




# BIJLAGENBUNDEL EINDVERSLAG AFSTUDEREN

TESTSYSTEEM PERS COMMUNICATIE

Jarno Meijer	2221557
Bedrijf:	Goss Contiweb
Bedrijfsbegeleiders:	Paul Hoeijmakers Paul Slaats
Begeleidend docent:	Frank van Gennip
Plaats, datum:	Boxmeer, 8-6-2017

  
**CONTIWEB**

  
**Fontys**

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

## Inhoudsopgave

Bijlage 1: Kosten en onderdelenoverzicht

Bijlage 2: Harting stekkers


Bijlage 3: Eplan tekeningen en schetsen


Bijlage 4: Software

Bijlage 5: Communicatie cyclus

Bijlage 6: Test overzicht

Bijlage 7: Testplan en resultaten

	Document: AFST_TB Bijlagenbundel Kaft en inhoudsopgave v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 1 van 1


	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---


## Bijlage 1: Kosten en onderdelenoverzicht

In de figuren B1-1 en B1-2 is de huidige bestellijst weergegeven. In tegenstelling tot het kostenoverzicht in het tussenverslag is hier enkel de kabelset van de CD en CR weergegeven. De onderdelen die horen bij de andere kabelsets en nog niet besteld gaan worden zijn weergegeven in figuur B1-3. Veel kosten zijn nog onbekend of onzeker, de redenen hiervoor zullen uitgewerkt worden in de volgende alinea.

In figuur B1-1 is de main switch groen gemarkeerd, in de onderdelen database staat hiervoor een prijs van €0,-. Dit wil zeggen dat dit onderdeel nog nooit besteld is en dat daardoor de prijs nog niet bekend is, deze kosten zullen dus nog toegevoegd moeten worden. In het geel zijn twee kosten gemarkeerd, de behuizing en de arbeidskosten. De kosten van de behuizing zijn direct overgenomen van voorgaand onderzoek, deze onderdelen staan niet in de onderdelen database en de exacte kosten zijn daarom onbekend. Bij de arbeidskosten is met een intern tarief van €50,- per uur gerekend, het is echter onduidelijk hoeveel uren er al exact aan dit project gewerkt zijn en daarnaast is nog niet begonnen aan de realisatie. In figuur B1-1 is "Bestelling Harting" rood gemarkeerd, deze bestelling is uitgebreid weergegeven in figuur B1-2. In figuur B1-2 zijn de Harting en Pflitsch onderdelen rood gemarkeerd. Deze onderdelen staan niet in de onderdelen database van Goss Contiweb en de prijzen hiervan zijn dus nog niet bekend. Voor de Harting onderdelen zijn prijzen van Mouser gebruikt, dit is een particuliere verkoper en de prijzen die hier gehanteerd worden zullen hoger zijn dan wanneer Goss Contiweb deze direct bij Harting inkoopt. Daarnaast is in overleg met de bedrijfsbegeleiders een ruim bedrag van €15,- per Pflitsch wartel aangehouden.


De daadwerkelijke kosten zullen waarschijnlijk dus lager uit vallen dan €6.656,35. De totaalprijs van de overige kabelsets zal ook lager uit vallen dan de momenteel berekende €1.293,59.


	Document: AFST_TB Bijlage 1 Kosten en onderdelenoverzicht v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 1 van 4

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

WH	Aantal	Productnummer	Bedrijf	Naam	Omschrijving	stukprijs	Prijs
				<b>IO station</b>			€ 752,46
WH095832307	1	6ES7155-6AA00-0BN0	SIEMENS	Interface Module	SIMATIC ET 200SP	€ 184,61	€ 184,61
WH095832309	1	6ES7193-6BP00-0DA0	SIEMENS	AO Base		€ 16,40	€ 16,40
WH095832334	1	6ES7135-6HB00-0DA1	SIEMENS	AO Module		€ 169,74	€ 169,74
WH095832311	7	6ES7193-6BP20-0BB0	SIEMENS	DO Base		€ 12,78	€ 89,46
WH095832306	7	6ES7132-6HD00-0BB1	SIEMENS	DO Module		€ 41,75	€ 292,25
				<b>HMI</b>			€ 1.807,80
WH095833712	1	6AV7240-0BB14-2AA0	SIEMENS	HMI/IPC	477D, 15"	€ 1.807,80	€ 1.807,80
				<b>Power Supply</b>			€ 164,17
WH095832357	1	6EP1334-3BA10	SIEMENS	Power supply		€ 119,37	€ 119,37
WH095200717	1	PISA 11.CLASS2	PULSPower	Pisa	4x3,7A	€ 44,80	€ 44,80
				<b>PLC</b>			€ 836,45
WH095833717	1	6AU1410-2AD00-0AA0	SIEMENS	Controll Unit	D410-2 DP/PN	€ 731,49	€ 731,49
WH095832783	1	6AU1400-2PA23-0AA0	SIEMENS	CF card		€ 104,96	€ 104,96
WH095833681	1	6SL3055-0AA00-3PA1	SIEMENS	Terminal Module	SINAMICS S120 TM41	€ 193,67	€ 193,67
				<b>Noodstop</b>			€ 30,65
WH095262298	1	M22-PVT45PQ (121462)	EATON	Drukknop		€ 13,21	€ 13,21
WH095262297	1	M22-XGPVQ (231273)	EATON	Beschermkap		€ 6,40	€ 6,40
WH095262283	1	M22-A (216374)	EATON	Adapter		€ 1,00	€ 1,00
WH095262292	2	M22-CK02 (107899)	EATON	N/C Contact	(Dubbel)	€ 5,02	€ 10,04
WH095200834	1	53111	EATON	Main Switch	3P	€ -	€ -
				<b>Behuizing</b>			€ 269,53
Ontbreekt	1	MPG06R5	ELDON	Lessenaar		209,31	209,31
Ontbreekt	1	LCR471	ELDON	Wielen	2 met rem, 2 zonder	60,22	60,22
				<b>Stekkers/kabels</b>			€ 451,62
WH095832670	0,5m	2 CE6147	LAPP	RJ45 (kort)		€ 3,87	€ 7,74
WH095833668	0,6m	6SL3060-4AU00-0AA0	SIEMENS	Drive CLiQ			€ 12,80
				Bestelling HARTING			€ 406,60
WH095145362	7	1020,2	CONTA-CLIP	Klemkaart 2x2	Doorlusbaar	82,75/100	€ 5,79
WH095147916	2	2828	CONTA-CLIP	Eindstop		65,76/100	€ 1,32
WH095147417	5	2087	CONTA-CLIP	verbinding 2x		24,56/100	€ 1,23
WH095147433	4	2089	CONTA-CLIP	verbinding 4x		41,67/100	€ 1,67
WH095145394	2	2269,2	CONTA-CLIP	Klemkaart 3x2	Doorlusbaar	149,36/100	€ 2,99
WH095147028	1	2699,2	CONTA-CLIP	Afscherming		€ 0,24	€ 0,24
WH095239003	1	4564	CONTA-CLIP	DIN TS35 rails	2m	€ 6,11	€ 6,11
WH095042021	1	1418080	ABL-SURSUM	Stekkerkop		€ 1,59	€ 1,59
WH095076283	5m	1026725	HARTING	Voedings kabel	3 aders, mag anders	0,71/m	€ 3,55
				<b>Arbeidskosten</b>			€ 2.150,00
	40			Assemblage uren		€ 50,00	€ 2.000,00
	3			Voorbereidings uren		€ 50,00	€ 150,00
				<b>Totaal</b>			€ 6.656,35


Figuur B1-1 Totale kostenoverzicht


	Document: AFST_TB Bijlage 1 Kosten en onderdelenoverzicht v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 2 van 4

	Teststelsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-----------------------------------	---

Groep	WH	Aantal	Productnummer	Bedrijf	Naam	Omschrijving	stukprijs	Prijs
H1 huis	NB	1	<a href="#">09 30 024 0301</a>	HARTING	H1 Casing huis		€ 25,76	€ 25,76
	NB	1	<a href="#">09 14 024 0313</a>	HARTING	H1 Frame f		€ 22,22	€ 22,22
	NB	3	<a href="#">09 14 017 3101</a>	HARTING	H1 17 pin f		€ 5,07	€ 15,21
	NB	1	<a href="#">09 14 003 3101</a>	HARTING	H1 3 pin f		€ 6,51	€ 6,51
	NB	1	<a href="#">09 14 001 3111</a>	HARTING	H1 adapter f		€ 5,76	€ 5,76
	NB	1	<a href="#">09 14 008 3116</a>	HARTING	H1 8 pin (com) f			€ -
	NB	1	<a href="#">09 14 000 9950</a>	HARTING	H1 Dummie		€ 4,08	€ 4,08
H1 hood	NB	1	<a href="#">19 30 024 0427</a>	HARTING	H1 Casing M32		€ 28,45	€ 28,45
	NB	1	<a href="#">09 14 024 0303</a>	HARTING	H1 Frame m		€ 22,22	€ 22,22
	NB	2	<a href="#">09 14 017 3001</a>	HARTING	H1 17 pin m		€ 5,56	€ 11,12
	NB	1	<a href="#">09 14 001 3011</a>	HARTING	H1 adapter m		€ 6,44	€ 6,44
	NB	1	<a href="#">09 14 008 3016</a>	HARTING	H1 8 pin (com) m			€ -
H2 huis	NB	1	<a href="#">09 30 010 0301</a>	HARTING	H2 Casing huis		€ 22,60	€ 22,60
	NB	1	<a href="#">09 14 010 0313</a>	HARTING	H2 Frame f		€ 21,14	€ 21,14
	NB	1	<a href="#">09 14 001 4721</a>	HARTING	H2 Ethernet f		€ 43,38	€ 43,38
	NB	1	<a href="#">09 14 002 3101</a>	HARTING	H2 Profibus f		€ 7,68	€ 7,68
H2 hood	NB	1	<a href="#">19 30 010 1421</a>	HARTING	H2 Casing M25		€ 14,52	€ 14,52
	NB	1	<a href="#">09 14 010 0303</a>	HARTING	H2 Frame m		€ 21,14	€ 21,14
	NB	1	<a href="#">09 14 002 3001</a>	HARTING	H2 Profibus m		€ 7,31	€ 7,31
Kabels H1	WH095094713	1*xm	<a href="#">1026711</a>	LAPP	5 aders		0,65/m	€ 5,20
	WH095094715	1*xm	<a href="#">1026713</a>	LAPP	12 aders		1,47/m	€ 11,76
Kabels H2	WH095832220	1*xm	6XV1830-0EH10	SIEMENS	Profibus	kabel	0,76/m	€ 6,08
	WH095832203	2	6ES7972-0BB11-0XA0	SIEMENS	Profibus	connector	€ 34,01	€ 68,02
Wartel H1	NB	1	23254dm1x7/1x10,5	PFLITSCH	7+10,5		€ 15,00	€ 15,00
Wartel H2	NB	1	22553st9	PFLITSCH	6,5/9,5		€ 15,00	€ 15,00
								€ 406,60

Figuur B1-2 Kostenoverzicht "Bestelling HARTING", stekkerset voor de CD en CR


	Document: AFST_TB Bijlage 1 Kosten en onderdelenoverzicht v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 3 van 4

	Teststelsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-----------------------------------	---

Groep	WH	Aantal	Productnummer	Bedrijf	Naam	Omschrijving	stukprijs	Prijs
H1 hood	NB	1	<a href="#">19 30 024 0467</a>	HARTING	H1 Casing 2M32		€ 50,22	€ 50,22
	NB	7	<a href="#">19 30 024 0427</a>	HARTING	H1 Casing M32		€ 28,45	€ 199,15
	NB	2	<a href="#">19 30 024 0428</a>	HARTING	H1 Casing M40		€ 33,79	€ 67,58
	NB	10	<a href="#">09 14 024 0303</a>	HARTING	H1 Frame m		€ 22,22	€ 222,20
	NB	17	<a href="#">09 14 017 3001</a>	HARTING	H1 17 pin m		€ 5,56	€ 94,52
	NB	4	<a href="#">09 14 003 3001</a>	HARTING	H1 3 pin m		€ 7,53	€ 30,12
H2 hood	NB	2	<a href="#">19 30 010 1421</a>	HARTING	H2 Casing M25		€ 14,52	€ 29,04
	NB	2	<a href="#">09 14 010 0303</a>	HARTING	H2 Frame m		€ 21,14	€ 42,28
	NB	2	<a href="#">09 14 001 4623</a>	HARTING	H2 Ethernet m		€ 5,75	€ 11,50
	NB	1	<a href="#">09 14 002 3001</a>	HARTING	H2 Profibus m		€ 7,31	€ 7,31
Kabels H1	WH095094712	3*xm	<a href="#">1026710</a>	LAPP	4 aders		0,54/m	€ 12,96
	WH095094715	2*xm	<a href="#">1026713</a>	LAPP	12 aders		1,47/m	€ 23,52
	WH095094716	3*xm	<a href="#">1026714</a>	LAPP	18 aders		2,18/m	€ 52,32
	WH095094717	5*xm	<a href="#">1026715</a>	LAPP	25 aders		3,04/m	€ 121,60
	-	4*xm	-	-	2300	niet meerekenen	-	-
Kabels H2	WH095832669	2*xm	CE6154	LAPP	RJ45	Ethernet (10m)	13,08	€ 26,16
	WH095832220	1*xm	6XV1830-0EH10	SIEMENS	Profibus	kabel	0,76/m	€ 6,08
	WH095832203	3	6ES7972-0BB11-0XA0	SIEMENS	Profibus	connector	€ 34,01	€ 102,03
Wartels H1	NB	2	23254dm1x7/1x10,5	PFLITSCH	7+10,5		€ 15,00	€ 30,00
	NB	3	23254st13	PFLITSCH	9,0\13,0		€ 15,00	€ 45,00
	NB	1	24056dm2x15	PFLITSCH	2*15		€ 15,00	€ 15,00
	NB	1	24055dm1x7/1x14	PFLITSCH	7+14		€ 15,00	€ 15,00
	NB	2	23254st16	PFLITSCH	11,5\15,5		€ 15,00	€ 30,00
	NB	2	23254um2x7	PFLITSCH	2*7		€ 15,00	€ 30,00
Wartels H2	NB	1	22553st8	PFLITSCH	5\8		€ 15,00	€ 15,00
	NB	1	22553dm1x6/1x8	PFLITSCH	6+8		€ 15,00	€ 15,00
								€ 1.293,59

Figuur B1-3 Kostenoverzicht overige stekkersets

(Goss Contiweb, 2017)(Harting, 2017) (Lapp, 2017) (Mouser, 2017) (Pulspower, 2017)  
 (Siemens, 2017)

	Document: AFST_TB Bijlage 1 Kosten en onderdelenoverzicht v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 4 van 4


## Bijlage 2: Harting stekkers

De droger “Dryer\_Manroland\_Pecom\_HW” heeft maar liefst 42 IO aansluitingen, daartegenover hebben de rollenwisselaars “CD” en “CR” slechts 8 IO aansluitingen. Wanneer deze aansluitingen stuk voor stuk aangesloten moeten worden op het testsysteem zal dit veel tijd kosten en daarnaast een redelijke kans op fouten veroorzaken. Om deze reden zijn tijdens voorgaande stage “Han-Modular” stekkers van Harting naar voren gekomen. Dit zijn stekkers bestaande uit meerdere modules, deze modules kunnen gekozen worden uit een breed assortiment. Hierdoor is het mogelijk om veel verschillende aansluitingen te combineren in één stekker. Deze stekker zal in één keer aangesloten kunnen worden en enkel de benodigde aansluitingen zullen aangesloten zijn.

In deze bijlage zullen veel verschillende termen gebruikt gaan worden. Om dit overzichtelijk te houden worden deze termen in deze alinea uitgewerkt. In figuur B2-1 is weergegeven wat bedoeld word met een aantal van deze begrippen. Met housing en hood zullen de weergegeven componenten bedoeld worden inclusief modules. Met stekker zal de combinatie van housing en hood bedoeld worden. Met kabelset wordt de combinatie van alle onderdelen tussen het testsysteem en de rollenwisselaar of droger bedoeld, dit bevat de hood, de wartels en alle kabels die door deze wartels gaan.



Figuur B2-1 Leesrichting: Housing, Hood, Modules, Wartel, Kabel

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---


Het ontwerpen van de Harting stekkers is in verschillende stappen gegaan. Als eerste moest er gekeken worden welke modules er nodig en beschikbaar waren. In totaal zijn er 42 IO aansluitingen nodig, er wordt hier uitgegaan van 0,75 mm<sup>2</sup> aders. De module met de meeste pinnen voor 0,75 mm<sup>2</sup> aders heeft 17 pinnen, van deze module zijn er 3 nodig om het gewenste aantal IO aansluitingen te krijgen. Daarnaast hebben sommige machines een 230 volt voeding nodig, hiervoor zijn 3 aders van 1,5 mm<sup>2</sup> nodig. Er is gekozen voor één module met exact 3 pinnen. Ook moet bij een deel van de rollenwisselaars een encoder signaal gegeven worden, hiervoor wordt een module met 8 communicatie pinnen gebruikt. Verder zijn er nog één Ethernet en één Profibus module nodig, de Profibus module heeft het formaat van twee gewone modules. In figuur B2-5 is aangegeven welke pinnen verbindingen er bij welke machine aanwezig moeten zijn, groen geeft aan dat de verbinding nodig is, rood geeft aan dat de module niet in de kabelset gebruikt gaat worden. Geel geeft aan dat de module aanwezig is maar dat de verbinding niet gebruikt gaat worden. In figuur B2-3 is overzichtelijker weergegeven welke modules er in welke kabelset aanwezig zullen zijn.

De tweede stap was het opdelen naar twee Harting stekkers, er is ruimte nodig voor 8 modules en er kunnen maximaal 6 modules in één stekker. Bij het verdelen van de modules is gekeken naar welke verdeling resulteerde in het kleinste aantal kabelsets. Hieruit kwamen verschillende combinaties waarbij hetzelfde aantal kabelsets over bleven. Met deze combinaties is gekeken naar welke wartels gebruikt konden worden, hieruit bleek dat bij één verdeling aanzienlijk minder wartels nodig had. Dit was wanneer de ethernet en Profibus in een aparte stekker gecombineerd werden. De eerste stekker bestaat uit de drie IO modules met 17 pinnen, de module met 3 pinnen voor de 230 volt voeding en de module met 8 pinnen voor het encoder signaal. Er is echter geen Harting stekker voor vijf modules, daarom is hiervoor een stekker voor zes modules gekozen. De plek voor de zesde module zal opgevuld worden met een dummie module, dit is een module zonder enige vorm van aansluiting. Een afbeelding van deze stekker is weergegeven in figuur B2-2. De tweede stekker bestaat uit de ethernet en de Profibus module, deze zullen samen een stekker van 3 modules delen. Voor deze tweede stekker zijn slechts 4 combinaties mogelijk, Ethernet en Profibus, enkel Ethernet, enkel Profibus en geen van beide. Bij de laatste combinatie is geen kabelset nodig. De verschillende types van de "CD" en "CR" hebben hetzelfde aantal en type verbindingen en zullen dus dezelfde kabelset gebruiken, hierdoor zijn er voor stekker één 11 kabelsets nodig. als we daar de 3 kabelsets van stekker twee bij op tellen zijn er in totaal 14 verschillende kabelsets nodig.


De verschillende kabelsets zijn weergegeven in figuur B2-3, hierbij is aangegeven welke modules er in welke kabelsets aanwezig zullen zijn en welke machines de kabelsets gaan gebruiken.



*Figuur B2-2 Harting stekker één, Leesrichting: 8 pinnen COM, dummie, 3 pinnen 1,5 mm<sup>2</sup>, (3x) 17 pinnen 0,75 mm<sup>2</sup>*


	Document: AFST_TB Bijlage 2 Harting Stekkers v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 2 van 5



	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

De meeraderige kabels die bij Goss Contiweb gebruikt worden zijn van LAPP. Omdat deze kabels flexibel moeten zijn om goed aangesloten te kunnen worden en daarnaast goed tegen beweging moeten kunnen is gekozen voor de “Ölflex Chain 809”. Deze worden aangeboden tot en met 25 aders. Voor elke kabelset zijn de kabels zo gekozen dat er geen of zo weinig mogelijk aders ongebruikt blijven. Bij verschillende kabelsets zijn er ongebruikte aders, deze aders zullen aan de machinekant weggewerkt moeten worden. Een voorbeeld zou zijn om deze aders af te knippen tot waar de buitenkabel gestript is of door deze met een krimpkous tegen de zijkant van de kabel te bevestigen. Hiermee zou het duidelijk moeten zijn dat deze kabels niet aangesloten moeten worden en het voorkomt ook dat deze problemen als kortsluiting veroorzaken.

Het aantal kabels en de dikte hiervan verschilt per kabelset, hierop zullen de hoods en wartels aangepast moeten worden. Bij de keuze van de hoods is rekening gehouden met de voorkeuren van de bedrijfsbegeleiders. De wartels moesten aan de bovenkant van de hoods zitten en deze hoods moeten met double locking levers vastgezet worden in de housings, een voorbeeld hiervan is te zien in figuur 5-4. Hierbij bleven enkele opties over, één gat van M32, één gat van M40 of twee gaten van M40. Hierbij zijn wartels van Pflitsch gekozen, Pflitsch biedt een breed assortiment aan wartels voor enkele en meerdere kabels. De kabelset voor de “Dryer\_NG\_GOSS\_C550/C700” heeft drie kabels dus het gebruik van wartels voor meerdere kabels is noodzakelijk. Twee wartels voor elk één kabel zijn duurder dan één wartel voor twee kabels, om deze reden is er voor gekozen zo veel mogelijk kabels door één wartel te laten gaan. Door de dikte en hoeveelheid van de kabels is het enkel bij de “Dryer\_NG\_GOSS\_C550/C700” niet mogelijk deze door één wartel te laten gaan, dit is hierdoor ook de enige die de hood met twee gaten zal gaan gebruiken.

	Document: AFST_TB Bijlage 2 Harting Stekkers v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 3 van 5

	17 Pin (1)	17 Pin (2)	17 Pin (3)	3 Pin	8 Pin (COM)	
Kabelset 1.1						CD & CR
Kabelset 1.2						NG C550/C700
Kabelset 1.3						NG CL
Kabelset 1.4						NG CL CPS
Kabelset 1.5						Manroland RTE
Kabelset 1.6						Manroland HW
Kabelset 1.7						Basic
Kabelset 1.8						KBA E/A
Kabelset 1.9						ecochill
Kabelset 1.10						FB
Kabelset 1.11						FB_goss
	Ethernet	Profibus				
Kabelset 2.1			CD & CR			
Kabelset 2.2			NG C550/C700/CL/ CL CPS & FB_goss			
Kabelset 2.3			Manroland RTE/HW, Basic, KBA E & FB			


Figuur B2-3 Modules per kabelset

In figuur B2-4 zijn de gekozen kabels met diktes en bijbehorende wartels weergegeven. Bij de kabelsets 1.# is eerst het benodigde aantal aders weergegeven. Daarna is dit omhoog afgerond naar het eerste beschikbare aantal aders per kabel. Kabel 3 is de 3 aderige 1,5 mm<sup>2</sup> kabel voor de 230 volt voeding. Kabel 4 is een 5 aderige kabel voor het encoder signaal. Hierna zijn de gaten van de gekozen hoods per kabelset weergegeven en daarnaast de bijbehorende wartel van Pflitsch. Voor de kabelsets 2.# zijn enkel de diktes van de kabels opgenomen met de bijbehorende gaten en wartels.

	Kabel 1	Kabel 2	Kabel 1*	Kabel 2*	Dikte 1	Dikte 2	Dikte 3	Dikte 4	Gat 1	Wartel 1	Gat 2	Wartel 2
Kabelset 1.1	4		4		6.5			7.1	M25	23254um2x7		
Kabelset 1.2	22	9	25	12	14.8	10.3	6.9		M25	23254dm1x7/1x10,5	M25*	23254st16
Kabelset 1.3	10		12		10.3		6.9		M25	23254dm1x7/1x10,5		
Kabelset 1.4	9		12		10.3		6.9		M25	23254dm1x7/1x10,5		
Kabelset 1.5	15		18		12.2				M25	23254st13		
Kabelset 1.6	25	17	25	25	14.8	14.8			M40	24056dm2x15		
Kabelset 1.7	14		18		12.2				M25	23254st13		
Kabelset 1.8	24	4	25	4	14.8	6.5			M40	24055dm1x7/1x14		
Kabelset 1.9	11		12		10.3				M25	23254st13		
Kabelset 1.10	21		25		14.8				M25	23254st16		
Kabelset 1.11	2		4		6.5		6.9		M25	23254um2x7		
	Dikte Ethernet	Dikte Profibus	Gat	Wartel								
Kabelset 2.1	6,5	-	M25	22553st8								
Kabelset 2.2	6,5	8	M25	22553dm1x6/1x8								
Kabelset 2.3	-	8	M25	22553st9								

Figuur B2-4 Kabeldiktes en bijbehorende wartels


Alle onderdelen van het testsysteem zijn terug te vinden in Bijlage 1: Kosten en onderdelenoverzicht. Hierin is het bovenste gedeelte van figuur B1-2 gereserveerd voor de gebruikte Harting onderdelen.


	<b>Teststelsysteem Pers Communicatie</b>	<b>Goss Contiweb B.V.</b> <b>Ir. Wagterstraat 10</b> <b>5831 AZ, Boxmeer</b>
--	--	--

Harting 1	Pin	Functie	Module	CD & CR	NG C550/C700	NG CL	NG CL CPS	Manroland RTE	Manroland HW	Basic	KBA E	KBA A	Ecochill	FB	FB_goss
				Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel
IO1 (17 pin)	1.1	DI	PLC	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	-	1.1	1.1	1.1
	1.2	DI	PLC	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	-	1.2	1.2	1.2
	1.3	DI	PLC	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	-	1.3	1.3	1.3
	1.4	DI	PLC	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	-	1.4	1.4	1.4
	1.5	DI	PLC	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	-	1.5	1.5	1.5
	1.6	DI	PLC	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	-	1.6	1.6	1.6
	1.7	DI	PLC	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	-	1.7	1.7	1.7
	1.8	DI	PLC	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	-	1.8	1.8	1.8
	1.9	DI	PLC	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	-	1.9	1.9	1.9
	1.10	DI	PLC	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	-	1.10	1.10	1.10
	1.11	DI	PLC	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11	-	1.11	1.11	1.11
	1.12	DO	IO station	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	-	1.12	1.12	1.12
	1.13	DO	IO station	1.13	1.13	1.13	1.13	1.13	1.13	1.13	1.13	-	1.13	1.13	1.13
	1.14	DO	IO station	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	-	1.14	1.14	1.14
	1.15	DO	IO station	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	-	1.15	1.15	1.15
	1.16	DO	IO station	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	-	1.16	1.16	1.16
	1.17	DO	IO station	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17	-	1.17	1.17	1.17
IO2 (17 pin)	2.1	DO	IO station	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18	-	1.18	1.18	1.18
	2.2	DO	IO station	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19	-	1.19	1.19	1.19
	2.3	DO	IO station	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	-	1.20	1.20	1.20
	2.4	DO	IO station	1.21	1.21	1.21	1.21	1.21	1.21	1.21	1.21	-	1.21	1.21	1.21
	2.5	24I	PLC	1.22	1.22	1.22	1.22	1.22	1.22	1.22	1.22	-	1.22	1.22	1.22
	2.6	24O	Klemmenstrook	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	-	1.23	1.23	1.23
	2.7	24O	Klemmenstrook	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	-	1.24	1.24	1.24
	2.8	AO	IO station	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	-	1.25	1.25	1.25
	2.9	DI	PLC	2.1	2.1	2.1	2.1	2.1	2.1	2.1	2.1	-	2.1	2.1	2.1
	2.10	DI	PLC	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	-	2.2	2.2	2.2
	2.11	DI	PLC	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	-	2.3	2.3	2.3
	2.12	DI	PLC	2.4	2.4	2.4	2.4	2.4	2.4	2.4	2.4	-	2.4	2.4	2.4
	2.13	DI	PLC	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	-	2.5	2.5	2.5
	2.14	DO	IO station	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	-	2.6	2.6	2.6
	2.15	DO	IO station	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	-	2.7	2.7	2.7
	2.16	DO	IO station	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	-	2.8	2.8	2.8
	2.17	DO	IO station	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	-	2.9	2.9	2.9
IO3 (17 pin)	3.1	DO	IO station	2.10	2.10	2.10	2.10	2.10	2.10	2.10	2.10	-	2.10	2.10	2.10
	3.2	DO	IO station	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11	-	2.11	2.11	2.11
	3.3	DO	IO station	2.12	2.12	2.12	2.12	2.12	2.12	2.12	2.12	-	2.12	2.12	2.12
	3.4	DO	IO station	2.13	2.13	2.13	2.13	2.13	2.13	2.13	2.13	-	2.13	2.13	2.13
	3.5	DO	IO station	2.14	2.14	2.14	2.14	2.14	2.14	2.14	2.14	-	2.14	2.14	2.14
	3.6	DO	IO station	2.15	2.15	2.15	2.15	2.15	2.15	2.15	2.15	-	2.15	2.15	2.15
	3.7	DO	IO station	2.16	2.16	2.16	2.16	2.16	2.16	2.16	2.16	-	2.16	2.16	2.16
	3.8	DO	IO station	2.17	2.17	2.17	2.17	2.17	2.17	2.17	2.17	-	2.17	2.17	2.17
	3.9	24O	IO station	2.18	2.18	2.18	2.18	2.18	2.18	2.18	2.18	-	2.18	2.18	2.18
	3.10	24I	IO station	2.19	2.19	2.19	2.19	2.19	2.19	2.19	2.19	-	2.19	2.19	2.19
	3.11	24I	IO station	2.20	2.20	2.20	2.20	2.20	2.20	2.20	2.20	-	2.20	2.20	2.20
	3.12	24I	IO station	2.21	2.21	2.21	2.21	2.21	2.21	2.21	2.21	-	2.21	2.21	2.21
	3.13	F	-	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22	-	2.22	2.22	2.22
	3.14	F	-	2.23	2.23	2.23	2.23	2.23	2.23	2.23	2.23	-	2.23	2.23	2.23
	3.15	F	-	2.24	2.24	2.24	2.24	2.24	2.24	2.24	2.24	-	2.24	2.24	2.24
	3.16	F	-	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25	-	2.25	2.25	2.25
	3.17	-	-	NC	NC	NC	NC	NC	NC	NC	NC	-	NC	NC	NC
IO4 (6 pin)	4.1	2300	Klemmenstrook	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	-	3.1	3.1	3.1
	4.2	2300	Klemmenstrook	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	-	3.2	3.2	3.2
	4.3	2300	Klemmenstrook	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	-	3.3	3.3	3.3
Encoder module	5.1	Encoder	TM41	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	-	4.1	4.1	4.1
	5.2	Encoder	TM41	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2	-	4.2	4.2	4.2
	5.3	Encoder	TM41	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	-	4.3	4.3	4.3
	5.4	Encoder	TM41	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	-	4.4	4.4	4.4
	5.5	Encoder	TM41	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	-	4.5	4.5	4.5
	5.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	5.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	5.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Harting 2	pin	Functie	Module	CD & CR	NG C550/C700	NG CL	NG CL CPS	Manroland RTE	Manroland HW	Basic	KBA E	KBA A	Ecochill	FB	FB_goss
				Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel	Kabel
RJ45 Module	6.1	-	HMI	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1
	6.2	-	HMI	5.2	5.2	5.2	5.2	5.2	5.2	5.2	5.2	5.2	5.2	5.2	5.2
	6.3	-	HMI	5.3	5.3	5.3	5.3	5.3	5.3	5.3	5.3	5.3	5.3	5.3	5.3
	6.4	-	HMI	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4
	6.5	-	HMI	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5
	6.6	-	HMI	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6
	6.7	-	HMI	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7	5.7
	6.8	-	HMI	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.8
Profibus module	7.1	-	PLC	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1
	7.2	-	PLC	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2

Figuur B2-5 Pinnen en modules per machine


(Goss Contiweb, 2017)(Harting, 2017) (Lapp, 2017)

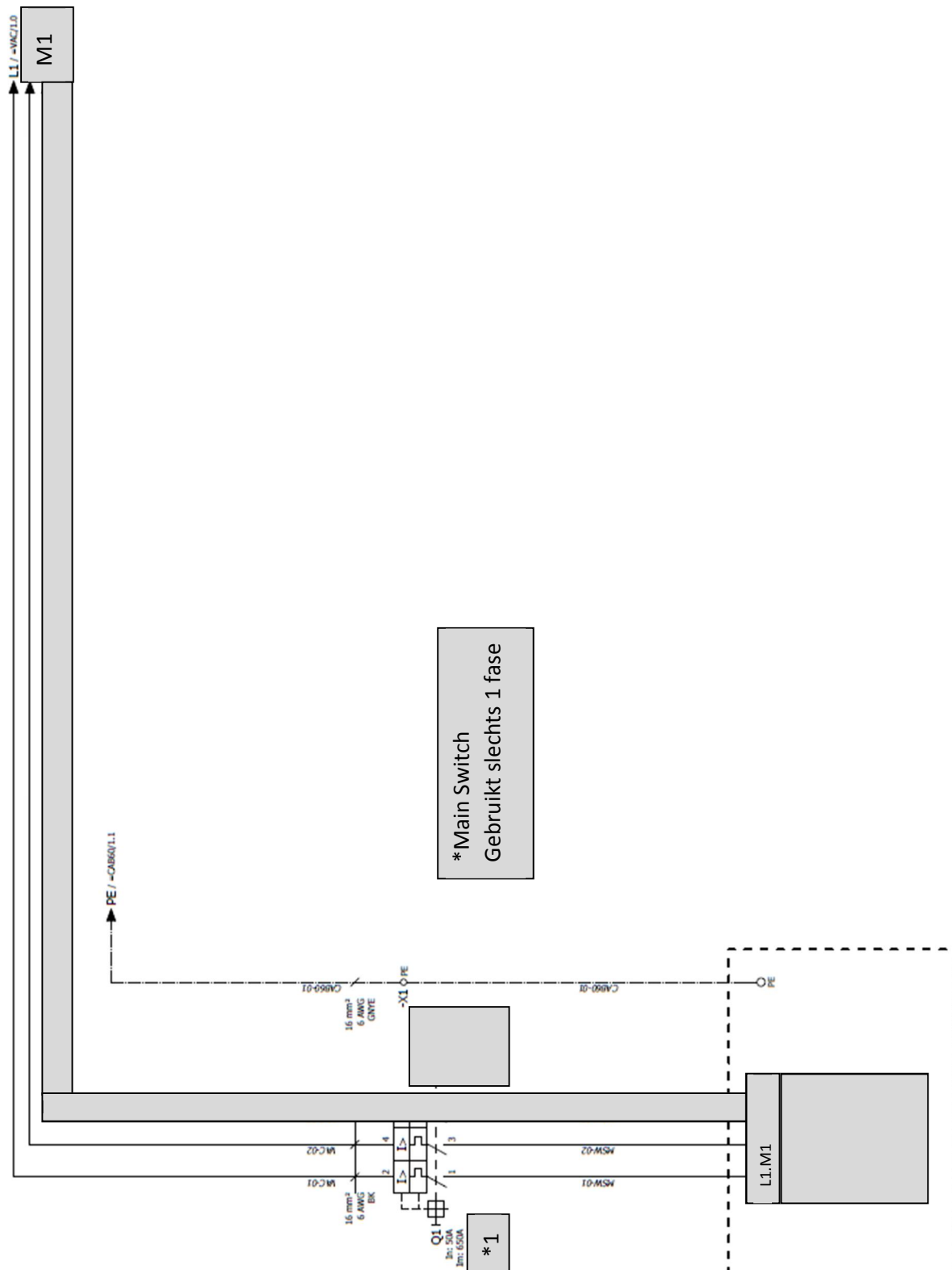
	<b>Document: AFST_TB Bijlage 2 Harting Stekkers v1</b>	<b>Auteur: Jarno Meijer</b>
	<b>Datum: 8-6-2017</b>	<b>Pagina 5 van 5</b>

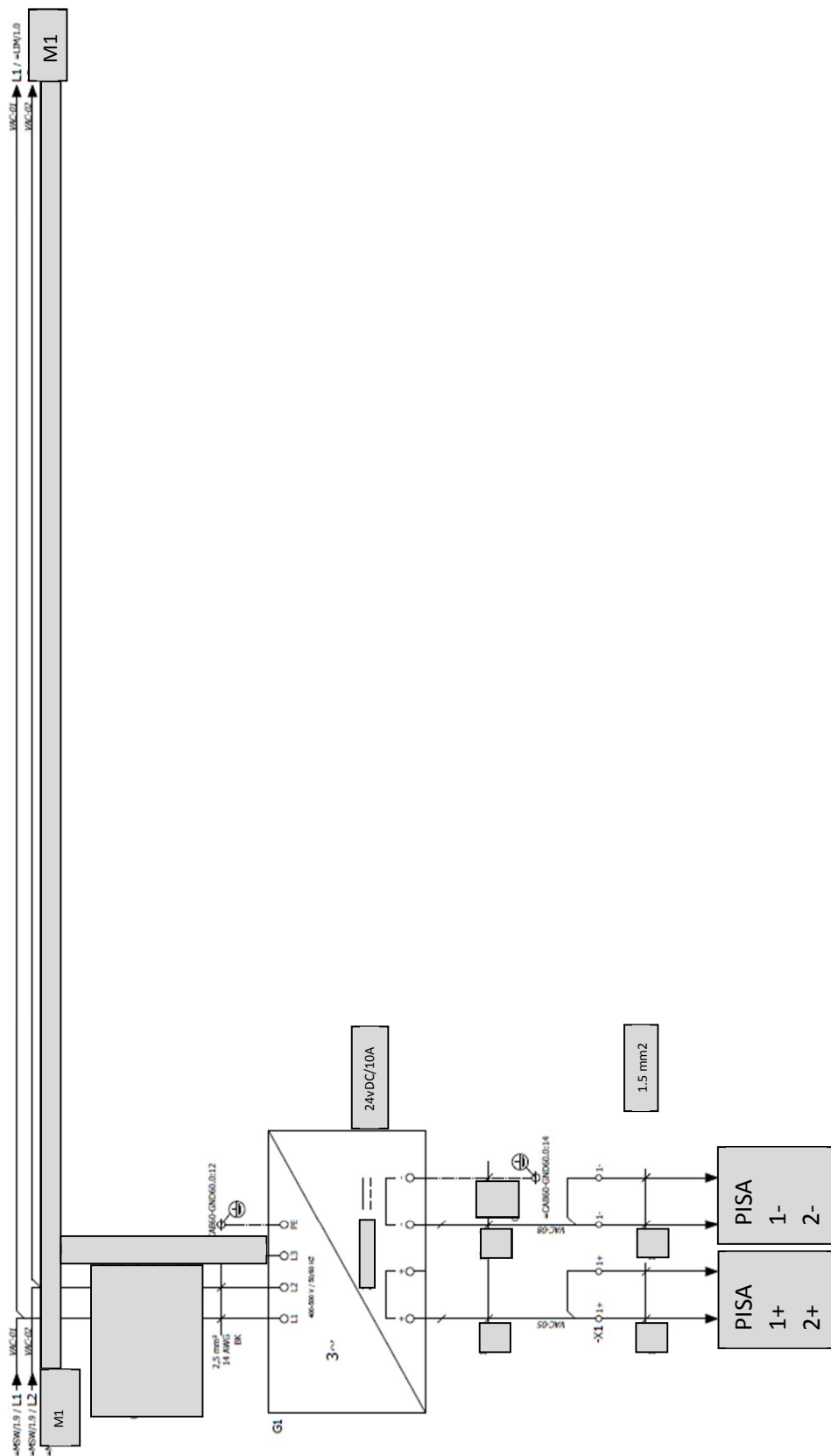
	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

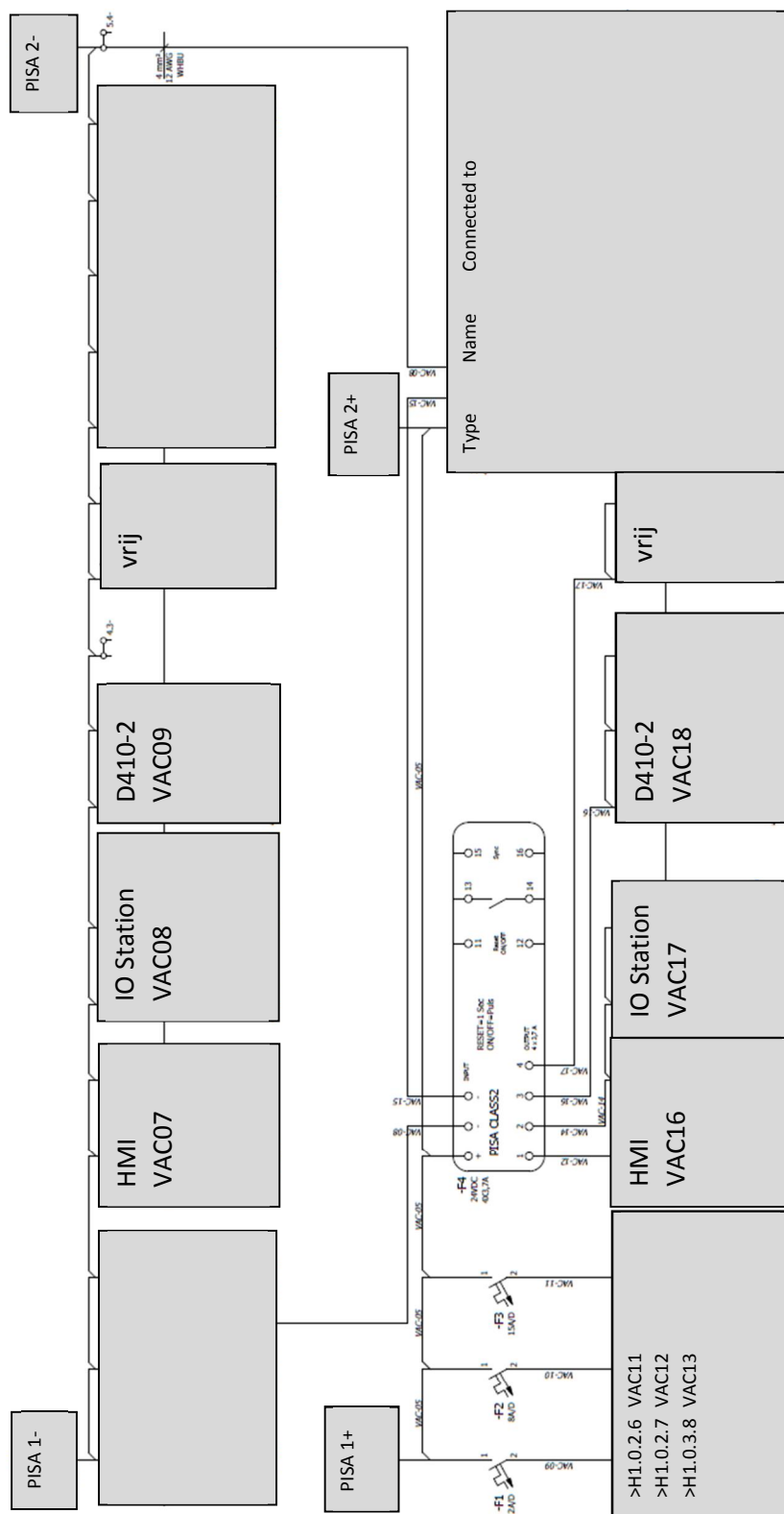
### Bijlage 3: Eplan tekeningen en schetsen

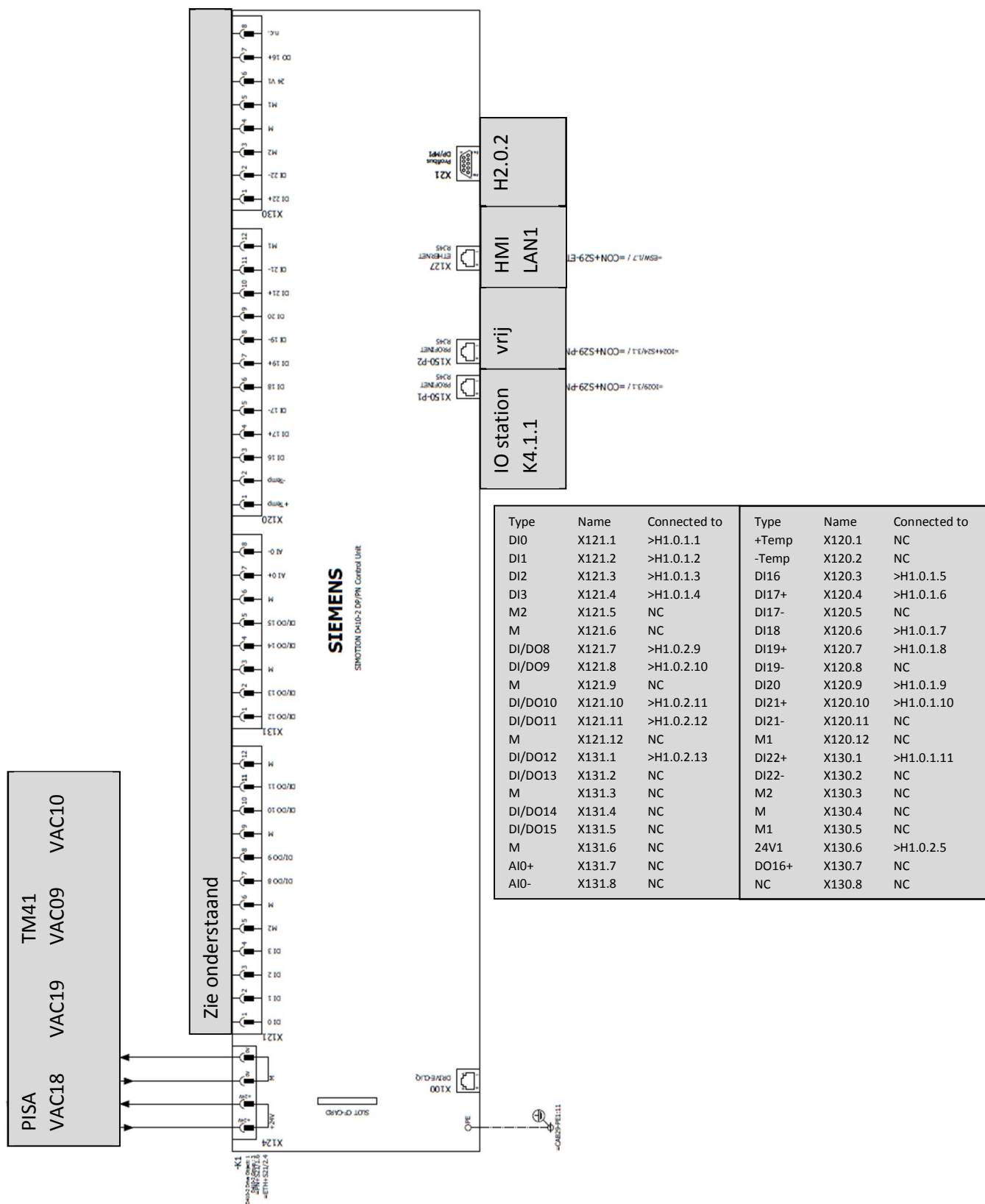
De elektrische schema's zijn uitgewerkt in het tekenpakket Eplan, hiervoor zijn eerst schetsen gemaakt van hoe deze tekeningen er uit zouden moeten gaan zien. Op de volgende pagina's zijn de verschillende elektrische schetsen te vinden, op de pagina's hierna de daadwerkelijke Eplan tekeningen. Bij de Eplan tekeningen is er voor gekozen om niet de vaste layout van dit verslag aan te houden aangezien hierdoor de tekeningen te ver verkleind en hierdoor moeilijk leesbaar worden.

	Document: AFST_TB Bijlage 3 Eplan tekeningen en schetsen v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 1 van 12

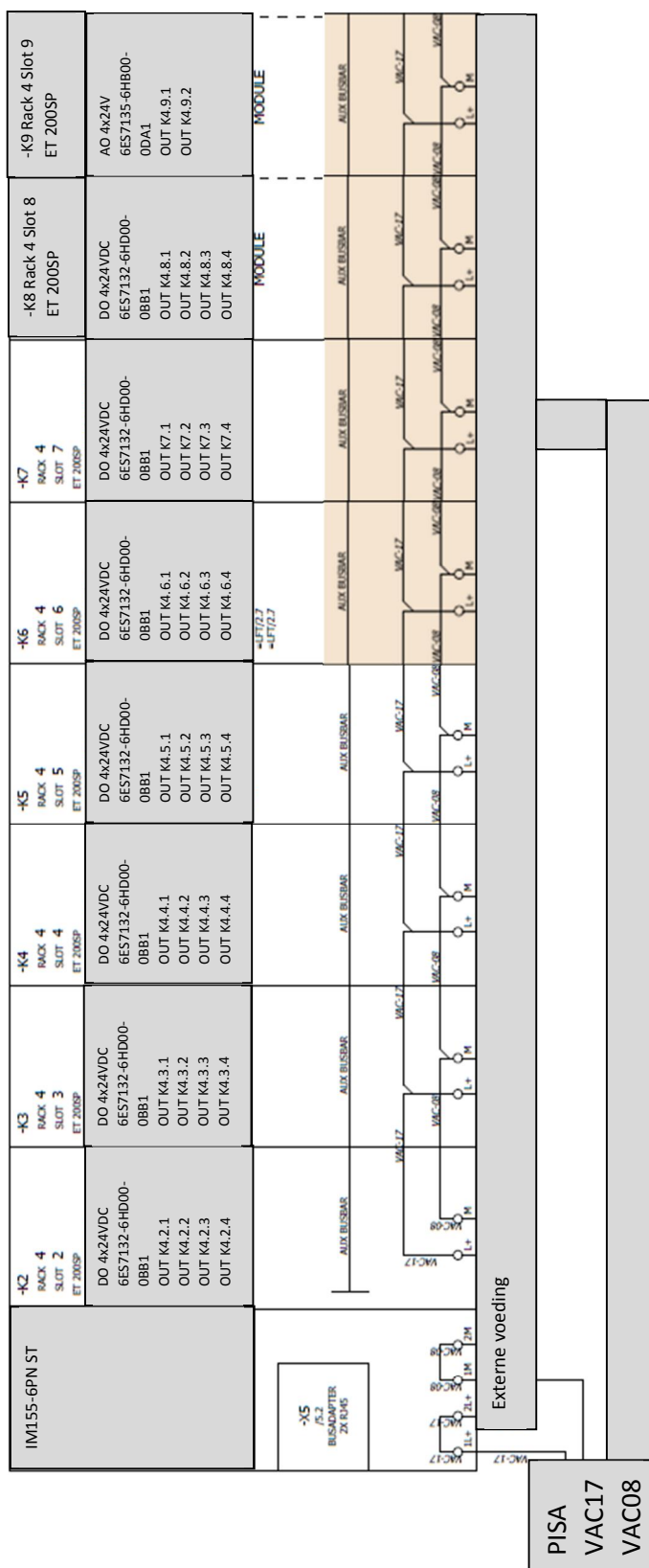








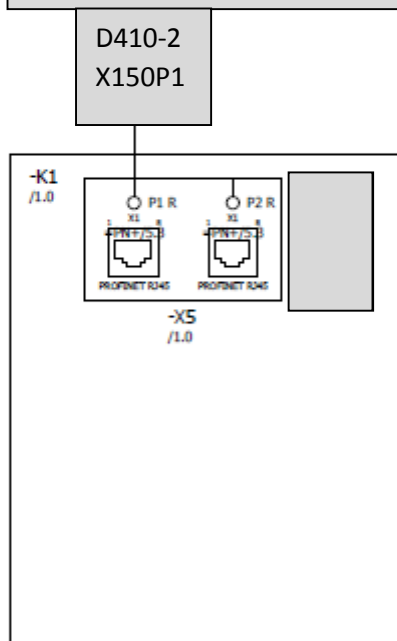


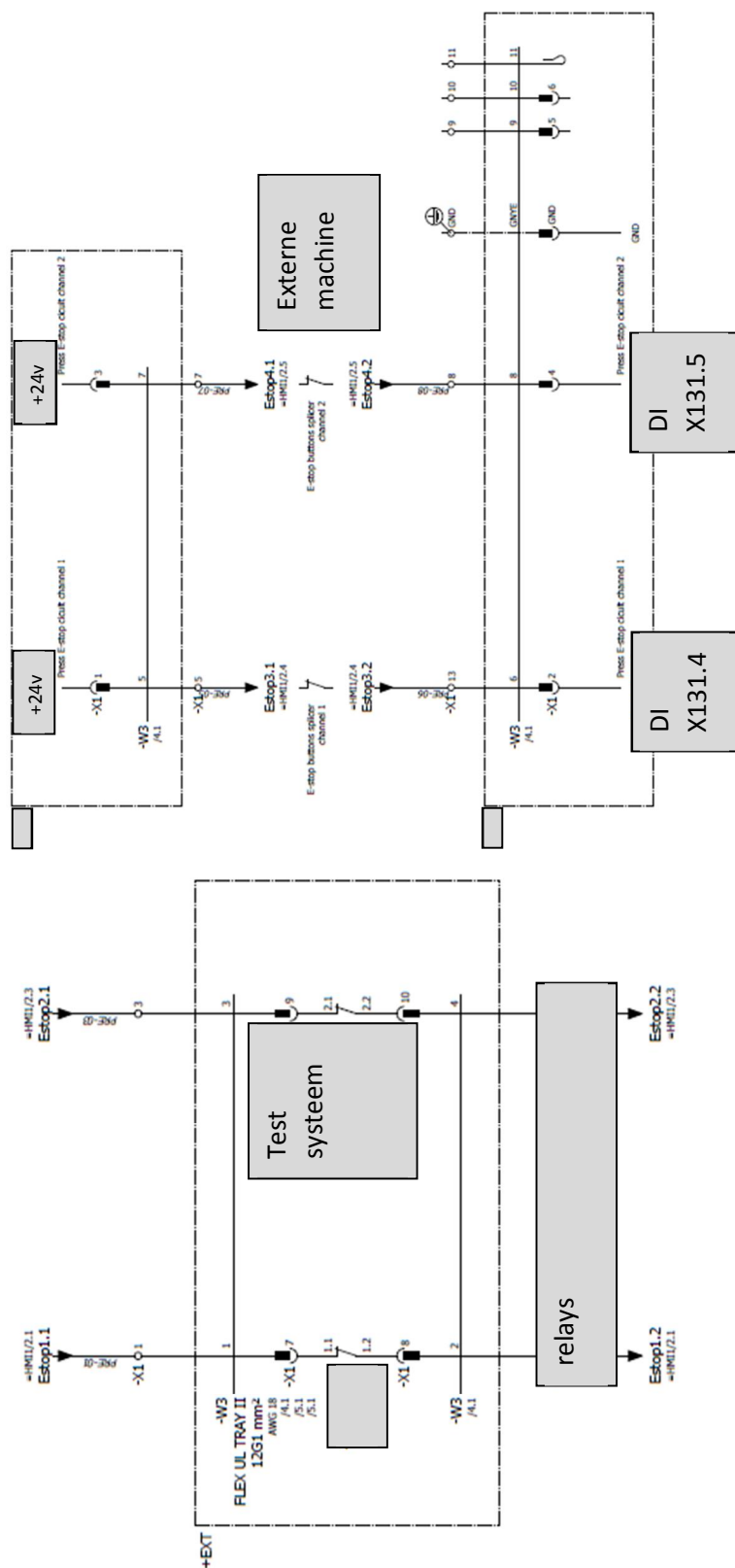


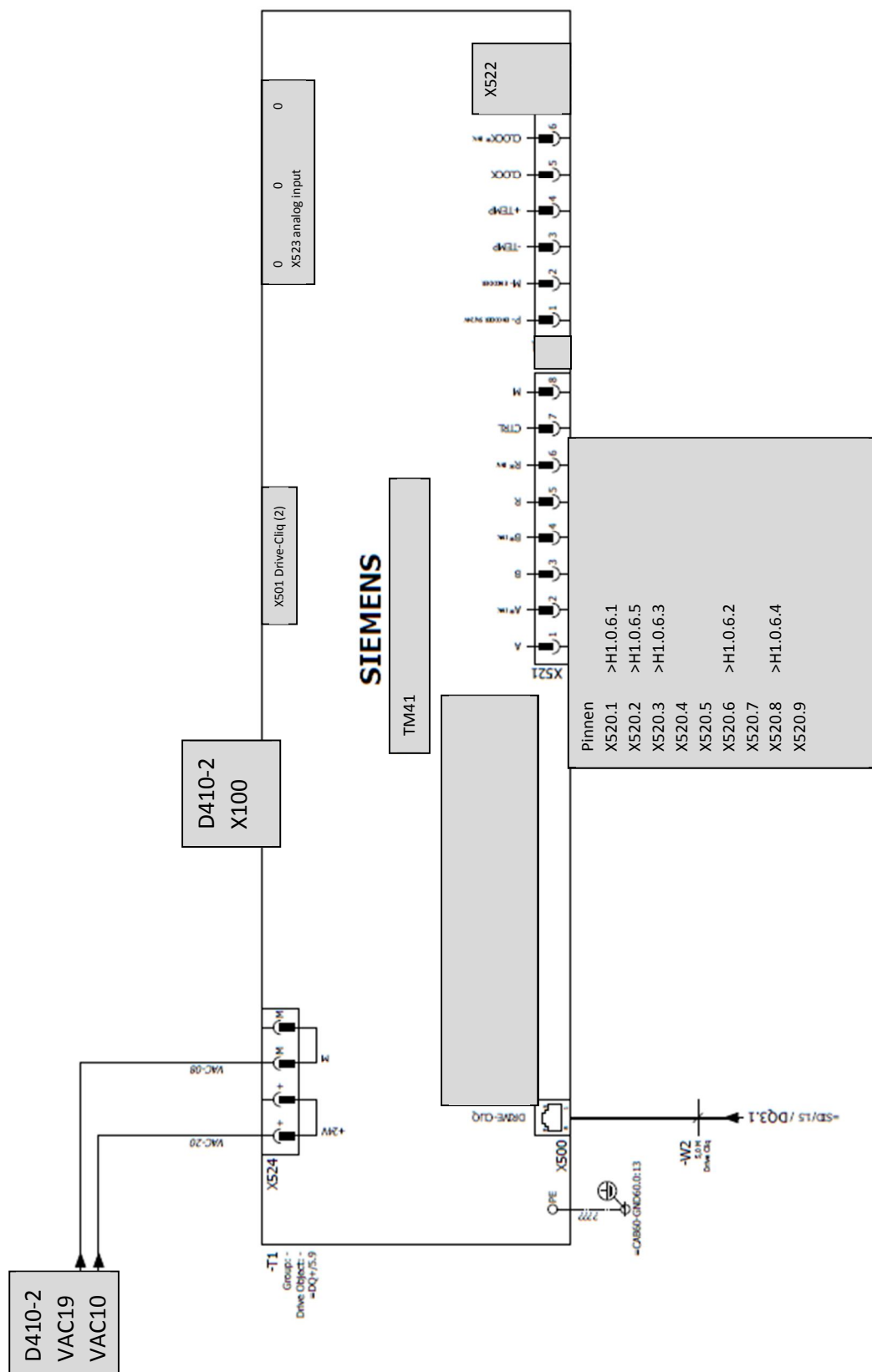
		voeding
K4.2.1	>H1.0.1.12	>1
K4.2.2	>H1.0.1.13	>1
K4.2.3	>H1.0.1.14	> H1.0.3.10
K4.2.4	>H1.0.1.15	> H1.0.3.11
K4.3.1	>H1.0.1.16	>1
K4.3.2	>H1.0.1.17	>1
K4.3.3	>H1.0.2.1	>1
K4.3.4	>H1.0.2.2	>1
K4.4.1	>H1.0.2.3	>1
K4.4.2	>H1.0.2.4	>1
K4.4.3	>H1.0.2.14	>1
K4.4.4	>H1.0.2.15	>1
K4.5.1	>H1.0.2.16	>1
K4.5.2	>H1.0.2.17	>1
K4.5.3	>H1.0.3.1	>1
K4.5.4	>H1.0.3.2	>1
K4.6.1	>H1.0.3.3	>1
K4.6.2	>H1.0.3.4	>1
K4.6.3	>H1.0.3.5	>1
K4.6.4	>H1.0.3.6	>1
K4.7.1	>H1.0.3.7	>1
K4.7.2	NC	
K4.7.3	NC	
K4.7.4	NC	
K4.8.1	NC	
K4.8.2	NC	
K4.8.3	NC	
K4.8.4	NC	
K4.9.1	>H1.0.2.8	> H1.0.2.9
K4.9.2	NC	

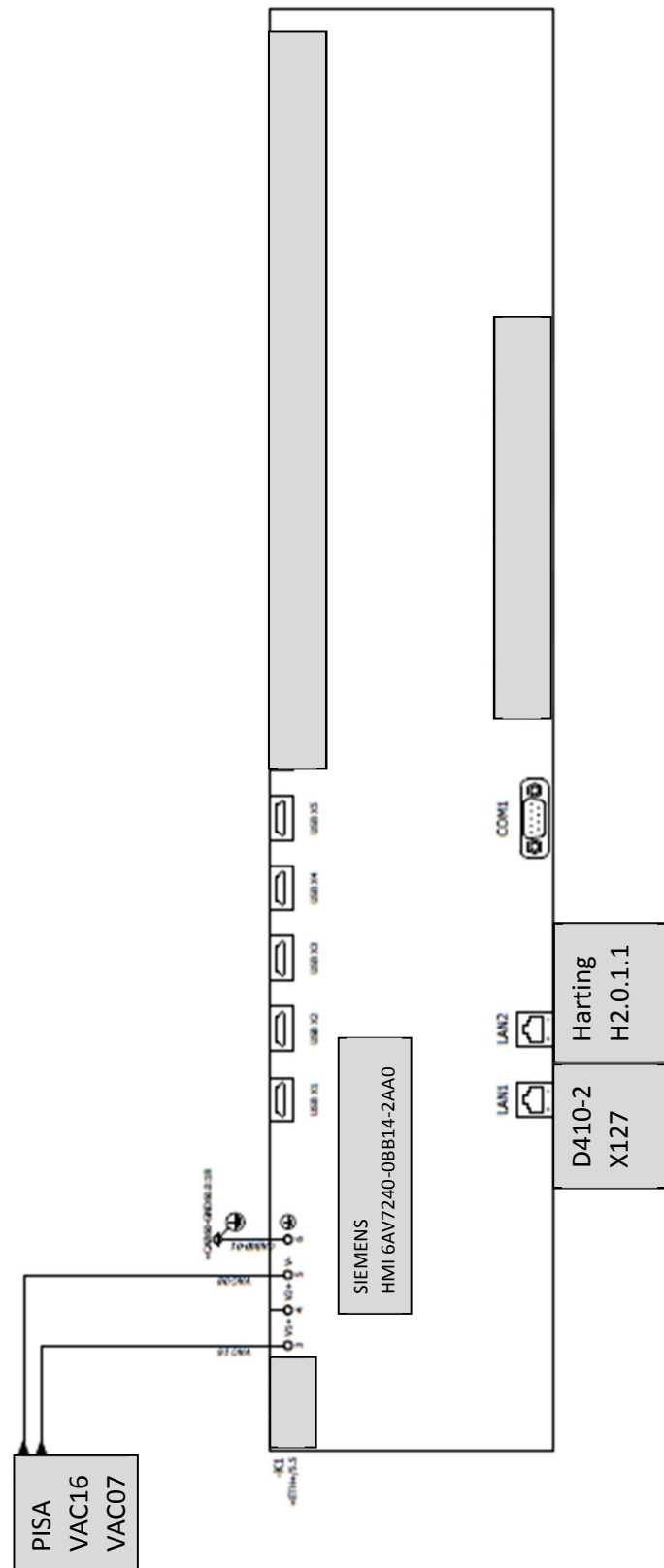
\*1 = H1.0.3.12

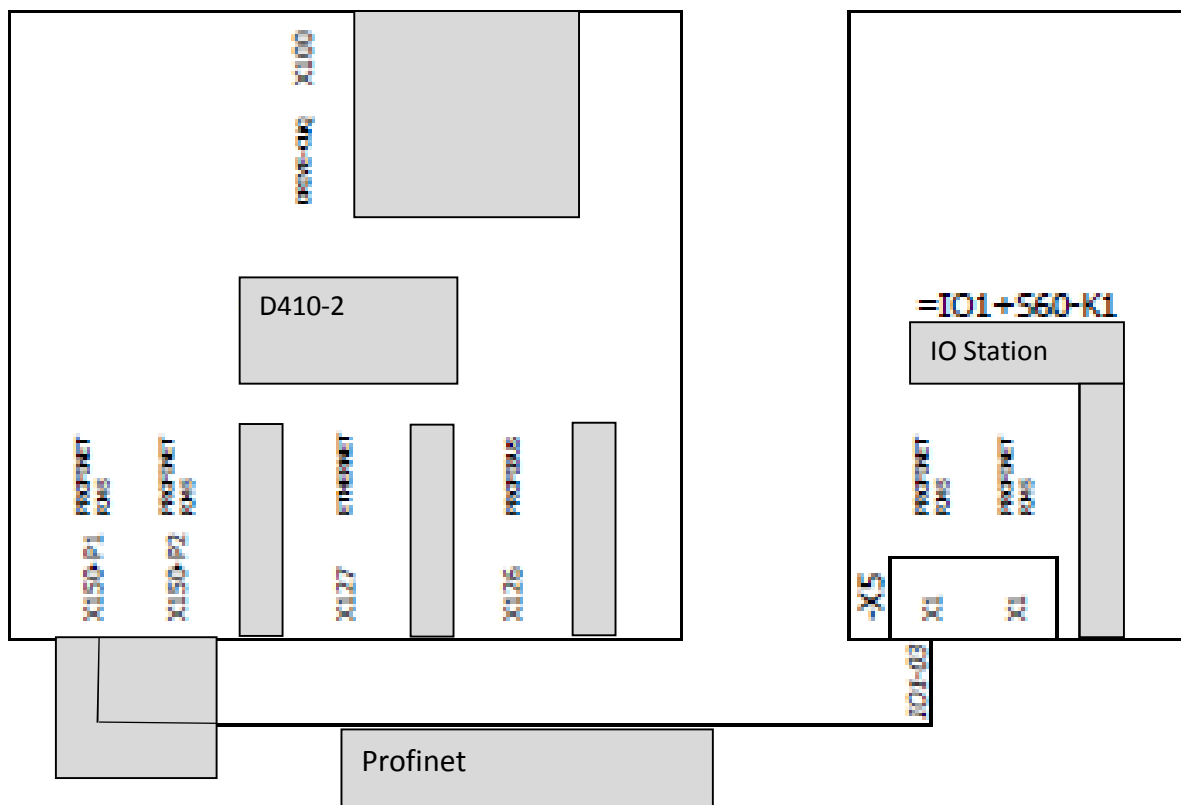
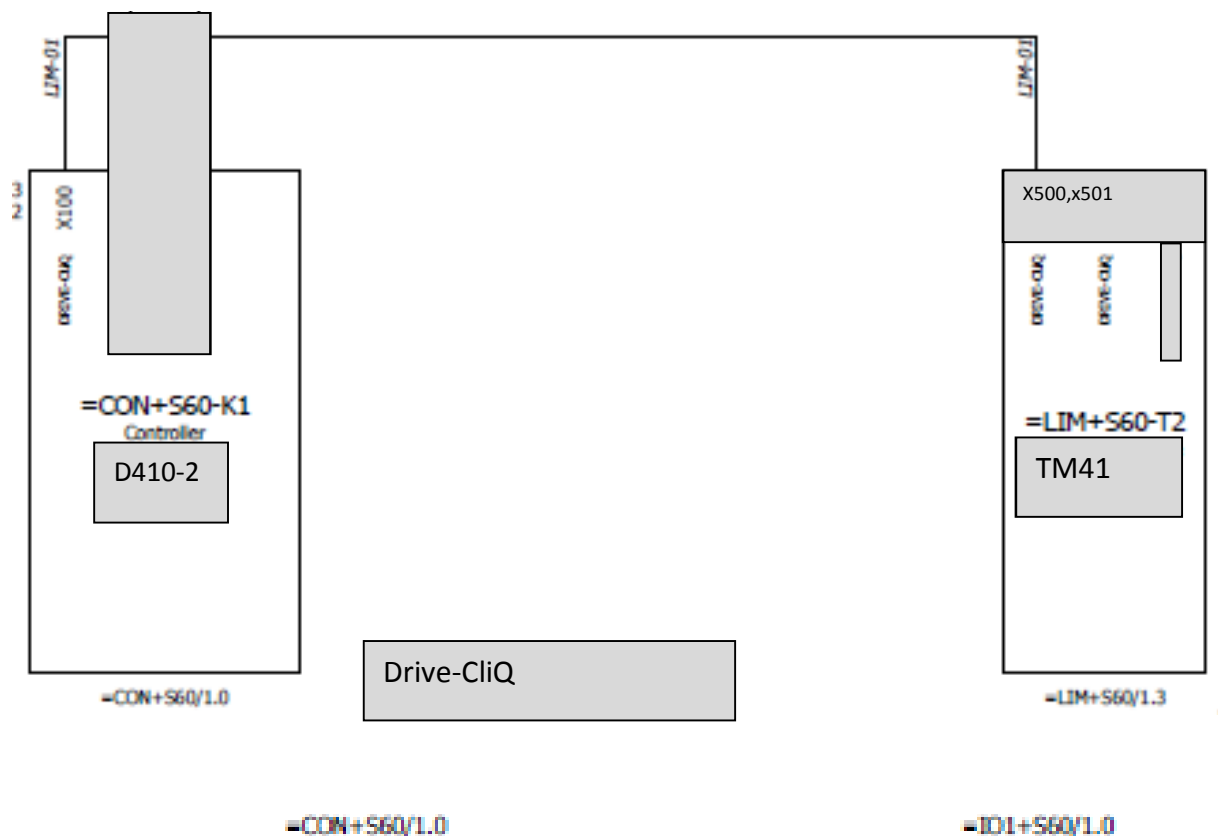
De outputs die nog niet gebruikt worden zullen nog geen voeding krijgen

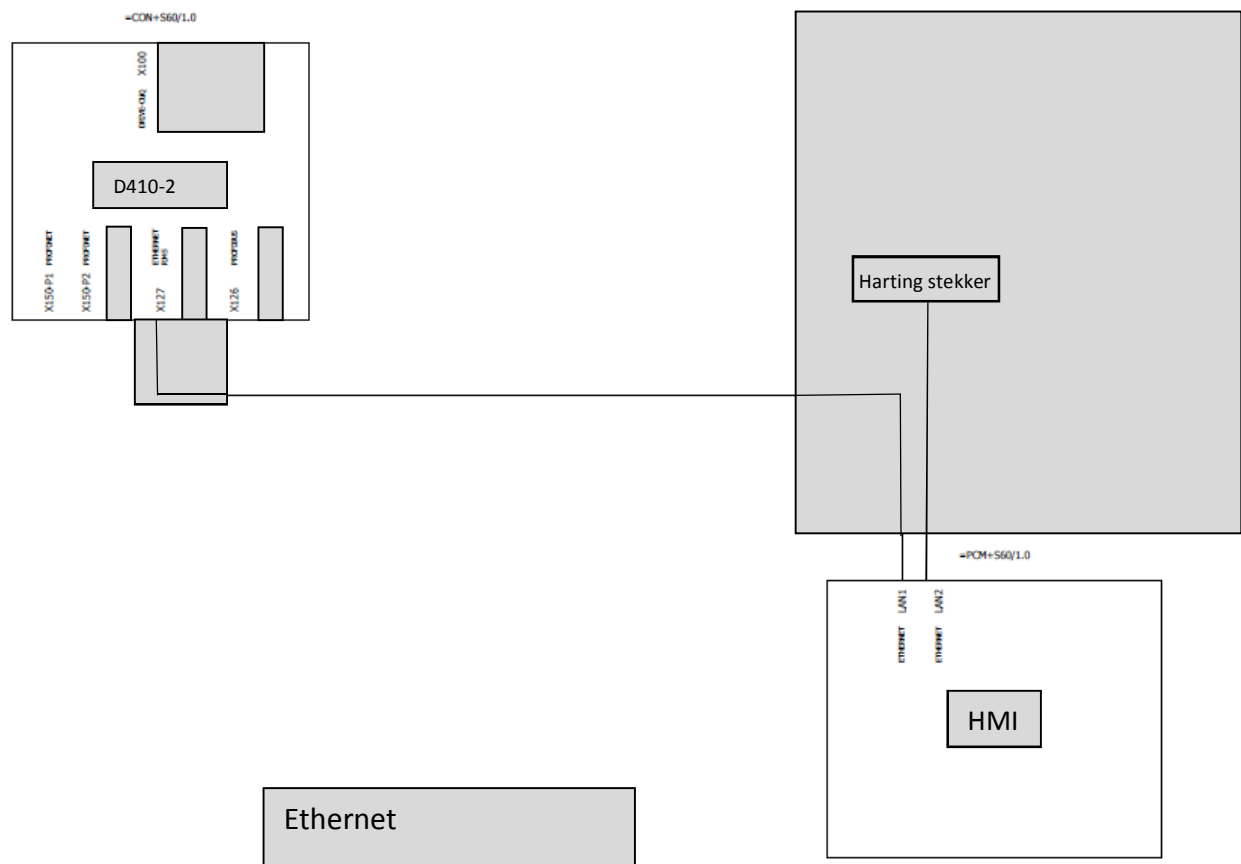












### Harting H2.0 (Housing)

<div data-bbox="186 1283 378 1400">           Harting H2.0.1 Ethernet         </div> <div data-bbox="186 1400 378 1628">           Harting H2.0.2 Dubbele module Profibus         </div>	<div data-bbox="571 1283 906 1355">           Harting H2.0.1 Ethernet   H2.0.1.1   HMI: LAN2         </div>	<div data-bbox="1062 1283 1396 1377">           Harting H2.0.2            Profibus   H2.0.2.1   &gt;D410-2            Profibus   H2.0.2.2   &gt;D410-2         </div>
--	---	---

## HARTING H1.0 (Housing)

Harting H1.0.1 17 pinnen
Harting H1.0.2 17 pinnen
Harting H1.0.3 17 pinnen
Harting H1.0.4 3 pinnen
Harting H1.0.5 Dummie
Harting H1.0.6 8 pinnen

Harting H1.0.1		
DI: IN	H1.0.1.1	>X121.1
DI: IN	H1.0.1.2	>X121.2
DI: IN	H1.0.1.3	>X121.3
DI: IN	H1.0.1.4	>X121.4
DI: IN	H1.0.1.5	>X120.3
DI: IN	H1.0.1.6	>X120.4
DI: IN	H1.0.1.7	>X120.6
DI: IN	H1.0.1.8	>X120.7
DI: IN	H1.0.1.9	>X120.9
DI: IN	H1.0.1.10	>X120.10
DI: IN	H1.0.1.11	>X130.1
DO: OUT	H1.0.1.12	>K4.2.1
DO: OUT	H1.0.1.13	>K4.2.2
DO: OUT	H1.0.1.14	>K4.2.3
DO: OUT	H1.0.1.15	>K4.2.4
DO: OUT	H1.0.1.16	>K4.3.1
DO: OUT	H1.0.1.17	>K4.3.2

Harting H1.0.2		
DO: OUT	H1.0.2.1	>K4.3.3
DO: OUT	H1.0.2.2	>K4.3.4
DO: OUT	H1.0.2.3	>K4.4.1
DO: OUT	H1.0.2.4	>K4.4.2
24I: IN	H1.0.2.5	>X130.6
24O: OUT	H1.0.2.6	>PISA1+
24O: OUT	H1.0.2.7	>PISA1+
AO: OUT	H1.0.2.8	>K4.9.1
AO: REV	H1.0.2.9	>K4.9 REV
DI: IN	H1.0.2.10	>X121.7
DI: IN	H1.0.2.11	>X121.8
DI: IN	H1.0.2.12	>X121.10
DI: IN	H1.0.2.13	>X121.11
DI: IN	H1.0.2.14	>X131.1
DO: OUT	H1.0.2.15	>K4.4.3
DO: OUT	H1.0.2.16	>K4.4.4
DO: OUT	H1.0.2.17	>K4.5.1
DO: OUT	H1.0.2.-	>K4.5.2

Harting H1.0.3		
DO: OUT	H1.0.3.1	>K4.5.2
DO: OUT	H1.0.3.2	>K4.5.3
DO: OUT	H1.0.3.3	>K4.5.4
DO: OUT	H1.0.3.4	>K4.6.1
DO: OUT	H1.0.3.5	>K4.6.2
DO: OUT	H1.0.3.6	>K4.6.3
DO: OUT	H1.0.3.7	>K4.6.4
DO: OUT	H1.0.3.8	>K4.7.1
24O: OUT	H1.0.3.9	>PISA1+
24I: IN	H1.0.3.10	>K4.2.3 IN
24I: IN	H1.0.3.11	>K4.2.4 IN
24I: IN	H1.0.3.12	>K4. 2-8 IN
F	H1.0.3.13	
F	H1.0.3.14	
F	H1.0.3.15	
F	H1.0.3.16	
NC		
*F: FREE, BUT NOT USED NC by default		

Harting H1.0.4		
230O: OUT	H1.0.4.1	L1
230O: OUT	H1.0.4.2	M1
230O: OUT	H1.0.4.3	Aarde

Harting H1.0.5		
Dummie: geen aansluitingen		

Harting H1.0.6		
Encoder	H1.0.6.1	>Enc1
Encoder	H1.0.6.2	>Enc2
Encoder	H1.0.6.3	>Enc3
Encoder	H1.0.6.4	>Enc4
Reference	H1.0.6.5	>Rev5
NC		
NC		
NC		

Information Informatie

Project name Projectnaam

Equipment / Serial number

Machine type Machinetype

Customer name Klantnaam

Location Locatie

: Multi textbox

: Testbox 0001

: Pressconnections

: CONTIWEB

:



Electrical specification Elektrische specificatie

Incoming main power only with copper wire and according to local requirements  
Inkomende hoofdvoeding alleen met koper en conform lokale eisen

Enclosed Industrial control Panel

Mains type Nettype

Number of phases Aantal fase

Mains voltage Netspanning

Frequency Frequentie

Total full-load amperes Totale vollast stroom

Main fuse Hoofdzekering

Largest motor (FLA) Grootste motor (FLA)

Residual-current device value Aardlekschakelaar

Short-circuit current rating Kortsluitstroom

Control voltage Stuurspanning

Installed power Geinstalleerd vermogen

Installation according to EN60204-1 Installatie volgens EN60204-1

Installation according to UL508A / NFPA 79

Enclosure type rating

: TN/TT

: 3 Phase + PE

: 400 VAC + PE

: 50 Hz

I<sub>b</sub> : 19 A

I<sub>n</sub> : 25 A

: 0,38 A

: na

: 10 kA

: 24 VDC / 230 VAC

: 11 kW

: yes

: no

: IP 55










0	1	2	3	4	5	6	7	8	9																																																																																																																																																					
Panelshop																																																																																																																																																														
Structure code list																																																																																																																																																														
Panelshop	<table><tr><th>Code Code</th><th>Page description Paginabeschrijving</th></tr><tr><td>+MACH</td><td>MACHINE LOCATION MACHINE LOKATIE</td></tr><tr><td>+S01</td><td>MAIN CABINET HOOFDSCHAKELKAST</td></tr><tr><td>+S02</td><td>CABINET SCHAKELKAST</td></tr><tr><td>+OSL</td><td>ORDER SPECIFIC LENGTH CABLES ORDER SPECIFIEKE LENGTE KABELS</td></tr><tr><td>+OV</td><td>OVERVIEW I/O OVERZICHT I/O</td></tr><tr><td>+SPL</td><td>SPARE PARTS LIST RESERVE ONDERDELENLIJST</td></tr><tr><td>+CABLES</td><td></td></tr><tr><td>+H1</td><td></td></tr><tr><td>+H1.0</td><td></td></tr><tr><td>+H2</td><td></td></tr><tr><td>+H2.0</td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	Code Code	Page description Paginabeschrijving	+MACH	MACHINE LOCATION MACHINE LOKATIE	+S01	MAIN CABINET HOOFDSCHAKELKAST	+S02	CABINET SCHAKELKAST	+OSL	ORDER SPECIFIC LENGTH CABLES ORDER SPECIFIEKE LENGTE KABELS	+OV	OVERVIEW I/O OVERZICHT I/O	+SPL	SPARE PARTS LIST RESERVE ONDERDELENLIJST	+CABLES		+H1		+H1.0		+H2		+H2.0																														<table><tr><th>Code Code</th><th>Page description Paginabeschrijving</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	Code Code	Page description Paginabeschrijving																																																	<table><tr><th>Code Code</th><th>Page description Paginabeschrijving</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	Code Code	Page description Paginabeschrijving																																																			Panelshop
	Code Code	Page description Paginabeschrijving																																																																																																																																																												
	+MACH	MACHINE LOCATION MACHINE LOKATIE																																																																																																																																																												
	+S01	MAIN CABINET HOOFDSCHAKELKAST																																																																																																																																																												
	+S02	CABINET SCHAKELKAST																																																																																																																																																												
	+OSL	ORDER SPECIFIC LENGTH CABLES ORDER SPECIFIEKE LENGTE KABELS																																																																																																																																																												
	+OV	OVERVIEW I/O OVERZICHT I/O																																																																																																																																																												
	+SPL	SPARE PARTS LIST RESERVE ONDERDELENLIJST																																																																																																																																																												
	+CABLES																																																																																																																																																													
	+H1																																																																																																																																																													
	+H1.0																																																																																																																																																													
	+H2																																																																																																																																																													
	+H2.0																																																																																																																																																													
	Code Code	Page description Paginabeschrijving																																																																																																																																																												
Code Code	Page description Paginabeschrijving																																																																																																																																																													

=/8001

20

 <div>P.O. Box 203 5830 AE Boxmeer (NL) t +31 (0)485 597 111 e contiweb.info@contiweb.com Trade register Eindhoven 17167501</div>	Engineer Bergmans	Multi testbox Testbox 0001 CONTIWEB  Eplan 2.6.3	GENERAL INFORMATION ALGEMENE INFORMATIE Structure code list	Document: WH7001471 /-	= <b>INFO</b> + Page: 10 Total: 5/50
	Date Eng. 4-1-2017			Source:	
	Edit Eng. Mark Bergmans			Copy from:	
	Edit Date 29-5-2017			Retrofit:	
	Print date 1-6-2017			Service delivery:	





Panelshop

Information

Project name : WH7001471 /-

Function name : = MSW

Description 1 : MAIN SWITCH  
HOOFDSCHAKELAAR

Description 2 :

Description 3 :

Machine type : Pressconnections

Panelshop

Revision Information: MSW

Revision	Function	Modification	By	Date	Change nr.

Panelshop

# Panelshop

- $N / = VAC/2.0$
- $L1 / = VAC/2.0$

➔ PE / =CAB1/1.1

6 mm<sup>2</sup>—  
8 AWG  
BK

6 mm<sup>2</sup>  
8 AWG  
GNYE

-Q1  
In: 32A IEC / 30A UL  
Iq: 80kA IEC / 5 kA UL

○ PE

CON

L1

C  
PE

Main voltage connection  
Hoofdvoeding aansluiting

# Panelshop

$$=VAC+/1$$

# Panelshop



Panelshop

Information

Project name : WH7001471    /-

Function name : = VAC

Description 1 : CONTROL VOLTAGE  
STUURSPANNING

Description 2 :

Description 3 : CE & UL

Machine type : Pressconnections

Panelshop

Revision Information: VAC

Revision	Function	Modification	By	Date	Change nr.

Panelshop

Panelshop





# Panelshop

Panelshop

Information

Project name : WH7001471    /-

Function name :                    = CON

Description 1 : CONTROL SYSTEM  
BESTURING

Description 2 :

Description 3 :

Machine type : Pressconnections

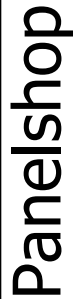
Panelshop

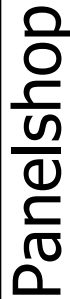
Revision Information: CON

Revision	Function	Modification	By	Date	Change nr.

Panelshop

Panelshop





Panelshop

Information

Project name : WH7001471 /-

Function name : = IO1

Description 1 : I/O SLAVE 1  
I/O SLAVE 1

Description 2 :

Description 3 :

Machine type : Pressconnections

Panelshop

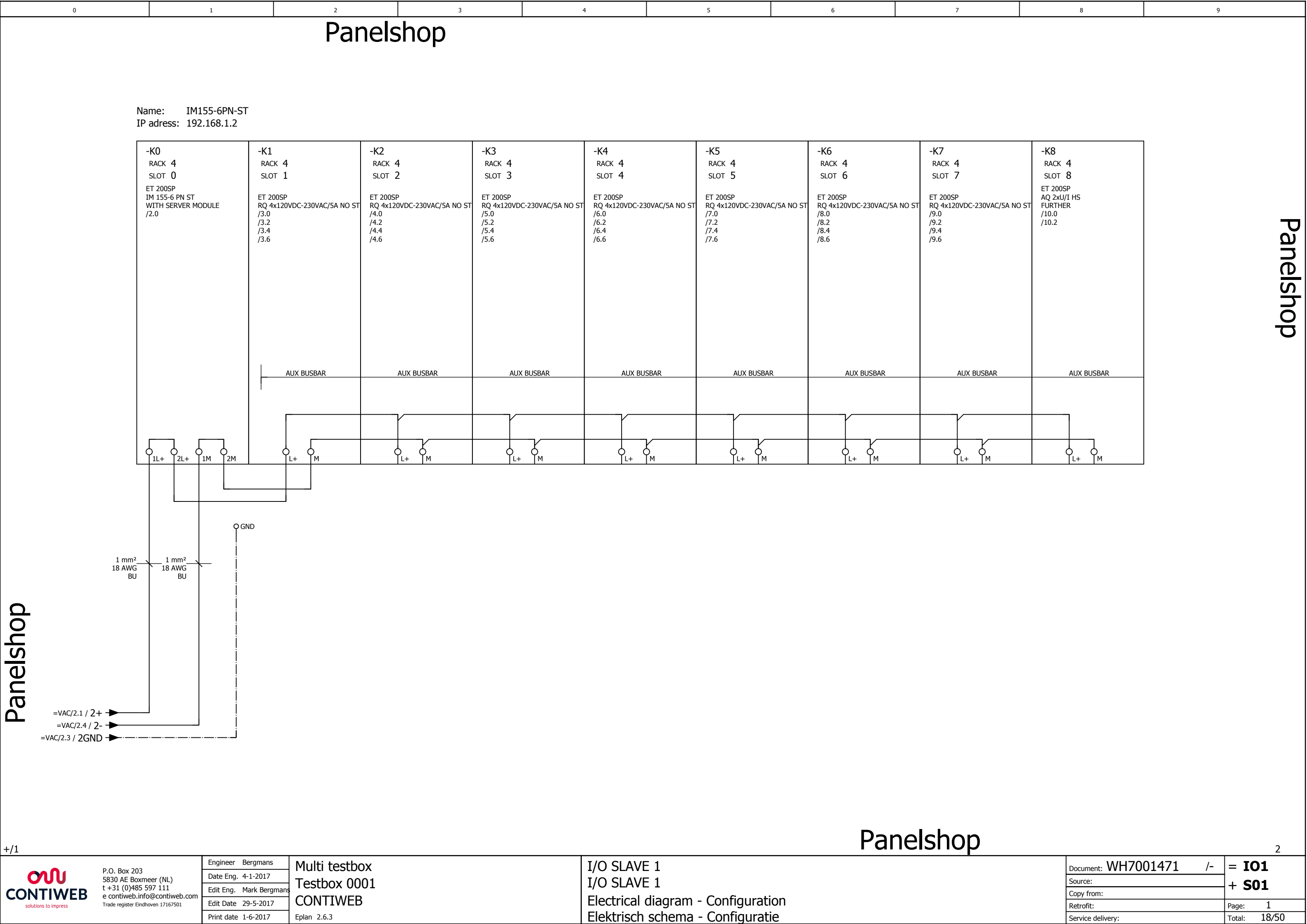
Revision Information: IO1

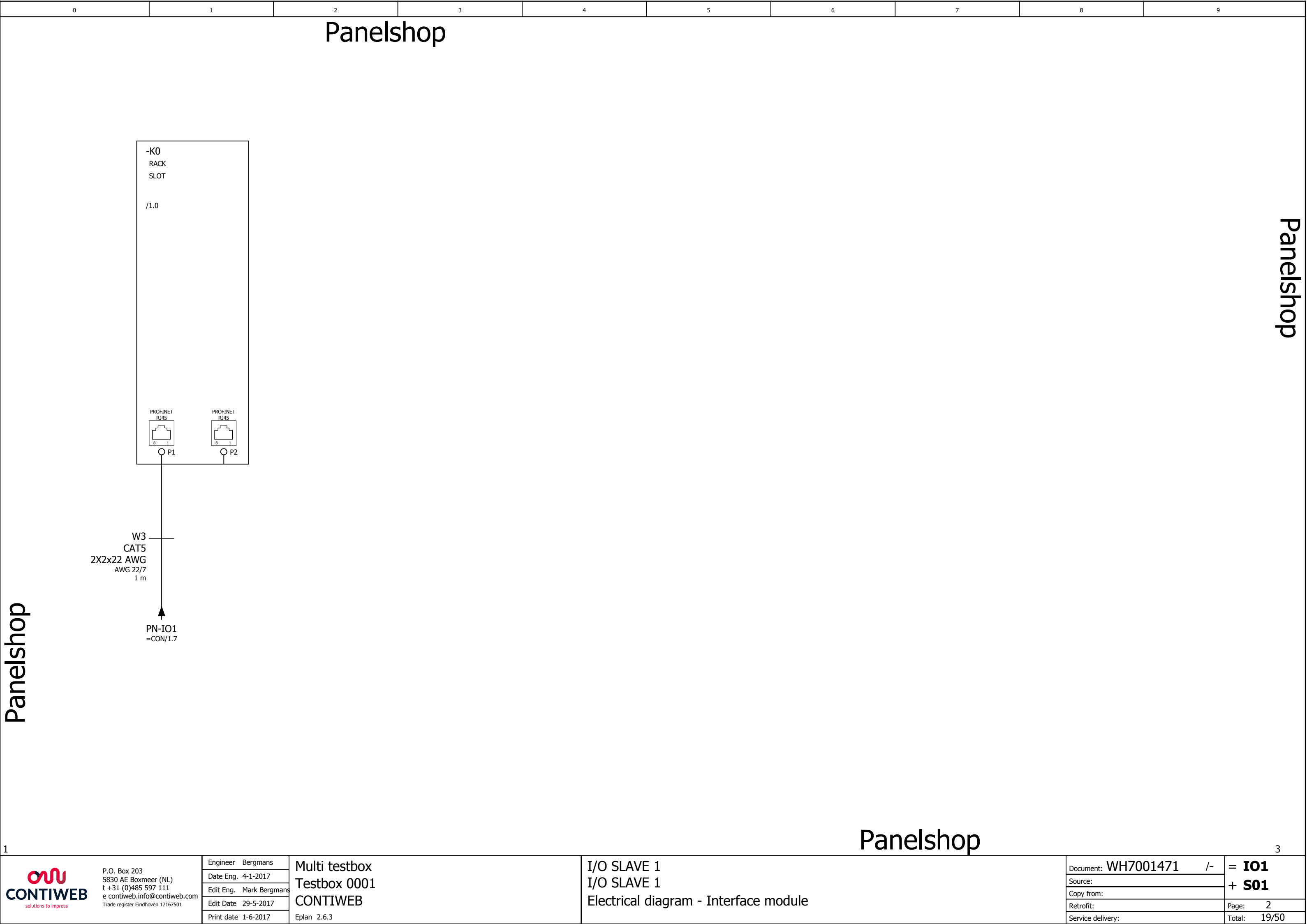
Revision	Function	Modification	By	Date	Change nr.

Panelshop

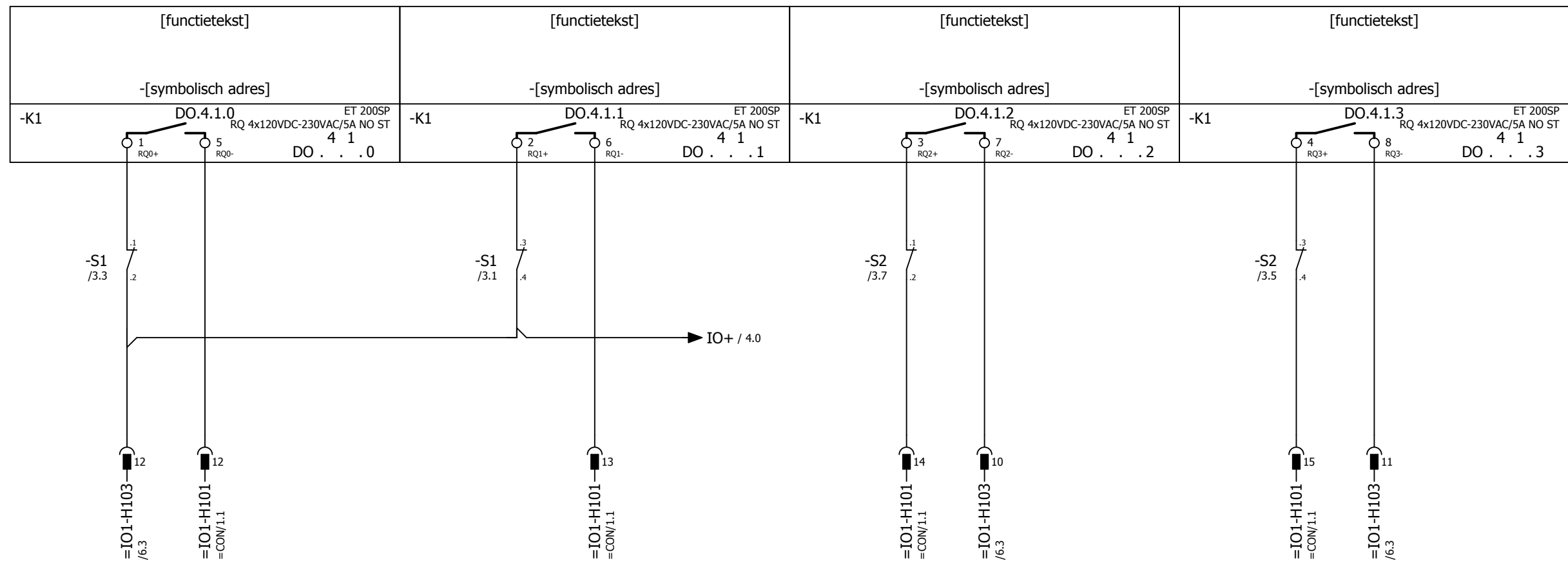
Panelshop







# Panelshop

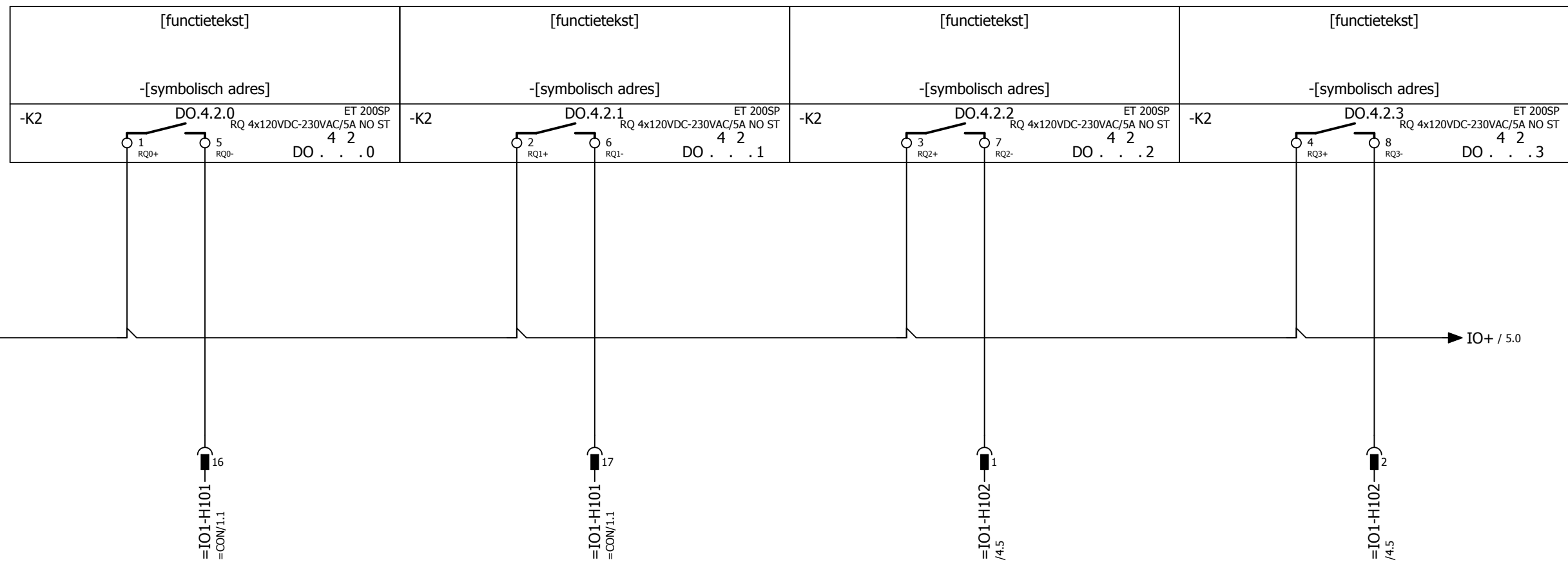


# Panelshop

# Panelshop

# Panelshop

# Panelshop



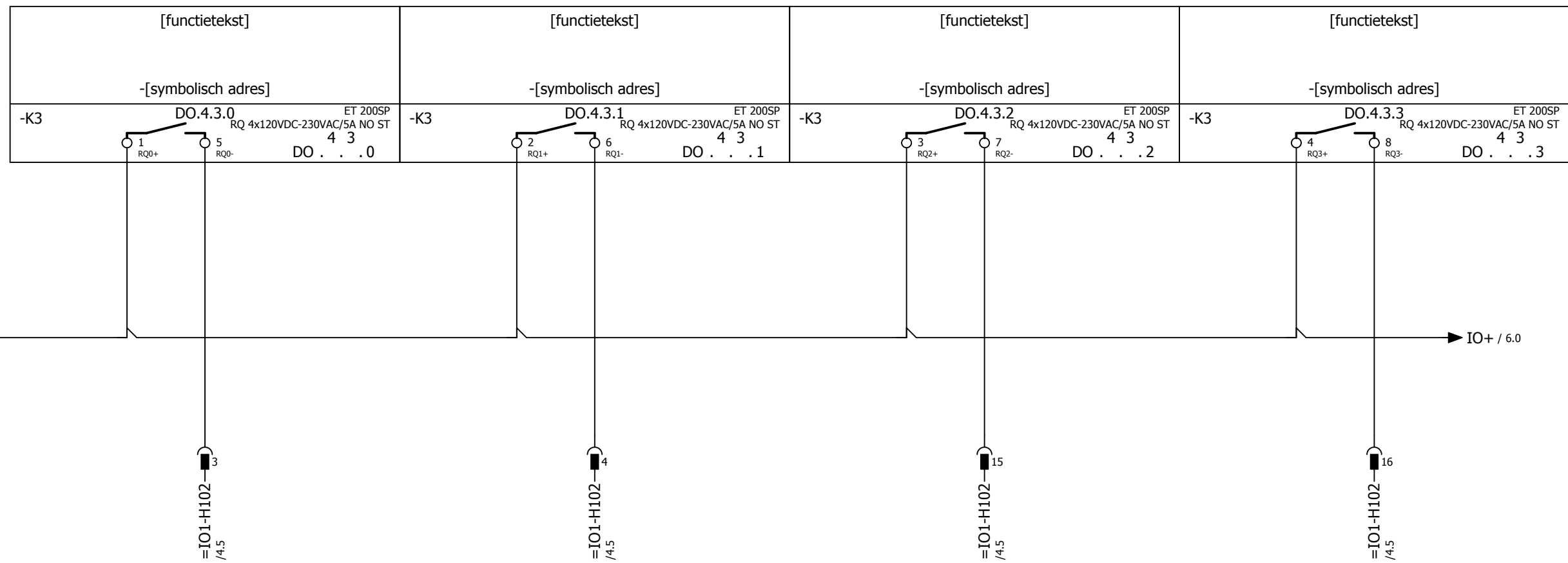
# Panelshop

# Panelshop

=IO1-H102 = Plug H102

# Panelshop

# Panelshop

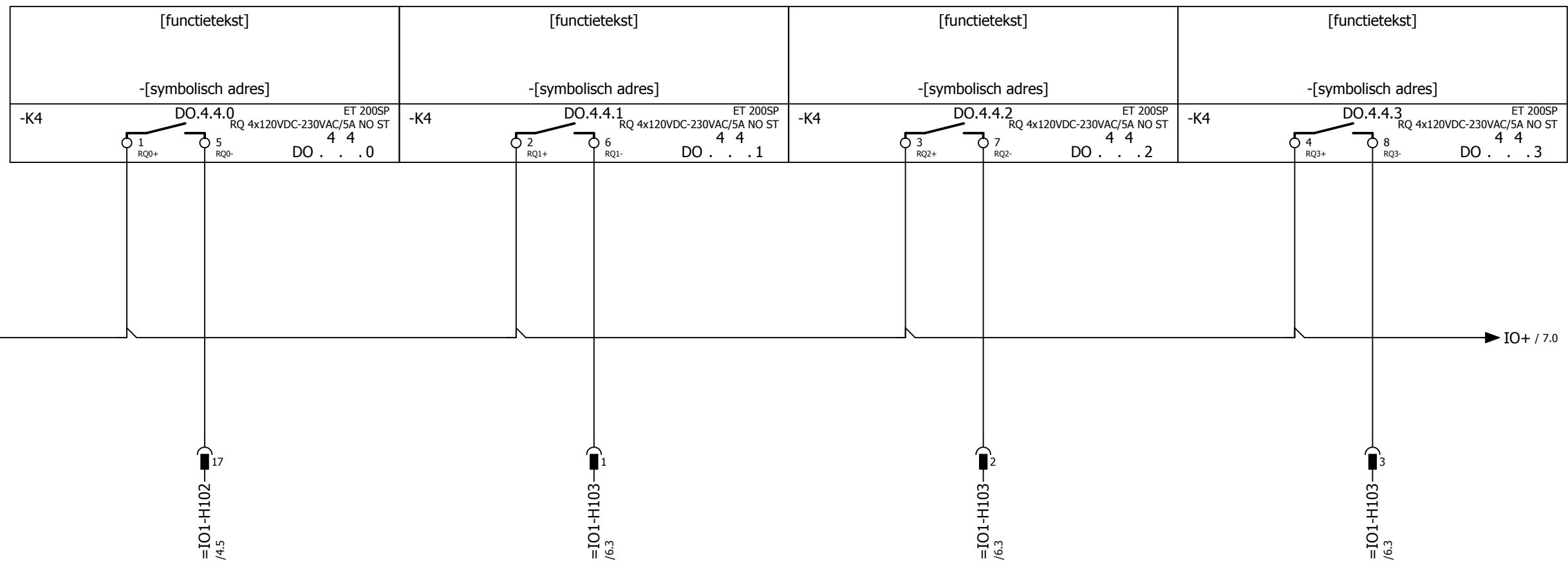


# Panelshop

# Panelshop

# Panelshop

# Panelshop

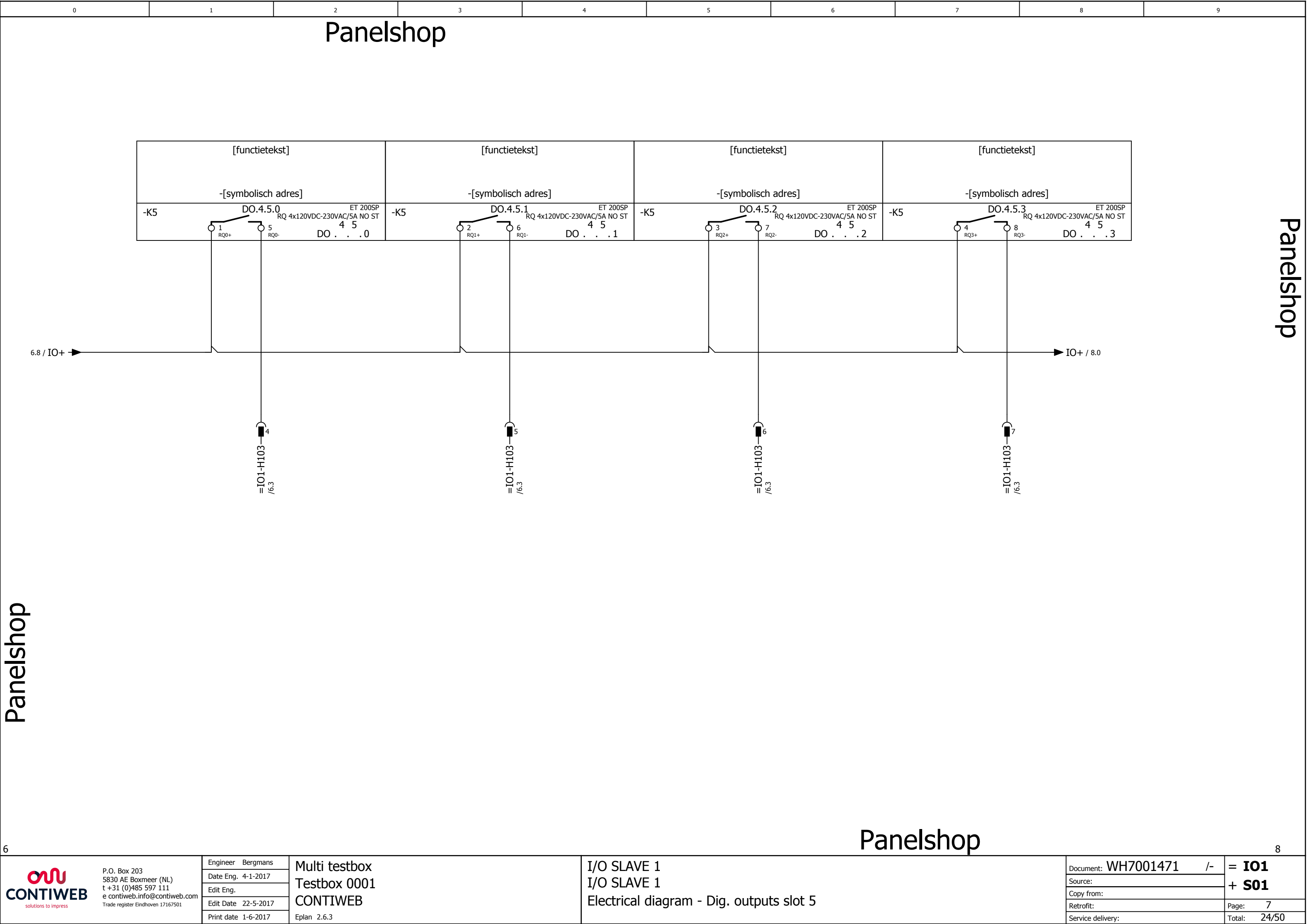


# Panelshop

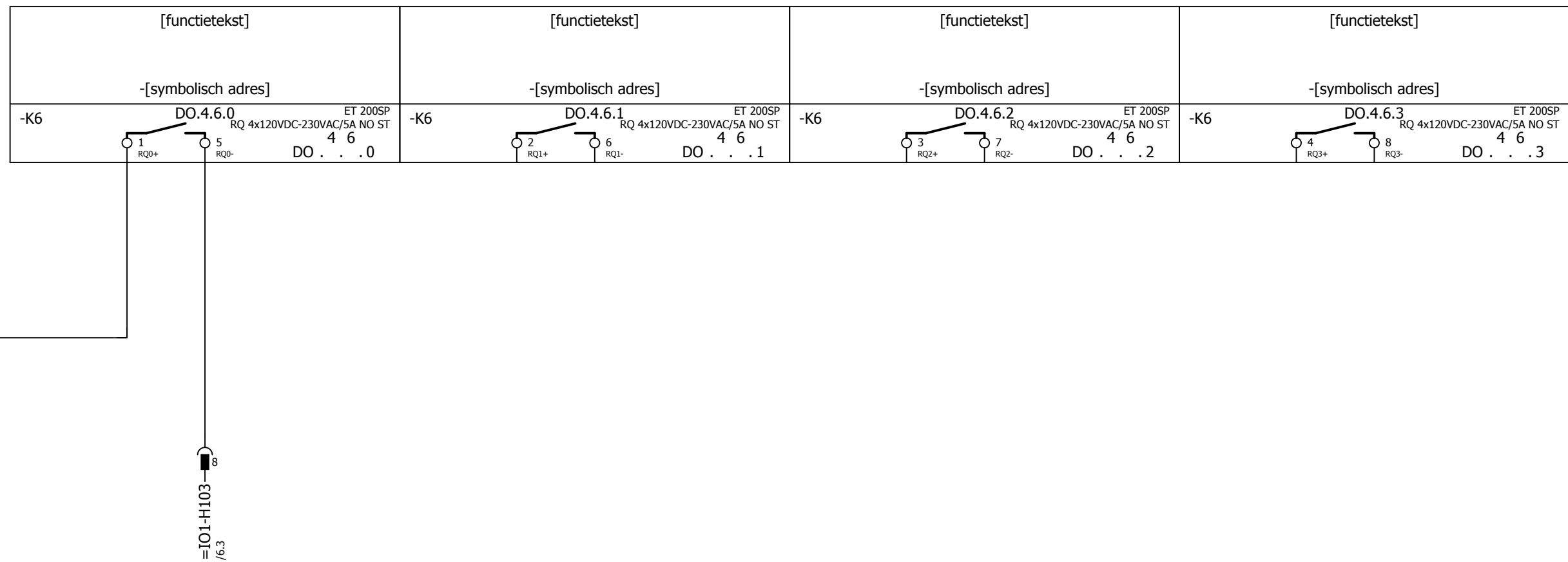
# Panelshop

=IO1-H103 = Plug H103

# Panelshop



# Panelshop



# Panelshop

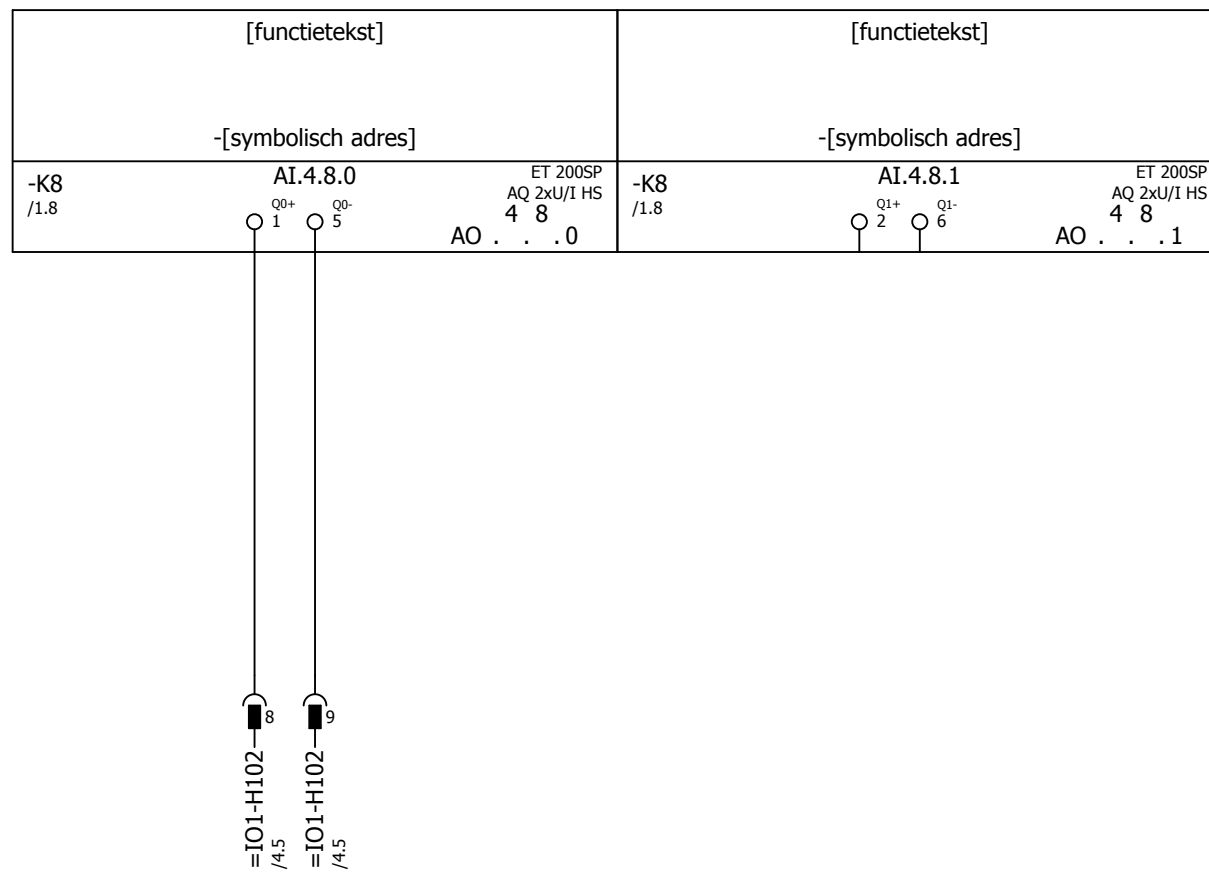
# Panelshop

# Panelshop





# Panelshop



# Panelshop

# Panelshop

# Panelshop

PRE+/1

Panelshop

Information

Project name : WH7001471    /-

Function name :                    = PRE

Description 1 : PRESS CONNECTIONS  
                             PERSVERBINDINGEN

Description 2 :

Description 3 :

Machine type : Pressconnections

Panelshop

Revision Information:    PRE

Revision	Function	Modification	By	Date	Change nr.

Panelshop

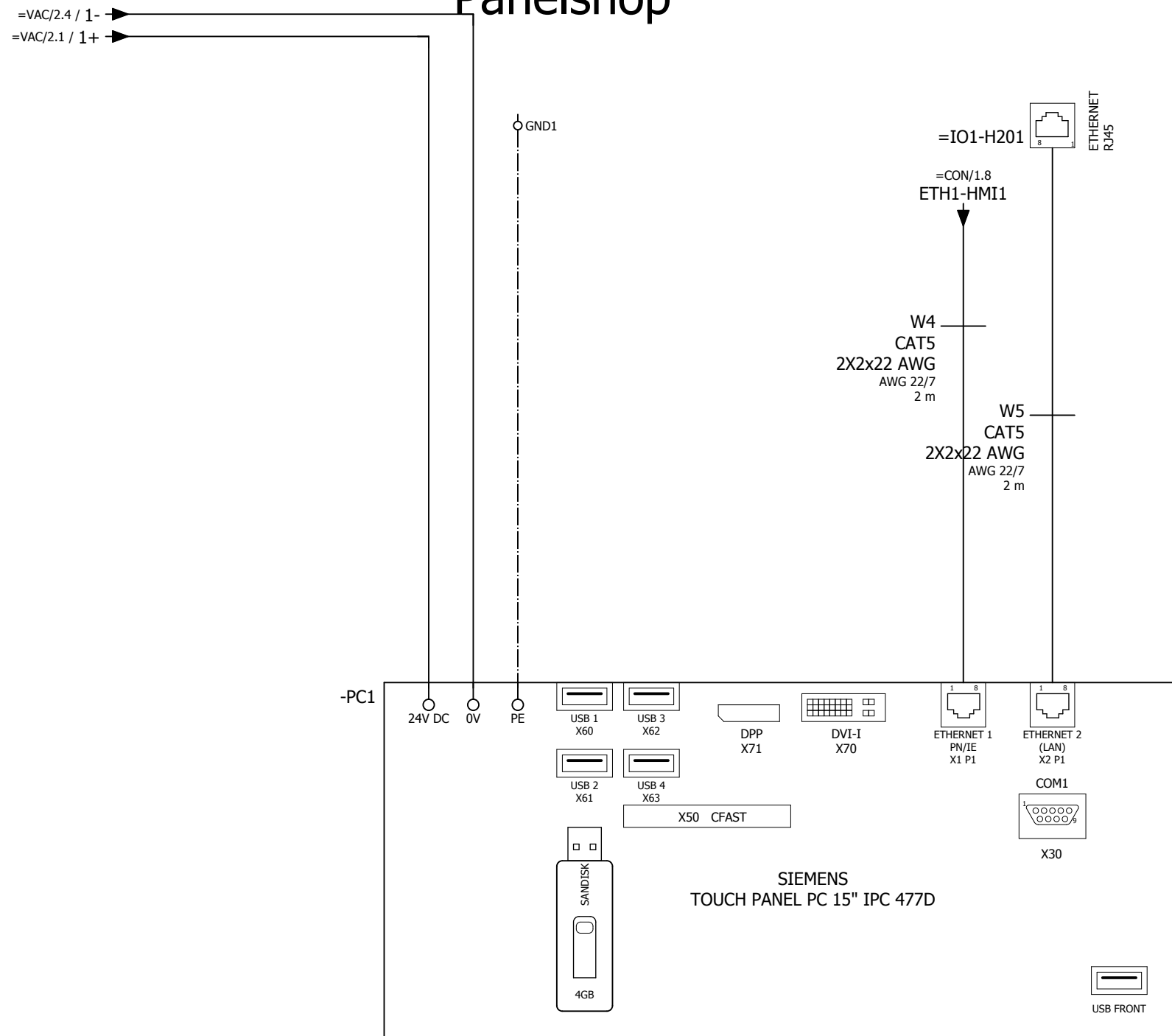
Panelshop



# Panelshop



# Panelshop



# Panelshop

# Panelshop

# Panelshop

8000


# Panelshop

# Panelshop Terminal diagram

# Panelshop

# Panelshop

$$= \text{PLC} + \text{OV} / 1$$

 <p>P.O. Box 203 5830 AE Boxmeer (NL) t +31 (0)485 597 111 e <a href="mailto:contiweb.info@contiweb.com">contiweb.info@contiweb.com</a> Trade register Eindhoven 17167501</p>	Engineer Bergmans	<p>Multi testbox</p> <p>Testbox 0001</p> <p>CONTIWEB</p> <p>Eplan 2.6.3</p>	<p>HUMAN MACHINE INTERFACE 1</p> <p>BEDIENINGSPANEEL 1</p> <p>Terminal connection list</p> <p>Klemmen aansluitlijst</p>	Document: WH7001471 /- = <b>HMI1</b>
	Date Eng. 4-1-2017			Source: + <b>S01</b>
	Edit Eng.			Copy from:
	Edit Date 16-5-2017			Retrofit: Page: 8000
	Print date 1-6-2017			Service delivery: Total: 33/50







Panelshop

Information

Project name : WH7001471    /-

Function name : = CAB1

Description 1 : CABINET OVERVIEW S01  
SCHAKELKASTOVERZICHT S01

Description 2 :

Description 3 :

Machine type : Pressconnections

Panelshop

Revision Information: CAB1

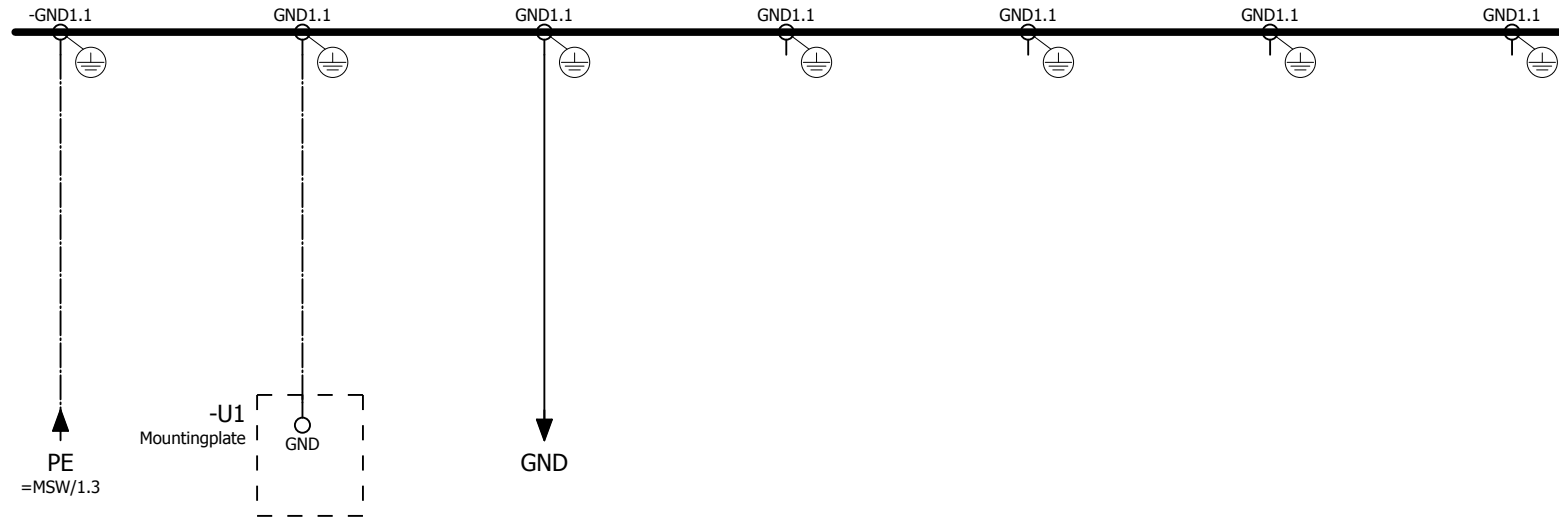
Revision	Function	Modification	By	Date	Change nr.

Panelshop

Panelshop

# Panelshop

Grounding bus bar  
40A 10mm<sup>2</sup>



# Panelshop

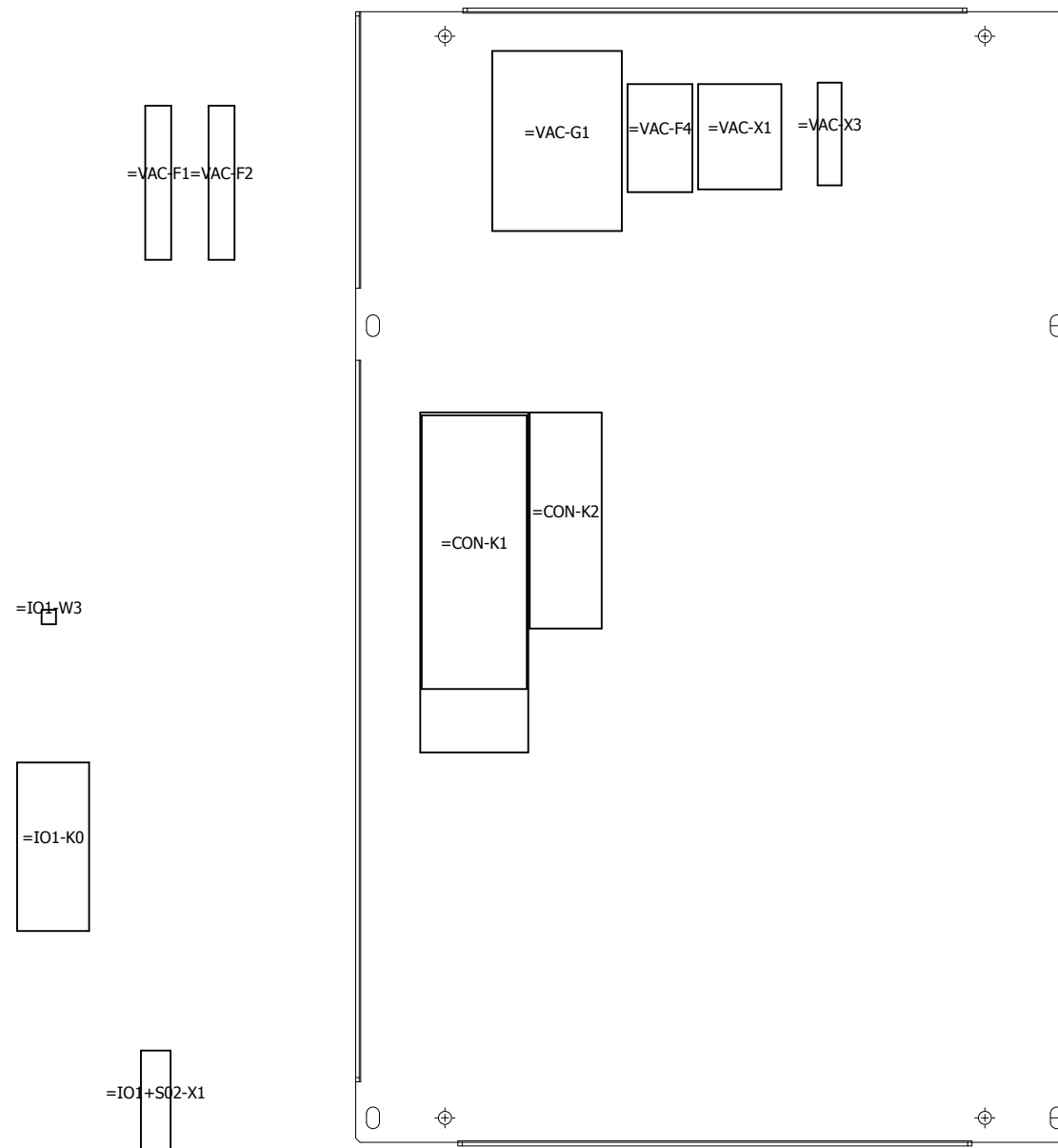
# Panelshop

# Panelshop

 $+/1$ 

2

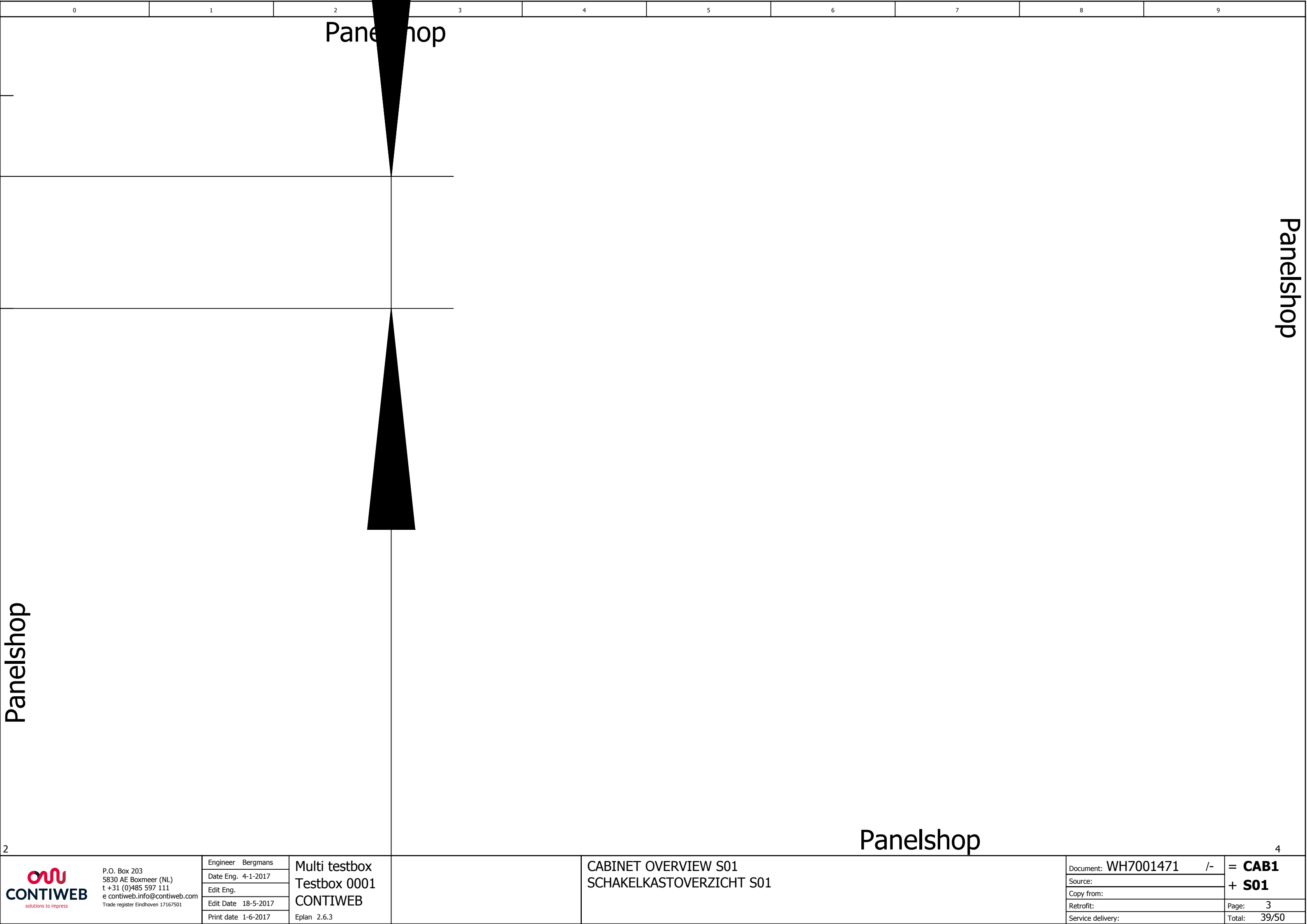
# Panelshop



# Panelshop

# Panelshop

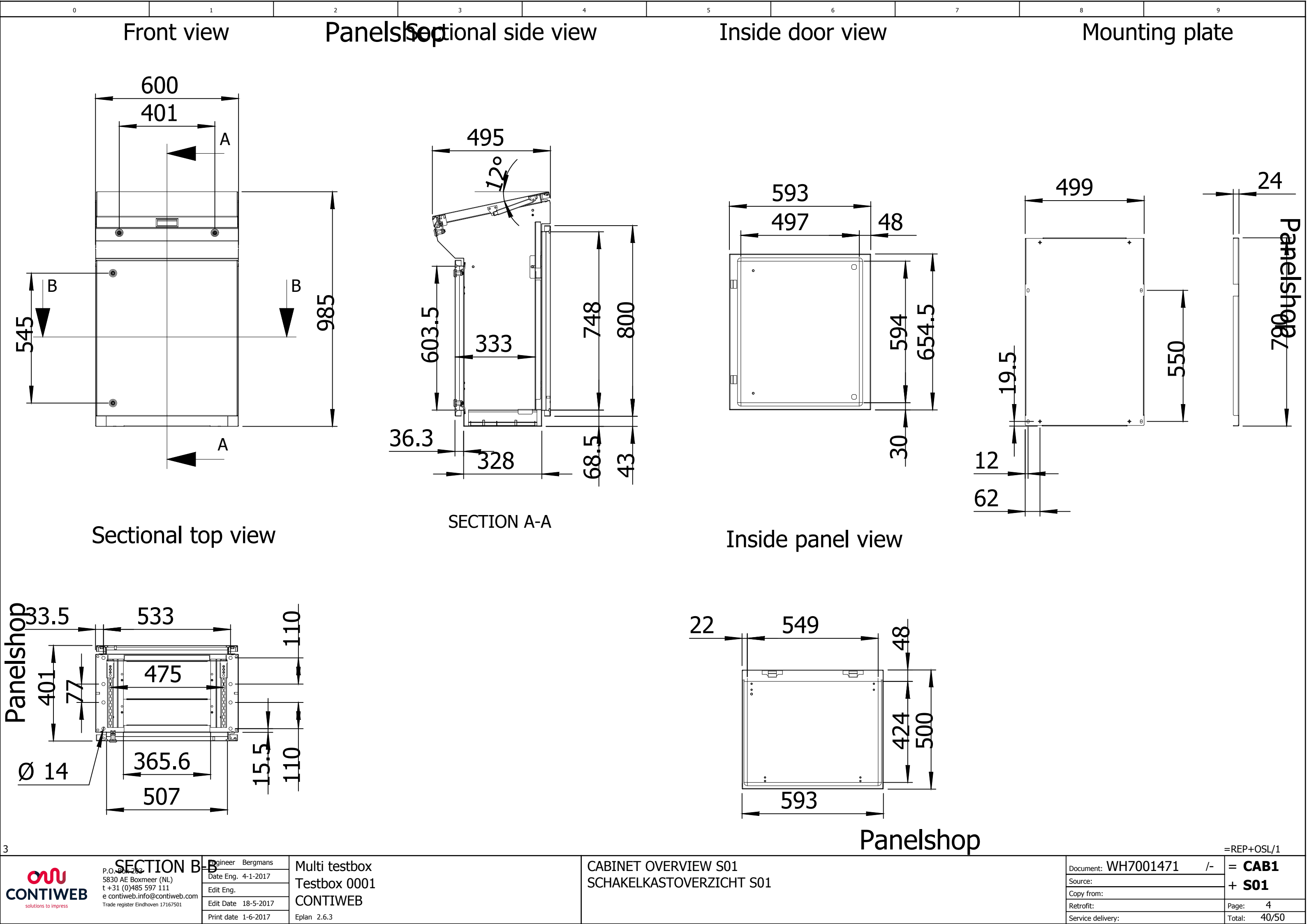
# Panelshop



Panelshop

Panelshop

Panelshop







Spare parts list  
Reserveonderdelenlijst

Panelshop

Cabinet Location	Component tag name	Drawing page	Component Order number	Description 1 (Englisch)	Description 1 (translated)	Length
		=CAB1+S01/2	WH095833716	MOUNTING PLATE	MOUNTING PLATE	
+S01	=MSW+S01-Q1	=MSW+S01/1	WH095200834	SWITCH-DISCONNECTOR 3P	SWITCH-DISCONNECTOR 3P	
+S01	=MSW+S01-Q1	=MSW+S01/1	WH095252331	COVER SWITCH-DISCONNECTOR	COVER SWITCH-DISCONNECTOR	
+S01	=VAC+S01-F1	=VAC+S01/2	WH095190208	CIRCUIT BREAKER	CIRCUIT BREAKER	
+S01	=VAC+S01-F2	=VAC+S01/2	WH095190208	CIRCUIT BREAKER	CIRCUIT BREAKER	
+S01	=VAC+S01-F3	=VAC+S01/2	WH095200717	CURRRENT LIMITTING MODULE	CURRRENT LIMITTING MODULE	
+S01	=VAC+S01-F4	=CAB1+S01/2	WH095200717	CURRRENT LIMITTING MODULE	CURRRENT LIMITTING MODULE	
+S01	=VAC+S01-G1	=VAC+S01/2	WH095832357	POWER SUPPLY UNIT	POWER SUPPLY UNIT	
+S01	=CON+S01-K1	=CON+S01/1	WH095833717	CONTROL UNIT	CONTROL UNIT	
+S01	=CON+S01-K1	=CON+S01/1	WH095833716	MOUNTING PLATE	MOUNTING PLATE	
+S01	=CON+S01-K1	=CON+S01/1	NA02WH095833656	CF CARD	CF CARD	
+S01	=CON+S01-K1	=CON+S01/1	WH095832783	FLASHDISK CF CARD	FLASHDISK CF CARD	
+S01	=CON+S01-K2	=CON+S01/2	WH095833730	TERMINAL MODULE	TERMINAL MODULE	
+S01	=CON+S01-W1	=CON+S01/2	WH095095305	CABLE, ETHERNET	CABLE, ETHERNET	1 m
+S01	=CON+S01-W1	=CON+S01/2	WH095832244	CONNECTOR	CONNECTOR	1 m
+S01	=CON+S01-W2	=CON+S01/2	WH095095308	CABLE, TWISTED PAIR	CABLE, TWISTED PAIR	2 m
+S01	=IO1+S01-K0	=IO1+S01/1	WH095832307	INTERFACE MODULE IM 155-6 PN ST	INTERFACE MODULE IM 155-6 PN ST	
+S01	=IO1+S01-K1	=IO1+S01/1	WH095832306	OUTPUT MODULE 20MM	OUTPUT MODULE 20MM	
+S01	=IO1+S01-K1	=IO1+S01/1	WH095832311	BASE UNIT (FURTHER) 20MM	BASE UNIT (FURTHER) 20MM	
+S01	=IO1+S01-K2	=IO1+S01/1	WH095832306	OUTPUT MODULE 20MM	OUTPUT MODULE 20MM	
+S01	=IO1+S01-K2	=IO1+S01/1	WH095832311	BASE UNIT (FURTHER) 20MM	BASE UNIT (FURTHER) 20MM	
+S01	=IO1+S01-K3	=IO1+S01/1	WH095832306	OUTPUT MODULE 20MM	OUTPUT MODULE 20MM	
+S01	=IO1+S01-K3	=IO1+S01/1	WH095832311	BASE UNIT (FURTHER) 20MM	BASE UNIT (FURTHER) 20MM	
+S01	=IO1+S01-K4	=IO1+S01/1	WH095832306	OUTPUT MODULE 20MM	OUTPUT MODULE 20MM	
+S01	=IO1+S01-K4	=IO1+S01/1	WH095832311	BASE UNIT (FURTHER) 20MM	BASE UNIT (FURTHER) 20MM	
+S01	=IO1+S01-K5	=IO1+S01/1	WH095832306	OUTPUT MODULE 20MM	OUTPUT MODULE 20MM	
+S01	=IO1+S01-K5	=IO1+S01/1	WH095832311	BASE UNIT (FURTHER) 20MM	BASE UNIT (FURTHER) 20MM	
+S01	=IO1+S01-K6	=IO1+S01/1	WH095832306	OUTPUT MODULE 20MM	OUTPUT MODULE 20MM	
+S01	=IO1+S01-K6	=IO1+S01/1	WH095832311	BASE UNIT (FURTHER) 20MM	BASE UNIT (FURTHER) 20MM	
+S01	=IO1+S01-K7	=IO1+S01/1	WH095832306	OUTPUT MODULE 20MM	OUTPUT MODULE 20MM	
+S01	=IO1+S01-K7	=IO1+S01/1	WH095832311	BASE UNIT (FURTHER) 20MM	BASE UNIT (FURTHER) 20MM	
+S01	=IO1+S01-K8	=IO1+S01/1	WH095832334	ANALOG OUTPUT MODULE	ANALOG OUTPUT MODULE	
+S01	=IO1+S01-K8	=IO1+S01/1	WH095832308	BASE UNIT (FURTHER)	BASE UNIT (FURTHER)	
+S01	=IO1+S01-S1	=IO1+S01/3	WH095262292	CONTACT BLOCK, FRONTMOUNT 2NC, SPRING	CONTACT BLOCK, FRONTMOUNT 2NC, SPRING	
+S01	=IO1+S01-S1	=IO1+S01/3	WH095262298	MUSHROOM BUTTON HEAD	MUSHROOM BUTTON HEAD	
+S01	=IO1+S01-S1	=IO1+S01/3	WH095262297	EMERGENCY STOP GUARDING RING	EMERGENCY STOP GUARDING RING	
+S01	=IO1+S01-S1	=IO1+S01/3	WH095262283	BODY / FIXING COLLAR	BODY / FIXING COLLAR	
+S01	=IO1+S01-S2	=IO1+S01/3	WH095262292	CONTACT BLOCK, FRONTMOUNT 2NC, SPRING	CONTACT BLOCK, FRONTMOUNT 2NC, SPRING	
+S01	=IO1+S01-W3	=IO1+S01/2	WH095095305	CABLE, ETHERNET	CABLE, ETHERNET	1 m
+S01	=IO1+S01-W3	=IO1+S01/2	WH095832244	CONNECTOR	CONNECTOR	1 m
+S01	=HMI1+S01-PC1	=HMI1+S01/1	WH095833712	PANEL PC SIMATIC IPC477D	PANEL PC SIMATIC IPC477D	
+S01	=HMI1+S01-PC1	=HMI1+S01/1	WH095832781	CFAST CARD	CFAST CARD	
+S01	=HMI1+S01-PC1	=HMI1+S01/1	WH095112002	CABLE SOCKET, RING, RED	CABLE SOCKET, RING, RED	
+S01	=HMI1+S01-PC1	=HMI1+S01/1	WH095832752	USB STICK	USB STICK	
+S01	=HMI1+S01-W4	=HMI1+S01/1	WH095095305	CABLE, ETHERNET	CABLE, ETHERNET	2 m
+S01	=HMI1+S01-W4	=HMI1+S01/1	WH095832244	CONNECTOR	CONNECTOR	2 m
+S01	=HMI1+S01-W5	=HMI1+S01/1	WH095095305	CABLE, ETHERNET	CABLE, ETHERNET	2 m
+S01	=HMI1+S01-W5	=HMI1+S01/1	WH095832244	CONNECTOR	CONNECTOR	2 m
+S01	=CAB1+S01-GND1.1	=CAB1+S01/1	WH095701009	EARTHRAIL	EARTHRAIL	

Panelshop

Panelshop

Panelshop

+OSL/1

+CABLES/1

Panelshop

Cable diagram

Cable name =CON+S01-W1			cable type CAT5					
function text CONTROL SYSTEM BESTURING			no. of conductors 2X2			cross-section 22		cable length 1
function text	Page / column	Target designation from	Connection point	conductor	Target designation to	Connection point	Page / column	function text
DRIVE-CLiQ	/1.1	-K1	X100:X100		-K2	X500:X500	/2.1	DRIVE-CLiQ
Cable name =CON+S01-W2			cable type TP for RS-485					
function text CONTROL SYSTEM BESTURING			no. of conductors 3x2			cross-section 24		cable length 2
function text	Page / column	Target designation from	Connection point	conductor	Target designation to	Connection point	Page / column	function text
	/2.5	=IO1-H106	1	WHBU	-K2	X520:1	/2.5	A
	/2.6	=IO1-H106	2	BUWH	-K2	X520:6	/2.6	A*
	/2.6	=IO1-H106	3	WHOG	-K2	X520:3	/2.6	B
	/2.6	=IO1-H106	4	OGWH	-K2	X520:8	/2.6	B*
	/2.6	=IO1-H106	5	WHGN	-K2	X520:2	/2.6	R
				GNWH				
				SH				
	/2.7	-W2			-K2	PE	/2.7	GND

Panelshop

Panelshop

Panelshop



Overview cable run list

Panelshop

Project : WH7001471

Overzicht kabelloop

#	Cable name	Cable Between:	
1	=CON+S01-W1 <small>2170886 CAT5 2X2x22 AWG</small>	=CON+S01-K1	=CON+S01-K2
2	=CON+S01-W2 <small>9843 TP for RS-485 3x2x24 AWG</small>	=IO1+S01-H106	=CON+S01-K2
3	=IO1+S01-W3 <small>2170886 CAT5 2X2x22 AWG</small>	=CON+S01-K1	=IO1+S01-K0
4	=HMI1+S01-W4 <small>2170886 CAT5 2X2x22 AWG</small>	=CON+S01-K1	=HMI1+S01-PC1
5	=HMI1+S01-W5 <small>2170886 CAT5 2X2x22 AWG</small>	=IO1+S01-H201	=HMI1+S01-PC1

Panelshop

Panelshop

Panelshop

## Panelshop

# Panelshop



# MAT

## Bill of materials Materiaallijst

# Panelshop

Pos	Tag	Component	Quantity	Description 1	Length	Description 2	Type number Article number	Manufacturer	Certificates	supl	Retro fit
1		NA00WH095832322	1	EARTH TERMINAL (TENSION-SPRING)		2,5 MM² GREEN-YELLOW (3 CONNECTIONS)	ZSL 2,5/3A GNYE CON.ZSL 2,5/3A_EARTH	CONTA CLIP	XCFR2.E95701	E	
2		NA00WH095145431	1	DOUBEL-LEVEL EARTH TERMINAL (TENSION-SPRING)		2 x 2,5 MM² GREEN-YELLOW (4 CONNECTIONS)	ZSLD 2,5 GNYE CON.ZSLD 2,5	CONTA CLIP	XCFR2.E95701	E	
3		WH095833716	1	MOUNTING PLATE MONTAGEPLAAT		BACKPLANE FOR SIMOTION D410	6AU1400-7AA05-0AA0 SIE.6AU1400-7AA05-0AA0	SIEMENS		E	
4	=MSW+S01 -Q1	WH095200834	1	SWITCH-DISCONNECTOR 3P HOOFDSCHAKELAAR		In: 32A IEC / 30A UL Iq: 80kA IEC / 5 kA UL	P1-32/EA/SVB-SW MOE.P1-32/EA/SVB-SW	MOELLER/EATON	NLRV.E36332	E	Panelshop
5	=MSW+S01 -Q1	WH095252331	1	COVER SWITCH-DISCONNECTOR		FOR P1-/EA/SVB	H-P1 MOE.H-P1	MOELLER/EATON	NLRV.E36332	E	
6	=VAC+S01 -F1	WH095190208	1	CIRCUIT BREAKER VERMOGENSCHAKELAAR		I=3A 1-POL. D CAR.	SE.60121 SE.60121_3A_D	SCHNEIDER	DIVQ.E215117	E	
7	=VAC+S01 -F2	WH095190208	1	CIRCUIT BREAKER VERMOGENSCHAKELAAR		I=3A 1-POL. D CAR.	SE.60121 SE.60121_3A_D	SCHNEIDER	DIVQ.E215117	E	
8	=VAC+S01 -F3	WH095200717	1	CURRRENT LIMITTING MODULE STROOMBEPERKINGS APPARAAT		24VDC 4X3,7A	PISA11.CLASS2 PULS.PISA11.CLASS2	PULS	NMTR.E198865	E	
9	=VAC+S01 -F4	WH095200717	1	CURRRENT LIMITTING MODULE STROOMBEPERKINGS APPARAAT		24VDC 4X3,7A	PISA11.CLASS2 PULS.PISA11.CLASS2	PULS	NMTR.E198865	E	
10	=VAC+S01 -G1	WH095832357	1	POWER SUPPLY UNIT VOEDING		SITOP 10A, 120/230-500VAC,24VDC	6EP1334-3BA00 SIE.6EP1334-3BA00	SIEMENS		E	
11	=VAC+S01 -X1	WH095147916	2	SCREW END STOP		FOR RAIL TS35	ES 35/K/ST BG CON.ES35/K/ST	CONTA CLIP	-	E	
12	=VAC+S01 -X1	WH095147950	1	MARKER HOLDER FOR END STOP		BEIGE	ZSchT 6 CON.ZSchT6	CONTA CLIP	-	E	
13	=VAC+S01 -X1	NA00WH095147921	2	END PLATE (FOR ZRK 2,5/2A)		BEIGE	ZAP 2,5/2A BG CON.ZAP 2,5/2A_END_PLATE	CONTA CLIP	-	E	
14	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
15	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
16	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
17	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
18	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
19	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
20	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
21	=VAC+S01 -X1	NA00WH095145421	1	TERMINAL (TENSION-SPRING)		2,5 MM² BEIGE (2 CONNECTIONS)	ZRK 2,5/2A BG CON.ZRK 2,5/2A	CONTA CLIP	XCFR2.E95701	E	
22	=VAC+S01 -X3	WH095147916	1	SCREW END STOP		FOR RAIL TS35	ES 35/K/ST BG CON.ES35/K/ST	CONTA CLIP	-	E	
23	=VAC+S01 -X3	WH095147950	1	MARKER HOLDER FOR END STOP		BEIGE	ZSchT 6 CON.ZSchT6	CONTA CLIP	-	E	
24	=VAC+S01 -X3	NA00WH095147920	1	END PLATE (FOR ZRK 4/3A)		BEIGE	ZAP 4/3A BG CON.ZAP 4/3A_END_PLATE	CONTA CLIP	-	E	
25	=VAC+S01 -X3	NA00WH095147921	1	END PLATE (FOR ZRK 2,5/2A)		BEIGE	ZAP 2,5/2A BG CON.ZAP 2,5/2A_END_PLATE	CONTA CLIP	-	E	
26	=VAC+S01 -X3	NA00WH095145420	1	TERMINAL (TENSION-SPRING)		4 MM² BEIGE (3 CONNECTIONS)	ZRK 4/3A BG CON.ZRK 4/3A	CONTA CLIP	XCFR2.E95701	E	
27	=VAC+S01 -X3	NA00WH095145420	1	TERMINAL (TENSION-SPRING)		4 MM² BEIGE (3 CONNECTIONS)	ZRK 4/3A BG CON.ZRK 4/3A	CONTA CLIP	XCFR2.E95701	E	
28	=CON+S01 -K1	WH095833717	1	CONTROL UNIT BESTURINGSEENHEID		SIMOTION D410-2 DP/PN	6AU1410-2AD00-0AA0 SIE.6AU1410-2AD00-0AA0	SIEMENS		E	


$$= \text{REP} + \text{CABLES} / 100$$

## Panelshop

2

 <p>P.O. Box 203 5830 AE Boxmeer (NL) t +31 (0)485 597 111 e contiweb.info@contiweb.com Trade register Eindhoven 17167501</p>	Engineer Bergmans	<p>Multi testbox Testbox 0001 CONTIWEB</p> <p>Eplan 2.6.3</p>	<p>MATERIAL LISTS</p> <p>Material list Materiaallijst</p>	Document: WH7001471 /- = <b>MAT</b>
	Date Eng. 4-1-2017			Source: +
	Edit Eng. Mark Bergmans			Copy from:
	Edit Date 29-5-2017			Retrofit: Page: 1
	Print date 1-6-2017			Service delivery: Total: 48/50



0		1		2		3		4		5		6		7		8		9	
Panelshop																			
MAT Bill of materials Materiaallijst																			
Pos	Tag		Component	Quantity	Description 1	Length	Description 2	Type number Article number		Manufacturer	Certificates	supl	Retro fit						
29	=CON+S01	-K1	WH095833716	1	MOUNTING PLATE MONTAGEPLAAT		BACKPLANE FOR SIMOTION D410	6AU1400-7AA05-0AA0 SIE.6AU1400-7AA05-0AA0		SIEMENS		E							
30	=CON+S01	-K1	NA02WH095833656	1	CF CARD CF KAAART		1 GB	6AU1 400-2PA01-0AA0 SIE.CF CARD_1GB_6AU1400-2PA01-0AA0		SIEMENS	C/UL60950 FILE E115352	E							
31	=CON+S01	-K1	WH095832783	1	FLASHDISK CF CARD		1 GB COMPACT FLASH CARD	6AU1400-2PA23-0AA0 SIE.6AU1400-2PA23-0AA0		SIEMENS	C/UL60950 FILE E115352	E							
32	=CON+S01	-K2	WH095833730	1	TERMINAL MODULE AANSLUITVOET		TM31 8DI-4DI/DO-2RO-2AI-2AO-1KTY	6SL3055-0AA00-3AA1 SIE.6SL3055-0AA00-3AA1		SIEMENS									
33	=CON+S01	-W1	WH095095305	1	CABLE, ETHERNET	1 m	ETHERLINE PN Cat.5 Y FLEX FC 2X2XAWG22/7	2170886 CABLE.ETH 2X2XAWG22/7		LAPP	DUZX.E236660	E							
34	=CON+S01	-W1	WH095832244	2	CONNECTOR VERBINDER		ETHERNET FAST CONNECT RJ45	09451511100 HAR.PROFINET_B		HARTING		E							
35	=CON+S01	-W2	WH095095308	1	CABLE, TWISTED PAIR	2 m	3X2XAWG 24	9843 BELD.TP_3PAIR_FOR_RS485		BELDEN		E							
37	=IO1+S01	-K0	WH095832307	1	INTERFACE MODULE IM 155-6 PN ST		ET200SP PROFINET	6ES7155-6AA00-0BN0 SIE.6ES7155-6AA00-0BN0		SIEMENS	NRAQ.E116536	E							
38	=IO1+S01	-K1	WH095832306	7	OUTPUT MODULE 20MM		4 OUTPUTS RQ 120VDC-230VAC/5A NO ST	6ES7132-6HD00-0BB0 SIE.6ES7132-6HD00-0BB0		SIEMENS	NRAQ.E116536	E							
39	=IO1+S01	-K1	WH095832311	7	BASE UNIT (FURTHER) 20MM		BU20-P12+A4+0B	6ES7193-6BP20-0BB0 SIE.6ES7193-6BP20-0BB0		SIEMENS	NRAQ.E116536	E							
40	=IO1+S01	-K2	WH095832306	1	OUTPUT MODULE 20MM		4 OUTPUTS RQ 120VDC-230VAC/5A NO ST	6ES7132-6HD00-0BB0 SIE.6ES7132-6HD00-0BB0		SIEMENS	NRAQ.E116536	E							
41	=IO1+S01	-K2	WH095832311	1	BASE UNIT (FURTHER) 20MM		BU20-P12+A4+0B	6ES7193-6BP20-0BB0 SIE.6ES7193-6BP20-0BB0		SIEMENS	NRAQ.E116536	E							
42	=IO1+S01	-K3	WH095832306	1	OUTPUT MODULE 20MM		4 OUTPUTS RQ 120VDC-230VAC/5A NO ST	6ES7132-6HD00-0BB0 SIE.6ES7132-6HD00-0BB0		SIEMENS	NRAQ.E116536	E							
43	=IO1+S01	-K3	WH095832311	1	BASE UNIT (FURTHER) 20MM		BU20-P12+A4+0B	6ES7193-6BP20-0BB0 SIE.6ES7193-6BP20-0BB0		SIEMENS	NRAQ.E116536	E							
44	=IO1+S01	-K4	WH095832306	1	OUTPUT MODULE 20MM		4 OUTPUTS RQ 120VDC-230VAC/5A NO ST	6ES7132-6HD00-0BB0 SIE.6ES7132-6HD00-0BB0		SIEMENS	NRAQ.E116536	E							
45	=IO1+S01	-K4	WH095832311	1	BASE UNIT (FURTHER) 20MM		BU20-P12+A4+0B	6ES7193-6BP20-0BB0 SIE.6ES7193-6BP20-0BB0		SIEMENS	NRAQ.E116536	E							
46	=IO1+S01	-K5	WH095832306	1	OUTPUT MODULE 20MM		4 OUTPUTS RQ 120VDC-230VAC/5A NO ST	6ES7132-6HD00-0BB0 SIE.6ES7132-6HD00-0BB0		SIEMENS	NRAQ.E116536	E							
47	=IO1+S01	-K5	WH095832311	1	BASE UNIT (FURTHER) 20MM		BU20-P12+A4+0B	6ES7193-6BP20-0BB0 SIE.6ES7193-6BP20-0BB0		SIEMENS	NRAQ.E116536	E							
48	=IO1+S01	-K6	WH095832306	1	OUTPUT MODULE 20MM		4 OUTPUTS RQ 120VDC-230VAC/5A NO ST	6ES7132-6HD00-0BB0 SIE.6ES7132-6HD00-0BB0		SIEMENS	NRAQ.E116536	E							
49	=IO1+S01	-K6	WH095832311	1	BASE UNIT (FURTHER) 20MM		BU20-P12+A4+0B	6ES7193-6BP20-0BB0 SIE.6ES7193-6BP20-0BB0		SIEMENS	NRAQ.E116536	E							
50	=IO1+S01	-K7	WH095832306	1	OUTPUT MODULE 20MM		4 OUTPUTS RQ 120VDC-230VAC/5A NO ST	6ES7132-6HD00-0BB0 SIE.6ES7132-6HD00-0BB0		SIEMENS	NRAQ.E116536	E							
51	=IO1+S01	-K7	WH095832311	1	BASE UNIT (FURTHER) 20MM		BU20-P12+A4+0B	6ES7193-6BP20-0BB0 SIE.6ES7193-6BP20-0BB0		SIEMENS	NRAQ.E116536	E							
52	=IO1+S01	-K8	WH095832334	1	ANALOG OUTPUT MODULE ANALOGUE UITGANGSKAART		AQ 2xU/I HS	6ES7135-6HB00-0DA1 SIE.6ES7135-6HB00-0DA1		SIEMENS	NRAQ.E116536	E							
53	=IO1+S01	-K8	WH095832308	1	BASE UNIT (FURTHER)		BU15-P16+A0+2B	6ES7193-6BP00-0BA0 SIE.6ES7193-6BP00-0BA0		SIEMENS	NRAQ.E116536	E							
54	=IO1+S01	-S1	WH095262292	2	CONTACT BLOCK, FRONTMOUNT 2NC, SPRING		2x NC, SPRING	M22-CK02 MOE.M22-CK02		MOELLER/EATON	NKCR.E29184	E							
55	=IO1+S01	-S1	WH095262298	1	MUSHROOM BUTTON HEAD PADDENSTOEL DRUKKNOP		D=45 RED TURN RELEASE	M22-PVT45P MOE.M22-PVT45P		MOELLER/EATON	NKCR.E29184	E							
56	=IO1+S01	-S1	WH095262297	1	EMERGENCY STOP GUARDING RING		YELLOW	M22-XGPV MOE.M22-XGPV		MOELLER/EATON	NKCR.E29184	E							
57	=IO1+S01	-S1	WH095262283	1	BODY / FIXING COLLAR		3 POSITIONS	M22-A MOE.M22-A		MOELLER/EATON	NKCR.E29184	E							
Panelshop																			
3																			
<div></div> <div>P.O. Box 203 5830 AE Boxmeer (NL) t +31 (0)485 597 111 e contiweb.info@contiweb.com Trade register Eindhoven 17167501</div>			Engineer Bergmans		Multi testbox Testbox 0001 CONTIWEB  Eplan 2.6.3			MATERIAL LISTS  Material list Materiaallijst			Document: WH7001471 /-		= <b>MAT</b> +  Page: 2 Total: 49/50						
			Date Eng. 4-1-2017								Source:								
			Edit Eng. Mark Bergmans								Copy from:								
			Edit Date 1-6-2017								Retrofit:								
			Print date 1-6-2017								Service delivery:								



# Panelshop

# Panelshop

## Bijlage 4: Software

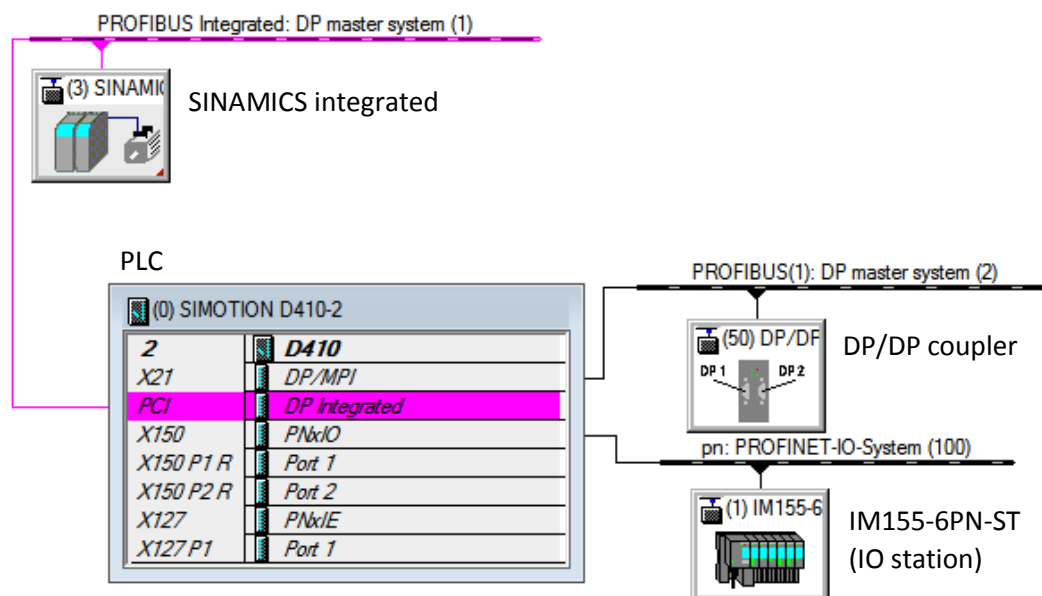
De software wordt uitgevoerd op twee devices, de PLC en de HMI. Hierbij is de functie van de PLC beperkt tot het ontvangen en doorsturen van de verschillende input, output en Profibus signalen. Zelf veranderd deze PLC niks, met uitzondering van het encoder signaal. De tests en de communicatie met de gebruiker worden dus uitgevoerd op de HMI. Momenteel wordt nog aan de software gewerkt, er moet onder andere nog het encoder signaal en de documentatie in de software toegevoegd worden. Daarnaast kan de code op veel punten nog verbeterd worden.

### PLC (ST)

Zoals hiervoor genoemd wordt de PLC enkel gebruikt voor het doorgeven van informatie, hoe dit gebeurt zal in deze paragraaf beschreven worden. De gebruikte taal in de PLC is Structured Tekst. In de volgende stukken zal eerst de hardware configuratie weergegeven worden, daarna zal ingegaan worden op de programma's zelf.

### Hardware configuratie

In figuur B4-1 is te zien hoe de verschillende hardware componenten met de PLC ((0)SIMOTION D410-2) verbonden zijn. In de figuren B4-2, B4-3 en B4-4, is weergegeven welke I/O adressen aan de verschillende outputs van deze componenten toegekend zijn en in Figuur B4-5 is de address list met de toegekende namen weergegeven. Enkel de toegekende namen zullen nog terug komen in de software.



Figuur B4-1 Configuratie componenten

(3) SINAMICS_Integrated					
Slot	Module	Telegram selection / default	I address	Q address	Comment
4	Drive Data	SIEMENS telegram 394, F2D-3/3	256...261		Control_Unit
5	Drive Data	SIEMENS telegram 394, F2D-3/3		256...261	Control_Unit
6	Drive Data				

Figuur B4-2 Adressen SINAMICS integrated

(50) DP/DP Coupler, Release 2					
Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	160	1 Byte Output consistent		262	
2	144	1 Byte Input consistent	262		
3	128	32 Bytes Output consistent		279...310	
4	128	32 Bytes Output consistent		311...342	
5	128	32 Bytes Output consistent		343...374	
6	175	16 Bytes Output consistent		263...278	
7	64	32 Bytes Input consistent	263...294		
8	64	32 Bytes Input consistent	295...326		
9	64	32 Bytes Input consistent	327...358		
10	159	16 Bytes Input consistent	359...374		


Figuur B4-3 Adressen DP/DP coupler

(1) IM155-6PN-ST								Pack Address	
Slot	Module	Order number	I address	Q address	Diagnostic addr...	C...	A...		
0	IM155-6PN-ST	6ES7 155-6AU00-0BNO			8179*				Full
X1	PN-IO				8178*				Full
X1 P1 R	Port 1				8181*				Full
X1 P2 R	Port 2				8180*				Full
1	DI16 x 24VDC ST	6ES7 131-6BH00-0BA0	0.0...1.7						Full
2	DQ16 x 24VDC/0.5A ST	6ES7 132-6BH00-0BA0		0.0...1.7					Full
3	RQ4 x 120VDC...230VAC/5A~	6ES7 132-6HD00-0BB0		2.0...2.3					Full
4	Server module	6ES7 193-6PA00-0AA0			8177*				Full

Figuur B4-4 Adressen IM155-6PN-ST (IO station)

	Name	I/O address	Read o	Data type	Array	Pro	Strategy	Display	Substitute		Control value
1	DPC_I_Slot10_0	PID 359		DWORD	1		Last value	HEX			
2	DPC_I_Slot10_10	PIB 369		BYTE	1		Last value	HEX			
3	DPC_I_Slot10_11	PIB 370		BYTE	1		Last value	HEX			
4	DPC_I_Slot10_12	PIW 371		WORD	1		Last value	HEX			
5	DPC_I_Slot10_14	PIW 373		WORD	1		Last value	HEX			
6	DPC_I_Slot10_4	PIW 363		WORD	1		Last value	HEX			
7	DPC_I_Slot10_6	PID 365		DWORD	1		Last value	HEX			
8	DPC_I_Slot7_0	PIB 263		BYTE	1		Last value	HEX			
9	DPC_I_Slot7_1	PIB 264		BYTE	1		Last value	HEX			
10	DPC_I_Slot9_28	PID 355		DWORD	1		Last value	HEX			
11	DPC_O_Slot1	PQB 262	<input type="checkbox"/>	BYTE	1		Last value	HEX		<input type="checkbox"/>	16#00
12	DPC_O_Slot3_0	PQB 279	<input type="checkbox"/>	BYTE	1		Last value	HEX		<input checked="" type="checkbox"/>	16#20
13	DPC_O_Slot3_2	PQB 281	<input type="checkbox"/>	BYTE	1		Last value	HEX		<input type="checkbox"/>	16#00
14	IOS_DI_Slot1a	PIB 0		BYTE	1		Substitut...	HEX	16#00		
15	IOS_DI_Slot1b	PIB 1		BYTE	1		Substitut...	HEX	16#00		
16	IOS_DO_Slot2a	PQB 0		BYTE	1		Substitut...	HEX	16#00	<input type="checkbox"/>	16#00
17	IOS_DO_Slot2b	PQB 1		BYTE	1		Substitut...	HEX	16#00	<input type="checkbox"/>	16#00
18	IOS_DO_Slot3	PQB 2		BYTE	1		Substitut...	HEX	16#00	<input type="checkbox"/>	16#00
19	PLC_DI_Word1	IN		WORD	1		Substitut...	HEX	16#00_00		
20	PLC_DI_Word2	IN		WORD	1		Substitut...	HEX	16#00_00		
21	PLC_DO_Byte2	OUT	<input type="checkbox"/>	BYTE	1		Substitut...	HEX	16#00	<input type="checkbox"/>	16#00
22	PLC_DO_Byte3	OUT	<input type="checkbox"/>	BYTE	1		Substitut...	HEX	16#00	<input type="checkbox"/>	16#00
23	PLC_DO_Byte4	OUT	<input type="checkbox"/>	BYTE	1		Substitut...	HEX	16#00	<input type="checkbox"/>	16#00
24	PLC_DO_X121	OUT	<input type="checkbox"/>	BYTE	1		Substitut...	HEX	16#00	<input type="checkbox"/>	16#00


Figuur B4-5 Address List

	Teststelsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
---	-----------------------------------	---

## Main

De complete code van de main is weergegeven in figuur B4-6. In dit figuur zijn drie programma's dicht geklapt (regels 46, 51 en 56), deze zijn "leeg" zoals weergegeven bij het programma "prg\_timefault\_task" (regel 41).

De PLC voert verschillende programma's uit zoals deze in het execution system geordend staan. Deze programma's zijn geschreven in de main. Op regel 13 wordt aangeroepen dat de main gebruik maakt van het programma "tcp\_connection". Onder het programma "prg\_background\_task" (regel 15) wordt het functie block "fb\_tcp\_connection" aangeroepen (regel 21), dit is het deel van het programma waar de daadwerkelijke functionaliteit geschreven staat. In het programma "prg\_timer\_task" worden verschillende variabelen gedefinieerd, deze worden echter niet gebruikt. Daarna volgen nog vier programma's die geen inhoud hebben, hierdoor zal de PLC direct doorgaan naar de volgende task.

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 3 van 40

```

1  INTERFACE
2      PROGRAM prg_background_task;
3      PROGRAM prg_timer_task;
4      PROGRAM prg_timefault_task;
5      PROGRAM prg_startup_task;
6      PROGRAM prg_shutdown_task;
7      PROGRAM prg_peripheralfault_task;
8
9  END_INTERFACE
10
11 IMPLEMENTATION
12
13     USES tcp_connection;
14
15     PROGRAM prg_background_task           //aanroepen tcp_connection
16
17         VAR
18             connection      : fb_tcp_connection;
19         END_VAR;
20
21         connection(i_ul6ServerPort := 1000);
22
23     END_PROGRAM // prg_background_task
24
25     PROGRAM prg_timer_task               //aanroepen variabelen, not used
26
27         VAR CONSTANT
28             TIMERTASK_INTERVAL           : INT := 10; // [ms]
29         END_VAR
30
31         VAR
32             counter_50ms                : DINT := 0;
33             counter_100ms               : DINT := 20;
34             counter_200ms               : DINT := 30;
35             counter_1500ms              : DINT := 40;
36         END_VAR
37     ;
38
39     END_PROGRAM // prg_timer_task
40
41     PROGRAM prg_timefault_task           //empty
42     ;
43
44     END_PROGRAM // prg_timefault_task
45
46     PROGRAM prg_startup_task             //empty
47
48
49
50
51     PROGRAM prg_shutdown_task            //empty
52
53
54
55
56     PROGRAM prg_peripheralfault_task     //empty
57
58
59
60
61 END_IMPLEMENTATION

```

Figuur B4-6 Main programma PLC software

## Genblk

Genblk is een afkorting voor generate blink. Wanneer het functie blok “fb\_genblk” (regel 9) aangeroepen wordt moet een tijd aan gegeven worden, het uitgangssignaal is een boolean. Stel de gegeven tijd is één seconde en het uitgangs signaal begint op false, dan zal de uitgang na één seconden true worden en één seconden later weer false. Dit proces zal zich blijven herhalen en de uitgang zal dus “knipperen”. Dit gebeurt doordat de volgende vier stappen herhaaldelijk doorlopen worden.

s\_indicator\_blink\_time is in dit voorbeeld 1 seconden

1. (regel 24) s\_indicator\_blink\_time = false: na 1 seconden zal on\_cycle.q true worden.
2. (regel 27) on\_cycle.q = true: s\_indicator\_blink\_time wordt true. (on\_cycle.q wordt false)
3. (regel 25) s\_indicator\_blink\_time = true: na 1 seconden zal off\_cycle.q true worden.
4. (regel 29) off\_cycle.q = true: s\_indicator\_blink\_time wordt false. (off\_cycle.q wordt false)
5. Terug naar stap 1

```

1  INTERFACE
2
3      FUNCTION_BLOCK fb_genblk;
4
5  END_INTERFACE
6
7  IMPLEMENTATION
8
9      FUNCTION_BLOCK fb_genblk
10
11         VAR_INPUT
12             s_indicator_blink_time : TIME;
13         END_VAR
14
15         VAR_OUTPUT
16             genblk_indicator_blink : BOOL;
17         END_VAR
18
19         VAR
20             on_cycle           : TON;
21             off_cycle          : TON;
22         END_VAR
23
24         on_cycle(in:=NOT genblk_indicator_blink, pt:=s_indicator_blink_time);
25         off_cycle(in:=genblk_indicator_blink, pt:=s_indicator_blink_time);
26
27         IF on_cycle.q THEN
28             genblk_indicator_blink := TRUE;
29         ELSIF off_cycle.q THEN
30             genblk_indicator_blink := FALSE;
31         END_IF;
32
33
34     END_FUNCTION_BLOCK // fb_genblk
35
36 END_IMPLEMENTATION

```

Figuur B4-7 Genblk programma PLC software

### Tcp\_connection

Het programma tcp connect bestaat uit 245 regels en is ingeklapt weergegeven in figuur B4-8, om dit overzichtelijk te houden zal dit programma in stukken besproken worden. Ten eerste maakt dit programma gebruik van het programma genblk wat in de vorige paragraaf besproken is. Op regel 75 wordt dit blink signaal gestart met een blink time van 100ms, de regel hierna wordt de output boolean gekoppeld aan de variabele second\_tick. De rest van het programma zal op de volgende pagina's besproken worden. Buiten de definities van de types en global constants valt alles onder het function block "fb\_tcp\_connection".

```

1  INTERFACE
2
3      FUNCTION_BLOCK fb_tcp_connection;
4
5  END_INTERFACE
6
7
8  IMPLEMENTATION
9
10     USES genblk;
11
12     TYPE
13
41
42     VAR GLOBAL CONSTANT
43
44
45     FUNCTION_BLOCK fb_tcp_connection
46
47     VAR INPUT
48
49
50
51
52     VAR
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75     genblk(s_indicator_blink_time := T#100ms);
76     second_tick(genblk.genblk_indicator_blink);
77
78     //Init,Open,Close Server
79     CASE open state OF
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118     // close connections
119     FOR i := 0 TO (MAX CONNECTIONS - 1) DO
120
121
122
123
124
125
126
127
128
129     // check on received data // alive test from client
130     FOR i := 0 TO (MAX CONNECTIONS - 1) DO
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211     aSendBuffer := ANYTYPE_TO_BIGBYTEARRAY (Data_Send, 0);
212
213     // Send data
214     FOR i := 0 TO (MAX CONNECTIONS - 1) DO
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243     END_FUNCTION_BLOCK // fb_tcp_connection
244
245 END_IMPLEMENTATION

```

Figuur B4-8 Tcp\_connection programma PLC software

#### Regel 12 tot 44: Defenitie types en global constants (Figuur B4-9)

In deze regels worden drie types en één variabele gedefinieerd. Het eerste type is “connection\_state\_def” (regel 13), hiermee wordt bijgehouden in welke state het programma is. Hierna volgen twee structs, de eerste is tcp\_connections\_rec (regel 19). Voor elk via TCP/IP verbonden device zal een copy van deze struct bijgehouden worden, naast het standaard TCP/IP struct word er bijgehouden of het device connected is, de verbinding wil verbreken en een waarde die verstuurd moet worden. De tweede struct (regel 26) is de opbouw van de ontvangen berichten. Wanneer de ontvangen data in deze struct gecast wordt is het direct mogelijk om de bytes uit te lezen zonder dat er eerst conversies of berekeningen gedaan hoeven worden. De globale constant “MAX\_CONNECTIONS” stelt vast hoeveel TCP/IP verbindingen er maximaal gemaakt mogen worden, deze staat ingesteld op 50.

```

12  TYPE
13      connection_state_def : {
14          INIT_SERVER := 1,
15          OPEN_SERVER,
16          CLOSE_SERVER
17      };
18
19  tcp_connections_rec : STRUCT //used for state connection
20      RetTcpOpenServer : StructRetTcpOpenServer;
21      connected        : BOOL;
22      req_disconnect    : BOOL;
23      RetTcpSend        : DINT;
24  END_STRUCT;
25
26  Data_Struct          : STRUCT //used for receiving messages
27      State             : BYTE;
28      // IO Station
29      IOS_D0_Slot2a     : BYTE;
30      IOS_D0_Slot2b     : BYTE;
31      IOS_D0_Slot3      : BYTE;
32      //PLC IO
33      PLC_D0_X121       : BYTE;
34      //Profibus
35      DPC_0_Slot1       : BYTE;
36      DPC_0_Slot3_0     : BYTE;
37      DPC_0_Slot3_2     : BYTE;
38  END_STRUCT
39
40  END_TYPE
41
42  VAR_GLOBAL CONSTANT
43      MAX_CONNECTIONS : UINT := 50;
44  END_VAR

```

Figuur B4-9 Regel 12 tot 44: Defenitie types en global constants



#### Regel 48 tot 73: Definitie variabele (Figuur B4-10)

Het functionblock “fb\_tcp\_connection” heeft één input (regel 49), hiermee wordt ingesteld via welke client port devices verbinding kunnen maken met de PLC. Wanneer dit afwijkt van de in de HMI aangegeven poort zal er geen verbinding opgezet kunnen worden. Onder variabelen worden verschillende variabelen en functies gedefinieerd, wanneer deze in de code gebruikt worden zullen deze behandeld worden.

```

48  VAR_INPUT
49    i_ul6ServerPort          : UINT := 1000; // Client port (connect from HMI)
50  END_VAR
51
52  VAR
53    open_state               : connection_state_def;
54    connection_id            : ARRAY[0.. MAX_CONNECTIONS - 1] OF tcp_connections_rec;
55    ul6ServerPort            : UINT;
56    sRetOpenServer           : StructRetTcpOpenServer;
57    i32RetCloseServer        : DINT;
58    i32RetCloseConnectionServer : DINT;
59    ul6RecvDataLength        : UINT;
60
61    sRetTcpReceive           : StructRetTcpReceive;
62    aRecvBuffer              : ARRAY[0..1023] OF BYTE; //recievebuffer systemfunktion
63    aSendBuffer              : ARRAY[0..1023] OF BYTE;
64    i                        : UINT;
65    empty_connection_id      : tcp_connections_rec;
66    second_tick              : R_TRIG;
67    genblk                   : fb_genblk;
68    receiveData              : data_struct;
69
70    word_to_2byte            : _word_to_2byte;
71    dword_to_4byte           : _DWORD_TO_4BYTE;
72    Data_Send                : ARRAY [0.. 26] OF BYTE;
73  END_VAR

```

Figuur B4-10 Regel 48 tot 73: Definitie variabele

#### Regel 79 tot 116: State case (Figuur B4-11)

Met de variabele “open\_state” van het type “connection\_state\_def” word gekozen welke van de drie states uitgevoerd wordt. Deze states zullen in de volgende alinea’s afzonderlijk uitgewerkt worden.

De PLC begint in de state “INIT\_SERVER”, deze zal één keer doorlopen worden waarna naar de “OPEN\_SERVER” gegaan wordt. Voordat dit gebeurt zal de geselecteerde poort in een nieuwe variabele gecast worden en zal het address array gevuld worden met lege adressen, hiermee wordt verzekerd dat er geen adressen van voorgaande tests in het array blijven staan.

Wanneer het programma in de “OPEN\_SERVER” state is zal verbinding gemaakt kunnen worden vanuit andere devices, het programma zal overwegend in deze state zijn. Bij regel 90 wordt het verbinden met een nieuw device voorbereid. Wanneer het resultaat van “sRetOpenServer.functionresult” (regel 96) gelijk is aan nul zal dit betekenen dat een nieuw device verbinding probeert te maken met de PLC. Hierna zal in het address array gezocht worden naar de eerste lege plek (regel 98 en 99). Hier worden de gegevens van de nieuwe verbinding opgeslagen en ook zal met een boolean aangegeven worden dat de plek niet meer leeg is. Voor de zekerheid zal de boolean “req\_disconnect false gemaakt worden zodat de verbinding niet onbedoeld direct gesloten wordt. Wanneer “sRetOpenServer.functionresult” lager is dan nul betekent dit dat er een error is opgetreden, de PLC zal naar de “CLOSE\_SERVER” state gaan.


In de state “CLOSE\_SERVER” zal de server gesloten worden (regel 113) en hierna zal het programma direct terug naar de “INIT\_SERVER” state gaan. Bij een fout zal de server dus direct gereset worden.

```

79  CASE open_state OF
80
81  INIT_SERVER:
82      ul6ServerPort      := i_ul6ServerPort;
83      open_state         := OPEN_SERVER;
84
85      FOR i := 0 TO (MAX_CONNECTIONS - 1) DO
86          connection_id[i] := empty_connection_id;
87      END_FOR;
88
89  OPEN_SERVER:
90      sRetOpenServer := _tcpopenserver(
91          port      := ul6ServerPort,
92          backlog   := MAX_CONNECTIONS,
93          nextcommand := IMMEDIATELY
94      );
95
96      IF sRetOpenServer.functionResult = 0 THEN          // ready
97          // Make connection
98          FOR i := 0 TO (MAX_CONNECTIONS - 1) DO
99              IF (connection_id[i].connected = FALSE) THEN
100                  connection_id[i].RetTcpOpenServer := sRetOpenServer;
101                  connection_id[i].connected         := TRUE;
102                  connection_id[i].req_disconnect     := FALSE;
103              END_IF;
104          END_FOR;
105      ELSIF sRetOpenServer.functionResult > 0 THEN      // busy
106          ;
107      ELSIF sRetOpenServer.functionResult < 0 THEN      // error
108          open_state := CLOSE_SERVER;
109      END_IF;
110
111  CLOSE_SERVER:
112      i32RetCloseServer := _tcpCloseServer( port := ul6ServerPort );
113      open_state        := INIT_SERVER;
114
115  END_CASE;
116

```

Figuur B4-11 Regel 79 tot 116: State case

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

#### Regel 118 tot 127: Close connections (Figuur B4-12)

In dit gedeelte wordt voor elke verbinding gekeken of er aangevraagd is om deze verbinding te verbreken, dit kan zowel door de PLC als het verbonden device aangevraagd zijn. Als dit bij een van de verbindingen zo is zal via de server deze verbinding verbroken worden. Hierna zal in het address een leeg address geplaatst worden.

```


118      // close connections
119      FOR i := 0 TO (MAX_CONNECTIONS - 1) DO
120      IF connection_id[i].req_disconnect THEN
121      // disconnect client
122      i32RetCloseConnectionServer := _tcpCloseConnection( connection_id[i].RetTcpOpenServer.connectionId );
123      IF i32RetCloseConnectionServer = 0 THEN
124      connection_id[i] := empty_connection_id;
125      END_IF;
126      END_FOR;
127

```

Figuur B4- 12 Regel 118 tot 127: Close connections

#### Regel 129 tot 209: Check on received data (Figuur B4-13)

Voor elk address wordt gekeken of deze verbonden is en er niet aangevraagd is om de verbinding te verbreken (regel 131). In regel 134 wordt de nieuwe data naar een bytearray gekopieerd, deze wordt in regel 141 naar de struct "receivedata" gecast. In de regels 142 tot 148 wordt de ontvangen data direct naar de output adressen gekopieerd. Hierna worden in de regels 150 tot 192 de verschillende inputs van de PLC gekopieerd naar het array "Data\_Send" wat vanuit de PLC naar de verbonden devices gestuurd wordt. Op verschillende plekken (regel 155, 162, 168, 174, 178, 186 en 190) moeten een word of double wordt eerst omgezet worden in afzonderlijke bytes. Dit word gedaan met de functies "word\_to\_2byte" en "dword\_to\_4byte". In regel 196 wordt gekeken of de lengte van de ontvangen data langer is dan nul, wanneer dit het geval is zal de ontvangen state byte terug gestuurd worden (dit is 250 of 255). Wanneer de lengte van de ontvangen data echter niet hoger is dan nul en "sRetOpenServer.functionresult" lager is dan nul zal aangevraagd worden om de verbinding te verbreken.


	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 10 van 40

```

129 // check on received data // alive test from client
130 FOR i := 0 TO (MAX_CONNECTIONS - 1) DO
131     IF connection_id[i].connected AND (NOT connection_id[i].req_disconnect) THEN
132
133         // reviewed data
134         sRetTcpReceive := _tcpReceive(
135             connectionId      := connection_id[i].RetTcpOpenServer.connectionId,
136             nextCommand       := IMMEDIATELY,
137             receiveVariable   := aRecvBuffer
138         );
139
140         ul6RecvDataLength := UDINT_TO_UINT(sRetTcpReceive.dataLength);
141         receiveddata := BICBYTEARRAY_TO_ANYTYPE(aRecvBuffer, 0);
142         IOS_D0_Slot2a := receiveddata.IOS_D0_Slot2a;
143         IOS_D0_Slot2b := receiveddata.IOS_D0_Slot2b;
144         IOS_D0_Slot3 := receiveddata.IOS_D0_Slot3;
145         PLC_D0_XI21 := receiveddata.PLC_D0_XI21;
146         DPC_0_Slot1 := receiveddata.DPC_0_Slot1;
147         DPC_0_Slot3_0 := receiveddata.DPC_0_Slot3_0;
148         DPC_0_Slot3_2 := receiveddata.DPC_0_Slot3_2;
149
150         //IO Station
151         Data_Send[1] := IOS_DI_Slot1a;
152         Data_Send[2] := IOS_DI_Slot1b;
153
154         //PLC IO
155         word_to_2byte (wordin := PLC_DI_Word1);
156         Data_Send[3] := word_to_2byte.byte0; // PLC_DI_XI31
157
158         //profibus
159         Data_Send[4] := DPC_I_Slot7_0;
160         Data_Send[5] := DPC_I_Slot7_1;
161         //DWord 9.28 PID 355
162         dword_to_4byte ( dwordIn := DPC_I_Slot9_28);
163         Data_Send[6] := dword_to_4byte.byte0;
164         Data_Send[7] := dword_to_4byte.byte1;
165         Data_Send[8] := dword_to_4byte.byte2;
166         Data_Send[9] := dword_to_4byte.byte3;
167         //DWord 10.0 PID 359
168         dword_to_4byte ( dwordIn := DPC_I_Slot10_0);
169         Data_Send[10] := dword_to_4byte.byte0;
170         Data_Send[11] := dword_to_4byte.byte1;
171         Data_Send[12] := dword_to_4byte.byte2;
172         Data_Send[13] := dword_to_4byte.byte3;
173         //Word 10.4
174         word_to_2byte (wordin := DPC_I_Slot10_4);
175         Data_Send[14] := word_to_2byte.byte0;
176         Data_Send[15] := word_to_2byte.byte1;
177         //DWord 10.6 PID 365
178         dword_to_4byte ( dwordIn := DPC_I_Slot10_6);
179         Data_Send[16] := dword_to_4byte.byte0;
180         Data_Send[17] := dword_to_4byte.byte1;
181         Data_Send[18] := dword_to_4byte.byte2;
182         Data_Send[19] := dword_to_4byte.byte3;
183         Data_Send[20] := DPC_I_Slot10_10;
184         Data_Send[21] := DPC_I_Slot10_11;
185         //Word 10.12
186         word_to_2byte (wordin := DPC_I_Slot10_12);
187         Data_Send[22] := word_to_2byte.byte0;
188         Data_Send[23] := word_to_2byte.byte1;
189         //Word 10.14
190         word_to_2byte (wordin := DPC_I_Slot10_14);
191         Data_Send[24] := word_to_2byte.byte0;
192         Data_Send[25] := word_to_2byte.byte1;
193
194         IF (ul6RecvDataLength > 0) THEN
195
196             IF receiveddata.State = 250 THEN
197                 Data_Send[0] := 250;
198             ELSIF receiveddata.State = 255 THEN
199                 Data_Send[0] := 255;
200             END_IF;
201
202             ELSIF (sRetTcpReceive.functionResult > 0) THEN // busy, empty
203                 ;
204             ELSIF (sRetTcpReceive.functionResult < 0) THEN // error, disconnect
205                 connection_id[i].req_disconnect := TRUE;
206             END_IF;
207             END_IF; // connection_id[i]
208         END_FOR; // i

```

Figuur B4-13 Regel 129 tot 209: Check on received data

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 211 tot 241: Send data (Figuur B4-14)


In voorgaande alinea zijn alle bytes in een array geplaatst, deze wordt in regel 211 gecast naar een bigbytearray "aSendBuffer". Voor elk address wordt wederom gekeken of deze verbonden is en er niet aangevraagd is om de verbinding te verbreken (regel 215). Wanneer door het device gevraagd is om data te verzenden (regel 217) of wanneer het blink signaal hoog is geworden (regel 229) zal de data verzonden worden. Hierna zal ook nog gekeken worden of een error ontstaan is, in dit geval zal aangevraagd worden om de verbinding te verbreken.


```

211      aSendBuffer := ANTYPE_TO_BIGBYTEARRAY (Data_Send, 0);
212
213      // Send data
214      FOR i := 0 TO (MAX_CONNECTIONS - 1) DO
215        IF connection_id[i].connected AND (NOT connection_id[i].req_disconnect) THEN
216
217          IF (connection_id[i].RetTcpSend > 0) THEN
218            connection_id[i].RetTcpSend := _tcpSend(
219              connectionId      := connection_id[i].RetTcpOpenServer.connectionId,
220              nextCommand       := IMMEDIATELY,
221              dataLength        := DINT_TO_UDINT(_sizeof(aSendBuffer)),
222              data              := aSendBuffer
223            );
224
225            IF (connection_id[i].RetTcpSend < 0) THEN // error, disconnect
226              connection_id[i].req_disconnect := TRUE;
227            END_IF;
228
229          ELSIF second_tick.q AND (connection_id[i].RetTcpSend = 0) THEN
230            connection_id[i].RetTcpSend := _tcpSend(
231              connectionId      := connection_id[i].RetTcpOpenServer.connectionId,
232              nextCommand       := IMMEDIATELY,
233              dataLength        := DINT_TO_UDINT(_sizeof(aSendBuffer)),
234              data              := aSendBuffer
235            );
236            IF (connection_id[i].RetTcpSend < 0) THEN // error, disconnect
237              connection_id[i].req_disconnect := TRUE;
238            END_IF;
239          END_IF;
240        END_IF;
241      END_FOR;

```

Figuur B4-14 Regel 211 tot 241: Send data

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 12 van 40

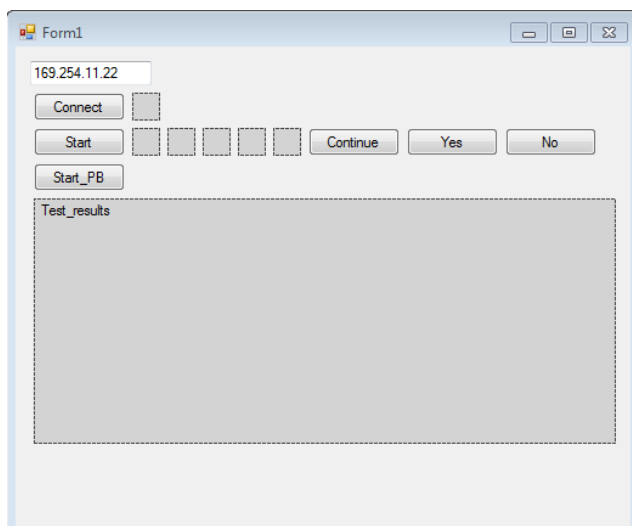
	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

## HMI (C#)


Op de HMI worden alle testen uitgevoerd en daarnaast wordt hier ook de user interface aangestuurd. De taal die op de HMI gebruikt wordt is C#. In de volgende stukken zal eerst ingegaan worden op de user interface en daarna op de code zelf.


## User Interface

In figuur B4-15 is de user interface weergegeven, in de rest van deze alinea wordt uitgewerkt wat alle knoppen en panels doen. Er is de keuze gemaakt om de tests eerst goed te laten functioneren en hierna pas te focussen op het uiterlijk van de user interface. Bovenaan is een textbox met een IP adres te zien, hierin moet het IP adres van de PLC ingesteld staan. Bij default staat deze goed, er wordt echter de vrijheid geboden om deze aan te passen. Hieronder zien we de knop connect, wanneer op connect gedrukt wordt zal de HMI verbinding proberen te maken met de PLC. Wanneer dit lukt zal de tekst veranderen in disconnect, als hier op gedrukt wordt zal de verbinding verbroken worden en zal de tekst weer terug veranderen in connect. Het grijze panel zal van kleur veranderen afhankelijk van de status van het de verbinding en het programma. Grijs betekend dat er nog geen verbinding is gemaakt of dat deze verbinding gestopt is met de disconnect knop. Groen/grijs knipperend betekend dat het programma in rust stand is. Oranje betekend dat het programma aan het testen is en rood betekend dat de verbinding verbroken is. Hieronder zien we twee knoppen "Start" en "Start\_PB", start word gebruikt om de noodstop test uit te voeren en bij start\_PB wordt het profibus deel getest. Deze laatste is vooral gebruikt voor het testen van de software. Rechts van de startknop zijn vijf panels, weergegeven deze worden gebruikt om de resultaten van de noodstop tests weer te geven. Elk panel is gelinkt aan één test behalve het derde panel, deze is aan test 3 en 4 gelinkt. Wanneer deze grijs zijn betekend dit dat de test nog niet is uitgevoerd, geel betekend dat deze test bezig is, groen dat de test geslaagd is en rood als de test mislukt is. Hiernaast staan de knoppen "continue", "Yes" en "No", deze worden gebruikt om vragen aan de gebruiker mee te beantwoorden. Onderin is een textbox weergegeven, hierin komen resultaten en vragen aan de gebruiker te staan.



Figuur B4-15 User Interface

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 13 van 40


	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

## Software

De totale code bestaat uit 1039 regels, om deze reden zal de software in stukken behandeld worden. In figuur B4-16 is de code ingeklapt weergegeven, elk ingeklapte deel zal in de volgende pagina's uitgewerkt worden. Daarnaast zijn in de figuur alle lijnen tussen lijn 20 en 99 weggehaald, dit is gedaan om de afbeelding overzichtelijk te houden. Dit stuk zal ook in de volgende pagina's uitgewerkt worden.

Ten eerste zullen de waardes gedefinieerd worden in de regels 21 tot 98. Hierna zal "Form1" (regel 100) uitgevoerd worden. Het programma zal wachten tot het event "btn\_connect\_Click" (regel 528) heeft plaatsgevonden. Hierna zal in "Initialize\_Connection" (regel 549) de verbinding met de PLC opgezet worden. Daarna zal de stap "Receive\_Messages" (regel 985) het daadwerkelijke programma lopen. Wanneer de verbinding wordt verbroken of als er een error plaatsvindt zal het programma doorgaan naar "Close\_Connection" (regel 972).

De overige deelprogramma's worden door deze deelprogramma's aangeroepen of zijn events die bovenstaande deelprogramma's beïnvloeden


	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 14 van 40

```


1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Net;
11 using System.Net.Sockets;
12 using System.IO;
13 using System.Threading;
14 using System.Media;
15
16 namespace AFST_T_PC_v1
17 {
18
19     public partial class Form1 : Form
20     {
21
22         public Form1()...
23
24         private void Send_Bytearray(int bytenumber, int newbyte)...)
25
26         private bool Bit_Is_True(int bytenumber, int bitnumber)...)
27
28         //test1, test if both Estop inputs are low, print result, go to test2 or end test
29         private void test1()... //private void test1()
30
31         //test2, test if both Estop inputs are low, print result, go to test3 or end test
32         private void test2()... //private void test2()
33
34         //turns the results found in test 3 and 4 into useable information
35         private void cross_table ()... //private void cross_table ()
36
37         //test5, test if both Estop inputs are low, print result, go to test6 or end test
38         private void test5()... //private void test5()
39
40         //test6, enable both Estop outputs, test if all non-Estop inputs are false
41         private void test6()... //private void test6()
42
43         private void testP_HB()...
44
45         private void testP_DV()...
46
47         private void test_profibus_Output(int select_byte, byte select_bit, string outputname)...)
48
49         private void test_profibus_Input(int select_byte, byte select_bit, string inputname)...)
50
51         private void test_profibus_Analog(int select_byte, int length_bit, string analogname, int value1, int value2)...)
52
53         //Save Exit, disconnect
54         public void On_Application_Exit(object sender, EventArgs e)...) //public void On_Application_Exit(object sender, EventArgs e)
55
56         //Connect/Disconnect
57         private void btn_connect_Click(object sender, EventArgs e)...) //private void btn_connect_Click(object sender, EventArgs e)
58
59         //Start connection
60         private void Initialize_Connection()... //private void Initialize_Connection()
61
62         //Receiving Messages
63         private void Receive_Messages()... //private void ReceiveMessages()
64
65         //End connection
66         private void Close_Connection()... //private void CloseConnection()
67
68         private void Show_Message()...
69
70         private void Enable_Continue_Pressed(bool Enabled)...)
71
72         private void btn_start_Click(object sender, EventArgs e)...) //private void btn_start_Click(object sender, EventArgs e)
73
74         private void btn_Continue_Click(object sender, EventArgs e)...)
75
76         private void Form1_Load(object sender, EventArgs e)...)
77
78         private void btn_start_PB_Click(object sender, EventArgs e)...) //private void btn_start_PB_Click(object sender, EventArgs e)
79
80         private void btn_yes_Click(object sender, EventArgs e)...)
81
82         private void btn_no_Click(object sender, EventArgs e)...)
83
84     } // Form1
85 } // AFST_T_PC_v1

```

Figuur B4- 16 Totale programma HMI

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 15 van 40



	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 21 tot 98: Definitie variabele (Figuur B4-17)

In de regels 21 tot 98 worden alle private variabelen en functies geïnitieerd, deze zullen hier onder benoemd worden. Wanneer deze gebruikt worden zal dieper op de functie ingegaan worden.

(regel 21) “nwStream”, (regel 22) “tcp\_Server”, (regel 23) “thrMessaging” en (regel 24) “ipAddr” worden gebruikt bij het instellen van de TCP/IP verbinding en zorgen ervoor dat deze verbinding blijft bestaan.

(regel 25) “tcp\_Connected”, met deze boolean wordt bijgehouden of de opstelling verbonden is via TCP/IP of dat deze verbinding verbroken is of verbroken moet worden

(regel 26) “Continue\_Pressed”, (regel 27) “Yes\_Pressed” en (regel 28) “No\_Pressed” houden bij of de bijbehorende knoppen ingedrukt zijn

(regel 29) “Heartbeat\_Active”, met deze boolean wordt bijgehouden of het heartbeat signaal verstuurd moet worden.

(regel 30) “db\_Out” Dit is een byte array waarin alle bytes die naar de PLC gestuurd worden opgeslagen worden.

(regel 31) “db\_In” Dit is een byte array waarin alle bytes die ontvangen zijn van de PLC opgeslagen worden.


(regel 32) “data\_test3\_4” In dit array worden de afzonderlijke resultaten van test 3 en test 4 opgeslagen, hieruit wordt later een conclusie getrokken.

(regel 33) “test\_state” en (regel 34) “test\_step” worden gebruikt om de voortgang van de tests bij te houden.

(regel 35) “test\_result”, in deze string worden de resultaten en instructies opgeslagen die op het user interface weergegeven worden.

(regel 37) “thrShow\_Message”, (regel 39) “thrEnable\_Continue” en (regel 41) “thrClose\_Connection” worden gebruikt om threats aan te kunnen roepen

De waardes in regel 45 tot 97 worden gebruikt om betekenis aan getallen te geven. In plaats van 8 van 5 wordt in de code nu bit3 van DCP\_I\_Slot7\_1 geschreven. Hierdoor is het veel duidelijker wat er bedoeld wordt en dit maakt de code veel overzichtelijker. Verschillende van de waardes worden niet of niet direct gebruikt, deze staan hier voor de compleetheid en omdat het mogelijk is dat deze later wel gebruikt gaan worden.


	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 16 van 40

```

21     private NetworkStream nwStream;
22     private TcpClient tcp_Server;
23     private Thread thrMessaging;
24     private IPAddress ipAddr;
25     private bool tcp_Connected;
26     private bool Continue_Pressed;
27     private bool Yes_Pressed;
28     private bool No_Pressed;
29     private bool Heartbeat_Active = true;
30     private byte[] db_Out = new byte[8];
31     private byte[] db_In = new byte[26];
32     private bool[] data_test3_4 = new bool[4];
33     private string test_state = "Idle";
34     private string test_step = "1";
35     private string test_result = "no testresult";
36     private delegate void dShowMessage();
37     private dShowMessage thrShow_Message;
38     private delegate void dEnableContinue(bool Enabled);
39     private dEnableContinue thrEnable_Continue;
40     private delegate void dCloseConnection();
41     private dCloseConnection thrClose_Connection;
42
43
44     //bit values
45     private byte no_bits = 0;
46     private byte bit0 = 1;
47     private byte bit1 = 2;
48     private byte bit2 = 4;           //currently not used
49     private byte bit3 = 8;           //currently not used
50     private byte bit4 = 16;
51     private byte bit5 = 32;          //currently not used
52     private byte bit6 = 64;
53     private byte bit7 = 128;
54     private byte all_bits = 255;
55
56     //db_In & db_Out, meaning byte position
57     private int Status = 0;
58
59     //db_In meaning byte position
60     private int IOS_DI_Slot1a = 1;
61     private int IOS_DI_Slot1b = 2;
62     //private int PLC_DI_X131 = 3;      //currently not used
63     private int DPC_I_Slot7_0 = 4;      //bit 0-7 used
64     private int DPC_I_Slot7_1 = 5;      //bit 0-4 used
65     private int DPC_I_Slot9_28 = 6;     //RUNNING REEL REMAINING LENGTH
66     //private int DPC_I_Slot9_29 = 7;
67     //private int DPC_I_Slot9_30 = 8;
68     //private int DPC_I_Slot9_31 = 9;
69     private int DPC_I_Slot10_0 = 10;    //SPLICE DIAMETER ACTUAL VALUE
70     //private int DPC_I_Slot10_1 = 11;
71     //private int DPC_I_Slot10_2 = 12;
72     //private int DPC_I_Slot10_3 = 13;
73     private int DPC_I_Slot10_4 = 14;    //WEB TENSION ACTUAL VALUE
74     //private int DPC_I_Slot10_5 = 15;
75     private int DPC_I_Slot10_6 = 16;    //RUNNING REEL DIAMETER
76     //private int DPC_I_Slot10_7 = 17;
77     //private int DPC_I_Slot10_8 = 18;
78     //private int DPC_I_Slot10_9 = 19;
79     private int DPC_I_Slot10_10 = 20;   //RUNNING REEL POSITION
80     private int DPC_I_Slot10_11 = 21;   //TIME TO SPLICE SELECTOR
81     private int DPC_I_Slot10_12 = 22;   //TIME TO SPLICE SECONDS
82     //private int DPC_I_Slot10_13 = 23;
83     private int DPC_I_Slot10_14 = 24;   //TIME TO SPLICE MINUTES
84     //private int DPC_I_Slot10_15 = 25;
85
86     //db_Out meaning byte position
87     //private int IOS_DO_Slot2a = 1;      //currently not used
88     private int IOS_DO_Slot2b = 2;
89     //private int IOS_DO_Slot3 = 3;        //currently not used
90     //private int PLC_DO_X121 = 4;        //currently not used
91     //private int DPC_O_Slot1 = 5;        //bit 0 used, currently not used
92     private int DPC_O_Slot3_0 = 6;       //bit 0-6 used
93     private int DPC_O_Slot3_2 = 7;       //bit 0 used
94
95     //meaning values Status bit
96     private int no_fault = 255;
97     private int new_data = 250;

```

Figuur B4-17 Regel 21 tot 98: Definitie variabele

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 100 tot 108: Form1 (Figuur B4-18)

In regel 102 wordt een eventhandler gestart waardoor het lopende programma onderbroken kan worden door events. In de regels 103 tot 104 worden threats aangeroepen zodat deze later in het programma gebruikt kunnen worden door andere threats. In regel 106 wordt het userinterface geïnitialiseerd, hierna zal deze gebruikt kunnen worden door de gebruiker. Op regel 107 wordt de tekst in de textbox geleegd zodat begonnen wordt met een lege textbox.

```

100 public Form1()
101 {
102     Application.ApplicationExit += new EventHandler(On_Application_Exit);
103     thrShow_Message = new dShowMessage(Show_Message);
104     thrEnable_Continue = new dEnableContinue(Enable_Continue_Pressed);
105     thrClose_Connection = new dCloseConnection(Close_Connection);
106     InitializeComponent();
107     l_test_result.Text = "";
108 }

```

Figuur B4-18 Regel 100 tot 108: Form1

### Regel 110 tot 115: Send\_Bytearray (Figuur B4-19)

Dit programma vraagt een bytenummer van het output array en de nieuwe byte die op deze plek moet komen. Deze byte wordt gewijzigd (regel 112) waarna het complete bytearray klaar wordt gezet in een buffer (regel 113) en hierna wordt de buffer leeg gemaakt en verstuurd (regel 114). Dit deelprogramma zal gebruikt worden wanneer nieuwe data vanuit de HMI naar de PLC gestuurd wordt.

```

110 private void Send_Bytearray(int bytenumber, int newbyte)
111 {
112     db_Out[bytenumber] = Convert.ToByte(newbyte);
113     nwStream.Write(db_Out, 0, db_Out.Length);
114     nwStream.Flush();
115 }

```

Figuur B4-19 Regel 110 tot 115: Send\_Bytearray

### Regel 117 tot 120: Bit\_Is\_True (Figuur B4-20)


Dit is een korte code maar deze wordt voornamelijk gebruikt in if state selectors, deze werden hierdoor erg onduidelijk. Om deze reden is hier een deelprogramma van gemaakt. Het deelprogramma vergelijkt de waarde van een byte in het array van ontvangen bytes met een bit waarde, wanneer de geselecteerde bit true is zal true uitgegeven worden door dit deelprogramma, anders zal false uitgegeven worden.


```

117 private bool Bit_Is_True(int bytenumber, int bitnumber)
118 {
119     return ((db_In[bytenumber] & bitnumber) != 0);
120 }

```

Figuur B4-20 Regel 117 tot 120: Bit\_Is\_True

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 18 van 40

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
---	-------------------------------	---

### Regel 122 tot 152: Test1 (Figuur B4-21)


In test 1 worden twee bits gecontroleerd (regel 125) afhankelijk hiervan zal één van de vier mogelijke states gekozen worden. Deze bestaan elk uit drie regels, de eerste regel verandert de kleur van het eerste testpanel. Wanneer het resultaat voldoet aan de verwachtingen zal dit panel groen worden, anders wordt deze rood. In de tweede regel wordt het test resultaat geschreven, dit is succesvol of een error. Deze errors hebben een uniek nummer, geven aan welk foute resultaat geconstateerd is en wat hier waarschijnlijk de oorzaak van is. In de derde regel wordt de volgende state geselecteerd, bij een fout zal naar de “end\_test” state gegaan worden en wanneer de test goed gaat zal naar test 2 gegaan worden. Na de if states staan nog drie regels. Op regel 149 wordt de threat gestart waarmee de tekst op de user interface aangepast wordt. Hierna staat in regel 150 een regel voor het testen van de software, deze is in de daadwerkelijke software verwijderd. Deze regel zorgt ervoor dat het programma na een fout alsnog naar de volgende test gaat. In regel 151 wordt het programma “Send\_Byte” aangeroepen hiermee zal de data vanuit de HMI naar de PLC gestuurd worden.


```

122 //test1, test if both Estop inputs are low, print result, go to test2 or end test
123 private void test1()
124 {
125     if (Bit_Is_True(IOS_DI_Slot1a, bit0) && Bit_Is_True(IOS_DI_Slot1a, bit1))
126     {
127         panel_test1.BackColor = System.Drawing.Color.Red;
128         test_result = "test1: (error101) both Estop inputs are high, shortcircuit or connected to power source expected";
129         test_state = "end_test";
130     }
131     else if (Bit_Is_True(IOS_DI_Slot1a, bit0))
132     {
133         panel_test1.BackColor = System.Drawing.Color.Red;
134         test_result = "test1: (error102) input Estop1 is high, shortcircuit or connected to power source expected";
135         test_state = "end_test";
136     }
137     else if (Bit_Is_True(IOS_DI_Slot1a, bit1))
138     {
139         panel_test1.BackColor = System.Drawing.Color.Red;
140         test_result = "test1: (error103) input Estop2 is high, shortcircuit or connected to power source expected";
141         test_state = "end_test";
142     }
143     else
144     {
145         panel_test1.BackColor = System.Drawing.Color.LightGreen;
146         test_result = "test1: succesfull";
147         test_state = "test2_active";
148     }
149     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message);}
150     test_state = "test2_active"; //for testing
151     Send_Bytearray(Status, no_fault);
152 } //private void test1()

```

Figuur B4-21 Regel 122 tot 152: Test1

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 19 van 40

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 154 tot 184: Test2 (Figuur B4-22)


Test 2 functioneert exact hetzelfde als test 1, wel worden andere waarden gebruikt. In plaats van het eerste panel wordt tweede panel gebruikt om aan te geven of de test goed gegaan is. Daarnaast zijn de error en succes berichten anders en is de volgende state test 3.

```

154 //test2, test if both Estop inputs are low, print result, go to test3 or end test
155 private void test2()
156 {
157     if (Bit_Is_True(IOS_DI_Slot1a, bit0) && Bit_Is_True(IOS_DI_Slot1a, bit1))
158     {
159         panel_test2.BackColor = System.Drawing.Color.Red;
160         test_result += "\ntest2: (error201) both Estop inputs are high, shortcircuit to other output(s) expected";
161         test_state = "end_test";
162     }
163     else if (Bit_Is_True(IOS_DI_Slot1a, bit0))
164     {
165         panel_test2.BackColor = System.Drawing.Color.Red;
166         test_result += "\ntest2: (error202) input Estop1 is high, shortcircuit to another output expected";
167         test_state = "end_test";
168     }
169     else if (Bit_Is_True(IOS_DI_Slot1a, bit1))
170     {
171         panel_test2.BackColor = System.Drawing.Color.Red;
172         test_result += "\ntest2: (error203) input Estop2 is high, shortcircuit to another output expected";
173         test_state = "end_test";
174     }
175     else
176     {
177         panel_test2.BackColor = System.Drawing.Color.LightGreen;
178         test_result += "\ntest2: succesfull";
179         test_state = "test3_active";
180     }
181     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
182     test_state = "test3_active"; //for testing
183     Send_Bytearray(Status, no_fault);
184 } //private void test2()

```

Figuur B4-22 Regel 154 tot 184: Test2

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 20 van 40

### Regel 186 tot 241: Cross\_table (Figuur B4-23)


Uit test 3 en 4 komen vier warden, in deze cross table wordt uit deze warden één conclusie getrokken. Op regel 190 tot 193 worden de vier waarden gecombineerd in één waarde. Hierna wordt in regel 195 gekeken of de gewenste resultaten behaald zijn. Als dit het geval is zal het derde panel groen gemaakt worden en zal na de cross table naar de volgende state gegaan worden. Anders zal dit panel rood worden en zal naar de state “end\_test” gegaan worden. Als laatste wordt in regel 206 tot 239 de bijbehorende string gezocht, dit is of succes of een error met geconstateerde fout en waarschijnlijke oorzaak. Deze string zal toegevoegd worden aan de tekst op het user interface.

```

186 //turns the results found in test 3 and 4 into useable information
187 private void cross_table ()
188 {
189     int casenumber = 0;
190     if (data_test3_4[0]) { casenumber = 8; };
191     if (data_test3_4[1]) { casenumber = casenumber + 4; };
192     if (data_test3_4[2]) { casenumber = casenumber + 2; };
193     if (data_test3_4[3]) { casenumber = casenumber + 1; };
194
195     if (casenumber == 9)
196     {
197         panel_test3_4.BackColor = System.Drawing.Color.LightGreen;
198         test_state = "Activate_Estop";
199     }
200     else
201     {
202         panel_test3_4.BackColor = System.Drawing.Color.Red;
203         test_state = "end_test";
204     };
205
206     switch(casenumber)
207     {
208         //FFFF
209         case 0: test_result += "\ntest3 and 4: (error301) break in cable 1 and 2 expected \nother possibilities: Estop pressed or malfunctioning power source"; break;
210         //FFFF
211         case 1: test_result += "\ntest3 and 4: (error302) break in cable 1 expected"; break;
212         //FFTF
213         case 2: test_result += "\ntest3 and 4: (error303) break in cable 1 and crossed wires expected"; break;
214         //FFTF
215         case 3: test_result += "\ntest3 and 4: (error304) break in cable 1 and shortcircuit after break expected"; break;
216         //FTTF
217         case 4: test_result += "\ntest3 and 4: (error305) break in cable 2 and crossed wires expected"; break;
218         //FTTF
219         case 5: test_result += "\ntest3 and 4: (error306) break and shortcircuit before break expected, possibly crossed wires"; break;
220         //FTTF
221         case 6: test_result += "\ntest3 and 4: (error307) crossed wires expected"; break;
222         //FTTF
223         case 7: test_result += "\ntest3 and 4: (error308) this error should not be possible"; break;
224         //TFFF
225         case 8: test_result += "\ntest3 and 4: (error309) break in cable 2 expected"; break;
226         //TFFF-NO ERROR
227         case 9: test_result += "\ntest3 and 4: succesfull"; break;
228         //TFTF
229         case 10: test_result += "\ntest3 and 4: (error310) break and shortcircuit before break expected"; break;
230         //TFTF
231         case 11: test_result += "\ntest3 and 4: (error311) this error should not be possible"; break;
232         //TFTF
233         case 12: test_result += "\ntest3 and 4: (error312) break in cable 2 and shortcircuit after break expected"; break;
234         //TFTF
235         case 13: test_result += "\ntest3 and 4: (error313) this error should not be possible"; break;
236         //TFTF
237         case 14: test_result += "\ntest3 and 4: (error314) this error should not be possible"; break;
238         //TTTT
239         case 15: test_result += "\ntest3 and 4: (error315) shortcircuit expected"; break;
240     } //switch(casenumber)
241 } //private void cross_table ()

```

Figuur B4-23 Regel 186 tot 241: Cross\_table

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 243 tot 279: Test5 (Figuur B4-24)


Aan het begin van test 5 wordt ervoor gezorgd dat de continue knop niet meer ingedrukt kan worden (regel 248). Hierna wordt “Continue\_Pressed” false gemaakt, dit wordt gedaan omdat op de continue knop gedrukt moet worden om in dit deelprogramma te komen. Hierna functioneert de test net als test 1 en 2. Er wordt naar twee bits gekeken (regel 250) waarna één van de vier mogelijke states geselecteerd wordt. In elke state wordt een panel gekleurd, een string op het user interface geschreven en de volgende state geselecteerd. Hierna wordt de threat voor het schrijven op het user interface geactiveerd en wordt een bericht naar de PLC gestuurd. In deze test is echter de regel 275 uit de code gehaald, voor het testen kan deze stap terug gezet worden waardoor altijd naar de volgende deelttest gegaan zal worden.


```

243 //test5, test if both Estop inputs are low, print result, go to test6 or end test
244 private void test5()
245 {
246     if (db_In[Status] == new_data)
247     {
248         if (this.thrEnable_Continue != null) { this.Invoke(this.thrEnable_Continue, new Object[] { false }); }
249         Continue_Pressed = false;
250         if (Bit_Is_True(IOS_DI_Slot1a, bit0) && Bit_Is_True(IOS_DI_Slot1a, bit1))
251         {
252             panel_test5.BackColor = System.Drawing.Color.Red;
253             test_result += "\ntest5: (error501) both Estop inputs are high, malfunctioning Estop expected or Estop not pressed";
254             test_state = "end_test";
255         }
256         else if (Bit_Is_True(IOS_DI_Slot1a, bit0))
257         {
258             panel_test5.BackColor = System.Drawing.Color.Red;
259             test_result += "\ntest5: (error502) input Estop1 is high, malfunctioning Estop expected";
260             test_state = "end_test";
261         }
262         else if (Bit_Is_True(IOS_DI_Slot1a, bit1))
263         {
264             panel_test5.BackColor = System.Drawing.Color.Red;
265             test_result += "\ntest5: (error503) input Estop2 is high, malfunctioning Estop expected";
266             test_state = "end_test";
267         }
268         else
269         {
270             panel_test5.BackColor = System.Drawing.Color.LightGreen;
271             test_result += "\ntest5: succesfull";
272             test_state = "Deactivate_Estop";
273         };
274         if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
275         //test_state = "test6_active"; //for testing
276         Send_Bytearray(Status, no_fault);
277     }
278     else { db_Out[IOS_DO_Slot2b] = bit4; Send_Bytearray(Status, new_data); };
279 } //private void test5()

```

Figuur B4-24 Regel 243 tot 279: Test5

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 22 van 40

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 281 tot 303: Test6 (Figuur B4-25)


Bij test 6 wordt in de regels 286 en 287 wederom gezorgd dat de knop continue niet meer ingedrukt kan worden en dat de boolean “Continue\_Pressed” false wordt. Hierna wordt gekeken of één of meer inputs hoog zijn (regel 286), afhankelijk hiervan wordt het laatste panel gekleurd en een string met succes of error naar het voorscherm geschreven. In regel 298 wordt de threat voor het updaten van de tekst op het voorscherm aangeroepen. Aangezien dit de laatste test van de noodstop test is zal na deze test altijd naar de “end\_test” state gegaan worden. Als laatste wordt wederom een bericht naar de PLC gestuurd.

```

281 //test6, enable both Estop outputs, test if all non-Estop inputs are false
282 private void test6()
283 {
284     if (db_In[Status] == new_data)
285     {
286         if (this.thrEnable_Continue != null) { this.Invoke(this.thrEnable_Continue, new Object[] { false }); }
287         Continue_Pressed = false;
288         if (Bit_Is_True(IOS_DI_Slot1a, all_bits))
289         {
290             panel_test6.BackColor = System.Drawing.Color.Red;
291             test_result += "\ntest6: a non-Estop input is high, shortcircuit to Estop output expected";
292         }
293         else
294         {
295             panel_test6.BackColor = System.Drawing.Color.LightGreen;
296             test_result += "\ntest6: succesfull";
297         }
298         if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
299         test_state = "end_test";
300         Send_Bytearray(Status, no_fault);
301     }
302     else { db_Out[IOS_DO_Slot2b] = (byte)(bit6 + bit7); Send_Bytearray(Status, new_data); }
303 } //private void test6()

```

Figuur B4-25 Regel 281 tot 303: Test6

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 23 van 40



### Regel 305 tot 351: TestP\_HB (Figuur B4-26)


De eerste deeltest van de Profibus tests is het testen van het heartbeat signaal. Hierbij wordt in regel 307 de heartbeat uitgeschakeld, deze zou er voor zorgen dat de test niet goed functioneert. Doordat er een delay in de reactie van de te testen machine zit zou er eigenlijk een delay in de state overgangen moeten zitten. Dit is momenteel niet het geval en als replacement zijn hier extra steps toegepast (regel 315 en 327) om dit delay op te vangen, dit zal echter nog aangepast moeten worden. In step 1 (regel 308) wordt het heartbeat signaal laag gemaakt en wordt de data naar de PLC gestuurd. Er zijn twee step 2s, één voor wanneer de heartbeat response hoog is en één voor wanneer deze laag is. Wanneer deze hoog is zal het heartbeat signaal hoog worden en naar step 3 gegaan worden, wanneer deze laag is zal naar de state "end\_test" gegaan worden. Er zijn ook twee step 3s, één voor wanneer de heartbeat response hoog is en één voor wanneer deze laag is. Wanneer deze laag is zal de test geslaagd zijn, de step zal terug op 1 gezet worden, de volgende test zal geselecteerd worden en het heartbeat signaal wordt weer geactiveerd. Wanneer de heartbeat response hoog is zal naar de state "end\_test" gegaan worden.

```

305 private void testP_HB()
306 {
307     Heartbeat_Active = false;
308     if (test_step == "1")
309     {
310         db_Out[DPC_O_Slot3_0] = no_bits;
311         Send_Bytestream(Status, new_data);
312         //System.Threading.Thread.Sleep(100);
313         test_step = "1.5";
314     }
315     else if (test_step == "1.5")
316     {
317         test_step = "2";
318         //change to wait 500 ms
319     }
320     else if (test_step == "2" && Bit_Is_True(DPC_I_Slot7_0, bit0))
321     {
322         db_Out[DPC_O_Slot3_0] = bit0;
323         Send_Bytestream(Status, new_data);
324         //System.Threading.Thread.Sleep(100);
325         test_step = "2.5";
326     }
327     else if (test_step == "2.5")
328     {
329         test_step = "3";
330         //change to wait 500 ms
331     }
332     else if (test_step == "2" && !(Bit_Is_True(DPC_I_Slot7_0, bit0)))
333     {
334         //error
335         panel_test1.BackColor = System.Drawing.Color.Red;
336         test_state = "end_test";
337     }
338     else if (test_step == "3" && !(Bit_Is_True(DPC_I_Slot7_0, bit0)))
339     {
340         panel_test1.BackColor = System.Drawing.Color.Aquamarine;
341         test_state = "testP_OV_active";
342         test_step = "1";
343         Heartbeat_Active = true;
344     }
345     else if (test_step == "3" && Bit_Is_True(DPC_I_Slot7_0, bit0))
346     {
347         //error
348         panel_test1.BackColor = System.Drawing.Color.Red;
349         test_state = "end_test";
350     }
351 }

```

Figuur B4-26 Regel 305 tot 351: TestP\_HB

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 353 tot 394: TestP\_DV (Figuur B4-27)


Bij deze test zal de webtool gebruikt moeten worden om na te gaan of de waarde "IODataValid" hoog is. Wanneer dit niet het geval is zal een error gegenereerd worden en zal de test beëindigd worden. Wanneer deze waarde wel hoog is zal gevraagd worden om één DIL switch om te schakelen en dit nogmaals te controleren, wanneer de waarde gelijk blijft zal een error verstuurd worden. Wanneer de error veranderd zal gevraagd worden om de switch weer terug te zetten en dit voor een laatste keer te controleren, hierna zal naar de volgende test gegaan worden. Aan het eind van dit programma (regel 391) wordt de threat aangeroepen om de tekst naar de user interface te schrijven. Daarnaast worden ook de waardes van de knoppen yes en no false gemaakt (regel 392 en 393) om te voorkomen dat deze hoog blijven tussen de stappen.

```

353 private void testP_DV()
354 {
355     if (test_step == "1")
356     {
357         test_result = "Is PRE_PST_GEN_DI_IODataValid high? (yes/no)";
358         test_step = "2";
359     }
360     else if (test_step == "2" && No_Pressed)
361     {
362         test_result += "\nError: waarschijnlijk DIL switch"; /**
363         test_state = "end_test";
364     }
365     else if (test_step == "2" && Yes_Pressed)
366     {
367         test_result += "\nSwitch DIL DP1 DIA to off.\nIs PRE_PST_GEN_DI_IODataValid high? (yes/no)";
368         test_step = "3";
369     }
370     else if (test_step == "3" && No_Pressed)
371     {
372         test_result += "\nSwitch DIL DP1 DIA to on.\nIs PRE_PST_GEN_DI_IODataValid high? (yes/no)";
373         test_step = "4";
374     }
375     else if (test_step == "3" && Yes_Pressed)
376     {
377         test_result += "\nError: waarschijnlijk werkt switch niet correct"; /**
378         test_state = "end_test";
379     }
380     else if (test_step == "4" && No_Pressed)
381     {
382         test_result += "\nError: waarschijnlijk werkt switch niet correct"; /**
383         test_state = "end_test";
384     }
385     else if (test_step == "4" && Yes_Pressed)
386     {
387         panel_test2.BackColor = System.Drawing.Color.Aquamarine;
388         test_step = "1";
389         test_state = "testPO_1_active";
390     }
391     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
392     Yes_Pressed = false;
393     No_Pressed = false;
394 }

```

Figuur B4-27 Regel 353 tot 394: TestP\_DV

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 25 van 40

### Regel 396 tot 431: test\_profibus\_Output (Figuur B4-28)


De verschillende outputs worden met dit programma getest. De outputs worden hoog gemaakt (regel 400) waarna gevraagd wordt of deze hoog geworden is op de webtool. Wanneer dit niet het geval is zal een error gegenereerd worden en zal de test beëindigd worden, wanneer dit wel het geval is zal deze waarde weer laag gemaakt worden en zal gevraagd worden of deze ook op de webtool als laag aangegeven wordt. Wanneer dit niet het geval is zal een error gegenereerd worden en wanneer dit wel het geval is zal naar de volgende test gegaan worden. Aan het eind van dit programma (regel 427) wordt de threat aangeroepen om de tekst naar de user interface te schrijven. Daarnaast worden ook de waardes van de knoppen yes en no false gemaakt (regel 428 en 429) om te voorkomen dat deze hoog blijven tussen de stappen.

```

396 private void test_profibus_Output(int select_byte, byte select_bit, string outputname)
397 {
398     if (test_step == "1")
399     {
400         db_Out[select_byte] = select_bit;
401         Send_Bytearray(Status, new_data);
402         test_result = "Is " + outputname + " high? (yes/no)";
403         test_step = "2";
404     }
405     else if (test_step == "2" && Yes_Pressed)
406     {
407         db_Out[select_byte] = no_bits;
408         Send_Bytearray(Status, new_data);
409         test_result += "\nIs " + outputname + " high? (yes/no)";
410         test_step = "3";
411     }
412     else if (test_step == "2" && No_Pressed)
413     {
414         test_result += "\nError: I1.2.1"; /**
415         test_state = "end_test";
416     }
417     else if (test_step == "3" && Yes_Pressed)
418     {
419         test_result += "\nError: I1.3.1"; /**
420         test_state = "end_test";
421     }
422     else if (test_step == "3" && No_Pressed)
423     {
424         test_step = "1";
425         test_state = "next";
426     }
427     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
428     Yes_Pressed = false;
429     No_Pressed = false;
430 }
431

```

Figuur B4-28 Regel 396 tot 431: test\_profibus\_Output

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---


### Regel 433 tot 463: test\_profibus\_Input (Figuur B4-29)

De verschillende outputs worden via dit programma getest. Er wordt gevraagd om de input op de webtool hoog te maken. Dit zal door het testsysteem gedetecteerd worden, wanneer dit niet het geval is zal op de knop yes gedrukt moeten worden. Wanneer dit gedetecteerd word zal gevraagd worden om de waarde laag te maken, wanneer op yes gedrukt wordt zal een error gegenereerd moeten worden. Wederom zal nu automatisch gedetecteerd worden of deze waarde laag is of er zal weer op de yes knop gedrukt moeten worden. Wanneer dit gedetecteerd word zal naar de volgende test gegaan worden, wanneer op yes gedrukt wordt zal een error gegenereerd moeten worden. Aan het eind van dit programma (regel 460) wordt de threat aangeroepen om de tekst naar de user interface te schrijven. Daarnaast worden ook de waardes van de knoppen yes en no false gemaakt (regel 461 en 462) om te voorkomen dat deze hoog blijven tussen de stappen.

```

433 private void test_profibus_Input(int select_byte, byte select_bit, string inputname)
434 {
435     if (test_step == "1")
436     {
437         test_result = "Make " + inputname + " true, if no response pres yes";
438         test_step = "2";
439     }
440     else if (test_step == "2" && Bit_Is_True(select_byte, select_bit))
441     {
442         test_result += "\nMake " + inputname + " false, if no response pres yes";
443         test_step = "3";
444     }
445     else if (test_step == "2" && Yes_Pressed)
446     {
447         test_result += "\nError: I1.3.1"; /**
448         test_state = "end_test";
449     }
450     else if (test_step == "3" && !(Bit_Is_True(select_byte, select_bit)))
451     {
452         test_step = "1";
453         test_state = "next";
454     }
455     else if (test_step == "3" && Yes_Pressed)
456     {
457         test_result += "\nError: I1.3.2"; /**
458         test_state = "end_test";
459     }
460     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
461     Yes_Pressed = false;
462     No_Pressed = false;
463 }
```

Figuur B4-29 Regel 433 tot 463: test\_profibus\_Input

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 27 van 40

#### Regel 465 tot 514: test\_profibus\_Analog (Figuur B4-30)


Er zijn drie type analoge signalen die ontvangen worden: byte, word en double wordt. Deze worden in de regels 467 tot 485 omgezet naar een int. Hierna vind de test zelf plaats. Eerst wordt gevraagd om de waarde op de webtool op "value1" in te stellen. Dit wordt automatisch door het testsysteem gedetecteerd, wanneer dit niet het geval is moet op de knop yes gedrukt worden. Wanneer de juiste waarde ontvangen wordt zal gevraagd worden om de waarde op de webtool in te stellen op "value2", wanneer op yes gedrukt wordt zal een error gegenereerd worden en zal de test beëindigd worden. Wanneer de juiste waarde ontvangen wordt zal naar de volgende test gegaan worden, wanneer op yes gedrukt wordt zal een error gegenereerd worden en zal de test beëindigd worden.

```

465 private void test_profibus_Analog(int select_byte, int length_bit, string analogname, int value1, int value2)
466 {
467     int received_value = 0;
468     byte[] received_bytes = new byte[4];
469     Single received_float = 0;
470     if (length_bit == 8) { received_value = db_In[select_byte];}
471     else if (length_bit == 16)
472     {
473         received_bytes[0] = db_In[select_byte];
474         received_bytes[1] = db_In[select_byte + 1];
475         received_value = BitConverter.ToInt32(received_bytes, 0);
476     }
477     else if (length_bit == 32)
478     {
479         received_bytes[0] = db_In[select_byte];
480         received_bytes[1] = db_In[select_byte + 1];
481         received_bytes[2] = db_In[select_byte + 2];
482         received_bytes[3] = db_In[select_byte + 3];
483         received_float = BitConverter.ToSingle(received_bytes, 0);
484         received_value = (int)received_float;
485     }
486     if (test_step == "1")
487     {
488         test_result += "Change " + analogname + " to " + value1 + ", if no response pres yes";
489         test_step = "2";
490     }
491     else if (test_step == "2" && received_value == value1)
492     {
493         test_result += "\nChange " + analogname + " to " + value2 + ", if no response pres yes";
494         test_step = "3";
495     }
496     else if (test_step == "2" && Yes_Pressed)
497     {
498         test_result += "\nError: I1.3.1"; /**
499         test_state = "end_test";
500     }
501     else if (test_step == "3" && received_value == value2)
502     {
503         test_step = "1";
504         test_state = "next";
505     }
506     else if (test_step == "3" && Yes_Pressed)
507     {
508         test_result += "\nError: I1.3.2"; /**
509         test_state = "end_test";
510     }
511     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
512     Yes_Pressed = false;
513     No_Pressed = false;
514 }

```

Figuur B4-30 Regel 465 tot 514: test\_profibus\_Analog

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

### Regel 516 tot 525: On\_Application\_Exit (Figuur B4-31)

Als de applicatie afgesloten wordt en het programma nog verbonden is via TCP/IP zal deze verbinding verbroken moeten worden, dit wordt geregeld in dit deelprogramma.

```

516         //Save Exit, disconnect
517     public void On_Application_Exit(object sender, EventArgs e)
518     {
519         if (tcp_Connected == true)
520         {
521             tcp_Connected = false;
522             nwStream.Close();
523             tcp_Server.Close();
524         }
525     } //public void On_Application_Exit(object sender, EventArgs e)

```

Figuur B4-31 Regel 516 tot 525: On\_Application\_Exit

### Regel 527 tot 546: btn\_connect\_Click (Figuur B4-32)


Dit event wordt uitgevoerd wanneer op de knop connect gedrukt wordt. De kleuren van de testpanels en de tekst in de textbox worden terug gezet op de startwaarde. Hierna wordt de verbinding gestart. Mocht er al een verbinding zijn dan zal doormiddel van deze knop de verbinding verbroken worden.

```

527         //Connect/Disconnect
528     private void btn_connect_Click(object sender, EventArgs e)
529     {
530         panel_connect.BackColor = System.Drawing.Color.LightGray;
531         panel_test1.BackColor = System.Drawing.Color.LightGray;
532         panel_test2.BackColor = System.Drawing.Color.LightGray;
533         panel_test3_4.BackColor = System.Drawing.Color.LightGray;
534         panel_test5.BackColor = System.Drawing.Color.LightGray;
535         panel_test6.BackColor = System.Drawing.Color.LightGray;
536         if (tcp_Connected == false)
537         {
538             test_result = "";
539             if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
540             Initialize_Connection();
541         }
542         else
543         {
544             Close_Connection();
545         }
546     } //private void btn_connect_Click(object sender, EventArgs e)

```

Figuur B4-32 Regel 527 tot 546: btn\_connect\_Click

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 29 van 40

#### Regel 548 tot 576: Initialize\_Connection (Figuur B4-33)

In dit programma wordt geprobeerd een TCP/IP verbinding op te zetten. Wanneer dit lukt zal de knop connect veranderen in de disconnect knop. Ook zal een eerste bericht naar de PLC gestuurd worden wat er voor zorgt dat er continu berichten tussen de HMI en PLC verstuurd worden in reactie op elkaar. Als laatste wordt de thread “Receive\_Messages” gestart. Wanneer geen verbinding gemaakt kan worden zal een error op het user interface weergegeven worden.

```


548 //Start connection
549 private void Initialize_Connection()
550 {
551     ipAddr = IPAddress.Parse(tb_IP.Text);
552     tcp_Server = new TcpClient();
553     try
554     {
555         tcp_Server.Connect(ipAddr, 1000);
556         btn_connect.Text = "Disconnect";
557         tcp_Connected = true;
558         tb_IP.Enabled = false;
559         btn_start.Enabled = true;
560
561         nwStream = tcp_Server.GetStream();
562         nwStream.ReadTimeout = 500;
563         nwStream.Flush();
564
565         Send_ByteArray(Status, no_fault);
566
567         thrMessaging = new Thread(new ThreadStart(Receive_Messages));
568         thrMessaging.Start();
569     }
570     catch
571     {
572         panel_connect.BackColor = System.Drawing.Color.Red;
573         test_result = "ERROR: Unable to connect";
574         if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
575     }
576 } //private void Initialize_Connection()

```

Figuur B4-33 Regel 548 tot 576: Initialize\_Connection

#### Regel 578 tot 968: Receive\_Messages (Figuur B4-34)

Op de volgende pagina's is de code van dit deel weergegeven, dit is de kern van dit programma. Met de regels 581 en 583 wordt de verbinding uitgelezen tot het eerste bericht van de PLC ontvangen wordt. Wanneer dit het geval is zal doorgedaan worden naar de while loop. Hierin wordt als eerste een bericht uitgelezen, mocht er binnen een halve seconden geen nieuw bericht ontvangen worden dan zal de verbinding als verloren beschouwd worden en zal de verbinding vanuit de HMI verbroken worden (regel 956). Wanneer het heartbeat signaal aan staat deze bij elke doorloop gewijzigd worden (regel 591). Verder bestaat dit programma uit een switch met cases, veel van deze cases komen overeen en deze zullen per groep besproken worden. Wanneer er geen tests uitgevoerd worden zal het systeem in de stand “Idle” (regel 599) staan, deze laat het verbindings panel groen/grijs knipperen. De cases “test1\_active” (regel 606), “test2\_active” (regel 612), “test5\_active” (regel 652) en “test6\_active” (regel 666) roepen enkel het bijbehorende testprogramma aan. In “test3\_active” (regel 618) worden twee ontvangen bits opgeslagen in één data array. Hierna worden in “test4\_active” (regel 630) nog twee waarden aan dit array toegevoegd en zal met behulp van de cross table een conclusie getrokken worden uit deze vier waarden. In de cases “Activate\_Estop” (regel 644) en “Deactivate\_Estop” (regel 658) wordt gevraagd om de noodstop in te drukken al dan niet uit te trekken en hierna op continue te drukken.


	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
---	-------------------------------	---

De Profibus tests zijn op te delen in vier groepen. De eerste groep bestaat uit "testP\_HB\_active" (regel 682) en "testP\_DV\_active" (regel 687), deze cases roepen enkel het bijbehorende programma aan. De tweede groep bestaat uit alle "testPO\_###\_active" cases (regel 692 tot 772), hierin wordt waar nodig de heartbeat uitgeschakeld en daarnaast wordt het programma "test\_profibus\_Output" aangeroepen. Wanneer "test\_state" de waarde next krijgt zal naar de volgende deelttest gegaan worden en de heartbeat zal weer ingeschakeld worden. De derde groep bestaat uit alle "testPI\_###\_active" cases (regel 774 tot 880) hierin wordt enkel het programma "test\_profibus\_Input" aangeroepen. Wanneer "test\_state" de waarde next krijgt zal naar de volgende deelttest gegaan worden. De laatste groep bestaat uit alle "testPA\_###\_active" cases (regel 882 tot 952) hierin wordt enkel het programma "test\_profibus\_Analog" aangeroepen. Wanneer "test\_state" de waarde next krijgt zal naar de volgende deelttest gegaan worden.

```

578 //Receiving Messages
579 private void Receive_Messages()
580 {
581     nwStream.Read(db_In, 0, db_In.Length);
582
583     if (db_In[Status] == no_bits) { return; }
584     while (tcp_Connected)
585     {
586         try
587         {
588             //ConResponse = srReceiver.Read();
589             nwStream.Read(db_In, 0, db_In.Length);
590
591             if (Heartbeat_Active)
592             {
593                 if (Bit_Is_True(DPC_I_Slot7_0, bit0)){ db_Out[DPC_O_Slot3_0] = bit0; }
594                 else if (db_In[Status] == new_data) { db_Out[DPC_O_Slot3_0] = no_bits; }
595             }
596
597             switch (test_state)
598             {
599                 case "Idle":
600                     Send_Bytearray(Status, no_fault);
601                     if (panel_connect.BackColor == Color.LightGreen) { panel_connect.BackColor = System.Drawing.Color.LightGray; }
602                     else { panel_connect.BackColor = System.Drawing.Color.LightGreen; };
603                     Send_Bytearray(Status, new_data);
604                     break;
605
606                     //test1, disable all outputs, measure if all inputs are false
607                 case "test1_active":
608                     if (db_In[Status] == new_data) { test1(); }
609                     else { db_Out[IOS_DO_Slot2b] = bit4; Send_Bytearray(Status, new_data); };
610                     break;
611
612                     //test2, disable the Estop outputs, enable all other outputs, measure if Estop inputs are false
613                 case "test2_active":
614                     if (db_In[Status] == new_data) { test2(); }
615                     else { db_Out[IOS_DO_Slot2b] = bit4; Send_Bytearray(Status, new_data); }; // Send(2, 000) changed to Send(2, 016)
616                     break; // for testing, no other relevant outputs
617
618                     //test3, enable Estop output 1, disable all other outputs, measure and save Estop inputs, continue to test 4
619                 case "test3_active":
620                     if (db_In[Status] == new_data)
621                     {
622                         data_test3_4[0] = (Bit_Is_True(IOS_DI_Slot1b, bit6));
623                         data_test3_4[1] = (Bit_Is_True(IOS_DI_Slot1b, bit7));
624                         test_state = "test4_active";
625                         Send_Bytearray(Status, no_fault);
626                     }
627                     else { db_Out[IOS_DO_Slot2b] = bit6; Send_Bytearray(Status, new_data); }; // for testing only 1 output is aviable,
628                     break; //Send(2, 016) (IOS_DO_Slot2_a4) is used for both

```


	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 31 van 40



```

629
630 //test4, enable Estop output 2, disable all other outputs, measure Estop inputs, use cross table
631 case "test4_active":
632     if (db_In[Status] == new_data)
633     {
634         data_test3_4[2] = (Bit_Is_True(IOS_DI_Slot1b, bit6));
635         data_test3_4[3] = (Bit_Is_True(IOS_DI_Slot1b, bit7));
636         cross_table();
637         if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
638         test_state = "Activate_Estop"; //for testing
639         Send_Bytearray(Status, no_fault);
640     }
641     else { db_Out[IOS_DO_Slot2b] = bit7; Send_Bytearray(Status, new_data); }; // for testing only 1 output is aviable, Send(2, 0
642     break;
643
644 case "Activate_Estop":
645     test_result += "\n\nActivate Estop and press button \"Continue\"\n";
646     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
647     if (this.thrEnable_Continue != null) { this.Invoke(this.thrEnable_Continue, new Object[] { true }); }
648     test_state = "test5_active";
649     Send_Bytearray(Status, no_fault);
650     break;
651
652 //test5, ask to activate Estop button, Enable Estop outputs, measure if Estop inputs are false, repeat for each Estop button
653 case "test5_active":
654     if (Continue_Pressed) { test5(); }
655     else { Send_Bytearray(Status, no_fault); }
656     break;
657
658 case "Deactivate_Estop":
659     test_result += "\n\nDeactivate Estop and press button \"Continue\"\n";
660     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
661     if (this.thrEnable_Continue != null) { this.Invoke(this.thrEnable_Continue, new Object[] { true }); }
662     test_state = "test6_active";
663     Send_Bytearray(Status, no_fault);
664     break;
665
666 //test6, remind to deactivate the Estop buttons, enable Estop outputs, measure if other outputs are false
667 case "test6_active":
668     if (Continue_Pressed) { test6(); }
669     else { Send_Bytearray(Status, no_fault); }
670     break;
671
672 case "end_test":
673     test_result += "\n\nTest completed";
674     if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
675     if (this.thrEnable_Continue != null) { this.Invoke(this.thrEnable_Continue, new Object[] { false }); }
676     Continue_Pressed = false;
677     Send_Bytearray(IOS_DO_Slot2b, no_bits);
678     test_state = "Idle";
679     test_step = "1";
680     break;
681


```


	Teststelsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-----------------------------------	---

```

682     case "testP_HB_active": // heartbeat
683         if (db_In[Status] == new_data) { testP_HB(); }
684         else { Send_Bytestream(Status, new_data); };
685         break;
686
687     case "testP_DV_active": //I/O data valid
688         if (db_In[Status] == new_data) { testP_DV(); }
689         else { Send_Bytestream(Status, new_data); };
690         break;
691
692     case "testPO_1_active": //Clear Error Messages
693         if (db_In[Status] == new_data)
694         {
695             Heartbeat_Active = false;
696             test_profibus_Output(DPC_O_Slot3_0, bit2, "PRE_PST_GEN_DI_ClearFaults");
697             if (test_state == "next")
698             {
699                 test_state = "testPO_2_active";
700                 Heartbeat_Active = true;
701             }
702         }
703         else { Send_Bytestream(Status, new_data); }
704         break;
705
706     case "testPO_2_active": //Press Faulted
707         if (db_In[Status] == new_data)
708         {
709             Heartbeat_Active = false;
710             test_profibus_Output(DPC_O_Slot3_0, bit1, "PRE_PST_GEN_DI_PressFaulted");
711             if (test_state == "next")
712             {
713                 test_state = "testPO_3_active";
714                 Heartbeat_Active = true;
715             }
716         }
717         else { Send_Bytestream(Status, new_data); }
718         break;
719
720     case "testPO_3_active": //Press Quick Stop
721         if (db_In[Status] == new_data)
722         {
723             Heartbeat_Active = false;
724             test_profibus_Output(DPC_O_Slot3_0, bit5, "PRE_PST_GEN_DI_PressQuickStop");
725             if (test_state == "next")
726             {
727                 test_state = "testPO_4_active";
728                 Heartbeat_Active = true;
729             }
730         }
731         else { Send_Bytestream(Status, new_data); }
732         break;

```


	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 33 van 40

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

```

733
734
735     case "testPO_4_active": //Motion Starting
736         if (db_In[Status] == new_data)
737         {
738             Heartbeat_Active = false;
739             test_profibus_Output(DPC_O_Slot3_0, bit4, "PRE_PST_UNW_DI_MotionStarting");
740             if (test_state == "next")
741             {
742                 test_state = "testPO_5_active";
743                 Heartbeat_Active = true;
744             }
745         }
746         else { Send_Bytearray(Status, new_data); }
747         break;
748
749     case "testPO_5_active": //Remote Splice Request
750         if (db_In[Status] == new_data)
751         {
752             test_profibus_Output(DPC_O_Slot3_2, bit0, "PRE_PST_SPL_DI_RemoteSpliceRequest");
753             if (test_state == "next")
754             {
755                 test_state = "testPO_6_active";
756             }
757         }
758         else { Send_Bytearray(Status, new_data); }
759         break;
760
761     case "testPO_6_active": //Web Tension Selected
762         if (db_In[Status] == new_data)
763         {
764             Heartbeat_Active = false;
765             test_profibus_Output(DPC_O_Slot3_0, bit3, "PRE_PST_UNW_DI_WebTensionSelection");
766             if (test_state == "next")
767             {
768                 test_state = "testPI_1_active";
769                 Heartbeat_Active = true;
770             }
771         }
772         else { Send_Bytearray(Status, new_data); }
773         break;
774
775     case "testPI_1_active": //Clear Error Messages
776         if (db_In[Status] == new_data)
777         {
778             test_profibus_Input(DPC_I_Slot7_0, bit7, "PRE_PST_GEN_DO_ClearFaults");
779             if (test_state == "next") { test_state = "testPI_2_active"; }
780         }
781         else { Send_Bytearray(Status, new_data); }
782         break;

```

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 34 van 40

```

783 case "testPI_2_active": //Local Emergency Stop Released
784   if (db_In[Status] == new_data)
785   {
786     test_profibus_Input(DPC_I_Slot7_1, bit0, "PRE_PST_GEN_DO_LocalEmergencyReleased");
787     if (test_state == "next") { test_state = "testPI_3_active"; }
788   }
789   else { Send_Bytearray(Status, new_data); }
790   break;
791
792 case "testPI_3_active": //Normal Stop
793   if (db_In[Status] == new_data)
794   {
795     test_profibus_Input(DPC_I_Slot7_0, bit6, "PRE_PST_GEN_DO_NormalStop");
796     if (test_state == "next") { test_state = "testPI_4_active"; }
797   }
798   else { Send_Bytearray(Status, new_data); }
799   break;
800
801 case "testPI_4_active": //Slice Imminent
802   if (db_In[Status] == new_data)
803   {
804     test_profibus_Input(DPC_I_Slot7_0, bit2, "PRE_PST_SPL_DO_SpliceImminent");
805     if (test_state == "next") { test_state = "testPI_5_active"; }
806   }
807   else { Send_Bytearray(Status, new_data); }
808   break;
809
810 case "testPI_5_active": //Splice Prepared
811   if (db_In[Status] == new_data)
812   {
813     test_profibus_Input(DPC_I_Slot7_1, bit2, "PRE_PST_SPL_DO_SplicePrepared");
814     if (test_state == "next") { test_state = "testPI_6_active"; }
815   }
816   else { Send_Bytearray(Status, new_data); }
817   break;
818
819 case "testPI_6_active": //Web Tension Low Present
820   if (db_In[Status] == new_data)
821   {
822     test_profibus_Input(DPC_I_Slot7_0, bit4, "PRE_PST_UMW_DO_WebTensionLowPresent");
823     if (test_state == "next") { test_state = "testPI_7_active"; }
824   }
825   else { Send_Bytearray(Status, new_data); }
826   break;
827
828 case "testPI_7_active": //Local Guards Closed
829   if (db_In[Status] == new_data)
830   {
831     test_profibus_Input(DPC_I_Slot7_1, bit3, "PRE_PST_GEN_DO_LocalGuardsClosed");
832     if (test_state == "next") { test_state = "testPI_8_active"; }
833   }
834   else { Send_Bytearray(Status, new_data); }
835   break;

```

```

828     case "testPI_7_active": //Local Guards Closed
829         if (db_In[Status] == new_data)
830             {
831                 test_profibus_Input(DPC_I_Slot7_1, bit3, "PRE_PST_GEN_DO_LocalGuardsClosed");
832                 if (test_state == "next") { test_state = "testPI_8_active"; }
833             }
834         else { Send_Bytearray(Status, new_data); }
835         break;
836
837     case "testPI_8_active": //Single Reel Mode Selected
838         if (db_In[Status] == new_data)
839             {
840                 test_profibus_Input(DPC_I_Slot7_1, bit1, "PRE_PST_SPL_DO_SingleReelModeSelected");
841                 if (test_state == "next") { test_state = "testPI_9_active"; }
842             }
843         else { Send_Bytearray(Status, new_data); }
844         break;
845
846     case "testPI_9_active": //Splice Active
847         if (db_In[Status] == new_data)
848             {
849                 test_profibus_Input(DPC_I_Slot7_0, bit3, "PRE_PST_SPL_DO_SpliceActive");
850                 if (test_state == "next") { test_state = "testPI_10_active"; }
851             }
852         else { Send_Bytearray(Status, new_data); }
853         break;
854
855     case "testPI_10_active": //Splice Knife Cut
856         if (db_In[Status] == new_data)
857             {
858                 test_profibus_Input(DPC_I_Slot7_1, bit4, "PRE_PST_SPL_DO_SpliceKnifeOut");
859                 if (test_state == "next") { test_state = "testPI_11_active"; }
860             }
861         else { Send_Bytearray(Status, new_data); }
862         break;
863
864     case "testPI_11_active": //Splicer Failure/Press Fast Stop Request
865         if (db_In[Status] == new_data)
866             {
867                 test_profibus_Input(DPC_I_Slot7_0, bit1, "PRE_PST_GEN_DO_SplicerFailure");
868                 if (test_state == "next") { test_state = "testPI_12_active"; }
869             }
870         else { Send_Bytearray(Status, new_data); }
871         break;
872
873     case "testPI_12_active": //Web Tension High Present
874         if (db_In[Status] == new_data)
875             {
876                 test_profibus_Input(DPC_I_Slot7_0, bit5, "PRE_PST_UNW_DO_WebTensionHighPresent");
877                 if (test_state == "next") { test_state = "testPA_1_active"; }
878             }
879         else { Send_Bytearray(Status, new_data); }
880         break;

```

```

881
882     case "testPA_1_active":
883         if (db_In[Status] == new_data)
884         {
885             test_profibus_Analog(DPC_I_Slot10_10, 8, "RunningReelPosition", 1, 2);
886             if (test_state == "next") { test_state = "testPA_2_active"; }
887         }
888         else { Send_Bytearray(Status, new_data); }
889         break;
890
891     case "testPA_2_active":
892         if (db_In[Status] == new_data)
893         {
894             test_profibus_Analog(DPC_I_Slot10_12, 16, "TimeToSpliceInSec", 1, 2);
895             if (test_state == "next") { test_state = "testPA_3_active"; }
896         }
897         else { Send_Bytearray(Status, new_data); }
898         break;
899
900     case "testPA_3_active":
901         if (db_In[Status] == new_data)
902         {
903             test_profibus_Analog(DPC_I_Slot10_4, 16, "WebTensionHighFeedbackLBS", 1, 2);
904             if (test_state == "next") { test_state = "testPA_4_active"; }
905         }
906         else { Send_Bytearray(Status, new_data); }
907         break;
908
909     case "testPA_4_active":
910         if (db_In[Status] == new_data)
911         {
912             test_profibus_Analog(DPC_I_Slot10_6, 32, "RunningReelDiameterINCH", 1, 2);
913             if (test_state == "next") { test_state = "testPA_5_active"; }
914         }
915         else { Send_Bytearray(Status, new_data); }
916         break;
917
918     case "testPA_5_active":
919         if (db_In[Status] == new_data)
920         {
921             test_profibus_Analog(DPC_I_Slot10_11, 8, "TimeToSpliceSelector", 1, 2);
922             if (test_state == "next") { test_state = "testPA_6_active"; }
923         }
924         else { Send_Bytearray(Status, new_data); }
925         break;
926
927     case "testPA_6_active":
928         if (db_In[Status] == new_data)
929         {
930             test_profibus_Analog(DPC_I_Slot10_14, 16, "TimeToSpliceInMin", 1, 2);
931             if (test_state == "next") { test_state = "testPA_7_active"; }
932         }
933         else { Send_Bytearray(Status, new_data); }
934         break;
935
936     case "testPA_7_active":
937         if (db_In[Status] == new_data)
938         {
939             test_profibus_Analog(DPC_I_Slot9_28, 32, "RunningReelRemainingLengthINCH", 1, 2);
940             if (test_state == "next") { test_state = "testPA_8_active"; }
941         }
942         else { Send_Bytearray(Status, new_data); }
943         break;
944
945     case "testPA_8_active":
946         if (db_In[Status] == new_data)
947         {
948             test_profibus_Analog(DPC_I_Slot10_0, 32, "SpliceDiameterFeedbackINCH", 1, 2);
949             if (test_state == "next") { test_state = "Idle"; }
950         }
951         else { Send_Bytearray(Status, new_data); }
952         break;
953
954     }
955 } //try
956 catch
957 {
958     if (tcp_Connected == true)
959     {
960         panel_connect.BackColor = System.Drawing.Color.Red;
961         Continue_Pressed = false;
962         test_result += "\n\nERROR: Connection lost";
963         if (this.thrShow_Message != null) { this.Invoke(this.thrShow_Message); }
964         if (this.thrClose_Connection != null) { this.Invoke(this.thrClose_Connection); }
965     }
966 } //catch
967 } //while (tcp_Connected)
968 } //private void ReceiveMessages()

```

Figuur B4-34 Regel 578 tot 968: Receive\_Messages

### Regel 972 tot 983: Close\_Connection (Figuur B4-35)

In dit programma zullen alle waarden die te maken hebben met de voortgang van het programma terug gezet worden op het begin positie. Daarnaast wordt in de regels 980 tot 982 de verbinding gesloten.

```

971 //End connection
972 private void Close_Connection()
973 {
974     tb_IP.Enabled = true;
975     btn_start.Enabled = false;
976     btn_Continue.Enabled = false;
977     btn_connect.Text = "Connect";
978     test_state = "Idle";
979
980     tcp_Connected = false;
981     nwStream.Close();
982     tcp_Server.Close();
983 } //private void Close_Connection()

```

Figuur B4-35 Regel 972 tot 983: Close\_Connection

### Regel 985 tot 988: Show\_Message (Figuur B4-36)

Dit programma wordt aangeroepen met een threat en verzekerd dat de tekst naar de userinterface geschreven wordt.

```

985 private void Show_Message()
986 {
987     l_test_result.Text = test_result;
988 }

```

Figuur B4-36 Regel 985 tot 988: Show\_Message

### Regel 990 tot 993: Enable\_Continue\_Pressed (Figuur B4-37)

Dit programma wordt aangeroepen met een threat en verzekerd dat het mogelijk wordt om de knop continue te gebruiken wanneer de input true is of dat deze niet meer gebruikt kan worden wanneer de input false is.

```

990 private void Enable_Continue_Pressed(bool Enabled)
991 {
992     btn_Continue.Enabled = Enabled;
993 }

```

Figuur B4-37 Regel 990 tot 993: Enable\_Continue\_Pressed

### Regel 995 tot 1004: btn\_start\_Click (Figuur B4-38)


Wanneer de knop start ingedrukt wordt zal de kleur van alle panels terug gezet worden op de start positie, de panel die de verbinding aangeeft wordt oranje om aan te geven dat het programma bezig is. Hierna wordt de eerste deelttest aangeroepen.

```

995 private void btn_start_Click(object sender, EventArgs e)
996 {
997     panel_test1.BackColor = System.Drawing.Color.LightGray;
998     panel_test2.BackColor = System.Drawing.Color.LightGray;
999     panel_test3_4.BackColor = System.Drawing.Color.LightGray;
1000     panel_test5.BackColor = System.Drawing.Color.LightGray;
1001     panel_test6.BackColor = System.Drawing.Color.LightGray;
1002     panel_connect.BackColor = System.Drawing.Color.Orange;
1003     test_state = "test1_active";
1004 } //private void btn_start_Click(object sender, EventArgs e)

```

Figuur B4-38 Regel 995 tot 1004: btn\_start\_Click

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

#### Regel 1006 tot 1009: btn\_Continue\_Click (Figuur B4-39)

Dit event zorgt ervoor dat de waarde “Continue\_Pressed” hoog wordt wanneer op de knop continue gedrukt wordt.

```

1006  private void btn_Continue_Click(object sender, EventArgs e)
1007  {
1008      Continue_Pressed = true;
1009  }
```

Figuur B4-39 Regel 1006 tot 1009: btn\_Continue\_Click

#### Regel 1011 tot 1014: Form1\_Load (Figuur B4-40)

Dit stuk is door het programma zelf gegenereerd en wordt niet bewust gebruikt. Wanneer dit stuk verwijderd wordt functioneert het user interface niet meer naar behoren, om deze reden is er voor gekozen om dit stuk nog niet te verwijderen tot hier een oplossing voor gevonden is.

```

1011  private void Form1_Load(object sender, EventArgs e)
1012  {
1013  }
1014  }
```

Figuur B4-40 Regel 1011 tot 1014: Form1\_Load

#### Regel 1016 tot 1025: btn\_start\_PB\_Click (Figuur B4-41)

Wanneer de knop start\_PB ingedrukt wordt zal de kleur van alle panels terug gezet worden op de start positie, de panel die de verbinding aangeeft wordt oranje om aan te geven dat het programma bezig is. Hierna wordt de eerste deelttest van de Profibus test aangeroepen.

```

1016  private void btn_start_PB_Click(object sender, EventArgs e)
1017  {
1018      panel_test1.BackColor = System.Drawing.Color.LightGray;
1019      panel_test2.BackColor = System.Drawing.Color.LightGray;
1020      panel_test3_4.BackColor = System.Drawing.Color.LightGray;
1021      panel_test5.BackColor = System.Drawing.Color.LightGray;
1022      panel_test6.BackColor = System.Drawing.Color.LightGray;
1023      panel_connect.BackColor = System.Drawing.Color.Orange;
1024      test_state = "testP_HB_active";
1025  } //private void btn_start_PB_Click(object sender, EventArgs e)
```

Figuur B4-41 Regel 1016 tot 1025: btn\_start\_PB\_Click


#### Regel 1027 tot 1030: btn\_yes\_Click (Figuur B4-42)

Dit event zorgt ervoor dat de waarde “Continue\_Yes” hoog wordt wanneer op de knop yes gedrukt wordt.


```

1027  private void btn_yes_Click(object sender, EventArgs e)
1028  {
1029      Yes_Pressed = true;
1030  }
```

Figuur B4-42 Regel 1027 tot 1030: btn\_yes\_Click

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 39 van 40



	Teststelsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-----------------------------------	---


#### Regel 1032 tot 1035: btn\_no\_Click (Figuur B4-43)

Dit event zorgt ervoor dat de waarde “No\_Pressed” hoog wordt wanneer op de knop no gedrukt wordt.

```

1032  private void btn_no_Click(object sender, EventArgs e)
1033  {
1034      No_Pressed = true;
1035  }
```

Figuur B4-43 Regel 1032 tot 1035: btn\_no\_Click

	Document: AFST_TB Bijlage 4 Software v3	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 40 van 40

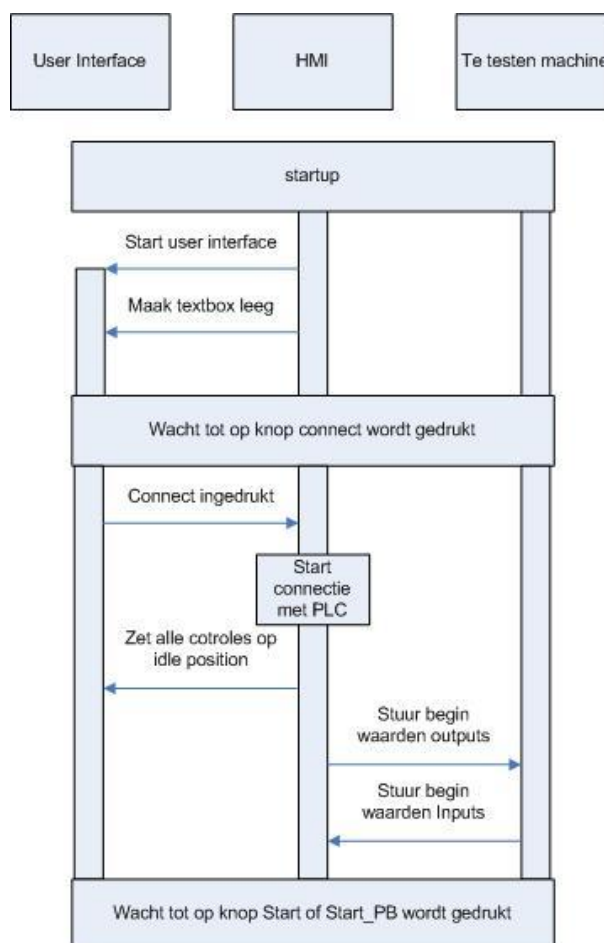
## Bijlage 5: Communicatie cyclus

In deze bijlage wordt voor elk deel van het programma weergegeven welke berichten er uitgewisseld worden, hierbij is uitgegaan van een test zonder fouten. Hierbij zijn enkel het User interface de HMI en de te testen machine weergegeven omdat dit de devices zijn die bepalen wat er verstuurd wordt. De PLC, TM41 en IO station zijn weg gelaten omdat deze enkel het signaal doorgeven en zelf niets aan het signaal veranderen. Ook wordt alle data continu doorgegeven, de signalen die weergegeven worden zijn enkel de wijzigingen.

Bij het opstarten wordt door de HMI de user interface opgestart, deze user interface wordt weergegeven op de HMI zelf. De reden dat dit een aparte kolom heeft is omdat hiermee aangegeven kan worden wat aan de gebruiker getoond wordt en welke responses door de gebruiker gegeven worden. Nadat de user interface opgestart is zal de textbox op de user interface leeg gemaakt worden. Hierna zal het programma niets uitvoeren tot er een event plaatsvindt. Het enige event wat op dit moment plaats kan vinden is dat de gebruiker op de knop connect drukt.

Wanneer op deze knop gedrukt is zal het programma de Ethernet verbinding met de PLC opbouwen. Nadat deze verbinding is opgebouwd worden alle controles op de user interface op de Idle positie gezet, dit houdt onder andere in dat de knoppen "Start" en "Start\_PB" beschikbaar worden. Hierna zullen alle uitgangswaarden van de HMI via de PLC naar de te testen machine gestuurd worden, als reactie hierop worden alle outputs van de te testen machine terug gestuurd naar de HMI. Hierna zal deze data continu heen en weer gestuurd worden.

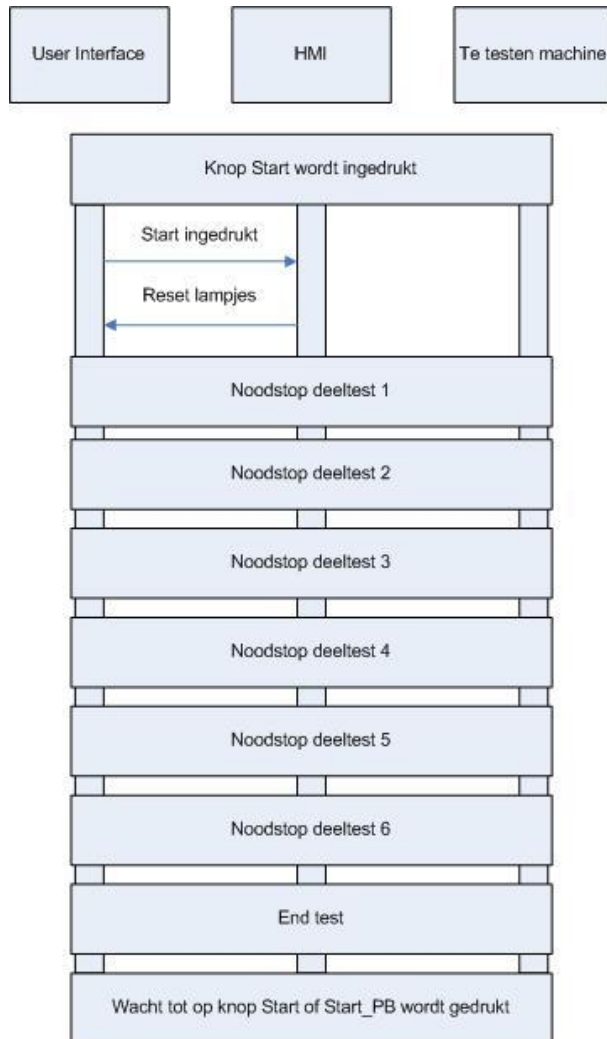
Het programma zal pas verder gaan wanneer op de knop "Start" of "Start\_PB" gedrukt wordt.



Figuur B5-1 Startup cyclus

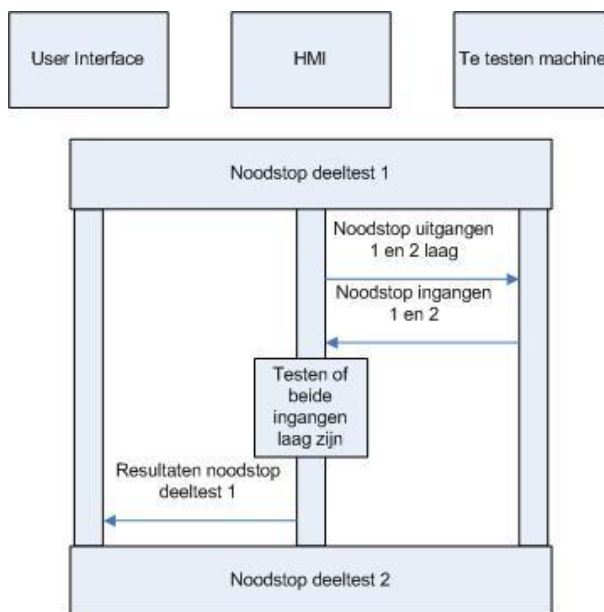
## Noodstoptest

Wanneer op de knop “Start” gedrukt wordt zullen alle testlampjes van de noodstop test uit gezet worden. Hierna zullen de noodstop deeltests 1 tot 6 uitgevoerd worden waarna naar de End test state gegaan wordt. Al deze tests en deze state worden uitgewerkt in deze paragraaf. Hierna zal het testprogramma weer wachten tot er weer op de knop “Start” of de knop “Start\_PB” gedrukt wordt.



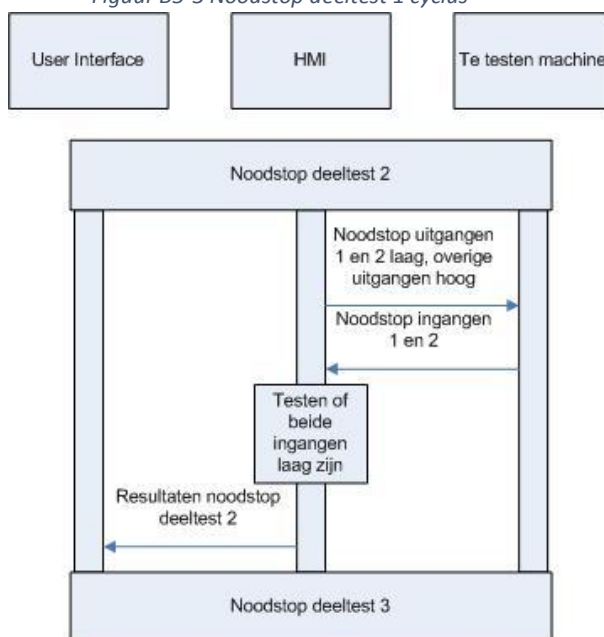
Figuur B5-2 Start noodstop cyclus

Vanuit deelttest 1 zullen beide noodstop uitgangen laag gemaakt worden, hierna zullen beide noodstop uitgangen uitgelezen worden. Hierna zal getest worden of beide input waarden laag zijn, wanneer dit het geval is zal naar de user interface gestuurd worden dat deelttest 1 succesvol was. Wanneer deze test succesvol is zal doorgedaan worden naar deelttest 2, als dit niet het geval is zal de naar de End test state gegaan worden.



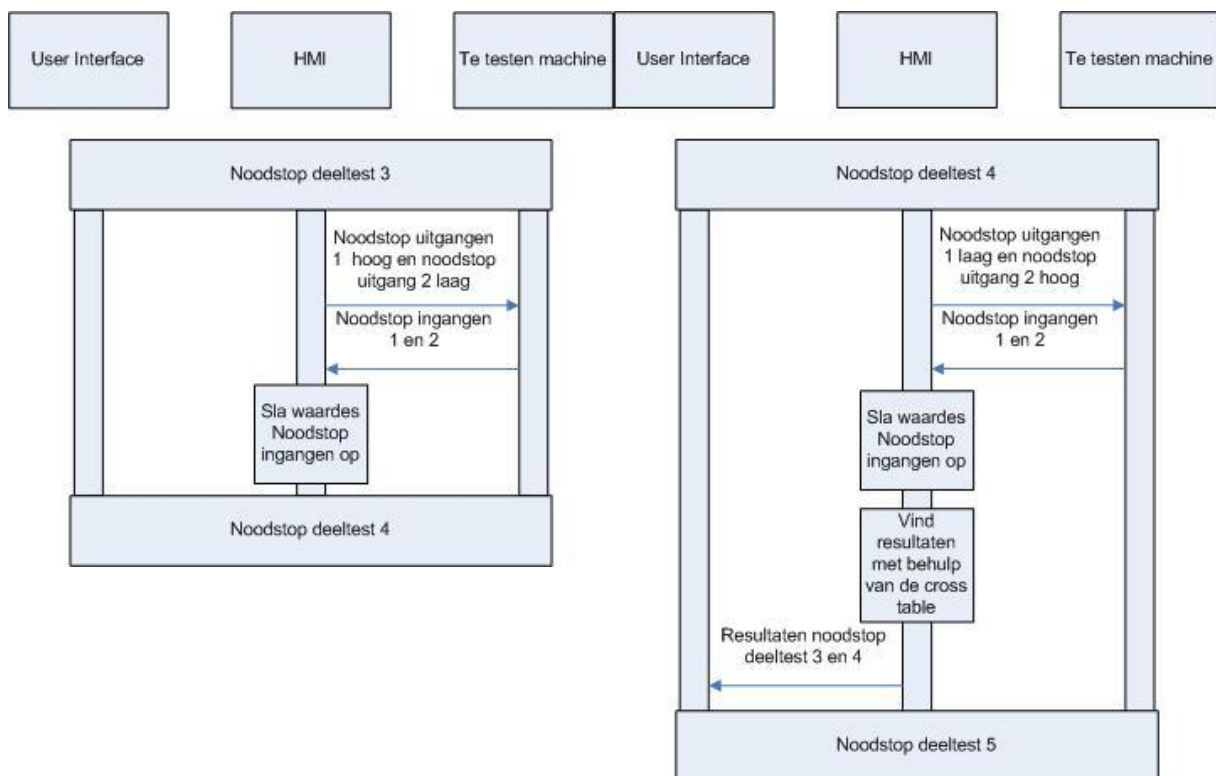
Figuur B5-3 Noodstop deelttest 1 cyclus

In Noodstop deelttest 2 zullen beide noodstop uitgangen wederom laag gemaakt worden echter worden hier de overige uitgangen hoog gemaakt. Hierna zullen beide noodstop ingangen weer ingelezen worden en zal er getest worden of deze beide weer laag zijn. Als dit het geval is zal weergegeven worden dat deelttest 2 succesvol was en zal doorgedaan worden naar deelttest 3, als dit niet het geval is zal naar de End test state gegaan worden.



Figuur B5-4 Noodstop deelttest 2 cyclus

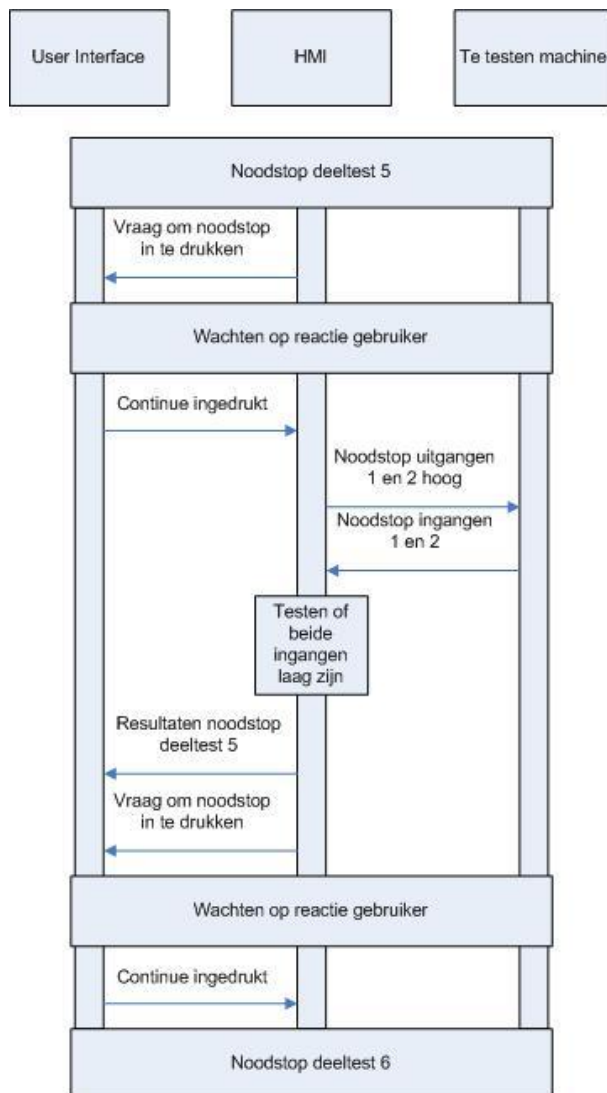
In deelttest 4 zal een conclusie getrokken worden uit de resultaten van deelttest 3 en deelttest 4 samen. In deelttest 3 wordt noodstop uitgang 1 hoog gemaakt en noodstop uitgang 2 laag. Hierna worden beide noodstop ingangen uitgelezen en opgeslagen. Hierna wordt in deelttest vier een vergelijkbare test uitgevoerd. Noodstop uitgang 1 wordt deze keer laag gemaakt en noodstop uitgang 2 wordt hoog. Hierna worden beide noodstop ingang waarden weer uitgelezen en opgeslagen. De opgeslagen waarden worden vergeleken met een cross table waar een conclusie uit volgt. Deze conclusie zal weergegeven worden op de user interface en wanneer geen error geconstateerd is zal doorgedaan worden naar deelttest 5, als wel een error geconstateerd is zal naar de stap End test gegaan worden.



Figuur B5-6 Noodstop deeltest 3 cyclus

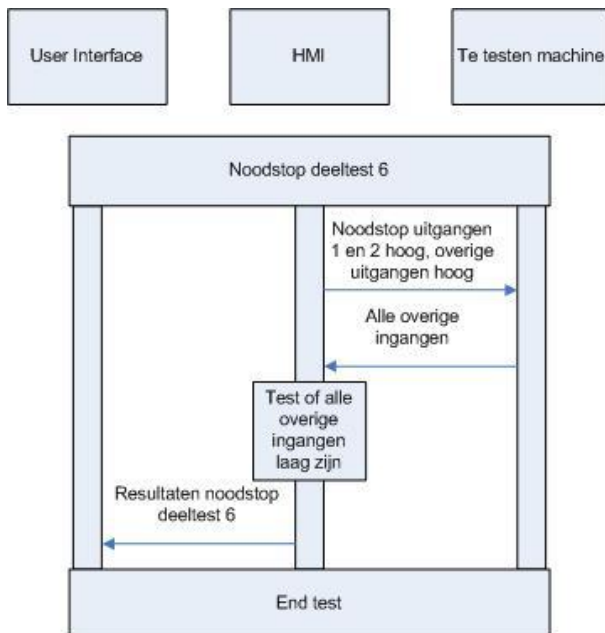
Figuur B5-5 Noodstop deeltest 4 cyclus

Aan het begin van deelttest 5 zal aan de gebruiker gevraagd worden om de noodstop in te drukken en wanneer dit gedaan is op continue te drukken. Wanneer op continue gedrukt is zullen beide noodstop uitgangen hoog gemaakt worden. Daarna zullen beide noodstop ingangen uitgelezen worden, wanneer deze beide laag zijn zal aangegeven worden dat de deelttest succesvol was. Hierna zal gevraagd worden om de noodstops uit te trekken en op continue te drukken. Wanneer op continue gedrukt wordt en de test succesvol was zal doorgedaan worden naar deelttest 6, is dit niet het geval dan zal naar de End test state gegaan worden.



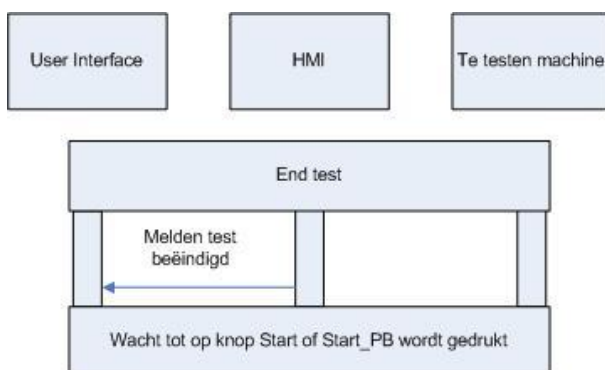
Figuur B5-7 Noodstop deelttest 5 cyclus

Deeltest 6 is de laatste deeltest van de noodstop tests, hierin worden beide noodstop uitgangen hoog gemaakt en alle overige uitgangen laag. Hierna zullen alle overige ingangen uitgelezen worden. Zijn deze allemaal laag dan zal aangegeven worden dat deeltest 6 succesvol was, na deze test zal sowieso naar de End test state gegaan worden.



Figuur B5-8 Noodstop deeltest 6 cyclus

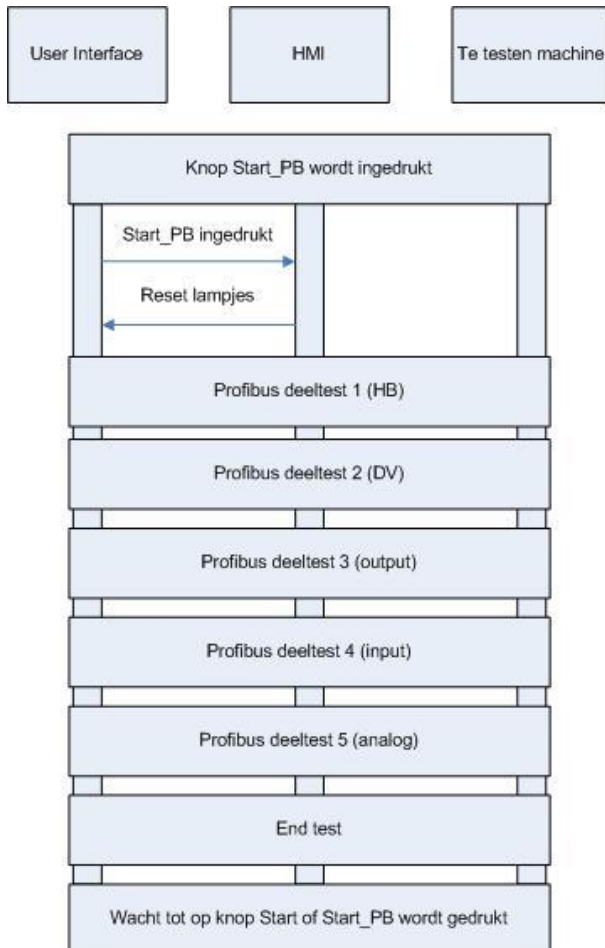
De state End test kan bereikt worden doordat de noodstop of Profibus test is afgerond of wanneer er een error in een van deze tests is geconstateerd. In de status End test zal op het user interface weergegeven worden dat de test beëindigd is. Hierna zal het programma wachten tot dat de knop "Start" of "Start\_PB" ingedrukt is.



Figuur B5-9 End test cyclus

## Profibustest

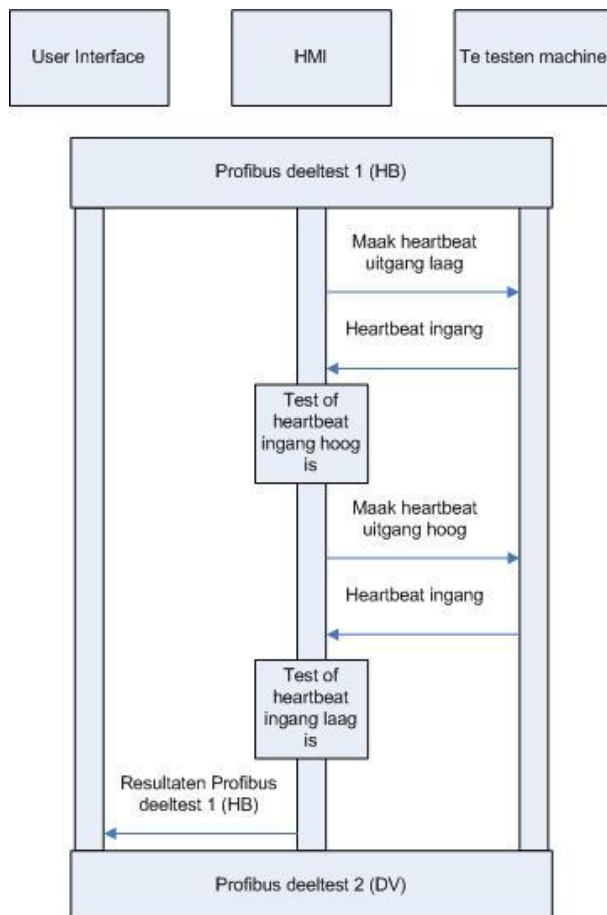
Wanneer op de knop “Start\_PB” gedrukt wordt zullen alle testlampjes van deze test uit gezet worden. Hierna zullen de Profibus deeltests 1 tot 5 uitgevoerd worden waarna naar de End test state gegaan wordt. Al deze tests en deze state worden uitgewerkt in deze paragraaf. Hierna zal het testprogramma weer wachten tot er weer op de knop “Start” of de knop “Start\_PB” gedrukt wordt.



Figuur B5-10 Start Profibus cyclus

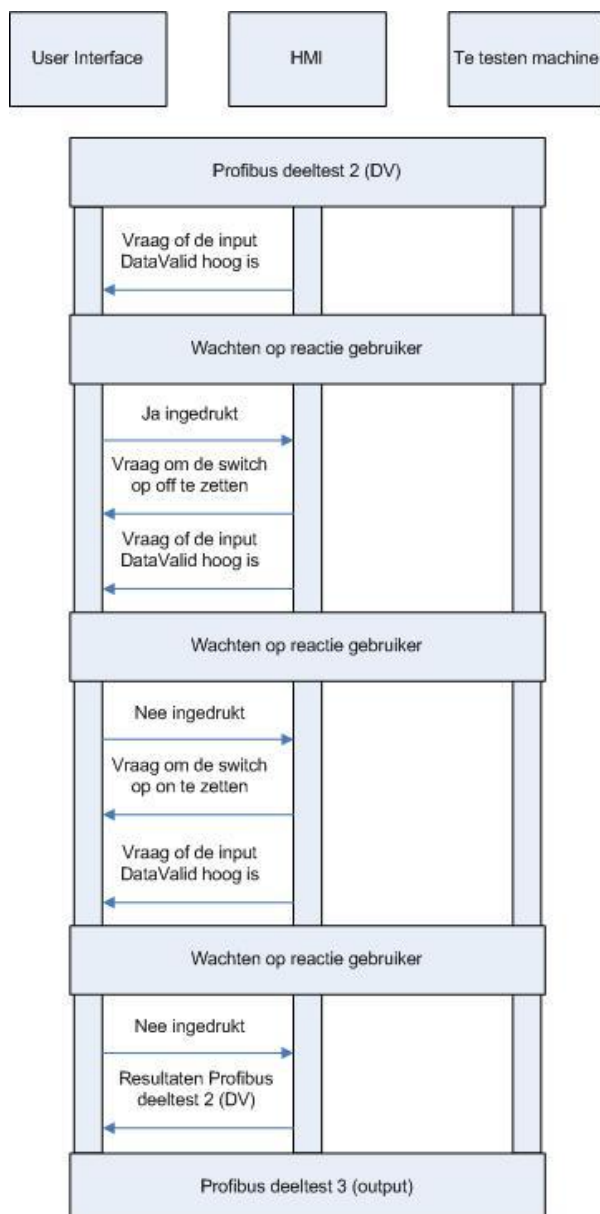


In Profibus deelttest 1 zal eerst de uitgaande heartbeat laag gemaakt worden, hierna zal na een halve seconde de heartbeat input ingelezen worden. Wanneer deze hoog is zal doorgedaan worden met de test, anders zal naar de End test step gegaan worden. Hierna zal de uitgaande heartbeat hoog gemaakt worden. Na een halve seconde zal de heartbeat input weer ingelezen worden, deze keer wordt getest of deze input laag is. Wanneer dit het geval is zal aangegeven worden dat deelttest 1 succesvol was en zal doorgedaan worden naar deelttest 2, als dit niet het geval is zal naar de End test state gegaan worden.



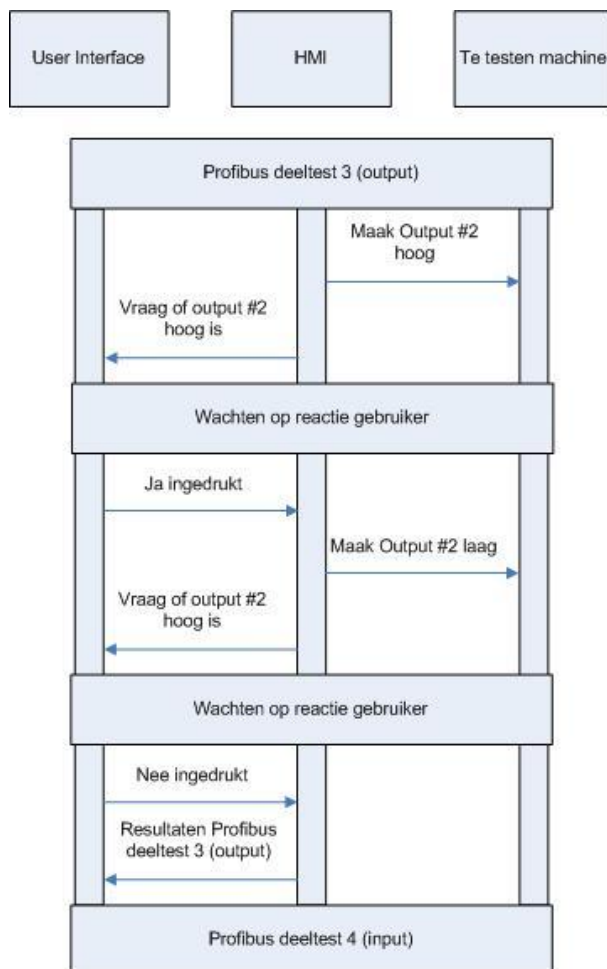
Figuur B5-11 Profibus deelttest 1 (HB) cyclus

Bij Profibus deelttest 2 zal eerst gevraagd worden of de DataValid input hoog is op de webtool. Wanneer op ja gedrukt wordt zal de test vervolgd worden, wanneer op nee gedrukt wordt zal naar de End test state gegaan worden. Hierna zal gevraagd worden om de DIL DP1 DIA switch op off te zetten en zal wederom gevraagd worden of de input DataValid hoog is. Wanneer nee geantwoord wordt zal doorgedaan worden met de test, als ja geantwoord wordt zal naar de End test state gegaan worden. Hierna zal gevraagd worden om de DIL DP1 DIA switch op on te zetten en zal wederom gevraagd worden of de input DataValid hoog is. Wanneer met ja geantwoord wordt zal aangegeven worden dat de test succesvol was en zal doorgedaan worden naar deelttest 3, als dit niet het geval is dan zal naar de End test state gegaan worden.

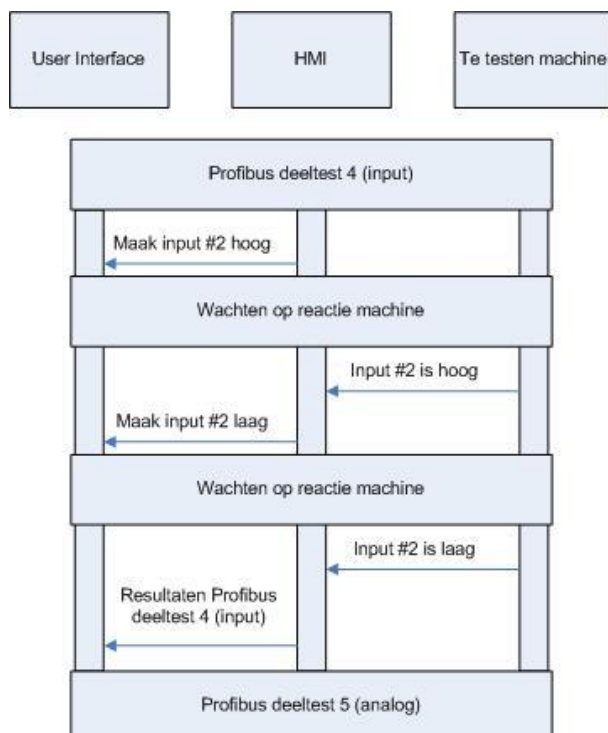


Figuur B5-12 Profibus deelttest 2 (DV) cyclus

Deeltest 3 wordt herhaald voor elke digitale Profibus output van het testsysteem. Als eerste wordt de digitale uitgang hoog gemaakt, hier aangeduid met #2. Hierna wordt aan de gebruiker gevraagd of deze output op de webtool staat aangegeven als hoog. Wanneer de gebruiker ja antwoord zal de output laag gemaakt worden en zal wederom gevraagd worden of deze als hoog aangegeven wordt, als dit niet het geval is zal naar de End test state gegaan worden. Wanneer door de gebruiker nu aangegeven wordt dat de output niet hoog is zal aangegeven worden op de user interface dat deeltest 3 succesvol is en zal doorgedaan worden naar de volgende output, als dit de laatste output was zal doorgedaan worden naar deeltest 4. Als dit niet het geval is zal naar de End test state gegaan worden.

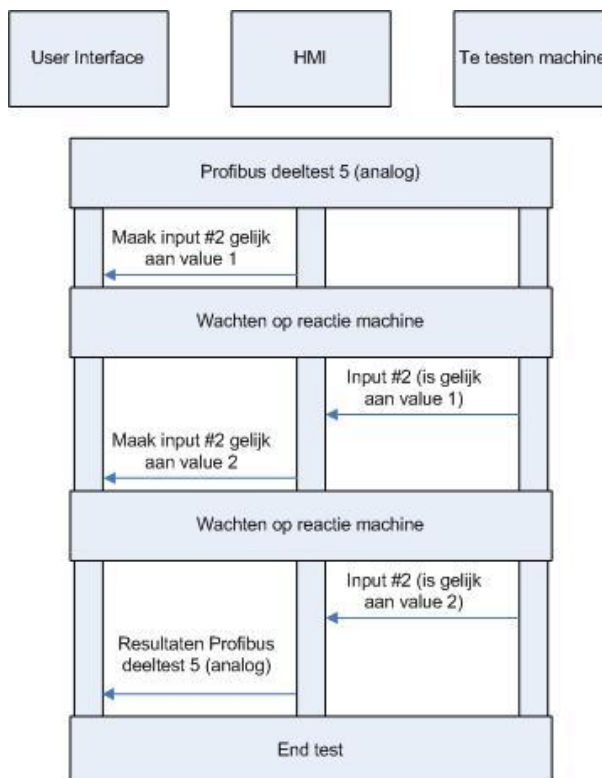


Figuur B5-13 Profibus deeltest 3 (output) cyclus



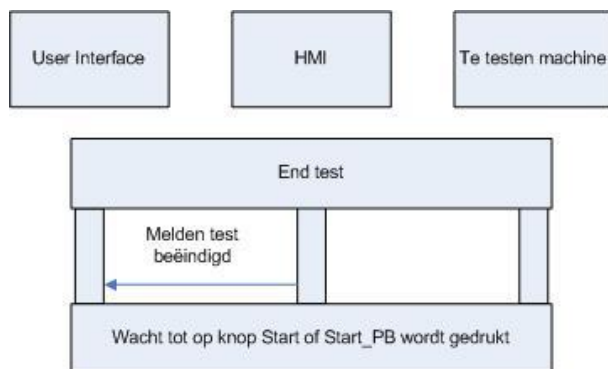
Figuur B5-15 Profibus deeltest 4 (input) cyclus

Deeltest 5 wordt herhaald voor elke analoge Profibus input van het testsysteem, hier aangeduid met #2. Als eerste zal gevraagd worden om de analoge input gelijk te maken aan een weergegeven waarde. Hierna zal gewacht worden tot gedetecteerd wordt dat de input deze waarde heeft. Hierna zal gevraagd worden om de input gelijk te maken aan een andere weergegeven waarde. Wanneer gedetecteerd wordt dat de input deze waarde heeft zal weergegeven worden dat deze test succesvol was en zal doorgedaan worden naar het testen van de volgende input. Was dit de laatste input dan zal doorgedaan worden naar de End test state. Wanneer het programma niet detecteert dat de input veranderd is zal door de gebruiker op continue gedrukt moeten worden. Hierna zal het programma naar de End test state gaan.



Figuur B5-14 Profibus deeltest 5 (analog) cyclus

Dit is de zelfde stap als beschreven bij de noodstop test. De state End test kan bereikt worden doordat de noodstop of Profibus test is afgerond of wanneer er een error in een van deze tests is geconstateerd. In de status End test zal op het user interface weergegeven worden dat de test beëindigd is. Hierna zal het programma wachten tot dat de knop “Start” of “Start\_PB” ingedrukt is.



Figuur B5-16 End test cyclus

## Bijlage 6: Test overzicht

In dit document zullen de verschillende tests van het testprogramma uitgewerkt worden. Hier zullen enkel de stappen behandeld worden er zal nog niet ingegaan worden op de code. Het eerste deel zal ingaan op de noodstop test, het tweede deel zal ingaan op de Profibus test. T zal staan voor true, hoog of waar, F zal staan voor false, laag of fout.

### Noodstop test


Noodstop deelttest 1 wordt gebruikt om een kortsluiting met een voeding uit te kunnen sluiten. Alle uitgang signalen van het testsysteem worden laag gemaakt hierna wordt gekeken of beide ingangen laag zijn. Is dit het geval dan wordt het testlampje groen gemaakt en gaat het programma verder naar de volgende test, is dit niet het geval dan zal een fout ontstaan. Na een fout zal het testlampje rood worden en daarnaast zal een error weergegeven worden, hierna zal de test beëindigd worden.

ErrorN101: Beide noodstop ingangen zijn hoog, kortgesloten aan een voedingsbron.

ErrorN102: Noodstop ingang 1 is hoog, kortgesloten aan voedingsbron.

ErrorN103: Noodstop ingang 2 is hoog, kortgesloten aan voedingsbron.

Noodstop deelttest 1	
Alle uitgangen worden laag gemaakt	
Beide noodstop ingangen worden uitgelezen	
TT:	test lamp 1 wordt rood
	laat errorN101 zien
	test beëindigen
TF:	test lamp 1 wordt rood
	laat errorN102 zien
	test beëindigen
FT:	test lamp 1 wordt rood
	laat errorN103 zien
	test beëindigen
FF:	test lamp 1 wordt groen
	laat succes test1 zien
	Ga naar noodstop deelttest 2

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

Noodstop deelttest 2 wordt gebruikt om een kortsluiting met een niet-noodstop ader uit te kunnen sluiten. Alle uitgangssignalen behalve de noodstop signalen worden hoog gemaakt en hierna wordt gekeken of beide ingangen laag zijn. Is dit het geval dan wordt het testlampje groen gemaakt en gaat het programma verder naar de volgende test, is dit niet het geval dan zal een fout ontstaan. Na een fout zal het testlampje rood worden en daarnaast zal een error weergegeven worden, hierna zal de test beëindigd worden.

ErrorN201: Beide noodstop ingangen zijn hoog, kortgesloten aan een niet-noodstop ader.


ErrorN202: Noodstop ingang 1 is hoog, kortgesloten aan een niet-noodstop ader.

ErrorN203: Noodstop ingang 2 is hoog, kortgesloten aan een niet-noodstop ader.

Noodstop deelttest 2	
Beide noodstop uitgangen laag maken, overige uitgangen hoog	
Beide noodstop ingangen worden uitgelezen	
TT:	test lamp 2 wordt rood
	laat errorN201 zien
	test beëindigen
TF:	test lamp 2 wordt rood
	laat errorN202 zien
	test beëindigen
FT:	test lamp 2 wordt rood
	laat errorN203 zien
	test beëindigen
FF:	test lamp 2 wordt groen
	laat succes test2 zien
	Ga naar noodstop deelttest 3

Noodstop deelttest 3 wordt in combinatie met noodstop deelttest 4 op verschillende problemen uit te kunnen sluiten. Deze problemen zijn kortsluiting tussen de noodstopaders, gekruiste noodstop aders en onderbreking in één of beide aders. In deze test zal het uitgangssignaal van noodstop 1 hoog gemaakt worden en het uitgangssignaal van noodstop 2 wordt laag. Hierna worden de resultaten van beide ingangen opgeslagen. Na deze test zal naar deelttest 4 gegaan worden

Noodstop deelttest 3	
Maak noodstop uitgang 1 hoog en alle andere uitgangen laag	
Sla de waarde op van noodstop ingang 1	
Sla de waarde op van noodstop ingang 2	
Ga naar noodstop deelttest 4	

	Document: AFST_TB Bijlage 6 Test overzicht v2	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 2 van 11

In noodstop deelttest 4 wordt het uitgangssignaal van noodstop 2 hoog gemaakt en het uitgangssignaal van noodstop 1 laag. Hierna worden beide noodstop ingangssignalen weer opgeslagen. De resultaten van deelttest 3 en 4 worden gecombineerd en door middel van een cross table wordt hier een conclusie uit getrokken. Is dit het gewenste resultaat dan wordt het testlampje groen gemaakt en gaat het programma verder naar de volgende test, is dit niet het geval dan zal een fout ontstaan. Na een fout zal het testlampje rood worden en daarnaast zal een error weergegeven worden, hierna zal de test beëindigd worden.

Noodstop deelttest 4	
Maak noodstop uitgang 2 hoog en alle andere uitgangen laag	
Sla de waarde op van noodstop ingang 1	
Sla de waarde op van noodstop ingang 2	
Gebruik de cross table om het juiste resultaat te vinden	
Fout:	test lamp 3 wordt rood
	laat errorN30X zien, afhankelijk van fout
	test beëindigen
Goed:	test lamp 3 wordt groen
	laat succes test3 en 4 zien
	Ga naar noodstop deelttest 5

ErrorN301: Beide aders onderbroken, overige problemen niet detecteerbaar.

ErrorN302: Ader 1 onderbroken.

ErrorN303: Ader 1 onderbroken en aders gekruist.

ErrorN304: Kortsluiting na breuk, mogelijk gekruist.

ErrorN305: Ader 2 onderbroken en aders gekruist.

ErrorN306: Kortsluiting voor breuk, mogelijk gekruist.

ErrorN307: Aders gekruist.

ErrorN308: Deze error zou niet mogelijk moeten zijn.

ErrorN309: Ader 2 onderbroken.

ErrorN310: Ader onderbroken en kortsluiting voor breuk, mogelijk gekruist.

ErrorN311: Deze error zou niet mogelijk moeten zijn.

ErrorN312: Ader 2 onderbroken en kortsluiting na breuk.

ErrorN313: Deze error zou niet mogelijk moeten zijn.

ErrorN314: Deze error zou niet mogelijk moeten zijn.

ErrorN315: Kortsluiting.

deelttest 3		deelttest4		
input 1	input 2	input 1	input 2	
F	F	F	F	errorN301
F	F	F	T	errorN302
F	F	T	F	errorN303
F	F	T	T	errorN304
F	T	F	F	errorN305
F	T	F	T	errorN306
F	T	T	F	errorN307
F	T	T	T	errorN308
T	F	F	F	errorN309
T	F	F	T	succes
T	F	T	F	errorN310
T	F	T	T	errorN311
T	T	F	F	errorN312
T	T	F	T	errorN313
T	T	T	F	errorN314
T	T	T	T	errorN315




In noodstop deelttest 5 wordt getest of de noodstoppen goed werken, deze test wordt herhaald voor elke noodstop. Om dit te testen worden beide noodstop uitgangssignalen hoog gemaakt, hierna wordt aan de gebruiker gevraagd of deze een noodstopknop wil indrukken. Wanneer de gebruiker aangeeft dat verder gegaan kan worden zullen beide noodstop ingangssignalen uitgelezen worden. Zijn beide ingangssignalen laag dan wordt het testlampje groen gemaakt en gaat het programma verder naar de volgende test, is dit niet het geval dan zal een fout ontstaan. Na een fout zal het testlampje rood worden en daarnaast zal een error weergegeven worden, hierna zal de test beëindigd worden. Aan het eind van deze test wordt aan de gebruiker gevraagd om de noodstopknoppen weer uit te trekken.

ErrorN501: Beide noodstop ingangen zijn hoog, noodstop functioneert niet, is niet ingedrukt of er zit een kortsluiting om de noodstop.

ErrorN502: Noodstop ingang 1 is hoog, slecht één van de contacten functioneert correct, de ander functioneert niet of hier zit een kortsluiting omheen.

ErrorN503: Noodstop ingang 2 is hoog, slecht één van de contacten functioneert correct, de ander functioneert niet of hier zit een kortsluiting omheen.

Noodstop deelttest 5	
Maak beide noodstop uitgang signalen hoog	
Vraag om de noodstop in te drukken	
Beide noodstop ingangen worden uitgelezen	
TT:	test lamp 5 wordt rood
	laat errorN501 zien
	test beëindigen
TF:	test lamp 5 wordt rood
	laat errorN502 zien
	test beëindigen
FT:	test lamp 5 wordt rood
	laat errorN503 zien
	test beëindigen
FF:	test lamp 5 wordt groen
	laat succes test5 zien
	Ga naar noodstop deelttest 6
vraag om de noodstop uit te trekken	

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

Als laatste wordt noodstop deeltest 6 uitgevoerd, hierna wordt gekeken of een overige input hoog is. Uitgangen en ingangen die niet laag gemaakt kunnen worden zullen hierbij niet meegerekend worden. Wanneer alle inputs laag zijn wordt het testlampje groen gemaakt en zal de test beëindigd worden, is dit niet het geval dan zal een fout ontstaan. Na een fout zal het testlampje rood worden en daarnaast zal een error weergegeven worden, hierna zal de test beëindigd worden.

ErrorN601: Eén of beide noodstopaders zijn kortgesloten aan één of meer niet-noodstop aders.

Noodstop Deeltest 6	
Maak beide noodstop outputs hoog en alle niet-noodstop outputs laag	
Alle overige ingangen worden uitgelezen	
&T:	test lamp 6 wordt rood
	laat errorN601 zien
	test beëindigen
&F:	test lamp 6 wordt groen
	laat succes test6 zien
	test beëindigen

Nadat een fout wordt geconstateerd of wanneer de noodstop test is afgerond zal het programma bij de stap “end test” terecht komen. Hierin zal aangegeven worden dat de test is afgelopen en alle waardes zullen terug gezet worden op hun start positie, hierdoor kan direct een nieuwe test gestart worden. Na deze stap zal het testsysteem naar een ruststand gaan.

End test
Laat zien test beëindigd
alle waardes terug zetten naar hun startpositie
Ga naar rust stand

## Profibustest


Bij deze tests wordt gebruik gemaakt van een webtool, hiermee kunnen de Profibus ingangen en uitgangen bestuurd en uitgelezen worden. Wat voor het testsysteem een output is, is voor de webtool een input en visa versa.

De Profibustest wordt gestart met het testen van de heartbeat. Om dit goed te testen wordt eerst de automatische heartbeat uitgeschakeld. Hierna wordt de heartbeat output laag gemaakt. 500ms hierna zal het ontvangen heartbeat signaal hoog moeten zijn, wanneer dit niet het geval is zal een error getoond worden en zal de test gestopt worden. Is dit signaal wel hoog dan zal het uitgaande heartbeat signaal hoog gemaakt worden. 500ms hierna zal het ontvangen heartbeat signaal laag moeten zijn, wanneer dit niet het geval is zal een error getoond worden en zal de test gestopt worden. Is dit wel het geval dan zal doorgedaan worden naar de volgende test.

ErrorP101: Het heartbeat signaal blijft laag, mogelijk is er geen verbinding.

ErrorP102: Het heartbeat signaal blijft hoog, deze error zou niet moeten kunnen gebeuren.

Profibus Deeltest HB	
schakel heartbeat uit	
Maak heartbeat laag	
Wacht 500ms	
lees heartbeat response uit	
T:	Doorgaan
F:	laat errorP101 zien
	test beëindigen
Maak heartbeat hoog	
Wacht 500ms	
lees heartbeat response uit	
T:	laat errorP101 zien
	test beëindigen
F:	ga naar Profibus deeltest DV
	Schakel heartbeat in

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---


Als tweede wordt Profibus deelttest DV getest, dit signaal wordt vanuit de DP/DP coupler naar de te testen machine gestuurd. De resultaten hiervan moeten uitgelezen worden op de Contiweb webtool. Als eerste wordt gevraagd of de waarde van DataValid hoog is, wanneer dit niet het geval is zal een error weergegeven worden en zal de test beëindigd worden en wanneer dit wel zo is zal de test doorgaan. Hierna zal gevraagd worden om de switch DIL DP1 DIA op off te zetten, hierdoor zal het bericht niet meer verstuurd worden. Dan wordt gevraagd of DataValid nog steeds hoog is, wanneer dit zo is zal een error weergegeven worden en zal de test beëindigd worden is dit niet zo dan zal de test doorgaan. Hierna zal gevraagd worden om de switch DIL DP1 DIA op on te zetten, hierdoor zullen de berichten weer verstuurd worden. Dan wordt gevraagd of DataValid weer hoog is, wanneer dit zo is zal doorgedaan worden naar de volgende test en wanneer dit niet zo is zal een error weergegeven worden en zal de test beëindigd worden.


ErrorP201: DataValid is laag, mogelijk is de switch niet goed ingesteld.

ErrorP202: DataValid blijft hoog, mogelijk is de switch defect of is de verkeerde switch omgezet.

ErrorP203: DataValid blijft laag, mogelijk is de switch defect of is de verkeerde switch omgezet.

Profibus deelttest DV	
vraag of DataValid hoog is	
ja:	doorgaan
nee:	laat errorP201 zien
	test beëindigen
vraag om de DIL DP1 DIA switch op off te zetten	
vraag of DataValid hoog is	
ja:	laat errorP202 zien
	test beëindigen
nee:	doorgaan
vraag om de DIL DP1 DIA switch op on te zetten	
vraag of DataValid hoog is	
ja:	Ga naar Profibus deelttest output 1
nee:	laat errorP203 zien
	test beëindigen

	Document: AFST_TB Bijlage 6 Test overzicht v2	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 7 van 11

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

De Profibus outputs worden getest met behulp van de webtool. De Profibus deelttest output #1 wordt uitgevoerd voor elke digitale Profibus output van het testsysteem, in de tabel naast de test is weergegeven wat de waardes #1 en #2 betekenen in elke afzonderlijke test. Eerst zal de geselecteerde output hoog gemaakt worden, daarna zal aan de gebruiker gevraagd worden om deze uit te lezen. Wanneer deze output hoog is zal doorgedaan worden, wanneer dit niet het geval is zal een error weergegeven worden en zal de test gestopt worden. Hierna zal de geselecteerde output laag gemaakt worden en daarna zal aan de gebruiker gevraagd worden om deze uit te lezen. Wanneer deze output hoog is zal een error weergegeven worden, is deze laag dan zal de volgende output getest worden. Na de laatste output zal naar de volgende test gegaan worden.

ErrorP301: De output wordt niet hoog, mogelijk is deze niet goed uitgelezen.

ErrorP302: De output wordt niet laag, mogelijk is deze niet goed uitgelezen.

Profibus deelttest output #1	
maak output #2 hoog	
vraag of output #2 hoog is	
ja:	doorgaan
nee:	laat errorP301 zien
	test beëindigen
maak output #2 laag	
vraag of output #2 laag is	
ja:	laat errorP302 zien
	test beëindigen
nee:	Ga naar #1+1

#1	#2
1	PRE_PST_GEN_DI_ClearFaults
2	PRE_PST_GEN_DI_PressFaulted
3	PRE_PST_GEN_DI_PressQuickStop
4	PRE_PST_UNW_DI_MotionStarting
5	PRE_PST_SPL_DI_RemoteSpliceRequest
6	PRE_PST_UNW_DI_WebTensionSelection

De Profibus inputs worden getest met behulp van de webtool. De Profibus deelttest input #1 wordt uitgevoerd voor elke digitale Profibus input van het testsysteem, in de tabel onder de test is weergegeven wat de waardes #1 en #2 betekenen in elke afzonderlijke test. Eerst zal gevraagd worden om de geselecteerde output hoog te maken, wanneer hier geen response op komt zal door de gebruiker op continue gedrukt moeten worden. Wanneer gedetecteerd wordt dat de input hoog wordt zal door het testsysteem naar het volgende deel van deze deelttest gegaan worden, wanneer op continue gedrukt wordt zal een error weergegeven worden en zal de test gestopt worden. Hierna zal gevraagd worden om de geselecteerde output laag te maken, wanneer hier geen response op komt zal door de gebruiker op continue gedrukt moeten worden. Wanneer op continue gedrukt wordt zal een error weergegeven worden en zal de test gestopt worden. Wanneer gedetecteerd wordt dat de input laag wordt zal de volgende input getest gaan worden, na de laatste input zal naar de volgende test gegaan worden.

ErrorP401: De input wordt niet hoog, mogelijk is deze niet goed ingesteld.

ErrorP402: De input wordt niet laag, mogelijk is deze niet goed ingesteld.

Profibus deelttest input #1	
vraag om input #2 hoog te maken, wanneer geen response druk op continue	
hoog:	doorgaan
continue:	laat errorP401 zien
	test beëindigen
vraag om input #2 laag te maken, wanneer geen response druk op continue	
laag:	Ga naar #1+1
continue:	laat errorP402 zien
	test beëindigen


#1	#2
1	PRE_PST_GEN_DO_ClearFaults
2	PRE_PST_GEN_DO_LocalEmergencyReleased
3	PRE_PST_GEN_DO_NormalStop
4	PRE_PST_SPL_DO_SpliceImminent
5	PRE_PST_SPL_DO_SplicePrepared
6	PRE_PST_UNW_DO_WebTensionLowPresent
7	PRE_PST_GEN_DO_LocalGuardsClosed
8	PRE_PST_SPL_DO_SingleReelModeSelected
9	PRE_PST_SPL_DO_SpliceActive
10	PRE_PST_SPL_DO_SpliceKnifeOut
11	PRE_PST_GEN_DO_SplicerFailure
12	PRE_PST_UNW_DO_WebTensionHighPresent

De analoge Profibus input signalen worden getest met behulp van de webtool. De Profibus deelttest analog #1 wordt uitgevoerd voor elk analoge Profibus input signaal van het testsysteem, in de tabel onder de test is weergegeven wat de waardes #1 en #2 betekenen in elke afzonderlijke test. Eerst zal gevraagd worden om de geselecteerde output gelijk te maken aan value1, wanneer hier geen response op komt zal door de gebruiker op continue gedrukt moeten worden. Wanneer gedetecteerd wordt dat de input gelijk is aan value1 zal door het testsysteem naar het volgende deel van deze deelttest gegaan worden, wanneer op continue gedrukt wordt zal een error weergegeven worden en zal de test gestopt worden. Hierna zal gevraagd worden om de geselecteerde output gelijk te maken aan value2, wanneer hier geen response op komt zal door de gebruiker op continue gedrukt moeten worden. Wanneer op continue gedrukt wordt zal een error weergegeven worden en zal de test gestopt worden. Wanneer gedetecteerd wordt dat de input gelijk is aan value2 zal de volgende input getest gaan worden, na de laatste input zal naar de test gestopt worden.

ErrorP501: De output wordt niet gelijk aan value1, mogelijk is deze niet goed ingesteld.


ErrorP502: De output wordt niet gelijk aan value2, mogelijk is deze niet goed ingesteld.

Profibus deelttest analog #1	
vraag om input #2 gelijk te maken aan value1, wanneer geen response druk op continue	
Value1:	doorgaan
continue:	laat errorP501 zien
	test beëindigen
vraag om input #2 gelijk te maken aan value2, wanneer geen response druk op continue	
Value2:	Ga naar #1+1
continue:	laat errorP502 zien
	test beëindigen
#1	#2
1	RunningReelPosition
2	TimeToSpliceInSec
3	WebTensionHighFeedbackLBS
4	RunningReelDiameterINCH
5	TimeToSpliceSelector
6	TimeToSpliceInMin
7	RunningReelRemainingLengthINCH
8	SpliceDiameterFeedbackINCH


	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

Nadat een fout wordt geconstateerd of wanneer de profibus test is afgerond zal het programma bij de stap “end test” terecht komen, dit is de zelfde stap als bij de noodstoptest. Hierin zal aangegeven worden dat de test is afgelopen en alle waardes zullen terug gezet worden op hun start positie, hierdoor kan direct een nieuwe test gestart worden. Na deze stap zal het testsysteem naar een ruststand gaan.

End test
Laat zien test beëindigd
alle waardes terug zetten naar hun startpositie
Ga naar rust stand

	Document: AFST_TB Bijlage 6 Test overzicht v2	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 11 van 11



	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

## Bijlage 7: Testplan en resultaten

In dit document zullen de testplannen van de hardware, de noodstoptest en de Profibustest weergegeven worden. Er zal niet ingegaan worden op de hardware en software zelf.

### Hardware

Om te kunnen concluderen dat de hardware van het testsysteem naar behoren werkt zullen alle devices en alle outputs functioneel getest moeten worden. Bij dit testplan zijn nog geen resultaten omdat de hardware nog niet gerealiseerd is.

Om de beginnen zal de hoofdschakelaar uitgeschakeld moeten zijn en de voedingsstekker nog niet aangesloten op een stopcontact. Wanneer de voedingsstekker wordt aangesloten op een stopcontact zal er nog niets moeten gebeuren, het HMI scherm mag nog niet aangaan en er mogen nog geen lampjes gaan branden op welke device dan ook.

Hierna moet de hoofdschakelaar ingeschakeld worden. De HMI moet opstarten en ook op de PISA, PLC, het IO Station en de TM41 moeten lampjes gaan branden en knipperen. Deze zullen later nog afzonderlijk getest worden.

De verschillende devices hebben een opstart procedure, om deze reden zal een korte tijd gewacht moeten wachten tot de devices gecontroleerd kunnen worden. Nadat de HMI is opgestart zou automatisch het testscherm moeten verschijnen.


Op het PISA element moet de led "Input Status" continu groen branden, de leds "output Failure 1-4" moeten continu uit staan.


Op de PLC moeten de leds "RDY" en "RUN/STOP" continu groen branden, de leds "OUT>5V/SY" en "SF/BF" moeten continu uit staan.

Op de hoofdmodule van het IO station moeten de leds "RN" en "PWR" continu branden, de leds "ER" en "MT" moeten continu uit staan. Links hiervan zit één Profinet module, hiervan moet de led "LK1" continu groen branden en de led "LK2" moet continu uit staan. Voor de rest van de modules geldt dat de leds "DIAG" en "PWR" continu groen moeten branden, naar de output leds zal later gekeken worden.

Op de TM41 moet de led "READY" continu groen branden.

Wanneer dit allemaal werkt naar behoren kunnen alle in en outputs getest gaan worden.

	Document: AFST_TB Bijlage 7 testplan en resultaten v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 1 van 7

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

Om alle outputs te kunnen testen worden deze vanuit de HMI aangestuurd, op de HMI zal SIMOTION SCOUT gebruikt moeten worden om de outputs direct aan te kunnen sturen.


De outputs op de PLC moeten allemaal hoog gemaakt worden, hierna kunnen deze uitgangen met behulp van een multimeter nagekeken worden. De negatieve testpin van de multimeter kan op verbinding X121.12 op de PLC gehouden worden, dit is de massa en zal als referentie gebruikt worden. De volgende pinnen moeten een hoog signaal geven en zullen stuk voor stuk uitgelezen moeten worden: X121.8, X121.9, X121.10, X121.11, X131.1, X131.2, X131.4 en X131.5. Hierna moet de negatieve testpin van de multimeter op verbinding X130.8 gehouden worden en de positieve testpin op X130.7, deze geïsoleerde output moet ook hoog zijn.


Het IO station heeft 7 digitale modules, voor elke machine moet deze test herhaald worden. Als eerste wordt de byte horende bij de module gelijk gemaakt aan 16#FF, hierdoor zouden alle outputs hoog moeten worden, dit wordt aangegeven door de leds. Hierna zal de byte gelijk gemaakt moeten worden aan 16#00, hierdoor zullen alle leds uit moeten gaan. Hierna zullen de volgende waarden aan de bytes gegeven moeten worden en zullen de afzonderlijke uitgangen oplopend aan moeten gaan: 12#01, 16#02, 16#04 en 16#08. Hierna moet de byte weer op 16#00 gezet worden.

Daarnaast heeft het IO station ook nog een analoge module. Hierbij zal eerst voor de ene output één van de bytes gelijk gemaakt moeten worden aan 16#FF, de led zal nu aan moeten gaan. Hierna moet de byte weer gelijk gemaakt worden aan 16#00 en zal dit gedeelte herhaald moeten worden voor de andere output.

Het encoder signaal zal getest moeten worden door deze aan te sluiten aan een rollenwisselaar of een proefopstelling bestaande uit een PLC met Siemens SMC30 encoder module. Hierna zal de encoder signal test uitgevoerd moeten worden, wanneer dit signaal goed ontvangen wordt kan geconcludeerd worden dat de TM41 en de verbindingen hiernaartoe correct werken.

De Profibus verbinding kan getest worden door deze aan te sluiten op een machine of een proefopstelling bestaande uit een PLC en een DP/DP coupler. Hierna kunnen met behulp van de webtool en SIMOTION SCOUT verschillende berichten op en neer gestuurd worden. Wanneer één bericht van het testsysteem naar de machine of proefopstelling en één bericht in tegengestelde richting werken zal geconcludeerd kunnen worden dat de Profibus verbinding correct is aangesloten. Als laatste is er ook een Ethernet verbinding, wanneer een computer of ander device aangesloten wordt op de ethernet aansluiting is de verbinding voldoende getest wanneer een IP ping succesvol uitgevoerd kan worden.

	Document: AFST_TB Bijlage 7 testplan en resultaten v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 2 van 7

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

## Noodstoptest

De noodstoptest zal getest worden door bij elke deelttest alle mogelijke resultaten aan te roepen. Dit wordt gedaan door de test uit te voeren op een machine of een proefopstelling. Omdat er bij de eerste test enkele foute conclusies getrokken werden is de code nagekeken en aangepast, hierna is een tweede test uitgevoerd. De resultaten van de beide tests zijn gemarkeerd met "(1)" voor de eerste test en "(2)" voor de tweede test.

Als eerst zullen alle teststappen uitgevoerd worden, het is hierbij de bedoeling dat de gebruiker alle reacties geeft zoals bij een normale test zou gebeuren. Wanneer alle stappen uitgevoerd worden en de juiste vragen en resultaten weergegeven worden kan geconcludeerd worden de test bij een goed werkende machine goed werkt. Hierna zullen nog alle "error" states getest moeten worden.

(1 en 2): Alle stappen werden doorlopen en alle verwachte berichten werden op het user interface weergegeven.

De deelttests stoppen de test wanneer een error waargenomen wordt, daarnaast volgen deze deelttests elkaar snel op waardoor de inputs tussen de tests niet gewijzigd kunnen worden. Om deze redenen moeten een aantal regels aan de software toegevoegd worden waardoor na een error doorgedaan wordt naar de volgende deelttest. Deze regels zijn in de code weergegeven als:

`//test_state = "test6_active"; //for testing` . Wanneer de "//" aan het begin van de regel weg gehaald wordt zal deze regel door het programma uitgevoerd worden. Na het testen moeten deze "//" weer terug geplaatst worden, de regels waar dit moet gebeuren zijn te herkennen aan "//for testing" wat achter elk van deze zinnen staat.

Wanneer kortsluiting gemaakt wordt tussen de voeding en beide noodstop inputs zullen de volgende errors ontstaan: errorN101, errorN201, errorN315 en errorN501.

(1 en 2) De gewenste errors worden weergegeven.

Wanneer kortsluiting gemaakt wordt tussen de voeding en noodstop input 1 zullen de volgende errors ontstaan: errorN102, errorN202, errorN311\* en errorN502.

(1) In plaatst van errorN202 wordt error N203 weergegeven, de error condities bleken omgewisseld.

(2) De gewenste errors worden weergegeven.

\*errorN311 geeft aan dat deze error niet mogelijk zou moeten zijn, dit komt doordat de test normaal voor deze error beëindigd zou worden.

Wanneer kortsluiting gemaakt wordt tussen de voeding en noodstop input 1 zullen de volgende errors ontstaan: errorN103, errorN203, errorN313\* en errorN503.


(1) In plaatst van errorN203 wordt error N202 weergegeven, de error condities bleken omgewisseld.


(2) De gewenste errors worden weergegeven.

\*errorN313 geeft aan dat deze error niet mogelijk zou moeten zijn, dit komt doordat de test normaal voor deze error beëindigd zou worden.

Maak nu een kortsluiting tussen de voeding en ingang 1 va de tweede IO kaart, hierdoor zal de volgende error ontstaan: errorN601.

(1 en 2) De gewenste error wordt weergegeven.

	Document: AFST_TB Bijlage 7 testplan en resultaten v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 3 van 7

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

In de volgende tests zal enkel gekeken worden naar “error3##” errors, er zullen ook andere errors optreden maar deze zijn in voorgaande tests getest. Waar bij de voorgaande errors van te voren erg duidelijk was welke tekst bij welke error hoort is dit hier redelijk onduidelijk. Om deze reden zal er bij elke test goed gelet moeten worden of de veroorzaakte situatie overeen komt met de weergegeven errortekst. Verwijder na elke test de kortsluitingen, verander verder enkel wat aangegeven wordt dat veranderd moet worden, ga niet terug naar de startpositie.

Verwijder beide noodstop aders uit de ingangen, hierdoor zal errorN301 optreden.

(1 en 2) De error wordt correct weergegeven.

Plaats enkel noodstop ader 2 terug in noodstop ingang 2, hierdoor zal errorN302 optreden.

(1 en 2) De error wordt correct weergegeven.

Plaats noodstop ader 2 nu in noodstop ingang 2, hierdoor zal errorN303 optreden.

(1 en 2) De error wordt correct weergegeven.

Maak nu kortsluiting tussen noodstop ingang 1 en 2, hierdoor zal errorN304 optreden.

(1) Correcte error wordt weergegeven, echter wordt aangegeven dat noodstop ader 2 is onderbroken in plaats van noodstop ader 1.

(2) De error wordt correct weergegeven.

Verwijder noodstop ader 2 en plaats noodstop ader 1 in noodstop ingang 2, hierdoor zal errorN305 optreden.

(1 en 2) De error wordt correct weergegeven.

Plaats nu noodstop ader 1 samen met noodstop ader 2 in noodstop ingang 2, hierdoor zal errorN306 optreden.

(1 en 2) De error wordt correct weergegeven.

Plaats nu noodstop ader 2 in noodstop ingang 1, zorg ervoor dat noodstopader 1 in noodstop ingang 2 blijft, hierdoor zal errorN307 optreden.

(1 en 2) De error wordt correct weergegeven.

Maak nu een kortsluiting tussen de voeding en noodstop ingang 2, hierdoor zal errorN308\* optreden.

(1 en 2) De error wordt correct weergegeven.


\*errorN308 geeft aan dat deze error niet mogelijk zou moeten zijn, dit komt doordat de test normaal voor deze error beëindigd zou worden.


Verwijder nu beide aders en sluit noodstop ader 1 aan op noodstop ingang 1, hierdoor zal errorN309 optreden.

(1 en 2) De error wordt correct weergegeven.

Sluit nu noodstop ader 2 samen met noodstop ader 1 aan op noodstop ingang 1, hierdoor zal errorN310 optreden.

(1 en 2) De error wordt correct weergegeven.

	Document: AFST_TB Bijlage 7 testplan en resultaten v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 4 van 7

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
---	-------------------------------	---


Verwijder noodstop ader 2 en maak een kortsluiting tussen noodstop ingang 1 en noodstop ingang 2, hierdoor zal errorN312 optreden.


(1 en 2) De error wordt correct weergegeven.

Sluit noodstop ader 1 nu aan op noodstop ingang 2 en maak een kortsluiting tussen de voeding en noodstop input 1, hierdoor zal errorN314\* optreden.

(1 en 2) De error wordt correct weergegeven.

\*errorN314 geeft aan dat deze error niet mogelijk zou moeten zijn, dit komt doordat de test normaal voor deze error beëindigd zou worden.

	Document: AFST_TB Bijlage 7 testplan en resultaten v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 5 van 7

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxtmeer
--	-------------------------------	--

## Profibustest

De Profibustest zal getest worden door bij elke deelttest alle mogelijke resultaten aan te roepen. Bij de eerste test zijn verschillende fouten naar voren gekomen, hierna zijn deze fouten aangepast en is een tweede test uitgevoerd. De resultaten van de beide tests zijn gemarkeerd met "(1)" voor de eerste test en "(2)" voor de tweede test.

Als eerst zullen alle teststappen uitgevoerd worden, het is hierbij de bedoeling dat de gebruiker alle reacties geeft zoals bij een normale test zou gebeuren. Wanneer alle stappen uitgevoerd worden en de juiste vragen en resultaten weergegeven worden kan geconcludeerd worden de test bij een goed werkende machine goed zal werken. Hierna zullen nog alle andere "error" states getest moeten worden.

(1): In de deelttest "Profibus deelttest output" zijn een yes en no state omgewisseld, hierdoor wordt onterecht een error weergegeven en wordt de test onterecht beëindigd.

(2): Alle stappen werden doorlopen en alle verwachte berichten werden op het user interface weergegeven.

Om niet voor elke error eerst door alle voorgaande deelttests te moeten gaan, met de kans om een fout te maken en hierdoor overnieuw te moeten beginnen, is gekozen om de deelttests afzonderlijk te testen. Dit wordt gedaan door de startknop van de Profibustest te linken aan de specifieke test in plaats van de eerste Profibus test. Wanneer dit gewijzigd moet worden zal dit aangegeven worden, na de test zal weer terug gezet moeten worden op "testP\_HB\_active". Om complete controle te hebben op de inputs van het testsysteem zal gebruik gemaakt worden van de webtool.

Maak het heartbeat signaal laag op de webtool, hierdoor zal errorP101 ontstaan.

(1 en 2): De gewenste error wordt weergegeven.

Maak het heartbeat signaal hoog op de webtool, hierdoor zal errorP102 ontstaan.

(1 en 2): De gewenste error wordt weergegeven.

Verander gelinkte test in "testP\_DV\_active", de opdrachten op het testsysteem hoeven niet uitgevoerd te worden.

Geef het antwoord nee, hierdoor zal errorP201 weergegeven worden.


(1 en 2): De gewenste error wordt weergegeven.


Geef tweemaal het antwoord ja, hierdoor zal errorP202 weergegeven worden.

(1 en 2): De gewenste error wordt weergegeven.

Geef eerst het antwoord ja en hierna tweemaal het antwoord nee, hierdoor zal errorP203 weergegeven worden.

(1 en 2): De gewenste error wordt weergegeven.

	Document: AFST_TB Bijlage 7 testplan en resultaten v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 6 van 7

	Testsysteem Pers Communicatie	Goss Contiweb B.V. Ir. Wagterstraat 10 5831 AZ, Boxmeer
--	-------------------------------	---

De volgende deelttest wordt 6x herhaald, het geeft geen meerwaarde om deze hier 6x te herhalen. Vervang het # teken voor cijfer 1 tot en met 6 afhankelijk van welke test uitgevoerd wordt. Verander gelinkte deelttest in "testPO\_#\_active", de opdrachten op het testsysteem hoeven niet uitgevoerd te worden.

Geef het antwoord nee, hierdoor zal errorP301 ontstaan.

(1 na eerste doorloop) De error is gelijk voor elke keer dat deze deelttest uitgevoerd wordt, deze zullen onderscheiden moeten kunnen worden. Deze fout zal optreden bij elke herhaling en daarom is direct doorgedaan naar het volgende deel van deze deelttest.

(2) Door de output naam toe te voegen in de error is deze nu wel uniek voor elke herhaling.

Geef tweemaal het antwoord nee, hierdoor zal errorP302 ontstaan.

(1 na eerste doorloop) Net als bij het vorige deel van deze test zijn de errors niet uniek, daarom is na de eerste doorloop direct doorgedaan naar de volgende deelttest.

(2) Door de output naam toe te voegen in de error is deze nu wel uniek voor elke herhaling.

Zorg ervoor dat alle outputs op de webtool zijn ingesteld op laag voor de test begonnen wordt. De volgende deelttest wordt 12x herhaald, het geeft geen meerwaarde om deze hier 12x te herhalen. Vervang het # teken door het cijfer 1 tot en met 12 afhankelijk van welke test uitgevoerd wordt. Verander gelinkte deelttest in "testPI\_#\_active", de opdrachten op het testsysteem hoeven niet uitgevoerd te worden.

Druk op continue, dit zal errorP401 veroorzaken.

(1 en 2) De gewenste error wordt weergegeven.

Maak de Output op de webtool hoog, druk op continue, dit zal errorP402 veroorzaken.

(1 en 2) De gewenste error wordt weergegeven.


Zorg ervoor dat de analoge waarden op de webtool niet staan ingesteld op value1, geen van de deelttests gebruikt de waarde 0. De volgende deelttest wordt 8x herhaald, het geeft geen meerwaarde om deze hier 8x te herhalen. Vervang het # teken door het cijfer 1 tot en met 8 afhankelijk van welke test uitgevoerd wordt. Verander gelinkte deelttest in "testPA\_#\_active", de opdrachten op het testsysteem hoeven niet uitgevoerd te worden.

Druk op continue, dit zal errorP501 veroorzaken.

(1 en 2) De gewenste error wordt weergegeven.

Maak de value gelijk aan value1 (weergegeven op de user interface), druk op continue, dit zal errorP502 veroorzaken.

(1 en 2) De gewenste error wordt weergegeven.

	Document: AFST_TB Bijlage 7 testplan en resultaten v1	Auteur: Jarno Meijer
	Datum: 8-6-2017	Pagina 7 van 7