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL		
4	DB2.DBW 26	"ProgramData".Current_JobSlot	HEX		
5	DB2.DBX 4.0	"ProgramData".Trigger	BOOL		
6	DB2.DBD 20	"ProgramData".TotalTriggerCount	HEX		
7	DB2.DBX 18.0	"ProgramData".Pass	BOOL		
8	DB2.DBD 10	"ProgramData".TotalPassedCount	HEX		
9	DB2.DBX 18.1	"ProgramData".Fail	BOOL		
10	DB2.DBD 14	"ProgramData".TotalFailedCount	HEX		
11	DB2.DBD 6	"ProgramData".TotalInspCount	HEX		
12	DB2.DBX 4.1	"ProgramData".ResetCounters	BOOL		
13	DB3.DBW 0	"UserIO".NumberOfParts	HEX		
14	DB2.DBX 34.1	"ProgramData".ResetError	BOOL		
15	DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL		
16					
17	DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL		
18	DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL		
19					
20					

Um eine Live-Verbindung zur SPS herzustellen, klicken Sie in der Symbolleiste auf das Symbol zum **Überwachen der Variable**. Dadurch werden die Daten von der SPS in das Dialogfenster übernommen. Die Titelleiste oben erscheint daraufhin blau, und in der Statusleiste unten wird der Ausführungsfortschritt in Form eines grünen Balkens angezeigt.

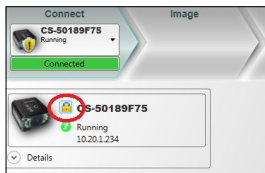


Change Job	Switch jobs from one slot to another
Get Current Job Slot	Displays the slot of the currently running job
Trigger an Inspection	Causes the camera to cycle through its inspection
Reset Counters	Resets the TotalTriggerCount, TotalPassedCount, TotalFailedCount, and TotalInspCount to 0, and also sends a command to the camera to reset its internal counters.
Reset an Error condition	If an error occurs during run time, this will reset it, so demo may be continued.

Hier können nun folgende Vorgänge getestet werden

Jobs ändern und aktuellen Job-Slot abrufen

Folgen Sie der Anleitung zum Speichern eines Jobs im Kameraspeicher, die Sie im Abschnitt „Einrichtung der CS50-Software“ finden. Damit dieser Test korrekt durchgeführt werden kann, laden Sie einen weiteren Job in **Slot 2** der Kamera. Laden Sie beispielsweise den Job „Circle\_LocatorDemo“. Geben Sie die Gerätesteuerung frei, indem Sie auf die Registerkarte **Verbinden** klicken und dann, wie im Bild unten dargestellt, das Schlosssymbol zu **geschlossen** ändern. Klicken Sie anschließend erneut auf die Registerkarte **Ausführen**, damit der neue Job richtig ausgeführt wird.



Rufen Sie die „VAT\_1“-Elemente **ChangeJob** (Job ändern) und **New\_JobSlot** (Neuer Job-Slot) auf. Klicken Sie dann neben „New\_JobSlot“ in die Spalte **Modify value** (Wert bearbeiten), geben Sie den Wert „2“ ein und drücken Sie die **Eingabetaste**. Klicken Sie in der Zeile „ChangeJob“ auf den Wert in der Spalte „Modify value“ und geben Sie „TRUE“ (WAHR) ein. Klicken Sie in der Leiste auf das im Bild unten markierte Symbol.

Address	Symbol	Display format	Status value	Modify value
1	DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false
2	DB2.DBW 32	"ProgramData".New_JobSlot	HEX	W#16#0001

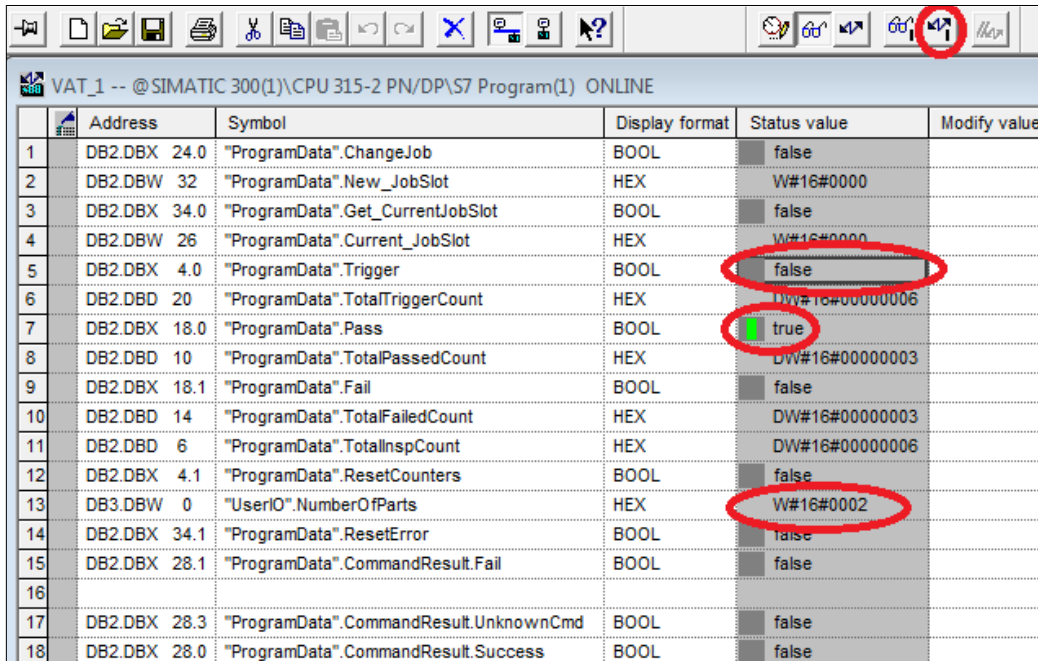
Statt des aktuellen Jobs wird daraufhin der Job in **Slot 2** geöffnet. Um sicherzustellen, dass dadurch alle Änderungen gelöscht werden, die Sie in der Spalte **Modify value** (Wert bearbeiten) vorgenommen haben, klicken Sie in der Zeile **Get\_CurrentJobSlot** (Aktuellen Job-Slot abrufen) auf den Wert in der Spalte „Modify value“. Geben Sie **TRUE** (WAHR) ein und klicken Sie dann erneut auf das im Bild markierte Symbol.

Address	Symbol	Display format	Status value	Modify value
1	DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false
2	DB2.DBW 32	"ProgramData".New_JobSlot	HEX	W#16#0002
3	DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL	false
4	DB2.DBW 26	"ProgramData".Current_JobSlot	HEX	W#16#0002

Der aktuelle Job erscheint nun rechts neben **Current\_JobSlot** (Aktueller Job-Slot). Sie können diese Vorgehensweise wiederholen, indem Sie wieder zu Slot 1 wechseln und überprüfen, ob als aktueller Job-Slot tatsächlich 1 angezeigt wird.

### Inspektion auslösen und Zählerstände zurücksetzen

Vergewissern Sie sich, dass als aktueller Job **Slot 1** ausgewählt ist. Löschen Sie alle Änderungen, die Sie in der Spalte **Modify value** (Wert bearbeiten) vorgenommen haben. Falls Sie die Inspektion mitverfolgen möchten, vergewissern Sie sich, dass die CS50-Software aktiv und mit dem Gerät verbunden ist. Überprüfen Sie auch, ob der Job ausgeführt wird. Falls Sie die CS50-Software nicht ausführen möchten, können Sie die Trigger anhand der LEDs am Gerät überwachen. Ändern Sie den Wert in der Spalte „Trigger“ in **TRUE** (WAHR) und klicken Sie dann erneut auf das im Bild markierte Symbol.



	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false	
2	DB2.DBW 32	"ProgramData".New_JobSlot	HEX	W#16#0000	
3	DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL	false	
4	DB2.DBW 26	"ProgramData".Current_JobSlot	HEX	W#16#0000	
5	DB2.DBX 4.0	"ProgramData".Trigger	BOOL	false	
6	DB2.DBW 20	"ProgramData".TotalTriggerCount	HEX	DW#16#00000006	
7	DB2.DBX 18.0	"ProgramData".Pass	BOOL	true	
8	DB2.DBW 10	"ProgramData".TotalPassedCount	HEX	DW#16#00000003	
9	DB2.DBX 18.1	"ProgramData".Fail	BOOL	false	
10	DB2.DBW 14	"ProgramData".TotalFailedCount	HEX	DW#16#00000003	
11	DB2.DBW 6	"ProgramData".TotalInspCount	HEX	DW#16#00000006	
12	DB2.DBX 4.1	"ProgramData".ResetCounters	BOOL	false	
13	DB3.DBW 0	"UserIO".NumberOfParts	HEX	W#16#0002	
14	DB2.DBX 34.1	"ProgramData".ResetError	BOOL	false	
15	DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL	false	
16					
17	DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL	false	
18	DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL	false	

Der Wert für **NumberOfParts** (Anzahl der Teile) ändert sich nun in **2** oder **5**. Die Booleschen Werte für „Pass“ (Erfolg) und „Fail“ (Fehlschlag) signalisieren, ob die Inspektion erfolgreich war oder nicht. Außerdem werden die entsprechenden Zählerstände erhöht. Jedes Mal, wenn Sie auf die im Bild markierte Schaltfläche klicken, wird eine Inspektion gestartet und damit zum nächsten Bild gewechselt. Die Inspektionsergebnisse sind abwechselnd Erfolge und Fehlschläge.

## Zähler zurücksetzen

Um die Zählerstände zurückzusetzen, löschen Sie alle Einstellungen in der Spalte „Modify value“ (Wert bearbeiten) und ändern Sie dann die entsprechende Einstellung in der Zeile **ResetCounters** (Zähler zurücksetzen) zu **TRUE** (WAHR). Klicken Sie

	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false	
2	DB2.DBW 32	"ProgramData".New_JobSlot	HEX	W#16#0000	
3	DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL	false	
4	DB2.DBW 26	"ProgramData".Current_JobSlot	HEX	W#16#0000	
5	DB2.DBX 4.0	"ProgramData".Trigger	BOOL	false	
6	DB2.DBD 20	"ProgramData".TotalTriggerCount	HEX	DW#16#00000000	
7	DB2.DBX 18.0	"ProgramData".Pass	BOOL	true	
8	DB2.DBD 10	"ProgramData".TotalPassedCount	HEX	DW#16#00000000	
9	DB2.DBX 18.1	"ProgramData".Fail	BOOL	false	
10	DB2.DBD 14	"ProgramData".TotalFailedCount	HEX	DW#16#00000000	
11	DB2.DBD 6	"ProgramData".TotalInspCount	HEX	DW#16#00000000	
12	DB2.DBX 4.1	"ProgramData".ResetCounters	BOOL	false	true
13	DB3.DBW 0	"UserIO".NumberOfParts	HEX	W#16#0002	
14	DB2.DBX 34.1	"ProgramData".ResetError	BOOL	false	
15	DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL	false	
16					
17	DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL	false	
18	DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL	false	
19					

anschließend wieder auf die im Bild markierte Schaltfläche. Alle im Bild unten blau markierten Zählerstände werden nun auf **0** zurückgesetzt.

Wenn Sie diese Einstellung erneut löschen, den Wert für „Trigger“ auf **TRUE** (WAHR) setzen und noch einmal auf die im Bild markierte Schaltfläche klicken, starten sowohl der SPS-Zähler als auch der Zähler der CS50-Software bei **1**.

## Fehler zurücksetzen

Sollte einer der unten dargestellten „TRUE“-Werte einen Fehler zurückgeben, können Sie diesen Fehler beheben, indem Sie in der Zeile **ResetError** (Fehler zurücksetzen) „TRUE“ (WAHR) eingeben und dann wieder auf die Schaltfläche klicken. Setzen Sie den Wert dann auf **FALSE** (FALSCH) zurück und klicken Sie erneut auf die Schaltfläche. Sämtliche Fehler in den unten dargestellten Werten müssten nun beseitigt werden.

DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL	false
DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL	false
DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL	false



# Demo PROFINET I/O- Kreislokalisierung mit STEP 7 und CS50- Sensor

Dieses Kapitel beschreibt die Anwendung des di-soric-SPS-Democodes in einem Vision-Job mit Testobjekt.

Die PROFINET I/O-Demodateien befinden sich im Installationsordner der CS50-Software: **C:\di-soric\Vscape\Tutorials and Samples\CS50 Sensor\PROFINET Demo\Step7\_Demos\Circle\_LocatorDemo**.

Öffnen Sie die Datei **Circle\_LocatorDemo.avp** und laden Sie sie auf die Kamera herunter.

#### Hinweise:

- Bevor das Kommunikationsprotokoll der Kamera in dieser Umgebung verwendet werden kann, muss es für PROFINET I/O aktiviert werden. Informationen zum Aktivieren und Umschalten von Kommunikationsprotokollen finden Sie in **Kapitel 1 Industrieprotokolle aktivieren**.
- Bei CS50-Software-Jobs wird die PROFINET I/O-Kommunikation zwischen Kamera und SPS mithilfe der di-soric Link-Funktion koordiniert.

**Wichtig:** Mit PROFINET I/O können die IP-Adresse und die Subnetzmaske der Kamera von der SPS automatisch zugewiesen werden. Dazu darf dem CS50-Sensor jedoch beim Neustart so lange keine IP-Adresse zugewiesen werden, bis die SPS in den Betriebsmodus wechselt. In der Zwischenzeit wird der CS50-Sensor im Netzwerk der CS50-Software nicht angezeigt.

---

## Überblick

---

In dieser Demonstration erfahren Sie, wie man einen gespeicherten Job auf die Kamera lädt und über **PROFINET I/O** eine Verbindung zu einer **Siemens S7-SPS** herstellt. Außerdem zeigen wir Ihnen anhand einiger Beispiele, wie man Programme ausführt, die über eine Schnittstelle mit der Kamera verbunden sind.

Während der Prüfung der PROFINET I/O-Kapazitäten werden folgende Schritte ausgeführt:

- **Start mit der CS50-Software**

Öffnen Sie einen Demo-Vision-Job in der CS50-Software. Bevor die Kamera mit der SPS verbunden wird, können Sie mithilfe der Option „Try Out“ (Ausprobieren) eine Vorschau aufrufen.

- **Vorbereitung der SPS**

Integrieren Sie die Kamera in die SPS-Umgebung mit der Software STEP 7 und der GSD-Datei.

- **Start der Kamera**

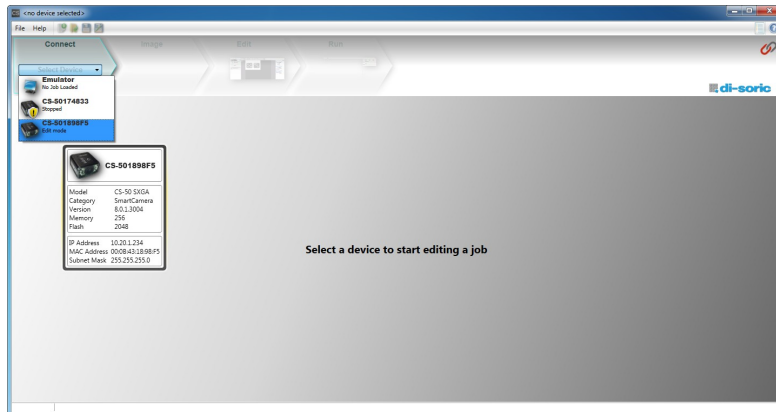
Lösen Sie die Kamera, während sie online ist, mit der SPS aus und beobachten Sie die Veränderungen im Inspektionsstatus, während die SPS die Parameter des Vision-Jobs neu konfiguriert.



## Einrichtung der CS50-Software

Vor dem Start der CS50-Software sollten Sie sicherstellen, dass entweder die Kamera mit der SPS verbunden ist oder sich SPS und Kamera beide im gleichen Netzwerk befinden. Vergewissern Sie sich, dass der PC, die SPS und die Kamera dieselbe Netzwerkkategorie und die entsprechenden Subnetzadressen aufweisen.

Starten Sie die CS50-Software und wählen Sie die Kamera aus.



Klicken Sie in der Ansicht „Image“ (Bild) auf **Load a Job** (Job laden). Suchen Sie dann den Ordner **C:\di-soric\Vscaple\Tutorials And Samples\CS50 Sensor\PROFINET demo\Step7\_Demos\Circle\_LocatorDemo**. Wählen Sie die Datei **Circle\_LocatorDemo.avp** aus.



Der Demo-Job enthält zwei Tools: **Locate Shape** (Form lokalisieren) und **Circle** (Kreis).

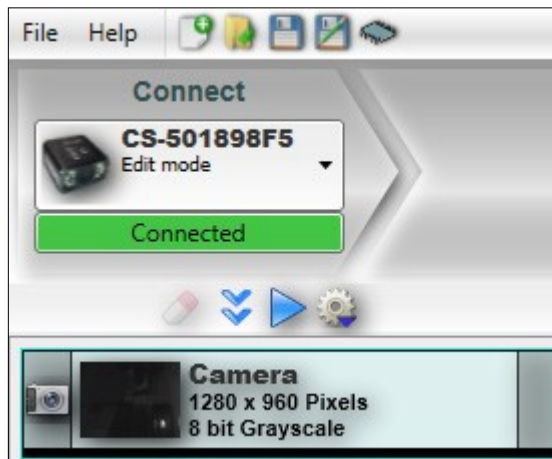


Hier eine Übersicht der PROFINET I/O-Struktur:

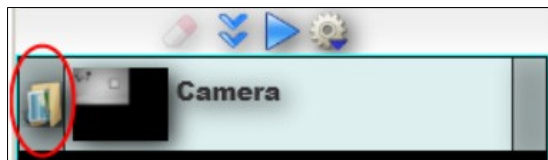
Tool Result	PLC IN
Instance Point1	"UserData".Instance1Point1(%DB2.DBD2)
Instance Point2	"UserData".Instance1Point2(%DB2.DBD6)
Instance Point3	"UserData".InstancePoint3(%DB2.DBD10)
Status	"UserData".Status[0](%DB2.DBX0.0)
RadiumDatum	"UserData".RadiusDatum(%DB2.DBD14)

Diese Daten werden in zyklischen Intervallen zwischen Kamera und SPS ausgetauscht.

Sobald der Job geladen ist, verknüpfen Sie die vorgeschichteten Bilder auf dem lokalen PC. Klicken Sie dazu auf der Schaltfläche **Camera** (Kamera) auf das Symbol ganz links, um ein Bild auszuwählen und zu laden.



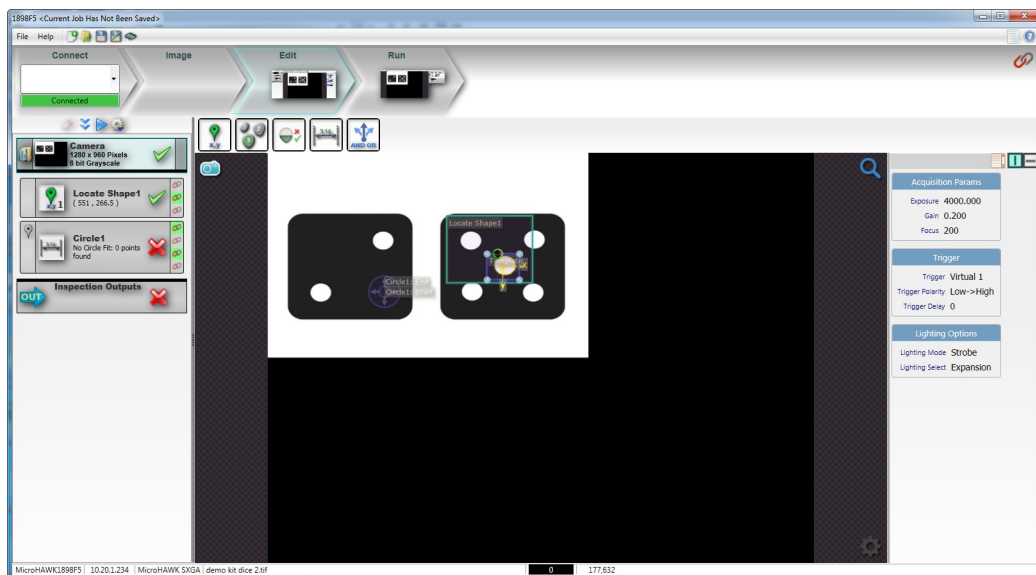
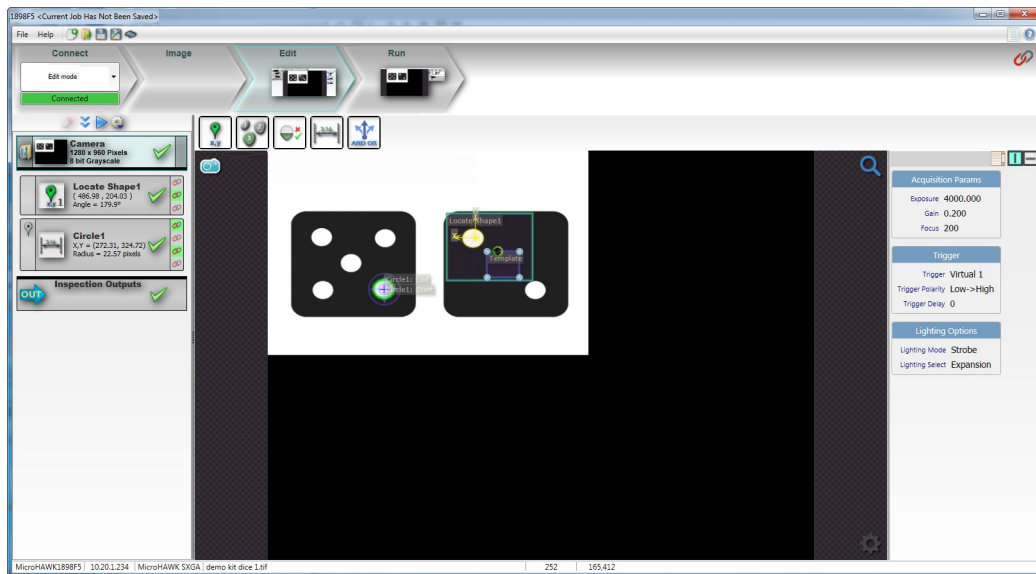
Daraufhin öffnet sich ein Fenster, in dem Sie eine Datei auswählen können. Navigieren Sie zu demselben Ordner, in dem der Demo-Job (PROFINET I/O Demo) geladen wurde. Sobald die Bilder gefunden wurden, ändert sich das Symbol: Anstelle der Kamera erscheint nun ein Ordner.



Klicken Sie auf das Symbol **Try Job Once** (Job einmal testen), um den gesamten Job mit dem geladenen Bild durchlaufen zu lassen.



Die Datei enthält zwei Bilder. Eines davon führt bei der Inspektion zu einem Erfolg, das andere zu einem Fehlschlag.



Klicken Sie nun oben in der Symbolleiste auf **Run** (Ausführen). Dadurch wird der Job auf die Kamera heruntergeladen. Der Job kann nun ausgeführt und getestet werden. Damit die Demo mit Job-Änderung ausgeführt werden kann, muss dieser Job allerdings in **Slot 1** geladen werden. Klicken Sie oben in der Symbolleiste wieder auf die Schaltfläche zum Öffnen der Ansicht **Edit** (Bearbeiten).

Klicken Sie auf das Slot-Symbol und wählen Sie „Slot 1“ oder **New Slot** (Neuer Slot) aus, falls derzeit keine Slots belegt sind.



Der Job und die Bilder werden nun im Flash-Speicher der Kamera gespeichert.

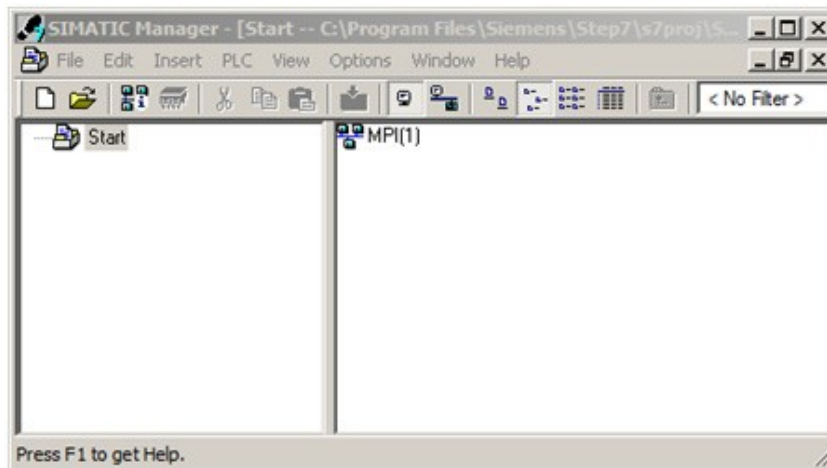
## Einrichtung von STEP 7

Die Anleitung in diesem Kapitel wurde auf der Basis der Software Siemens STEP 7 (Version 5.5 mit SP2) und einer ET200S-SPS (Katalognummer 6ES7 151-8AB01-0AB0, CPU-Version 3.2) erstellt. Getestet wurde mit einer 315-2 PN/DP-SPS (Katalognummer 6ES7-315-2EH13-0AB0, CPU-Version 2.6).

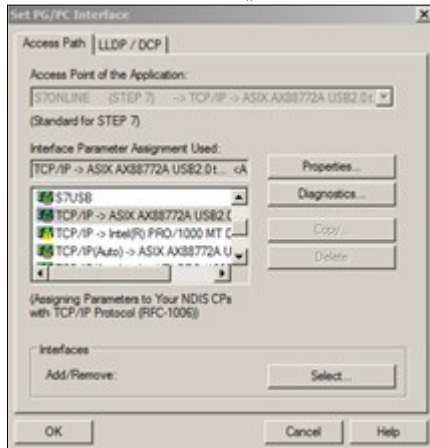
Starten Sie den SIMATIC-Manager vom Desktop aus.



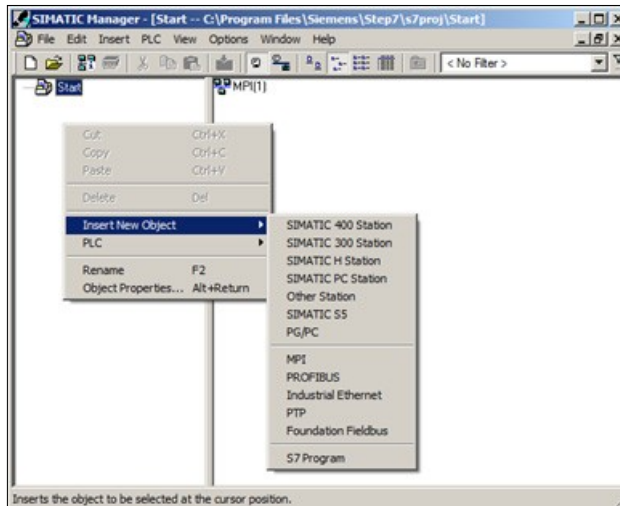
Erstellen Sie über das Menü **Projekt > Neu** ein neues Projekt. Legen Sie einen Speicherort für das Projekt fest, geben Sie einen Projektnamen ein und klicken Sie auf **OK**. Im Beispiel unten lautet der Projektname **Start**. Sobald das Projekt erstellt wurde, sehen Sie das unten abgebildete Fenster. Dabei handelt es sich um das Hauptfenster für das SPS-Programm und die Hardware-Einstellungen.



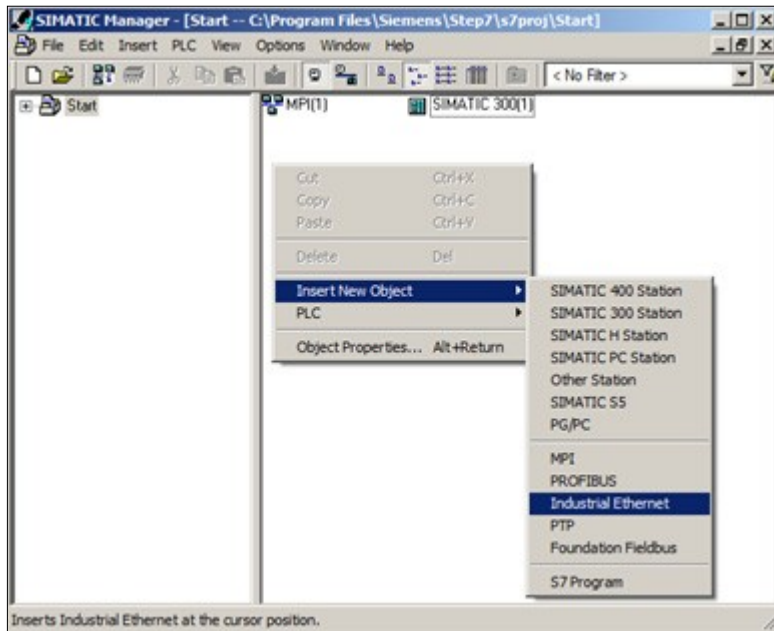
Falls der PC mehrere Netzwerkkarten hat, achten Sie darauf, für die SIMATIC-Software die richtige Karte auszuwählen. Wählen Sie im Menü „Extras“ und dann „PG/PC-Schnittstelle einstellen“ aus. Dadurch wird das Dialogfenster für die PG/PC-Schnittstelle mit einer Liste aller vorhandenen Netzwerkkarten geöffnet. Wählen Sie auf der Registerkarte „Zugriffsweg“ die NIC-Karte aus, die den Namensbestandteil „TCP/IP >“ enthält.



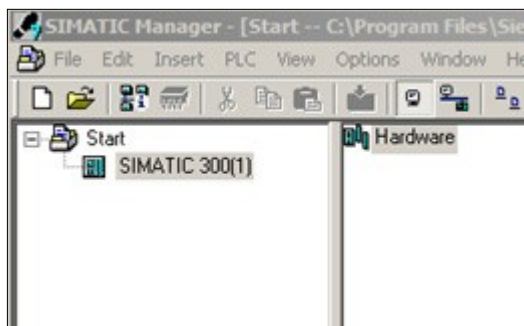
Klicken Sie mit der rechten Maustaste in das Dialogfenster des SIMATIC-Managers und wählen Sie „Neues Objekt einfügen“ aus. Hier können Sie nun die Basisstation festlegen. Wenn Sie beispielsweise eine ET200S konfigurieren möchten, wählen Sie die Station SIMATIC 300 aus, da diese auf der CPU-Serie 300 basiert.



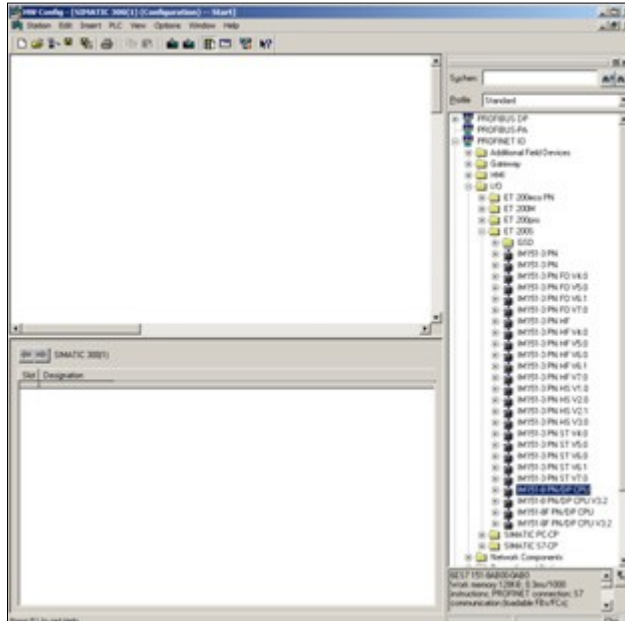
Nachdem Sie die gewünschte Station hinzugefügt haben, klicken Sie mit der rechten Maustaste und wählen Sie **Industrial Ethernet** aus.



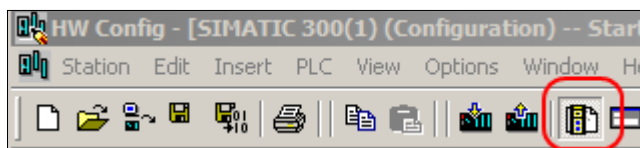
Maximieren Sie links in der Baumansicht den Knoten „Start“ und klicken Sie auf die Station. Auf der rechten Seite erscheint nun ein Hardware-Symbol.



Doppelklicken Sie auf **Hardware**, um das Dialogfenster **HW-Konfig** aufzurufen.



Vergewissern Sie sich, dass in der Symbolleiste oben das Symbol **Katalog** ausgewählt ist. Dadurch wird im rechten Fensterbereich eine Baumansicht mit allen vorhandenen Hardwaregeräten eingeblendet.



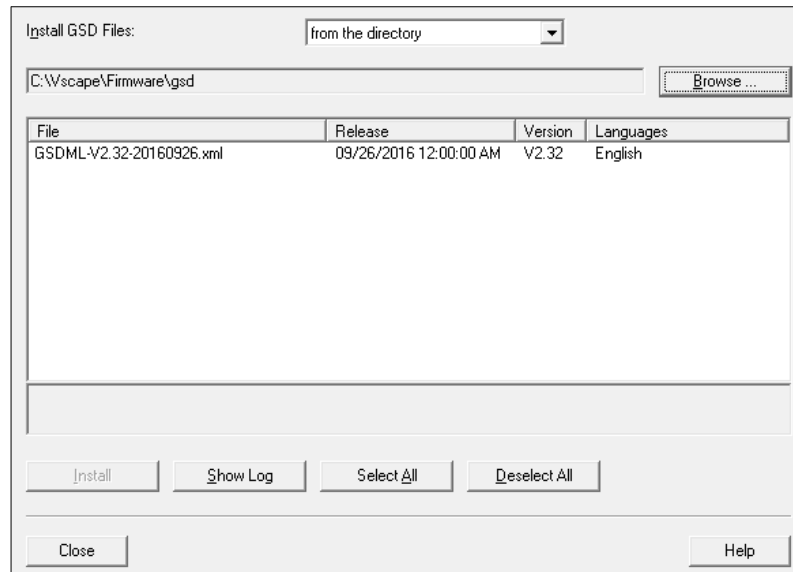
Falls die neueste GSDML-Datei noch nicht importiert wurde, folgen Sie der unten stehenden Anleitung zum Importieren.

### GSDML

- Installieren Sie die GSD-Datei über das Menü **Extras > GSD-Datei installieren**.
- Klicken Sie auf die Schaltfläche „Durchsuchen“ und wählen Sie den Ordner **\\di-soric\Vscape\Firmware\gsd\CS50 Sensor** aus, in dem sich die GSDML-Datei befindet.

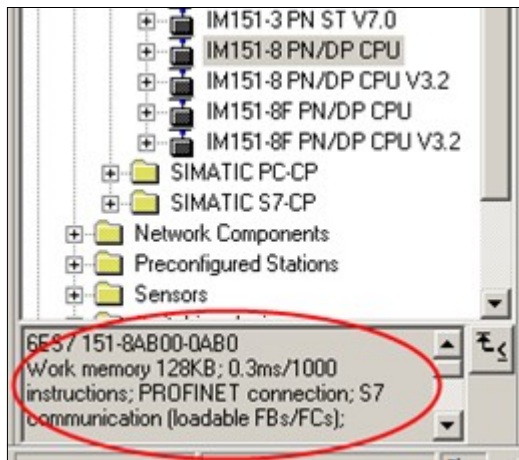


- Im mittleren Bereich des Bildschirms werden alle vorhandenen GSDML-Dateien angezeigt.



- Wählen Sie die gewünschte Datei aus und klicken Sie auf **Installieren**.
- Wenn Sie fertig sind, schließen Sie das Dialogfenster.
- Die Kamera müsste nun im Menü **Weitere Feldgeräte > Allgemein** unter „PROFINET I/O“ erscheinen.

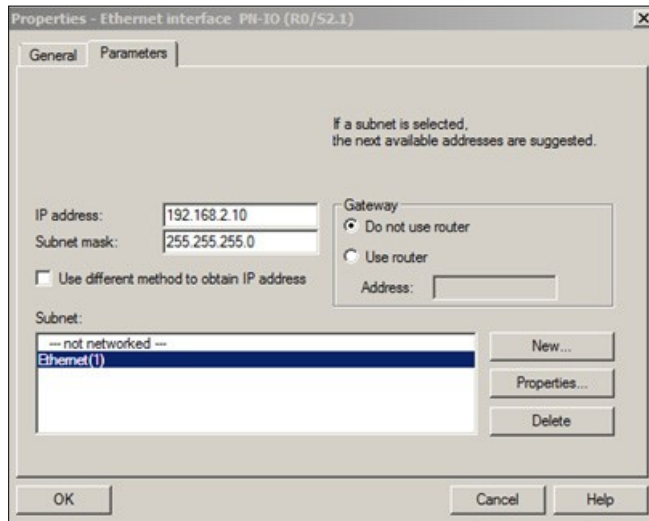
Fügen Sie die CPU aus der Katalogansicht hinzu, indem Sie sie herüberziehen oder darauf doppelklicken. Katalog- und Versionsnummer müssen exakt mit denen der SPS übereinstimmen. Die Katalognummer wird unten in der Ansicht eingeblendet.



Bei einigen CPU-Modulen muss zuvor ein allgemeines Rack hinzugefügt werden. Falls Ihre CPU ein Rack voraussetzt, werden Sie zum Hinzufügen eines Racks aufgefordert, bevor Sie das CPU-Modul einfügen können.

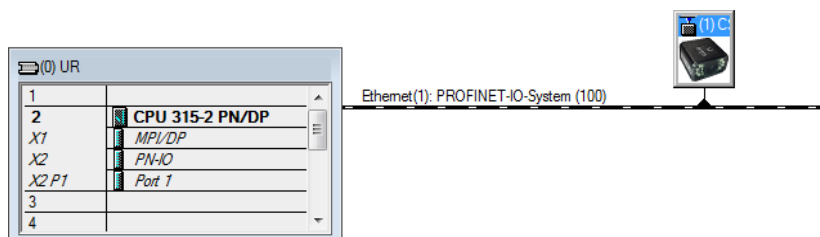
Sobald die CPU hinzugefügt wurde, erscheint ein Pop-up-Fenster, in dem Sie aufgefordert werden, die Eigenschaften bezüglich der IP-Informationen anzugeben.

Wählen Sie aus dem Listenfeld unten die Option **Ethernet(1)** aus und geben Sie die IP-Adresse der CPU ein.

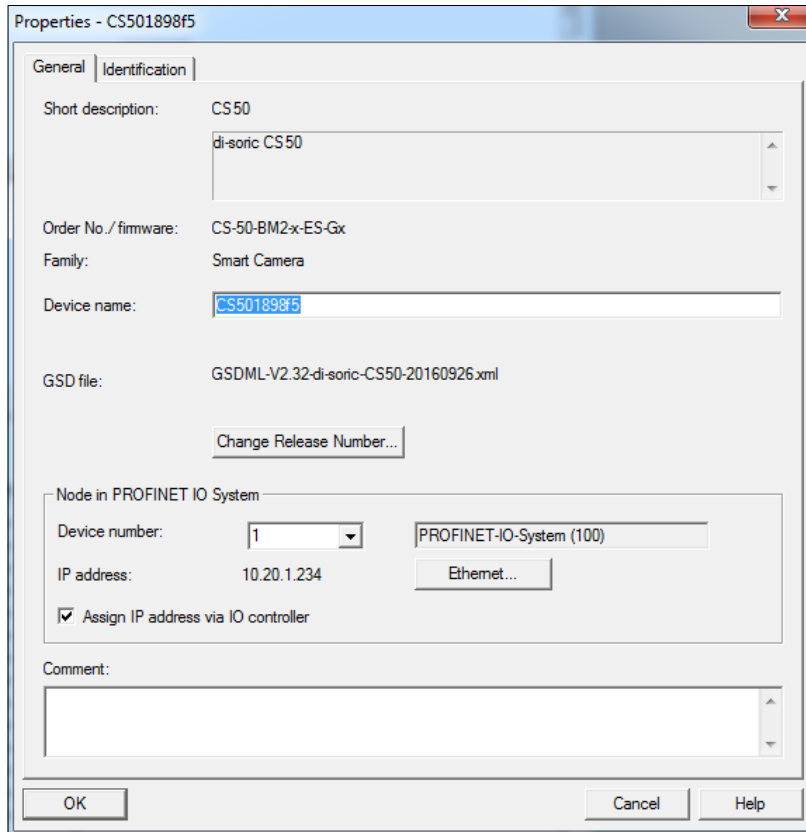


Nachdem Sie die CPU hinzugefügt haben, können Sie ggf. zusätzliche Erweiterungsmodule hinzufügen. Klicken Sie dazu im linken Abschnitt des Fensters auf die CPU. Daraufhin erscheint im unteren Abschnitt eine Liste aller vorhandenen Ports und Slots.

Fügen Sie die Kamera unter **PROFINET I/O > Weitere Feldgeräte > Allgemein > disoric** hinzu. Sie können die Kamera hinzufügen, indem Sie sie auf **Ethernet(1): PROFINET-I/O System (100)** ziehen oder indem Sie zunächst die Zeile **Ethernet(1)** auswählen und dann auf die Kamera doppelklicken. Sobald Sie die Kamera hinzugefügt haben, erscheint im Konfigurationsfenster ein entsprechendes Symbol.



Doppelklicken Sie auf das Symbol für die hinzugefügte Kamera, um ein Dialogfenster mit den Kameraeigenschaften zu öffnen. Im Feld „Gerätename“ können Sie entweder den vorhandenen Namen der Kamera beibehalten oder einen neuen, eindeutigen Namen eingeben.



Falls Sie einen neuen Namen eingegeben haben, muss das Gerät manuell aktualisiert werden. Wie das funktioniert, erfahren Sie im Abschnitt „Kameranamen aktualisieren“.

Klicken Sie auf das Kamerasymbol. Daraufhin öffnet sich das unten abgebildete Fenster, in dem Sie die Daten-Slot-Adresse zuordnen können. Achten Sie genau auf die richtigen Adresswerte, da diese Werte in der Demo-Anwendung benötigt werden. Angesichts einer unbegrenzten Zahl möglicher Kombinationen aus Modulen und Slot-Konfigurationen sind die Adresswerte bei jeder Einrichtung anders.

The screenshot shows the SIMATIC Manager configuration window for a SIMATIC 300 rack. A pop-up window titled '(0) UR' displays the following slot configuration:

1	
2	CPU 315-2 PN/DP
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
3	
4	

The main rack configuration shows a PROFINET-IO-System (100) connected to the rack. Below the rack, a table lists the hardware modules and their addresses:

Slot	M..	Order number	I address	Q address	Diagnostic address:	Comment
0	CS50	CS50-BM2-s-E			2043*	
X1	Interf				2042*	
X1 A	Port 1				2041*	
1	Status		0...1			
2	Contra			0...1		
3	Echo		264...265			
4	Echo			264...265		
5	Cmd A		256...259			
6	Cmd B			260...263		
7	Cmd A		260...263			
8	Cmd A			256...259		
9	State		266			
10						
11						
12						
13						
14						
15	Int In		267...286			
16						

In dieser Demo werden die Booleschen Daten **In** und **Float In** verwendet. Klicken Sie zum Auswählen rechts unter dem **PROFINET I/O CS50-Sensor** auf **Boolean In**. Dadurch wird der entsprechende Slot hervorgehoben. Ziehen Sie die Auswahl in diesen Slot. Wiederholen Sie diesen Schritt für „Float In“. Sie müssten nun folgende Tabelle sehen:

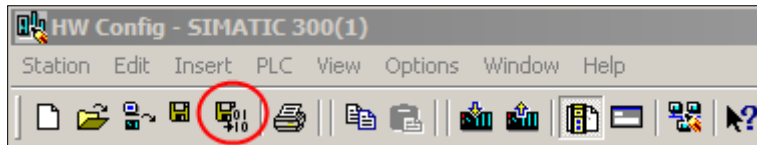
The screenshot shows the SIMATIC Manager interface for a SIMATIC 300(1) system. A hardware rack is displayed with the following components:

- Slot 1: UR
- Slot 2: CPU 315-2 PN/DP
- Slot X1: MPI/DP
- Slot X2: PN-IO
- Slot X2 P1: Port 1
- Slot 3: (Empty)
- Slot 4: (Empty)

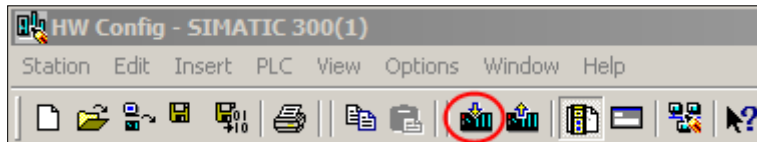
Below the rack, a table lists the modules and their addresses:

Slot	M...	Order number	I address	Q address	Diagnostic address:	Comment
0	CS50	CS50-BM2-x-E...			2043*	
X1	Interfa				2042*	
X1	Port 1				2041*	
1	Status		0...1			
2	Contra			0...1		
3	Echo		264...265			
4	Echo			264...265		
5	Cmd		256...259			
6	Cmd			260...263		
7	Cmd		260...263			
8	Cmd			256...259		
9	State		266			
10						
11						
12						
13						
14						
15	Int In		267...286			
16						

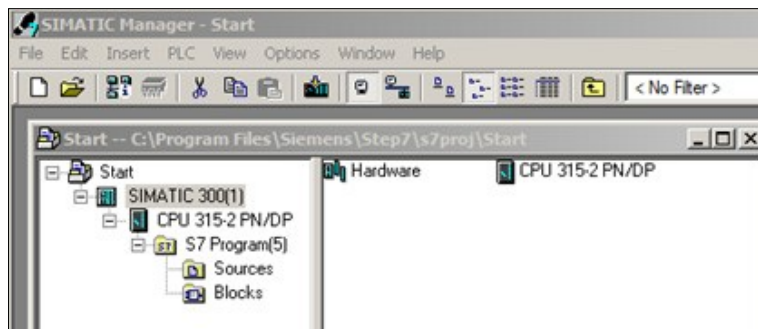
Nach Abschluss der Hardwarekonfiguration ist der nächste Schritt das Kompilieren und Herunterladen. Klicken Sie dazu in der Symbolleiste auf das Symbol zum Kompilieren und Speichern. Falls Ihnen bei der Konfiguration ein Fehler unterlaufen sein sollte, sehen Sie an dieser Stelle eine Warnmeldung.



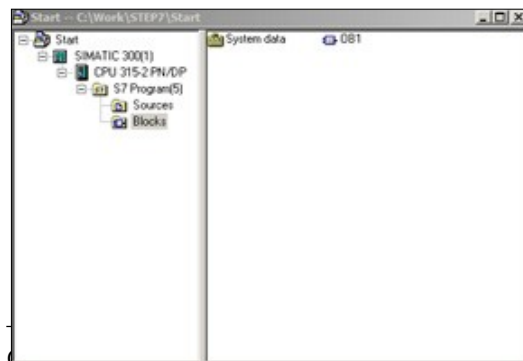
Klicken Sie nach dem Kompilieren in der Symbolleiste auf das Download-Symbol, damit die Daten an die SPS übermittelt werden.



Schließen oder minimieren Sie an dieser Stelle das Dialogfenster **HW-Konfig** und öffnen Sie das Fenster **SIMATIC-Manager**. Die CPU müsste nun neben dem Hardware-Symbol sowie in der Baumansicht links erscheinen. Klappen Sie in der Baumansicht den CPU-Knoten und dessen Unterknoten auf.



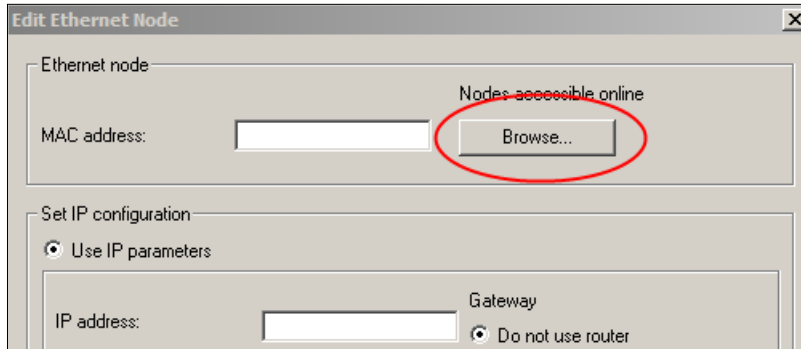
Um sich die Programmobjekte anzusehen, klicken Sie auf den Knoten „Blocks“ (Blöcke).



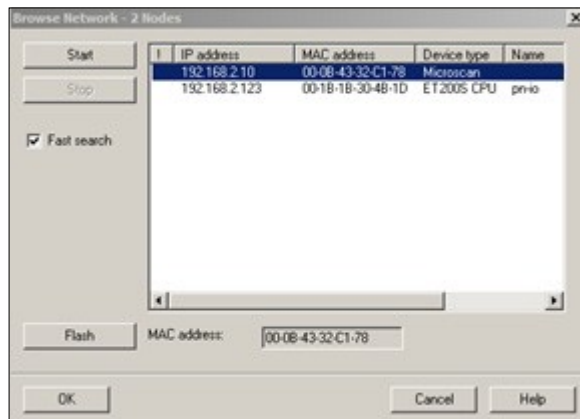
## Kameranamen aktualisieren

Im Dialogfenster „HW-Konfig“ öffnen Sie das Menü **SPS > Ethernet > Ethernet-Knoten bearbeiten** und wählen dort **Ethernet-Knoten bearbeiten** aus. Im Dialogfenster „SIMATIC-Manager“ öffnen Sie das Menü **SPS > Ethernet-Knoten bearbeiten** und wählen dort **Ethernet-Knoten bearbeiten** aus.

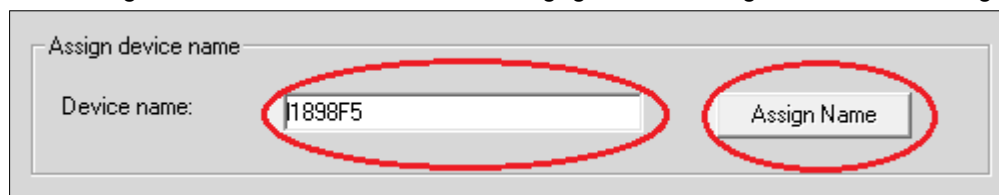
Um das Dialogfenster zur Auswahl des Knotens zu öffnen, klicken Sie auf **Durchsuchen**.



Wählen Sie die Kamera aus und klicken Sie auf **OK**.



Falls der **Gerätename** nicht mit dem Namen übereinstimmt, den Sie zuvor im Dialogfenster mit den Eigenschaften des CS50-Sensors eingegeben haben, geben Sie den richtigen

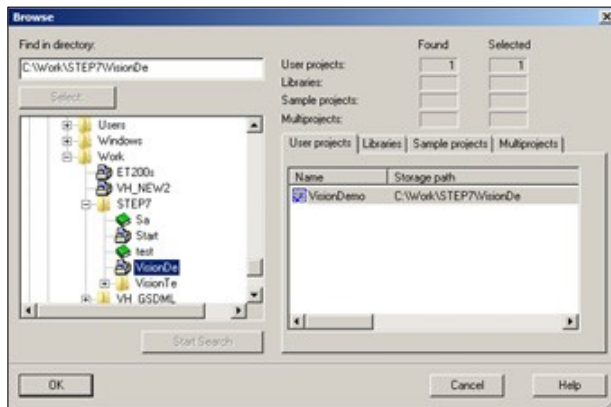


Gerätenamen ein. Klicken Sie dann auf **Name zuweisen**.

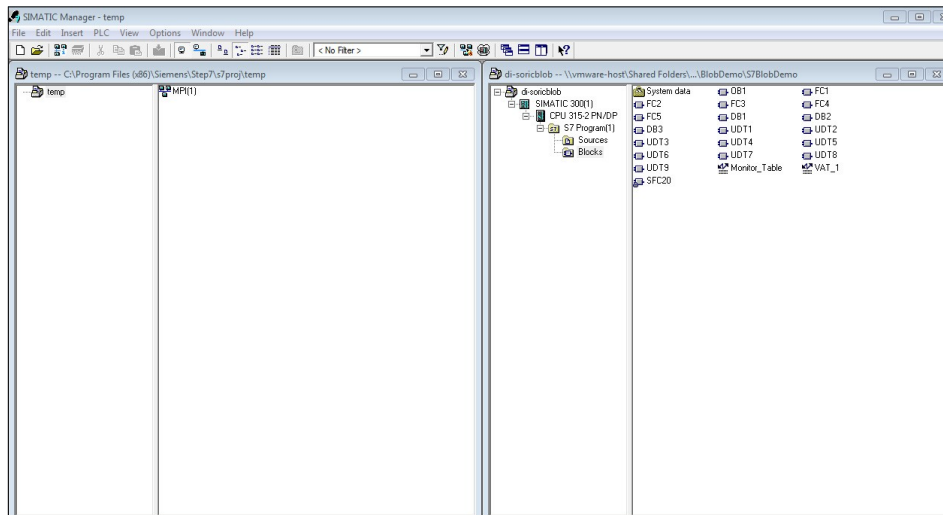


## Beispielprogramm importieren

Öffnen Sie das Beispielprogramm im SIMATIC-Manager über das Menü **Datei > Öffnen**. Klicken Sie auf **Durchsuchen** und suchen Sie das Programm BlobDemo. Es befindet sich im Ordner **\\di-soric\Vscope\Tutorials and Samples\CS50 Sensor\PROFINET Demo\Step7\_Demos\BlobDemo\S7BlobDemo\BlobDemo.s7p**.

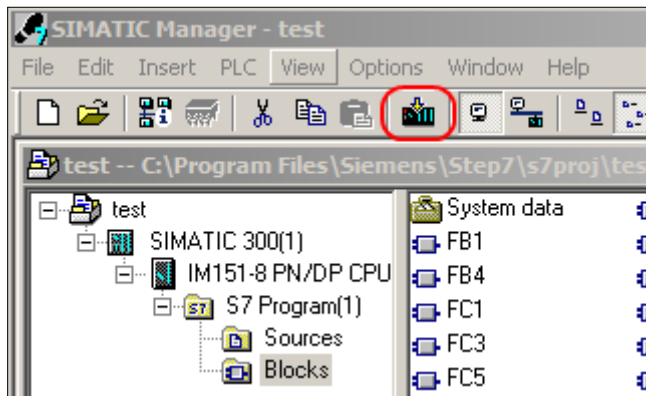


Teilen Sie den Bildschirm wie unten abgebildet, sodass auf einer Seite das Beispielprojekt und auf der anderen Seite das aktuelle Projekt zu sehen ist.

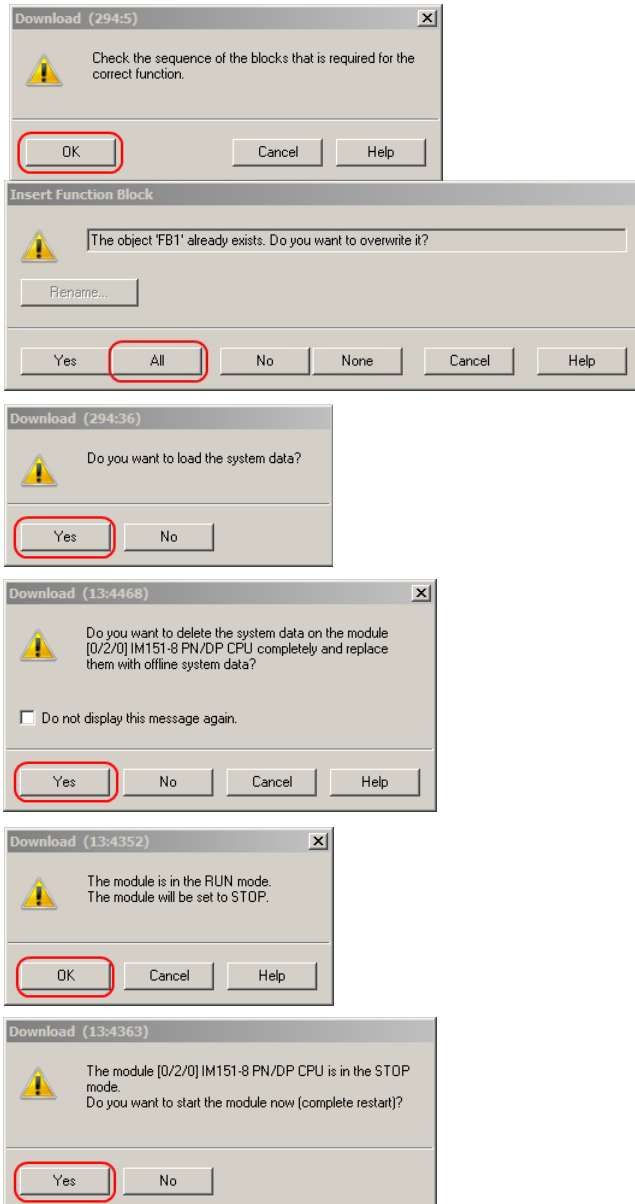


Wählen Sie die Objekte – außer dem Datenobjekt **System** – aus und kopieren Sie sie dann, indem Sie entweder mit der rechten Maustaste klicken und „Kopieren“ auswählen oder die Tastenkombination **Strg + C** verwenden. Fügen Sie die Objekte mit **Strg + V** oder Rechtsklick und der Option „Einfügen“ in das aktuelle Projekt ein. Wenn angezeigt wird, dass **OB1** bereits vorhanden ist, klicken Sie auf **Ja**, um dieses Objekt zu überschreiben. Sie haben nun das Beispielprogramm in das aktuelle Projekt importiert.

Wählen Sie im linken Bereich das Symbol **Blöcke** aus. Klicken Sie im Anschluss daran oben in der Symbolleiste auf das Download-Symbol. Dadurch werden die soeben kopierten Funktionen und Systemdaten auf die SPS heruntergeladen. Als Nächstes müssen Sie die Adressen aus der vorherigen Hardwareinstallation aktualisieren. Doppelklicken Sie dazu auf den Block **OB1**. Der **LAD/STL/FBD**-Editor wird daraufhin geöffnet. OB1 ist die Hauptroutine des SPS-Programms.



Scrollen Sie nach unten bis zum **Netzwerk 1**. Hier werden die Daten von der Kamera der lokalen Datenstruktur zugeordnet. Zur Übertragung der Eingangsdaten von der Kamera in die Programmstruktur dient die Funktion **FC1**. Klicken Sie auf die entsprechenden Nummern, um die Adressen der Hardware wie abgebildet anzugleichen. Klicken Sie in den sich öffnenden Pop-up-Fenstern auf folgende Schaltflächen:



**Hinweis:** Die Ansicht zur Adressenzuordnung des CS50-Sensormoduls, die von der Ansicht zur Hardwarekonfiguration abhängig ist, sieht so aus:

	Slot	Module	Order number	I address	Q address	Diagnost
	0	CS50189815	CS50-BM2-x-E			2043*
	X7	Interface				2042*
	X7.1	Port 1				2041*
0 - STATUS	1	Status		0..1		
	2	Control			0..1	
264 - ECHO_IN	3	Echo In		264..265		
	4	Echo Out			264..265	
CMDCODERS	5	Cmd Code Rest		256..259		
256 - LT	6	Cmd Code			260..263	
260 - CMDRET	7	Cmd Ret		260..263		
	8	Cmd Arg			256..259	
266 - STATE	9	State		266		
	10					
0 - CONTROL	11					
	12					
264 - ECHO_OUT	13					
	14					
260 - CMDCODE	15	Int In		267..286		
256 - CMDARG	16					

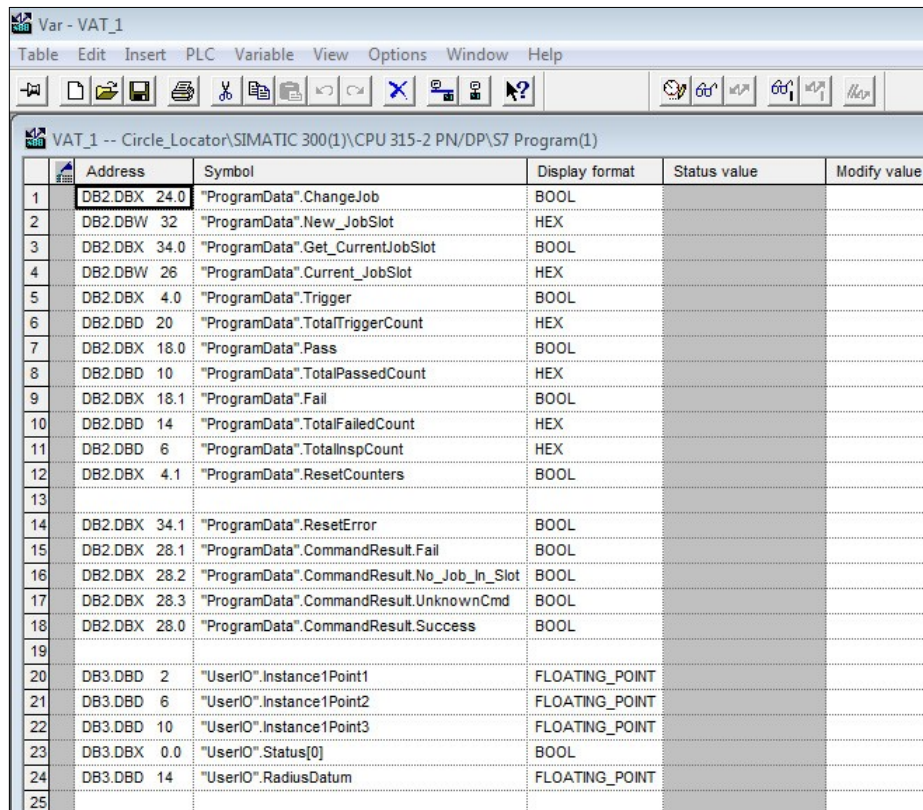
Damit alle Adressen übereinstimmen, müssen die Adressen auf der rechten Seite in **Netzwerk 1** kopiert werden. Beispiel: In der Hardwarekonfiguration wird für den Status die Adresse „0..1“ angezeigt. Kopieren Sie **0** und fügen Sie es unter „Netzwerk 1“ in den Wert für **STATUS** ein.

Zur korrekten Zuordnung der Ausgangsdaten wiederholen Sie diese Schritte für **Netzwerk 6**. Beachten Sie, dass alle Eingangsadressen in der Spalte **I-Adresse** und alle Ausgangsadressen in der Spalte **Q-Adresse** stehen. Speichern Sie die Daten auf dem PC und laden Sie sie dann auf die SPS herunter.

Führen Sie einen Master-Reset durch und schalten Sie die SPS in den Betriebsmodus. Vergewissern Sie sich, dass alle LEDs eine **korrekte Funktion** signalisieren.

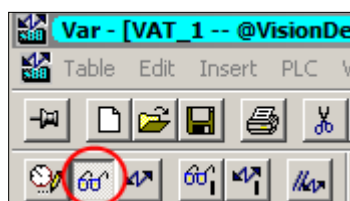
## Demo ausführen

Doppelklicken Sie im Dialogfenster **SIMATIC-Manager** auf das Symbol **VAT\_1**. Daraufhin wird ein Fenster mit der Variablen-tabelle für die Datentyp-Demo geöffnet. Maximieren Sie das darin angezeigte Dialogfenster, um den Sichtbereich zu vergrößern.



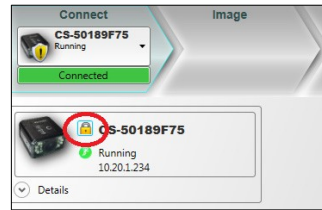
	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL		
2	DB2.DBW 32	"ProgramData".New_JobSlot	HEX		
3	DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL		
4	DB2.DBW 26	"ProgramData".Current_JobSlot	HEX		
5	DB2.DBX 4.0	"ProgramData".Trigger	BOOL		
6	DB2.DBD 20	"ProgramData".TotalTriggerCount	HEX		
7	DB2.DBX 18.0	"ProgramData".Pass	BOOL		
8	DB2.DBD 10	"ProgramData".TotalPassedCount	HEX		
9	DB2.DBX 18.1	"ProgramData".Fail	BOOL		
10	DB2.DBD 14	"ProgramData".TotalFailedCount	HEX		
11	DB2.DBD 6	"ProgramData".TotalInspCount	HEX		
12	DB2.DBX 4.1	"ProgramData".ResetCounters	BOOL		
13					
14	DB2.DBX 34.1	"ProgramData".ResetError	BOOL		
15	DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL		
16	DB2.DBX 28.2	"ProgramData".CommandResult.No_Job_In_Slot	BOOL		
17	DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL		
18	DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL		
19					
20	DB3.DBD 2	"UserIO".Instance1Point1	FLOATING_POINT		
21	DB3.DBD 6	"UserIO".Instance1Point2	FLOATING_POINT		
22	DB3.DBD 10	"UserIO".Instance1Point3	FLOATING_POINT		
23	DB3.DBX 0.0	"UserIO".Status[0]	BOOL		
24	DB3.DBD 14	"UserIO".RadiusDatum	FLOATING_POINT		
25					

Um eine Live-Verbindung zur SPS herzustellen, klicken Sie in der Symbolleiste auf das Symbol zum **Überwachen der Variable**. Dadurch werden die Daten von der SPS in das Dialogfenster übernommen. Die Titelleiste oben erscheint daraufhin blau, und in der Statusleiste unten wird der Ausführungsfortschritt in Form eines grünen Balkens angezeigt



Hier können nun folgende Vorgänge getestet werden:

Change Job	Switch jobs from one slot to another
Get Current Job Slot	Displays the slot of the currently running job
Trigger an Inspection	Causes the camera to cycle through its inspection.
Reset Counters	Resets the TotalTriggerCount, TotalPassedCount, TotalFailedCount, and TotalInspCount to 0, and also sends a command to the camera to reset its internal counters.
Reset an Error condition	If an error occurs during run time, this will reset it, so demo may be continued.



**Jobs ändern und aktuellen Job-Slot abrufen**

Folgen Sie der Anleitung zum Speichern eines Jobs im Kameraspeicher, die Sie im Abschnitt „Einrichtung der CS50-Software“ finden. Damit dieser Test korrekt durchgeführt werden kann, laden Sie einen weiteren Job in **Slot 2** der Kamera. Laden Sie beispielsweise den Job „Circle\_LocatorDemo“. Geben Sie die Gerätesteuerung frei, indem Sie auf die Registerkarte **Verbinden** klicken und dann, wie im Bild unten dargestellt, das Schlosssymbol zu **geschlossen** ändern. Klicken Sie anschließend erneut auf die Registerkarte **Ausführen**, damit der neue Job richtig ausgeführt wird.

Rufen Sie die „VAT\_1“-Elemente **ChangeJob** (Job ändern) und **New\_JobSlot** (Neuer Job-Slot) auf. Klicken Sie dann neben „New\_JobSlot“ in die Spalte **Modify value** (Wert bearbeiten), geben Sie den Wert „2“ ein und drücken Sie die **Eingabetaste**. Klicken Sie in der Zeile „ChangeJob“ auf den Wert in der Spalte „Modify value“ und geben Sie „TRUE“ (WAHR) ein. Klicken Sie in der Leiste auf das im Bild unten markierte Symbol.

Address	Symbol	Display format	Status value	Modify value
DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false	true
DB2.DBW 32	"ProgramData".New_JobSlot	HEX	W#16#0001	W#16#0002

Statt des aktuellen Jobs wird daraufhin der Job in **Slot 2** geöffnet. Um sicherzustellen, dass dadurch alle Änderungen gelöscht werden, die Sie in der Spalte **Modify value** (Wert bearbeiten) vorgenommen haben, klicken Sie in der Zeile **Get\_CurrentJobSlot** (Aktuellen Job-Slot abrufen) auf den Wert in der Spalte „Modify value“. Geben Sie **TRUE** (WAHR) ein und klicken Sie dann erneut auf das im Bild markierte Symbol.

Address	Symbol	Display format	Status value	Modify value
DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false	
DB2.DBW 32	"ProgramData".New_JobSlot	HEX	W#16#0002	
DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL	false	true
DB2.DBW 26	"ProgramData".Current_JobSlot	HEX	W#16#0002	

Der aktuelle Job erscheint nun rechts neben **Current\_JobSlot** (Aktueller Job-Slot). Sie können diese Vorgehensweise wiederholen, indem Sie wieder zu **Slot 1** wechseln und überprüfen, ob als aktueller Job-Slot tatsächlich **1** angezeigt wird.



## Inspektion auslösen und Zählerstände zurücksetzen

Vergewissern Sie sich, dass als aktueller Job **Slot 1** ausgewählt ist. Löschen Sie alle Änderungen, die Sie in der Spalte **Modify value** (Wert bearbeiten) vorgenommen haben. Falls Sie die Inspektion mitverfolgen möchten, vergewissern Sie sich, dass die CS50-Software aktiv und mit dem Gerät verbunden ist. Überprüfen Sie auch, ob der Job ausgeführt wird. Falls Sie die CS50-Software nicht ausführen möchten, können Sie die Trigger anhand der LEDs am Gerät überwachen. Ändern Sie den Wert in der Spalte „Trigger“ in **TRUE** (WAHR) und klicken Sie dann erneut auf das im Bild markierte Symbol.

	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false	
2	DB2.DBW 32	"ProgramData".New_JobSlot	HEX	W#16#0000	
3	DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL	false	
4	DB2.DBW 26	"ProgramData".Current_JobSlot	HEX	W#16#0000	
5	DB2.DBX 4.0	"ProgramData".Trigger	BOOL	false	
6	DB2.DBD 20	"ProgramData".TotalTriggerCount	HEX	DW#16#00000004	
7	DB2.DBX 18.0	"ProgramData".Pass	BOOL	true	
8	DB2.DBD 10	"ProgramData".TotalPassedCount	HEX	DW#16#00000002	
9	DB2.DBX 18.1	"ProgramData".Fail	BOOL	false	
10	DB2.DBD 14	"ProgramData".TotalFailedCount	HEX	DW#16#00000002	
11	DB2.DBD 6	"ProgramData".TotalInspCount	HEX	DW#16#00000004	
12	DB2.DBX 4.1	"ProgramData".ResetCounters	BOOL	false	
13					
14	DB2.DBX 34.1	"ProgramData".ResetError	BOOL	false	
15	DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL	false	
16	DB2.DBX 28.2	"ProgramData".CommandResult.No_Job_In_Slot	BOOL	false	
17	DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL	false	
18	DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL	false	
19					
20	DB3.DBD 2	"UserIO".Instance1Point1	FLOATING_POINT	486.0	
21	DB3.DBD 6	"UserIO".Instance1Point2	FLOATING_POINT	204.0	
22	DB3.DBD 10	"UserIO".Instance1Point3	FLOATING_POINT	179.0	
23	DB3.DBX 0.0	"UserIO".Status[0]	BOOL	false	
24	DB3.DBD 14	"UserIO".RadiusDatum	FLOATING_POINT	22.5	

Sie sehen nun verschiedene Änderungen wie in der Abbildung oben. Die Booleschen Werte für „Pass“ (Erfolg) und „Fail“ (Fehlschlag) signalisieren, ob die Inspektion erfolgreich war oder nicht. Außerdem werden die entsprechenden Zählerstände erhöht. Jedes Mal, wenn Sie auf die im Bild markierte Schaltfläche klicken, wird eine Inspektion gestartet und damit zum nächsten Bild gewechselt. Die Inspektionsergebnisse sind abwechselnd Erfolge und Fehlschläge.

**Zähler zurücksetzen**

Um die Zählerstände zurückzusetzen, löschen Sie alle Einstellungen in der Spalte „Modify value“ (Wert bearbeiten) und ändern Sie dann die entsprechende Einstellung in der Zeile **ResetCounters** (Zähler zurücksetzen) zu **TRUE** (WAHR). Klicken Sie anschließend wieder auf die im Bild markierte Schaltfläche. Alle im Bild unten blau markierten Zählerstände werden nun auf **0** zurückgesetzt.

Address	Symbol	Display format	Status value	Modify value
DB2.DBX 24.0	"ProgramData".ChangeJob	BOOL	false	
DB2.DBX 32	"ProgramData".New_JobSlot	HEX	W#16#0000	
DB2.DBX 34.0	"ProgramData".Get_CurrentJobSlot	BOOL	false	
DB2.DBX 26	"ProgramData".Current_JobSlot	HEX	W#16#0000	
DB2.DBX 4.0	"ProgramData".Trigger	BOOL	false	
DB2.DBD 20	"ProgramData".TotalTriggerCount	HEX	DV#16#00000000	
DB2.DBX 18.0	"ProgramData".Pass	BOOL	true	
DB2.DBD 10	"ProgramData".TotalPassedCount	HEX	DV#16#00000000	
DB2.DBX 18.1	"ProgramData".Fail	BOOL	false	
DB2.DBD 14	"ProgramData".TotalFailedCount	HEX	DV#16#00000000	
DB2.DBD 6	"ProgramData".TotalInspCount	HEX	DV#16#00000000	
DB2.DBX 4.1	"ProgramData".ResetCounters	BOOL	false	true
DB2.DBX 34.1	"ProgramData".ResetError	BOOL	false	
DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL	false	
DB2.DBX 28.2	"ProgramData".CommandResult.No_Job_In_Slot	BOOL	false	
DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL	false	
DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL	false	
DB3.DBD 2	"UserIO".Instance1Point1	FLOATING_POINT	486.0	
DB3.DBD 6	"UserIO".Instance1Point2	FLOATING_POINT	204.0	
DB3.DBD 10	"UserIO".Instance1Point3	FLOATING_POINT	179.0	
DB3.DBX 0.0	"UserIO".Status[0]	BOOL	false	
DB3.DBD 14	"UserIO".RadiusDatum	FLOATING_POINT	22.5	

Wenn Sie diese Einstellung erneut löschen, den Wert für „Trigger“ auf **TRUE** (WAHR) setzen und noch einmal auf die im Bild markierte Schaltfläche klicken, starten sowohl der SPS-Zähler als auch der Zähler der CS50-Software bei **1**.

**Fehler zurücksetzen**

Sollte einer der unten dargestellten „TRUE“-Werte einen Fehler zurückgeben, können Sie diesen Fehler beheben, indem Sie in der Zeile **ResetError** (Fehler zurücksetzen) „TRUE“ (WAHR) eingeben und dann wieder auf die Schaltfläche klicken. Setzen Sie den Wert dann auf **FALSE** (FALSCH) zurück und klicken Sie erneut auf die Schaltfläche. Sämtliche Fehler in den unten dargestellten Werten müssten nun beseitigt werden.

DB2.DBX 28.1	"ProgramData".CommandResult.Fail	BOOL	false
DB2.DBX 28.3	"ProgramData".CommandResult.UnknownCmd	BOOL	false
DB2.DBX 28.0	"ProgramData".CommandResult.Success	BOOL	false



# Serielle Befehle

Dieses Kapitel beschreibt die seriellen Befehle, die über den TCP- (Telnet-)Port, das AutoVISION-Terminal oder das HyperTerminal an die Kamera gesendet werden können.

**Serielle Befehlssyntax**

< > = Erforderliches Argument. Wird entsprechend ersetzt.

Beispiel:

**-u <DB\_Nutzername>** wird zu **-u av**, wobei **av** den Abschnitt **DB\_Nutzername** ersetzt.

| = Sich gegenseitig ausschließende Argumente. Wählen Sie eines dieser Argumente aus der Liste aus.

{ } = Trennt in Kombination mit | verschiedene Auswahlmöglichkeiten für ein Argument voneinander.

[ ] = Optionaler Parameter.

**Wichtig:** Sofern nicht anderweitig angegeben, antworten Befehle bei Erfolgen mit **!OK** und bei Fehlern mit **!ERROR**.

**GETIMAGE <-transfer=ymodem> [-format={jpg|png}] [-quality={0-100}] [-woi=links,oben,rechts,unten] [-inspection=n]**

Initiiert die serielle Übertragung des Inspektionsbildes (nur RS-232).

**Hinweis:** Mit diesem Befehl wird stets das jeweils letzte (aktuellste) Bild übertragen.

**-transfer=ymodem** ist derzeit nicht optional – nur das YMODEM-Protokoll wird unterstützt.

**-format={jpg|png}** definiert das Bildformat. Falls dieser Befehl übergangen wird, wird automatisch JPG als Bildformat festgelegt.

**-quality=n** legt die JPG-Komprimierungsqualität auf **n** fest. Dieser Wert darf höchstens 100 betragen. Falls kein Wert festgelegt wird, ist der Standardwert **80**.

**Hinweis:** Das Bildformat PNG gewährleistet eine verlustfreie Bildkomprimierung. Falls **format** als **PNG** definiert wird, erübrigt sich daher die Einstellung **quality**.

**woi=links,oben,rechts,unten** definiert einen rechteckigen Bereich des Bildes, der in das Ausgabebild aufgenommen wird. Falls dieser Befehl übergangen wird, wird der gesamte Inhalt des Bildspeichers ausgegeben.

**-inspection=n** definiert die Inspektion, von der ein Bild abgerufen werden soll. Das Bild stammt aus dem ersten Schnappschuss dieser Inspektion. Falls hier kein Wert festgelegt wird, wird das Bild aus der ersten Inspektion übernommen, die einen Schnappschuss enthält. Im unten stehenden Beispiel wird ein Bild von der Kamera mit folgenden Einstellungen abgerufen:

**Protokoll:** YMODEM; **Format:** PNG; **Qualität:** nicht zutreffend; **Inspektion:** zweite Inspektion.

**GETIMAGE –transfer=ymodem –format=png –inspection=2**

Im unten stehenden Beispiel wird ein Bild von der Kamera mit folgenden Einstellungen abgerufen: **Protokoll:** YMODEM; **Format:** JPG (Standardeinstellung); **Qualität:** 50; **Inspektion:** erste Inspektion (Standardeinstellung). **GETIMAGE –transfer=ymodem –quality=50**

---

## ONLINE

Alle Inspektionen werden gestartet.

## OFFLINE

Alle Inspektionen werden beendet.

## TRIGGER

Eine Inspektion wird ausgelöst.

### vt [n]

Es wird ein Impuls an einen virtuellen E/A-Punkt gesendet und dadurch eine Inspektion ausgelöst.

Beispiel:

#### vt 1

Der Impuls **VIO1** wird ausgegeben. Die Inspektion wird ausgeführt, sofern **VIO1** für die Inspektion als Trigger festgelegt wurde.

Der VIO-Index, falls angegeben, muss innerhalb des erlaubten Bereichs für virtuelle E/A-Punkte in Visionscape liegen. Die virtuelle E/A-Linie wird zunächst hoch, dann niedrig gesetzt.

Falls kein VIO-Index angegeben wird, wird automatisch VIO1 angenommen.

**Ausgabefehler:** Es wird **!ERROR** ausgegeben, gefolgt von der Fehlerursache.

Beispiel:

#### **!ERROR No such trigger**

Diese Fehlermeldung („Kein derartiger Trigger“) wird angezeigt, wenn der angegebene Index **n** außerhalb der Spanne der virtuellen Trigger liegt.

## REBOOT [-noload]

Das Gerät wird neu gestartet.

**-noload** = BOOT-Job wird nicht geladen.

**MEMAVAIL [-cp]**

Der verfügbare Speicherplatz wird für das Gerät oder den Koprozessor ausgegeben.

**MEMCONTIG [-cp]**

Der maximale Speicherblock wird für das Gerät oder den Koprozessor ausgegeben.

**MEMFRAGS [-cp]**

Speicherfragmente werden für das Gerät oder den Koprozessor ausgegeben.

**MEMINFO [-cp] [-v]**

Die Speicherübersicht mit Angaben zum verfügbaren Speicherplatz (avail), zum Speicherblock (contig) und zu Speicherfragmenten (frags) wird für das Gerät oder den Koprozessor ausgegeben. Wird als Text ausgegeben.

**VERSION**

Die Visionscape-Softwareversion wird ausgegeben.

**JOBSAVE [-slot=<n>]**

Der aktuelle Job wird in Slot *n* gespeichert.

**JOBLOAD [-slot=<n> [-r]**

Der Job aus Slot *n* wird geladen.

-r = Inspektionen werden gestartet.

**JOBDELETE {[-slot=<n>|-all}**

Der Job in Slot *n* wird gelöscht. Bei **-all** werden alle Jobs gelöscht.

**Wichtig:** Der aktuell im Kameraspeicher geladene Job wird nicht gelöscht.

---

## **JOBINFO [-slot=*n*] [-v]**

Die Job-Zusammenfassung oder Informationen zu Slot *n* werden abgerufen.

Falls für **JOBINFO** kein Argument angegeben wird, wird eine Liste aller Jobs im Gerät ausgegeben.

**-v = *n*** wird als Text ausgegeben. Bei dieser Option wird angezeigt, wie viel Speicherplatz durch Löschen des Jobs frei würde. Außerdem werden der gesamte Speicherplatz und der freie Speicherplatz angezeigt.

## **JOBBOOT [-slot=]*n*>**

Legt *n* als Job-Slot zum Hochfahren fest (nur RS-232).

## **JOBDOWNLOAD <-transfer=*y*modem>**

Der mittels der Übertragungsmethode gepackte .avz-Job wird heruntergeladen (nur RS-232).

## **SET <Tag-Name> <Wert>**

Damit wird der Wert eines globalen Tags definiert.

Der Tag-Name muss einem der vom Gerät unterstützten Tags entsprechen.

Der Wert kann Leerzeichen enthalten.

Der Befehl wird durch ein Wagenrücklauf- und/oder Zeilenvorschubzeichen beendet.

Der Wert kann eine Liste kommagetrennter Werte sein, mit der eine Tag-Sequenz definiert wird: Beim Senden von **SET int1 1, 2, 3** werden folgende Werte festgelegt: int1 = 1, int2 = 2, int3 = 3.

Der AVP-Dienst ermöglicht durch Angabe eines Schrägstriches (/) im symbolischen Namenspfad die Festlegung von Schritt- und Bezugspunktdateien aus dem Job-Baum. Bei **SET avp/insp1/snapshot1/acq1/gain 2.0**-Pfadern spielt die Groß-/Kleinschreibung keine Rolle. Sofern diese Pfade einmalig sind, ist keine vollständige Qualifikation erforderlich.

Mit **SET avp/acq1/gain 2.0** wird der gleiche Verstärkungswert festgelegt, sofern es nur eine Aufnahme gibt.

**Steuer-Tags des AVP-Dienstes, wie z. B. START, STOP und TRIGGER, verhalten sich wie Tastschalter. SET avp.start 1 entspricht dem Befehl ONLINE. avp.start wird jedes Mal sofort wieder zurückgesetzt und hat daher immer den Wert 0.**

**Anzeige erfolgreich ausgeführter Befehle:** Wurde der Befehl erfolgreich ausgeführt, wird **!OK** angezeigt, gefolgt von einem Echo des Befehls.

Beispiel: **!OK SET matchstring1**

**Ausgabefehler:** Bei einem Fehler wird **!ERROR** ausgegeben, gefolgt von der Fehlerursache.

Beispiel:

**!ERROR Tag matchstring66 not found**

## GET {Tag-Name|Dienst|Dienst.Tag-Name}

Damit wird der Wert eines globalen Tags abgerufen.

Der Tag-Name muss einem der vom Gerät unterstützten Tags entsprechen.

Der Befehl wird durch ein Wagenrücklauf- und/oder Zeilenvorschubzeichen beendet.

Wenn Sie einen Einzelwert aus einem Array wie z. B. **GET int1** abrufen möchten, müssen Sie einen Index angeben. Wird kein Index angegeben, wird das gesamte Array als Liste mit kommagetrennten Werten ausgegeben.

Mit **Get {Tag-Name|Dienst.Tag-Name|Dienst}** können Sie den Wert eines Tags im globalen Datendienst abrufen. Falls Sie den Wert eines Tags aus einem anderen Dienst abrufen möchten, setzen Sie Namen dieses Dienstes als Präfix vor den Tag-Namen. Beispiel: ein Befehl nach dem Muster **GET <Dienst.Tag-Name>**.

Mit **GET eip.input** können Sie z. B. die EIP-Eingangs-Assembly abrufen.

Der AVP-Dienst ermöglicht durch Angabe eines Schrägstriches (*/*) im symbolischen Namenspfad den Abruf von Schritt- und Bezugspunktdateien aus dem Job-Baum. Bei **GET avp/insp1/snapshot1/status**-Pfadern spielt die Groß-/Kleinschreibung keine Rolle. Sofern diese Pfade einmalig sind, ist keine vollständige Qualifikation erforderlich.

---

Mit `GET avp/snapshot1/status` wird das gleiche Ergebnis ausgegeben, sofern es nur eine Inspektion gibt.

Wenn `GET avp/snapshot1` auf einen Schritt angewandt wird, werden die Werte aller Bezugspunkte ausgegeben.

**Anzeige erfolgreich ausgeführter Befehle:** Wurde der Befehl erfolgreich ausgeführt, wird der im Tag gespeicherte Wert ausgegeben.

Beispiel: **ABCD**

**Ausgabefehler:** Bei einem Fehler wird „!ERROR“ ausgegeben, gefolgt von der Fehlerursache.

Beispiel:

**!ERROR Tag matchstring66 not found**

## INFO [Tag-Name|Dienst]

Damit werden Informationen zu einem Tag oder Dienst abgerufen.

Werden für **INFO** keine Argumente angegeben, wird eine Liste aller Dienste abgerufen.

Mit **INFO <Dienst>** wird eine Liste aller Tags in diesem Dienst abgerufen.

Mit **INFO <Dienst.Tag-Name>** werden die Attribute des angegebenen Tags sowie eine Liste aller Sub-Tags abgerufen.

Der AVP-Dienst ermöglicht durch Angabe eines Schrägstriches (*/*) im symbolischen Namenspfad den Abruf von Schritt- und Bezugspunktdateien aus dem Job-Baum. Bei **INFO avp/insp1/snapshot1/status**-Pfadern spielt die Groß-/Kleinschreibung keine Rolle. Sofern diese Pfade einmalig sind, ist keine vollständige Qualifikation erforderlich.

Mit **INFO avp/snapshot1/status** wird das gleiche Ergebnis ausgegeben, sofern es nur eine Inspektion gibt.

Wenn **INFO avp/snapshot1** auf einen Schritt angewandt wird, werden die Eigenschaften des Schritts, eine Liste aller untergeordneten Bezugspunkte und eine Liste aller Teilschritte ausgegeben. Teilschritte sind an einem nachgestellten Schrägstrich zu erkennen.

## **QUERYAUTOCAL**

Mit diesem Befehl werden die fotometrischen Einstellungen abgerufen: Verstärkung, Belichtung und Fokus.

## **AUTOCAL**

Die fotometrischen Einstellungen (Verstärkung, Belichtung und Fokus) werden automatisch kalibriert.

## **TARGET {0|1|off|on}**

Ziel-LEDs werden ein- oder ausgeschaltet.

## **CHECKSUM {BOOT | KERNEL | BOOTPARAM}**

Eine Prüfsumme für einen bestimmten Teil des Systems wird abgerufen.

## **HELP**

Eine Liste aller seriellen Befehle mit Beschreibung der korrekten Syntax und der jeweiligen Funktion wird abgerufen.