

# Studiengang Systemtechnik

## Vertiefungsrichtung Power & Control

# Diplom 2012

*Kilian Heinzmann*

*Instrumentierung einer  
hydraulischen Prüfbank*

Dozent Cécile Münch-Alligné

Experte Joseph Salamin

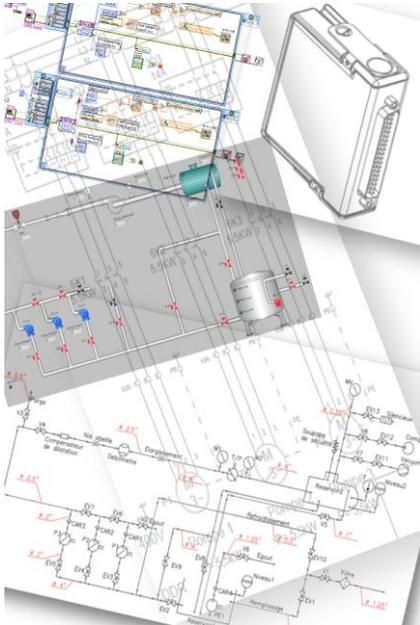
HES-HEVS-T (Sion)



EM000007488436

# Instrumentierung einer hydraulischen Prüfbank

Diplomand/in Kilian Heinzmann



## Ziel des Projekts

Ziel des Projekts ist die Planung & Dimensionierung der elektrischen Einheit / Steuerung, Erstellung des Elektroschemas, Auswahl geeigneter Automationshardware sowie die Strukturierung und Teilprogrammierung der Automationssoftware einer hydraulischen Prüfbank zur Leistungsmessung einer Mikroturbine.

## Methoden | Experimente | Resultate

Anhand von technischen Spezifikationen wie bspw. der gewünschte Messpräzision, Messgeschwindigkeit und Einsatzort, wurden die nötigen Sensoren im Bereich der Druckmessung, Durchflussmessung, Niveaumessung und Temperaturmessung ausgewählt. Die Auswahl der Automationshardware fiel aufgrund der Darstellungsmöglichkeiten und Umfang an Funktionen, auf die Hardware von National Instruments.

Es wurde ein NI cRio 9074 verwendet, auf welchem 1 analoge Eingangskarte, 1 analoge Ausgangskarte, 1 digitale Eingangskarte, 1 schnelle digitale Eingangskarte sowie 1 digitale Ausgangskarte eingesetzt wird. Das Resultat der Auswahl verschiedener Sensoren, & Automationskarten, ist ein vollständiges Elektroschema.

Der zweite Teil der Diplomarbeit ist die Grundlagenaufbereitung in Labview, der Automationssoftware von National Instruments.

Ein NI cRio kann auf 3 Ebenen, graphisch programmiert werden (FPGA - CPU - PC).

Im Rahmen der Arbeit wurde eine sinnvolle Strukturierung sowie eine Teilprogrammierung vorgenommen.

Bspw. wurde die Regelung aufgrund der Drehzahl, sowie eine Regelung über den Durchfluss implementiert. Der Algorithmus zum Bestimmen der Pumpengeschwindigkeit aufgrund von Impulsen ist ebenfalls ein Teil bereits entwickelter Software.

Diplomarbeit

| 2012 |

Studiengang  
Systemtechnik

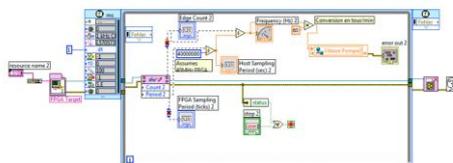
Anwendungsbereich  
Power & Control

Verantwortliche/r Dozent/in  
Dr Cecile Münch  
cecile.muench@hevs.ch

Assistenten  
Ing Meier Bastien  
Dr Hasmatuchi Vlad

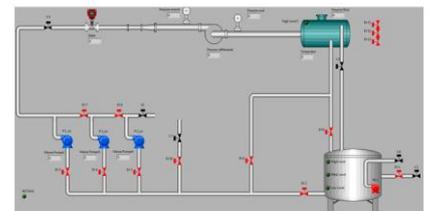
Experte  
Ing Salamin Joseph  
Hydro Exploitation

## RPM Berechnung (LABVIEW)



RPM Berechnung in Labview  
Ein Beispiel für die Berechnung von RPM auf der Grundlage von Impulsen. Diese Abbildung beinhaltet die Berechnung für jeweils 1 Geschwindigkeit

## HMI (LABVIEW)



Das HMI (Human Machine Interface) dient dem Anwender dazu, die Einstellungen vorzunehmen und den aktuellen Zustand der Maschine direkt zu erkennen.

# Diplomarbeit

## Instrumentierung einer hydraulischen Prüfbank

---

Hes-so Valais / Wallis Sion

Studiengang	: Power & Control
Student	: Kilian Heinzmann
Dozent/in	: Dr. Cécile Münch-Aligné
Ausführungsort	: HES-SO Valais
Abgabetermin	: 9 Juli 2012
mündliche Verteidigung	: September

### ABSTRACT

Die Diplomarbeit baut auf den Recherchen der Semesterarbeit auf. Ziel ist die Dimensionierung und Planung der Automatisierung sowie der elektrischen Komponenten der hydraulischen Test Bank.

## Inhaltsverzeichnis

<b>ABBILDUNGSVERZEICHNIS</b> .....	<b>3</b>
<b>ALLGEMEINE INFORMATIONEN</b> .....	<b>5</b>
<b>EINFÜHRUNG</b> .....	<b>5</b>
FUNKTIONSWEISE & AUFBAU .....	5
ZIELE.....	6
PLANING.....	6
<b>HARDWARE / ELEKTROTABLEAU</b> .....	<b>7</b>
EINFÜHRUNG .....	7
<i>Wahl geeigneter Software</i> .....	7
<i>Gewählte Beschriftung und Nummerierungen</i> .....	7
LEISTUNGSTEIL / PUISSANCE.....	8
<i>Leistungsanschluss Pumpen</i> .....	8
<i>24V Spannungsquellen, Steckdose Typ15, Spannungs-Kontrollanzeige</i> .....	9
<i>Bestimmung der Leistung der DC Versorgung</i> .....	9
<i>Elektroventile 230V</i> .....	10
<i>Entleerungspumpe, Leck Detektor &amp; Einschaltindikatoren</i> .....	11
STEUERUNG / COMMANDE.....	11
<i>Speisung SPS / Alimention CPU</i> .....	12
SPEISUNG BILDSCHIRM .....	13
1. KARTE / DIGITAL INPUTS.....	13
<i>Beschreibung der Karte</i> .....	13
<i>Eingesetzt Sensoren an NI 9425</i> .....	15
<i>Weitere Anschlüsse</i> .....	16
2.KARTE / DIGITAL OUTPUTS .....	18
<i>Beschreibung der Karte</i> .....	18
<i>Aktuatoren der analog Outputs</i> .....	19
3. KARTE / ANALOG INPUTS .....	21
<i>Beschreibung der Karte</i> .....	21
<i>Durchflussmesser</i> .....	22
<i>Drucksensoren</i> .....	23
<i>Temperaturmesssensor Reservoir 2</i> .....	24
<i>Differenzdrucksensor</i> .....	25
4. KARTE / ANALOG OUTPUTS .....	27
<i>Beschreibung der Karte</i> .....	27
5.KARTE / SCHNELLE DIGITAL INPUTS.....	29
<i>Beschreibung der Karte</i> .....	29
<i>Einweglichtschranke</i> .....	30
<b>AUTOMATION / LABVIEW</b> .....	<b>32</b>
ALLGEMEINE INFORMATIONEN .....	32
<i>Weshalb Labview?</i> .....	32
EINFÜHRUNG IN DIE PROGRAMMIERUNG MIT LABVIEW .....	33
<i>Grundlegender Aufbau von Labview</i> .....	33
<i>Variablen &amp; Datenlogging mit Citadel</i> .....	34
<i>Geschwindigkeit von Umgebungsvariablen</i> .....	38
STRUKTURIERUNG DES PROGRAMMS .....	39
INBETRIEBNAHME DES COMPACTRIO.....	41
REALISIERTE PROGRAMME.....	43
<i>Einführung</i> .....	43
<i>HMI</i> .....	43
<i>CPU</i> .....	46
<i>FPGA</i> .....	50

<b>TESTS</b> .....	<b>51</b>
<i>Test der Geschwindigkeitsmessung</i> .....	51
<i>Test der Speicherungsfunktion in eine Exceldatei</i> .....	53
<b>VERBESSERUNGEN</b> .....	<b>54</b>
<b>SCHLUSSFOLGERUNG</b> .....	<b>54</b>
<b>DANKSAGUNG</b> .....	<b>55</b>
<b>DATUM &amp; UNTERSCHRIFT</b> .....	<b>55</b>
<b>ANHÄNGE</b> .....	<b>56</b>
ELEKTROSCHEMA .....	57
LISTE DER EIN/AUSGÄNGE .....	58
SEMESTERARBEIT .....	59

## Abbildungsverzeichnis

Abbildung 1 Übersicht Testbank .....	5
Abbildung 2 Beschriftungsbeispiel Schema .....	7
Abbildung 3 Beschriftungsbeispiel Klemme .....	8
Abbildung 4 Leistungsteil / Pumpenanschluss .....	8
Abbildung 5 Leistungsteil / 24 Spannungsversorgung, Steckdose, Spannungsind. ....	9
Abbildung 6 Servomotor Belimo .....	10
Abbildung 7 Leistungsteil / Anschluss Elektroventil .....	10
Abbildung 8 Leistungsteil / Entleerungspumpe, Lecksensor, Betr. Status Indikator Pumpen .....	11
Abbildung 9 Steuerung / Pumpensteuerung .....	12
Abbildung 10 Steuerung / Speisung SPS .....	12
Abbildung 11 Steuerung / Speisung Bildschirm .....	13
Abbildung 12 Karte 1 / NI 9425 .....	13
Abbildung 13 Karte 1 / Anschlussbeschreibung .....	14
Abbildung 14 Karte 1 / Schema - angeschlossene Elemente .....	15
Abbildung 15 Karte 1 / Vibrationsgrenzschalter .....	15
Abbildung 16 Karte 1 / Anschluss Vibrationsgrenzschalter .....	16
Abbildung 17 Karte 1 / Schemaausschnitt Vibrationsgrenzschalter .....	16
Abbildung 18 Karte 1 / Einlesung der 100% Daten Elektroventil .....	17
Abbildung 19 Karte 1 Einlesung des Leakage -Signals .....	17
Abbildung 20 Karte 2 / NI 9477 .....	18
Abbildung 21 Karte 2 / Anschlussbeschreibung .....	18
Abbildung 22 Karte 2 / Schema - angeschlossene Elemente .....	19
Abbildung 23 Karte 2 / indirekte Ansteuerung Elektroventile .....	19
Abbildung 24 Karte 2 / man.-autom. Entleerung .....	20
Abbildung 25 Karte 3 / NI 9203 .....	21
Abbildung 26 Karte 3 / interner Schaltungsaufbau Karte .....	21
Abbildung 27 Karte 3 / Schema - angeschlossene Elemente .....	22
Abbildung 28 Karte 3 / Schema - Anschluss Durchflussmesser .....	23
Abbildung 29 Karte 3 / Drucksensor Cerabar PMC 131 .....	23
Abbildung 30 Karte 3 / Anschluss Drucksensor .....	24
Abbildung 31 Karte 3 / Temperatursensor .....	25
Abbildung 32 Karte 3 / Anschluss Temperatursensor .....	25
Abbildung 33 Karte 3 / Differenzdruckmesser Deltabar PMD55 .....	26
Abbildung 34 Karte 3 / Anschluss Differenzdruckmesser .....	26
Abbildung 35 Karte 4 / NI 9265 .....	27

Abbildung 36 Karte 4/ interner Aufbau Karte .....	27
Abbildung 37 Karte 4 / Anschluss Geschwindigkeitssteuerung .....	28
Abbildung 38 Karte 4 / ON-OFF Schaltung an Pumpenklemme.....	28
Abbildung 39 Karte 5 / NI 9411 .....	29
Abbildung 40 Karte 5 / Beschaltung Karte .....	29
Abbildung 41 Karte 5 / Einsatzort Geschwindigkeitssensor.....	30
Abbildung 42 Karte 5 / Geschwindigkeitssensor V180-2 Photoelectric .....	30
Abbildung 43 Karte 5 / Anschlussbeschreibung VSE 180-2 .....	31
Abbildung 44 Karte 5 / Schema - Anschluss Geschwindigkeitssensor .....	31
Abbildung 45 Bsp. Frontpanel.....	33
Abbildung 46 Bsp. Blockdiagramm .....	33
Abbildung 47 Projektauswahl.....	34
Abbildung 48 Projektexplorer .....	34
Abbildung 49 lokale Variable erstellen .....	35
Abbildung 50 Variablen & Library.....	35
Abbildung 51 Optionen Variablen .....	36
Abbildung 52 Citadel Speicherposition .....	36
Abbildung 53 Citadel Speichername Datenbank .....	37
Abbildung 54 Dateien von Citadel.....	37
Abbildung 55 hinzufügen von Variablen-Datenbank zu Citadel Datenbank .....	37
Abbildung 56 Protokollierung Variablen .....	38
Abbildung 57 Auslesen der Citadel - Protokollierungen.....	38
Abbildung 58 programmatischer Zugriff (Variablen) .....	39
Abbildung 59 Ordnerstruktur.....	39
Abbildung 60 Programmstruktur .....	40
Abbildung 61 Fehlermeldung cRIO.....	41
Abbildung 62 Einbindung cRio in Projekt.....	41
Abbildung 63 Hinzufügen des cRio zu Projekt.....	42
Abbildung 64 Auswahl - Scan o. FPGA Interface .....	42
Abbildung 65 Erstellung von Testkabeln.....	43
Abbildung 66 HMI Frontpanel .....	44
Abbildung 67 HMI Blockdiagramm.....	44
Abbildung 68 Regelungswahlschalter .....	45
Abbildung 69 Datumserstellung .....	45
Abbildung 70 Eingabeaufforderung .....	45
Abbildung 71 Erstellung Header .....	46
Abbildung 72 Speicherung von Daten .....	46
Abbildung 73 CPU Frontpanel .....	47
Abbildung 74 CPU Umgebungsvariablen bearbeiten .....	47
Abbildung 75 CPU Durchschnittsberechnung.....	48
Abbildung 76 Regelung.....	49
Abbildung 77 Geschwindigkeitssignalumwandlung .....	50
Abbildung 78 FPGA Frontpanel .....	50
Abbildung 79 FPGA Blockdiagramm .....	51
Abbildung 80 FPGA Funktionskontrolle .....	52
Abbildung 81 Test Geschwindigkeitsmessung 10Hz .....	52
Abbildung 82 Test Fontpanel .....	52
Abbildung 83 Test Geschwindigkeitsmessung 2.105 Mhz.....	53
Abbildung 84 Speicherung in Excel-File .....	53
Abbildung 85 Frontpanel - Dateipfad .....	54
Abbildung 86 beschriebenes Excel-File.....	54

## Allgemeine Informationen

Auftraggeber	: HES-SO Valais / Wallis
Ausführungsort	: HES-SO Valais / Wallis (Sion / Sitten)
Zeitraum	: 14 Mai 2012 – 9 Juli 2012
Studienrichtung	: FSI (filière system industriel)
Dozent	: Cécile Münch-Alligné
Experte	: Salamin Joseph
Assistent	: Bastien Meier , Hasmatuchi Vlad
No. DA	: pc/2012/38
Student	: Heinzmann Kilian

keine Vertrauliche Arbeit

## Einführung

### Funktionsweise & Aufbau

---

Der Testbank wird bei der HES-SO // Valais – Wallis in Sitten aufgebaut. Die Anlage erstreckt sich über 2 Etagen.

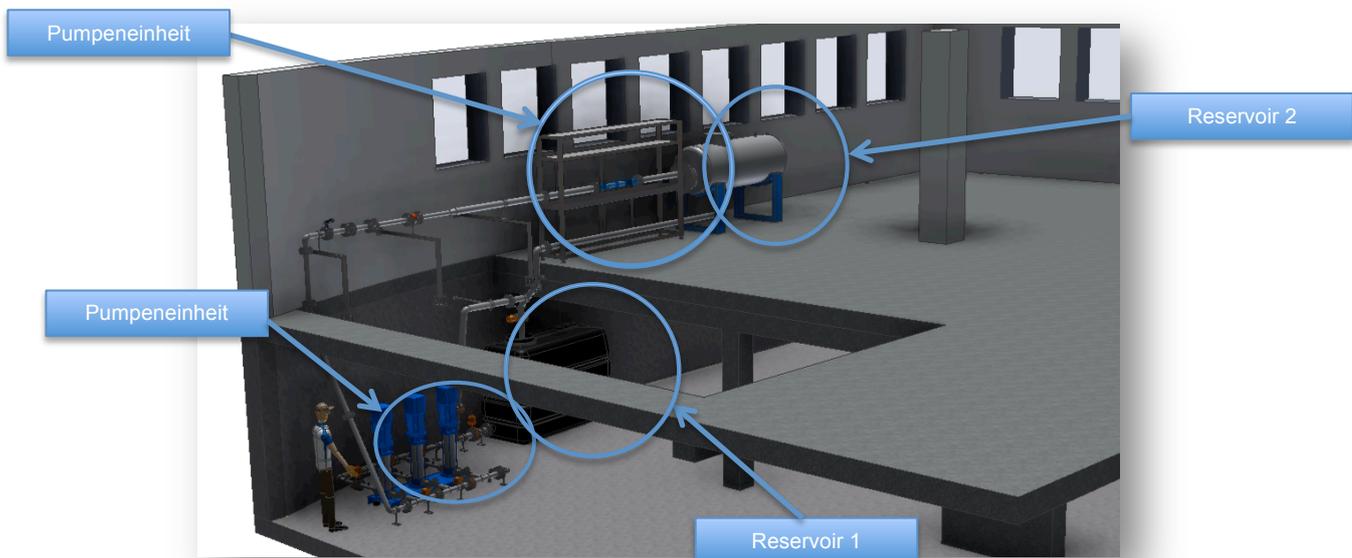


Abbildung 1 Übersicht Testbank



## Hardware / Elektrotabelleau

### Einführung

Das erhaltene Elektroschema enthielt die elektrische Steuerung des alten Test Banks. Der alte Test Bank wurde nur durch Schaltlogik automatisiert. Das Gehäuse der Installation sollte wiederverwendet werden.

Ziel dieser Neuentwicklung war, die Automatisierung durch einen Automaten. Aufgrund dieser Anforderung, reichte es nicht aus, die aktuelle Steuerung zu überarbeiten, sondern forderte eine grundlegende Neuentwicklung.

In den folgenden Abschnitten wird das Schema präsentiert und die nötige Entwicklung hierfür angefügt.

Die Entwicklung des Schemas erforderte mehrere Iterationen an Verbesserungen um einen sehr guten Prototyp zu garantieren. Dies bedeutete für die Zeichnung des Schemas, dass regelmässig alles überarbeitet und verbessert wurde und deutlich mehr Zeit eingesetzt wurde, als das reine Zeichnen erfordern würde.

Das gesamte Elektroschema inkl. Ein / Ausgangsliste ist im Anhand zu finden.

### Wahl geeigneter Software

Für die Realisierung des Elektroschemas standen folgende 3 Programme zur Debatte:

- PSpice
- Autocad Electrical
- Microsoft Visio

PSpice ist ein Programm welches für das Zeichnen und Entwickeln von Elektronikschaltung konzipiert wurde und bietet nicht so viele Möglichkeiten, ein traditionelles Elektroschema zu erstellen.

Das nächste Programm, Autocad Electrical, ist wohl das beste Programm dieser Auswahl für einen Elektroplaner. Es bietet alle Möglichkeiten ein Elektroschema zu erstellen. Es ist äusserst umfassend und enthält ebenfalls eine Simulationsengine. Aufgrund des riesigen Funktionsumfangs welcher das Programm bietet, ist eine gewisse Einarbeitungszeit nötig.

Microsoft Visio wird häufig für Schemas verwendet, da es bedeutend günstiger ist, als die speziellen Programme für die Entwicklung von Elektroschemas. Visio ist auch aufgrund der Verwandtschaft mit den anderen Office Tools wie Excel & Word, relativ schnell und intuitiv zu bearbeiten. Viele Bauteile sind schon vorhanden, einige wurden in der Schule erstellt.

### Gewählte Beschriftung und Nummerierungen

Sämtliche Bauteile werden einheitlich beschriftet.

Die Bauteile werden angefangen mit der Seitenzahl, typischen Kennbuchstaben und anschliessender auf eine Seite begrenzte Nummerierung beschriftet.

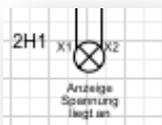


Abbildung 2 Beschriftungsbeispiel Schema

Dieses Element befindet sich dementsprechend auf der zweiten Seite und es handelt sich um eine Meldeeinrichtung, in diesem Fall eine Lampe. Es ist die erste Lampe auf dieser Seite.

Die Nummerierung der Klemme ist ebenfalls einheitlich und ist folgendermassen zu interpretieren:



Abbildung 3 Beschriftungsbeispiel Klemme

Diese Klemme befindet sich auf Seite 10, Zeile 5. Die ":1" gibt die Nummerierung der Klemmen dieses Bauteils an.

Es wird jeweils die erste Klemme mit der vollständigen Bezeichnung angegeben, die folgenden Klemmen eines Bauteils erhalten lediglich die weitere Nummerierung. Dies erhält die Übersichtlichkeit sowie einen maximalen Informationsgehalt.

## Leistungsteil / Puissance

### Leistungsanschluss Pumpen

Der Leistungsteil beinhaltet den Anschluss der Hauptaktoren unserer Testbank, die 3 5,5 kW Pumpen.

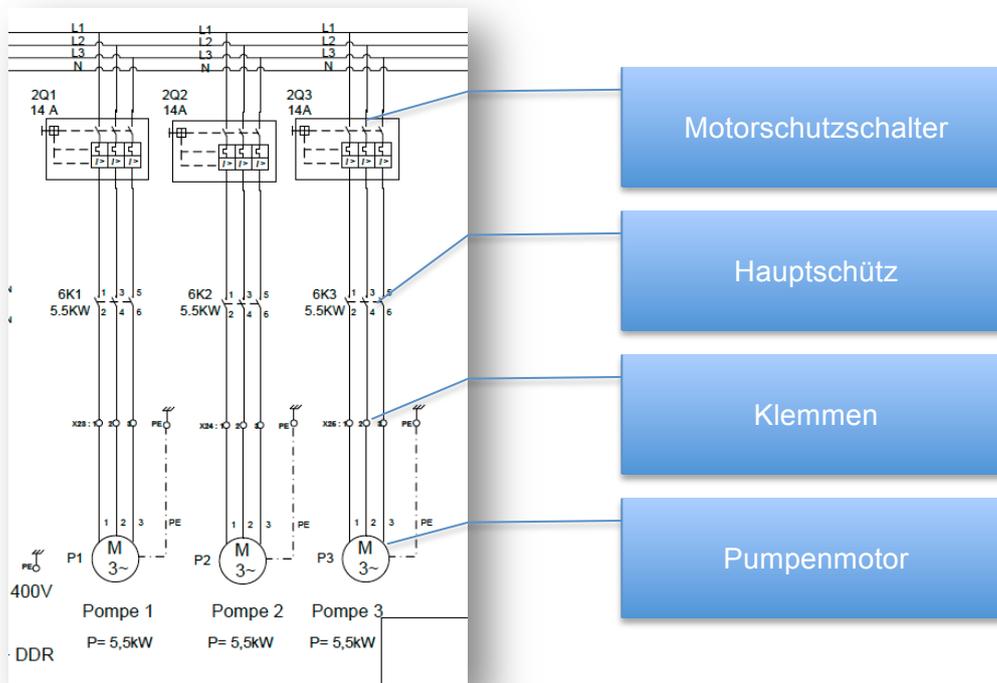


Abbildung 4 Leistungsteil / Pumpenanschluss

Die 3 Pumpen werden wie gewöhnliche Motoren angeschlossen. Sie werden durch eine Motorschutzsicherung geschützt und können durch die Hauptschütze, jeweils spannungsfrei geschaltet werden. Die Pumpen besitzen jedoch eine weitaus

grössere Anschlussvielfalt als den Anschluss der Leistung. Die Pumpen besitzen zudem einen Eingang zur Geschwindigkeitsregelung (weitere Informationen im Kapitel Analoge Outputs) sowie einen Anschluss um die Pumpe auf elektronischem Wege Ein, bzw. Auszuschalten (weitere Informationen im Kap. Digitale Outputs).

## 24V Spannungsquellen, Steckdose Typ15, Spannungs-Kontrollanzeige

Um den Automaten sowie verschiedene Sensoren und Karten anzuschliessen wurden 2 separate 24V Spannungsquellen eingesetzt.

Die Separation der Spannungsquellen dient der Trennung des Automaten von den herkömmlichen 24V Geräten.

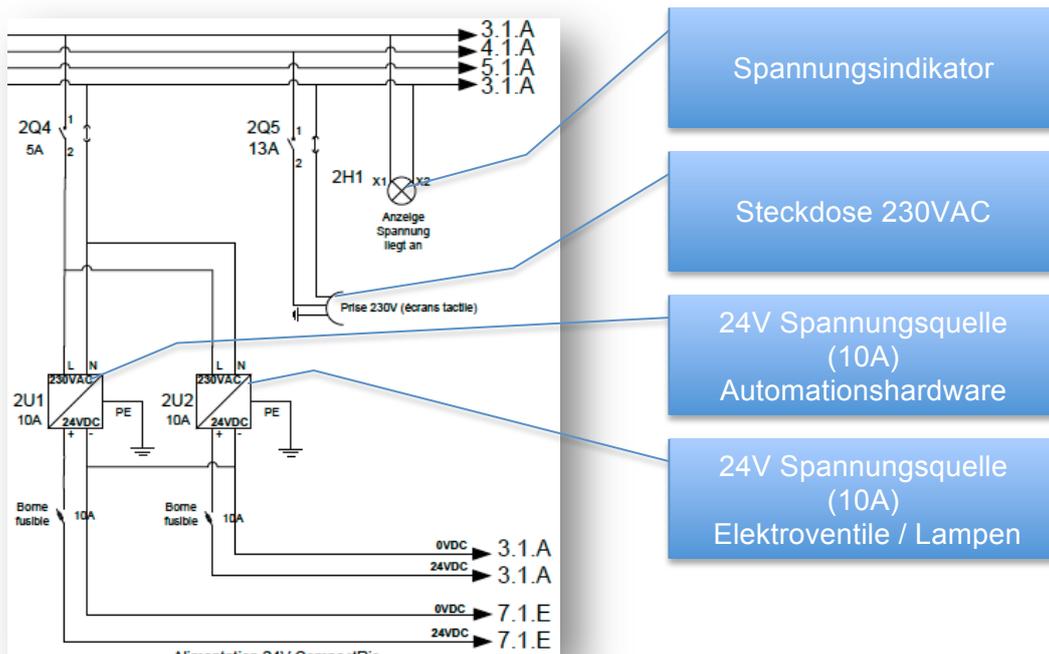


Abbildung 5 Leistungsteil / 24 Spannungsversorgung, Steckdose, Spannungsind.

Die Steckdose wurde für den Touch-Screen vorgesehen, welches ein handelsüblicher Bildschirm mit Touch-Funktion sein wird.

Ein entsprechender Bildschirm von National Instruments wäre kleiner und bedeutend teurer geworden als die aktuelle Wahl.

Der Spannungsindikator ist lediglich ein Element dass den Benutzer informiert ob die Hauptspannung anliegt.

## Bestimmung der Leistung der DC Versorgung

Um den maximalen Leistungsbezug durch die Verbraucher zu ermitteln, wurde näherungsweise die angegebene Leistung der Elemente zusammengezählt. Dieser Wert stellt das absolute Maximum dar, da nie alle Elemente gleichzeitig funktionieren werden.

Leistungen der verwendeten Elemente:

Glühlampe	: 1.2W * 16	= 19.2W
Elektroventil	: 20W * 5	= 100W
Jola (Auswertelektronik, Leckage-Sensor)	: 3W * 1	= 1W
Relais	: 5W * 13	= 65W
Liquifant Sensor	: <=1W * 4	= 4W

Druck / Temp/Geschwindigkeits Sensoren :  $5W * 7 = 35W$

Gesamt : 224.2 W

Zur Verfügung stehende Leistung :  $24V * 20A = 480W$

Anhand dieser Übersichtsrechnung ist mit keinen Problemen zu rechnen.

### Elektroventile 230V

Die Elektroventile werden von einem Servomotor der Marke Belimo angetrieben.

Die Spezifikationen dieses Motors :

- Rotationsmoment : 35Nm
- Spannung : 230VAC
- Leistungsaufnahme : 10W (Nominalmoment)



Abbildung 6 Servomotor Belimo

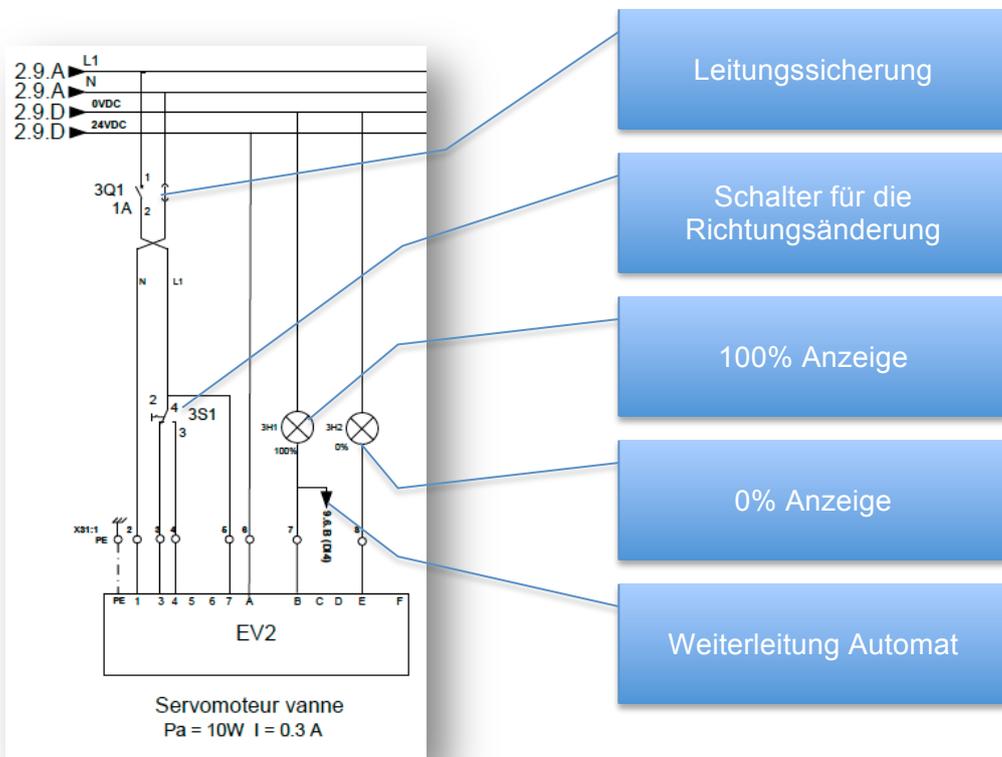


Abbildung 7 Leistungsteil / Anschluss Elektroventil

Der Belimo Servomotor wird jeweils durch eine Leitungssicherung geschützt. Diese Servomotoren werden von Hand geschaltet, da eine automatische Steuerung nicht verwendet wird. Um jedoch trotzdem ein Feedback auf der Visualisierung darzustellen, wird eine Weiterleitung der 100% Anzeige ebenfalls an den Automaten weitergegeben. Um dieses Signal vom Automat verarbeiten zu können, wird eine Spannung von 24V für die Leuchten verwendet.

## Entleerungspumpe, Leck Detektor & Einschaltindikatoren

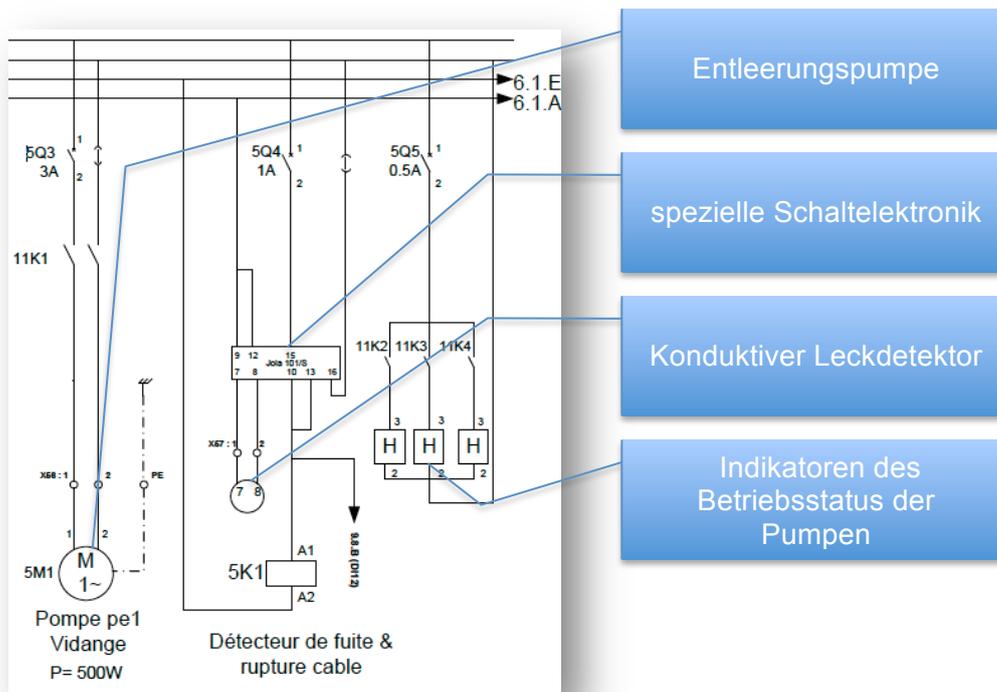


Abbildung 8 Leistungsteil / Entleerungspumpe, Lecksensor, Betr. Status Indikator Pumpen

Ebenfalls mit 230VAC angeschlossen, wird die Pumpe welche für das Entleeren zuständig ist. Die Pumpe der Steuerung befindet sich auf der Seite 11, digitale Outputs.

Der Leckage-Sensor wird durch eine spezielle Steuerung (Jola 101/S) gesteuert.

Die Auswertelektronik kann ebenfalls einen Kabelbruch detektieren. Die Detektion eines Kabelbruchs ist sehr wichtig und wird deshalb mit dem Alarmsignal eines Lecks gekoppelt.

Ein direktes Feedback des Betriebsstatus der Pumpen wird durch die 3 Meldeleuchten H1-H3 angezeigt. Diese Anzeige funktioniert rein elektrisch, um eine direkte Anzeige auch bei Ausfall der Automationshardware garantieren zu können. Die Steuerung wird der Ein / Aus-Signale erfolgt durch die digitalen Outputs der automatischen Steuerung.

## Steuerung / Commande

Die Beschriftung, Dokumentierung des Schemas hier im Bericht, wurde aufgrund der Übersichtlichkeit, auf einen Strang beschränkt. Die beiden folgenden Stränge sind identisch aufgebaut und schalten jeweils die anderen 2 Pumpen.

Der Einschaltbefehl für die Pumpen wird durch die Taste 6S1 gegeben. Das Relais 6K1 geht dabei in Selbsthaltung bis der Notstop, der Aus-Taster 6S4 oder die Motorschutzsicherung aktiviert wird.

Für die Kontrolle wird auch hier eine Lampe als Spannungsindikator eingesetzt.

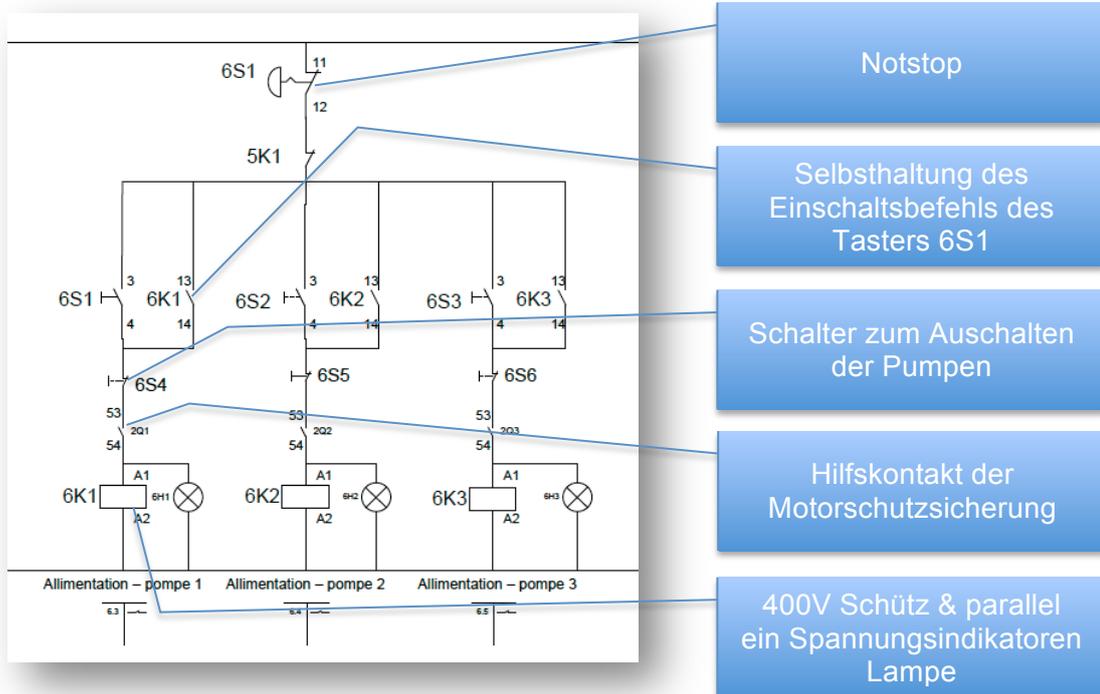


Abbildung 9 Steuerung / Pumpensteuerung

Unterhalb der Schütze werden die jeweils verwendeten Kontakte des Schützes notiert.

### Speisung SPS / Alimentation CPU

Der cRio 9074 benötigt einen Gleichspannungsanschluss mit 24V. Dieser wird durch eine separat geführte 24V sichergestellt. Die Spannungsquelle des cRio's versorgt ebenfalls die Karten des Automats. Der Einsatz von 2 separaten 24V Spannungsquellen sichert die Spannungsversorgung der Steuerung.

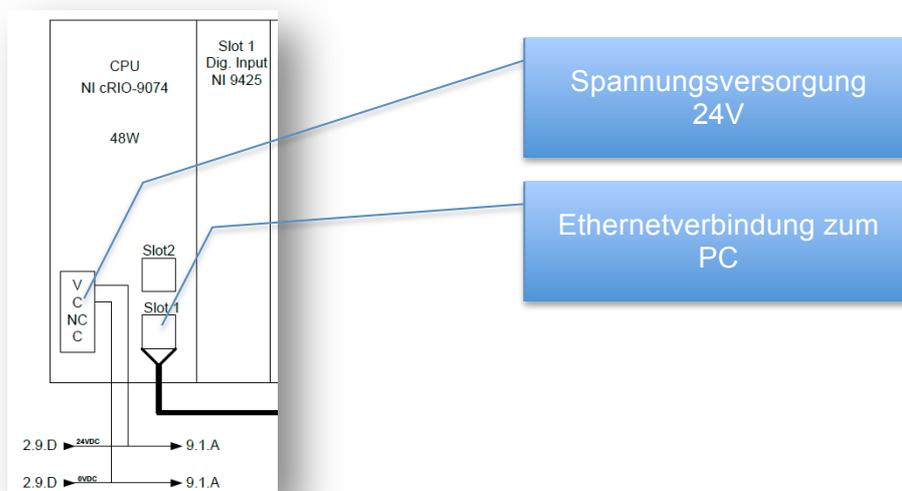


Abbildung 10 Steuerung / Speisung SPS

## Speisung Bildschirm

Anstelle eines Touch-Screens von National Instruments, welche im Prinzip schon einen kleinen PC darstellen, wurde ein normaler, handelsüblicher Touch-Screen ausgewählt. Ein spezifischer Touch-Screen von NI ist für die benötigten Funktionen überdimensioniert. Der ausgewählte Bildschirm muss mit 230V angeschlossen werden, eine geeignete Steckdose wurde im Schema auf Seite 2 bereits vorgesehen.

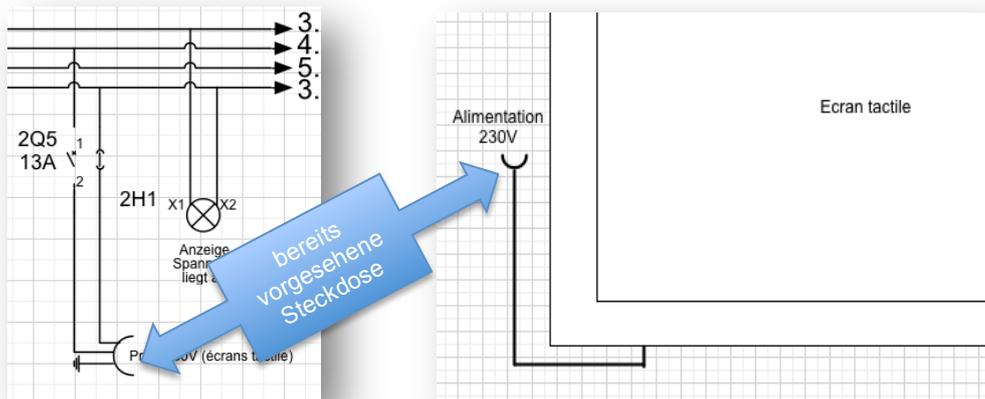
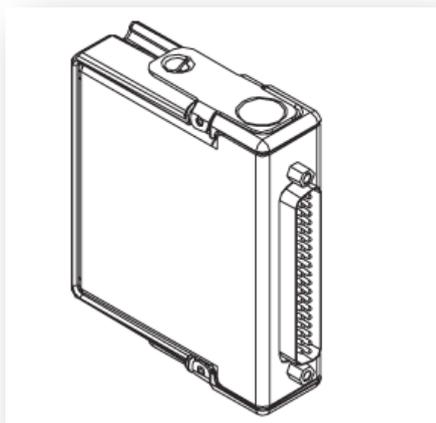


Abbildung 11 Steuerung / Speisung Bildschirm

## 1. Karte / Digital Inputs

### Beschreibung der Karte

Auf Seite 9 des Schemas wird die digitale Inputkarte NI 9425 dargestellt. Die Karte bietet 32 digitale Inputs.



NI 9425 mit D-Sub Connector

Abbildung 12 Karte 1 / NI 9425

Der D-Sub Anschluss, bietet die Möglichkeit die Karte ohne weiteres in sehr kurzer Zeit tauschen zu können, falls ein Defekt auftreten würde.

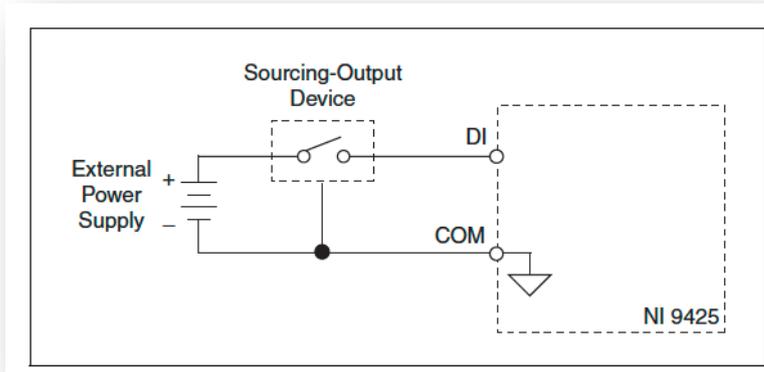


Abbildung 13 Karte 1 / Anschlussbeschreibung

Der Anschluss der Karte wird auf obigem Bild dargestellt. Ein Element welches ein "EIN/AUS" Signal generiert wird durch eine externe Spannungsquelle angeschlossen. Sobald das Element schaltet erhält die Karte ein logisches EIN. Intern wird das Signal nach der Analyse, zum COM aller Eingänge geschaltet was einen Stromfluss zulässt.

Grundlegende Eigenschaften der Karte:

- 32 digitale Eingänge
- Eingangsimpedanz von 30k Ohm
- Überspannungsschutz von 30VDC pro Kanal
- 1 $\mu$ s Reaktionszeit ("Setuptime")
- Update / Übertragungsrate von 0.007ms (ca. 140kHz)

Aufgrund der grossen Anzahl Eingänge eignet sich diese Karte sehr gut. Sie wird hauptsächlich zur Erfassung von Ventilstellung und Füllständen benötigt. Dies sind relativ langsame Elemente bei welchen die Updaterate von 140kHz reicht.

Die NI 9425 Karte hat noch einige Reserven bzgl. Ihrer Anzahl Eingänge. Aktuell werden 4 Füllstandmelder angeschlossen. Die Füllstandmelder dienen der Überwachung der Reservoirs. Das grosse Reservoir im Untergeschoss erhält 3 Messsensoren wobei das obere Reservoir nur einen Messsensor erhält um die oberste Position zu erfassen.

Eingesetzt Sensoren an NI 9425

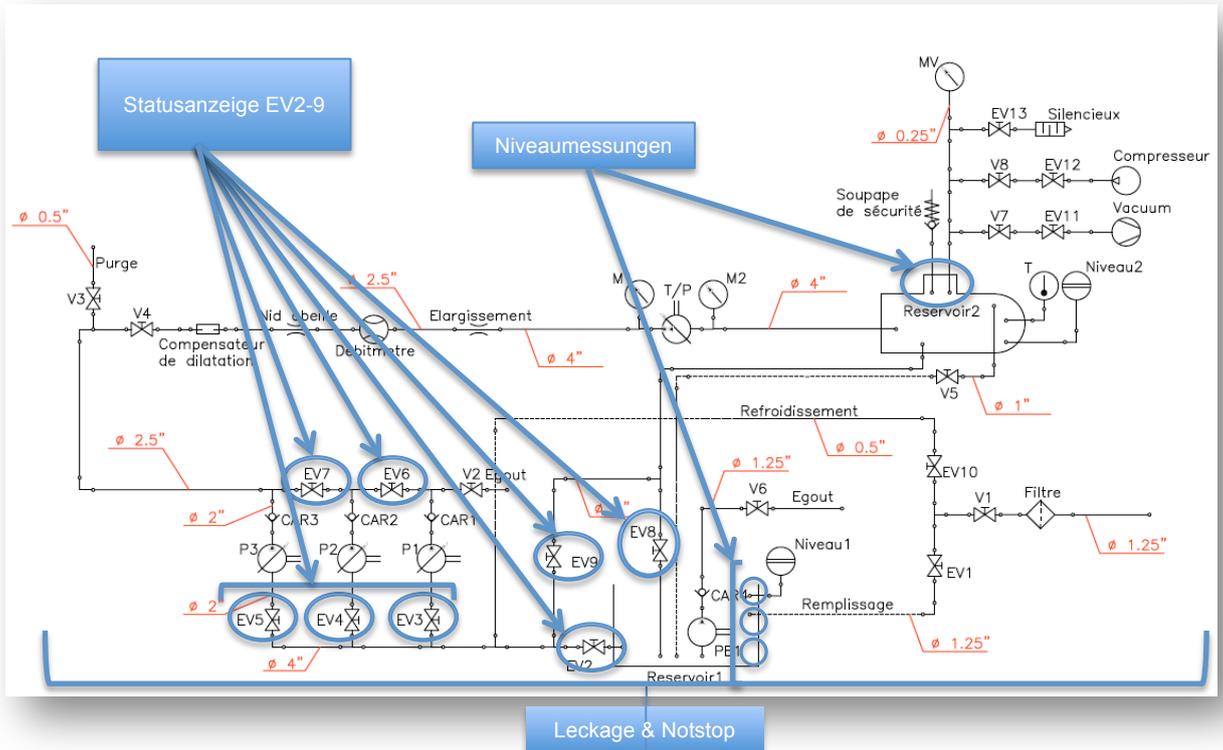


Abbildung 14 Karte 1 / Schema - angeschlossene Elemente

Für die Detektion des Füllstands werden eine Vibrationsgrenzschalter für Flüssigkeiten eingesetzt.

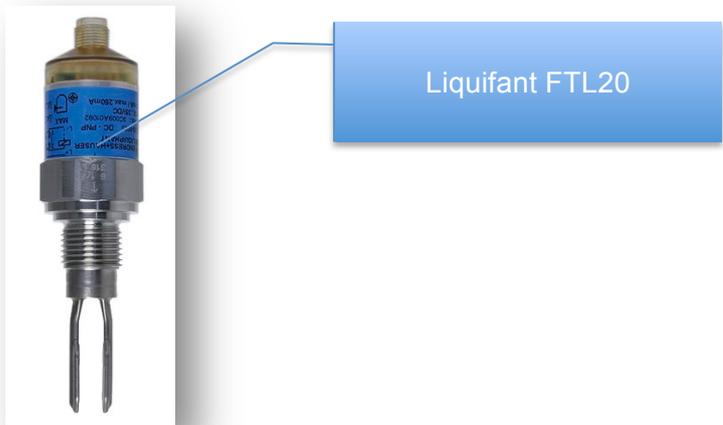


Abbildung 15 Karte 1 / Vibrationsgrenzschalter

Diese Sensoren vibrieren mit einer gewissen Frequenz. Werden die Gabeln in eine Flüssigkeit getaucht verändert sich diese Schwingfrequenz. Die Veränderung der schwingenden Frequenz wird durch die auswertende Elektronik erkannt und das Ausgangssignal entsprechend angepasst. Aufgrund der Tatsache, dass der Sensor in Reservoir 2 verwendet wird, musste auf den Prozessdruck geachtet werden. Dieser Sensor kann bei einem maximalen Prozessdruck von 40bar arbeiten.

Bei der Versorgungsspannung der Sensoren wurde allgemein auf die Verwendung mit 24VDC geachtet.  
 Der Anschluss dieses Sensors wird mit Hilfe eines M12-Steckers bewerkstelligt.

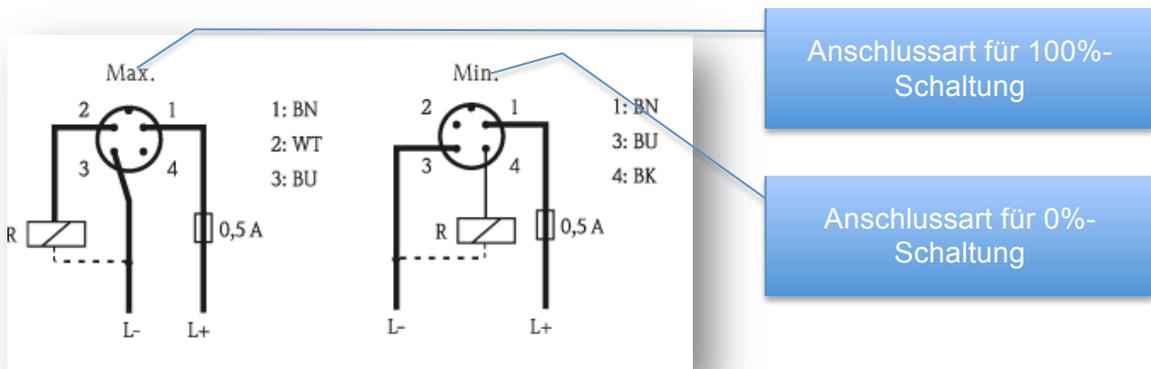


Abbildung 16 Karte 1 / Anschluss Vibrationsgrenzschalter

Um eine gewisse Einheit bei der Verkabelung zu erreichen, wird ausschliesslich mit der Max. Variante verkabelt. Dass alle Sensoren des gleichen Typs auf dieselbe Weise verdrahtet werden, bietet den Vorteil der schnellen und intuitiven Verkabelung / Tausch der Sensoren.

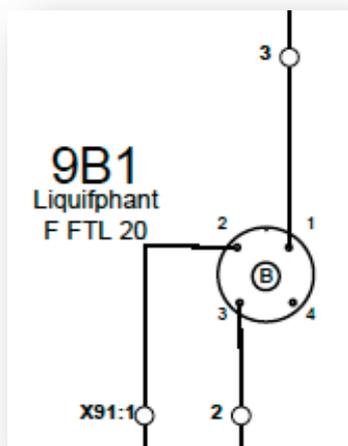


Abbildung 17 Karte 1 / Schemaausschnitt Vibrationsgrenzschalter

Die Bezeichnung "B" des Sensors bezieht sich auf die Betriebsmittelbezeichnung für Elemente welche nicht-elektrische Grössen in elektrische Grössen umwandeln.

### Weitere Anschlüsse

Sämtliche Elektroventile (Schema S.3-S.5) besitzen eine Positionsmeldung. Die 100% Position wird an die digitalen Inputs dieser Karte verbunden um auf der Visualisierung der Zustand der Ventile einzubinden was die Kontrolle der Anlage unterstützt. Die Anbindung dieser Elektroventile an das Automationssystem bzw. der Visualisierung kam erst in einer späteren Iteration als Verbesserungsmaßnahme hinzu.

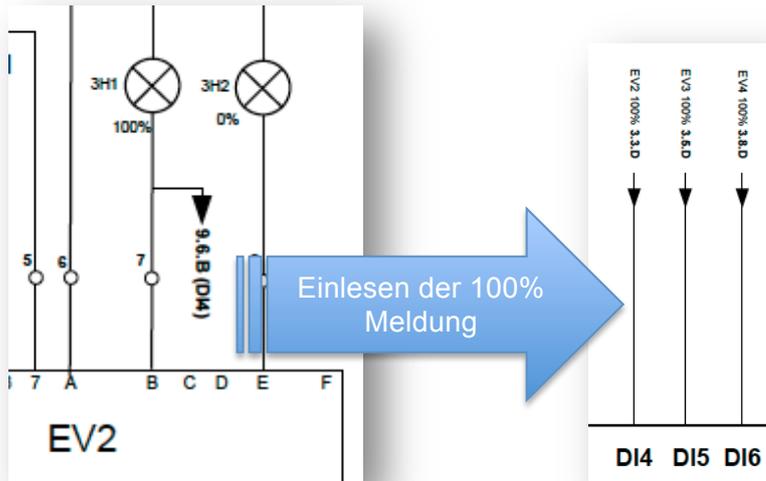


Abbildung 18 Karte 1 / Einlesung der 100% Daten Elektroventil

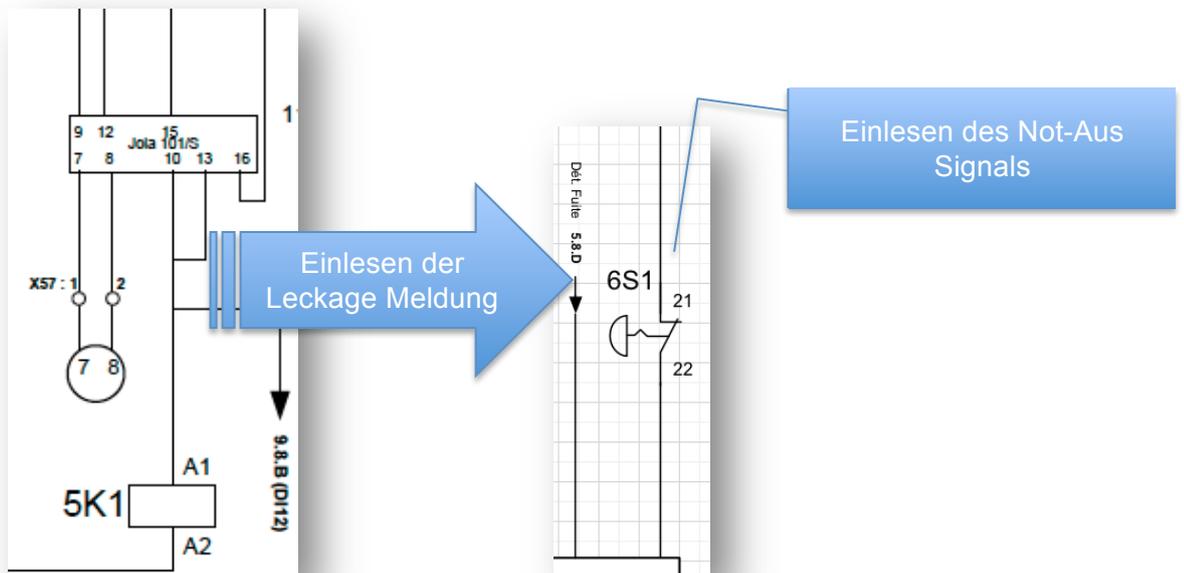


Abbildung 19 Karte 1 Einlesung des Leckage -Signals

Selbige Verbindung würde ebenfalls für die Alarmmeldung des Leckage-Sensors sowie des Not-Aus Schalters durchgeführt.

## 2.Karte / Digital Outputs

### Beschreibung der Karte

Die digitale Outputkarte NI 9477 befindet sich auf Seite 10 & 11 des Schemas. Die digitale Outputkarte dient zum Ein / Ausschalten von Aktuatoren. In unserem Fall werden die 24V Elektroventile durch ein Relais durch die digitalen Ausgänge geschaltet. Weiter sind die Steuerung der Entleerungspumpe sowie der EIN/AUS Kontakt der Pumpenmotoren, durch diese Karte gesteuert.

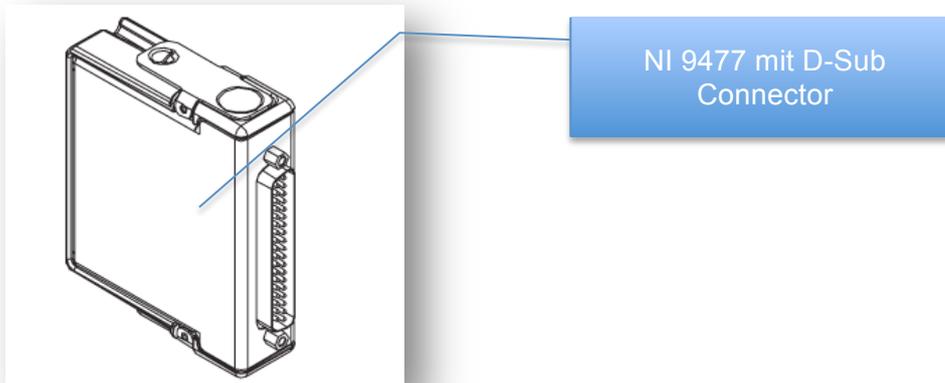


Abbildung 20 Karte 2 / NI 9477

Wie bei den digitalen Inputs wurde ein D-Sub Connector ausgewählt welcher einen schnellen Austausch der Karte zulässt, da nicht die einzelnen Drähte gelöst werden müssen.

Die NI 9477 bietet 32 digitale Ausgänge und bietet somit noch eine Reserve von 23 Outputs.

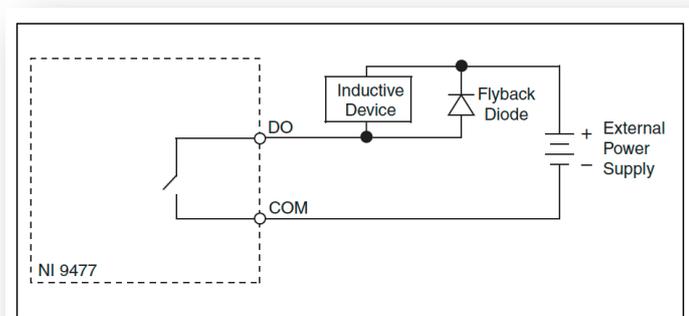


Abbildung 21 Karte 2 / Anschlussbeschreibung

Wie im obigen Bild angezeigt, ist die Funktion der digitalen Ausgänge das Schliessen eines Stromkreises. Der digitale Ausgang gibt keine Energie ab, wie man vom Namen ableiten könnte.

Grundlegende Eigenschaften der Karte :

- 32 digitale Eingänge
- 1A Ausgangsstrom pro Kanal
- Updaterate von 8us (125kHz)
- Ausbreitungsverzögerung max. 1us

### Aktuatoren der analog Outputs

Die digitalen Ausgänge steuern alle 24 V Elektroventile.

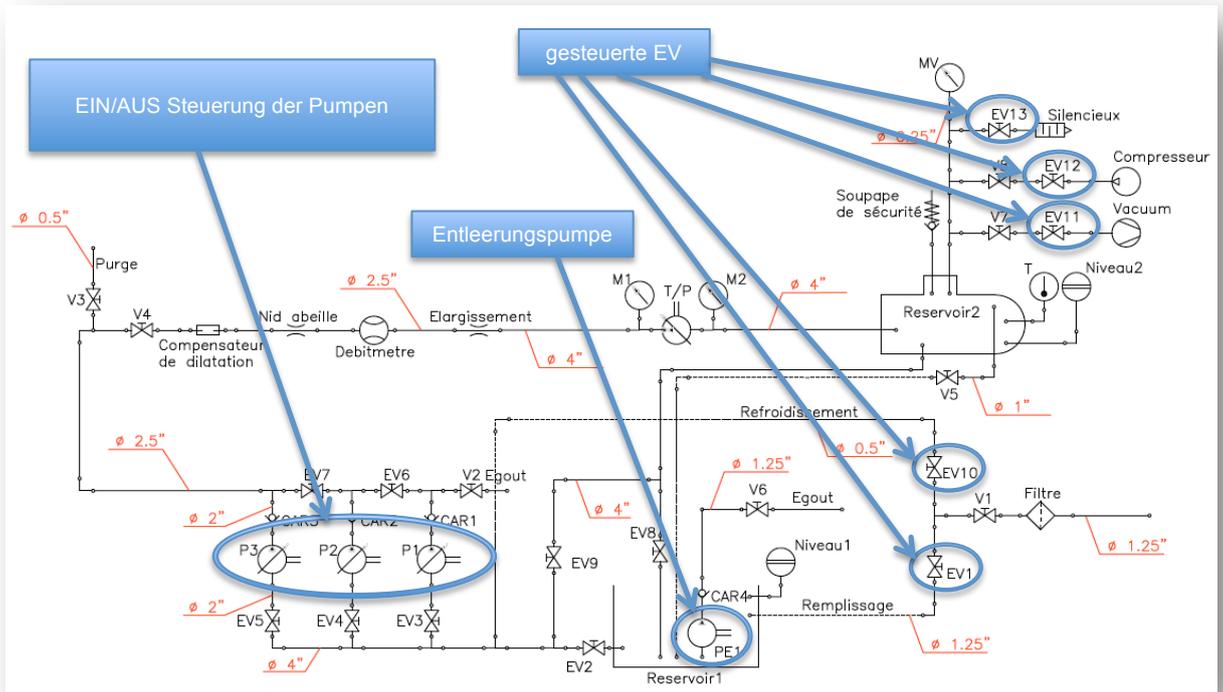


Abbildung 22 Karte 2 / Schema - angeschlossene Elemente

Da die Elektroventile 10W Leistung benötigen, was bereits die Hälfte der maximalen Kapazität auf 20 Kanälen ergibt, wurde der Umweg über kleine 24V Schütze bevorzugt.

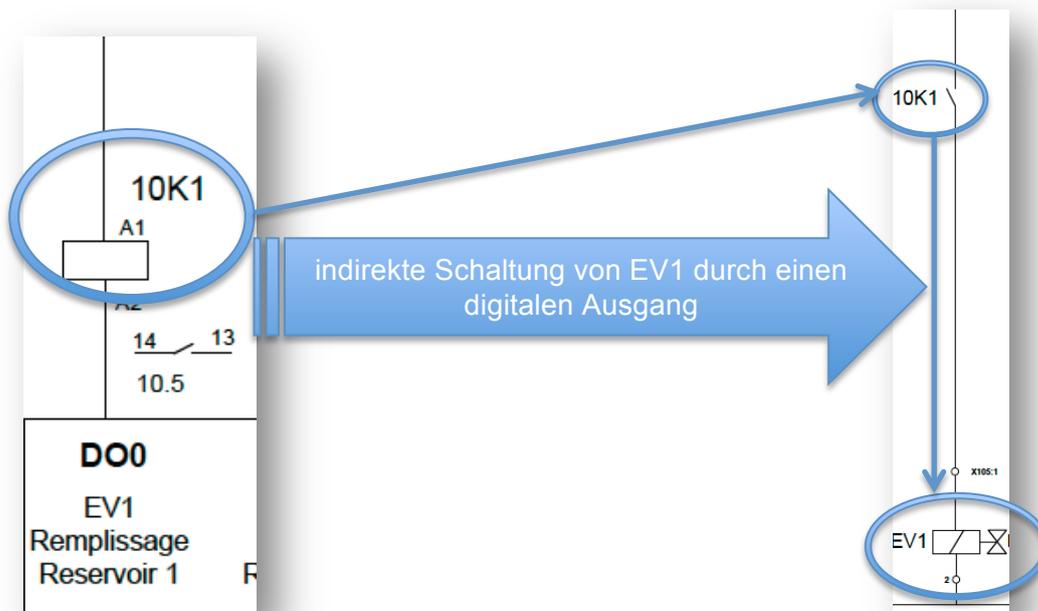


Abbildung 23 Karte 2 / indirekte Ansteuerung Elektroventile

Die obigen 2 Beispiele zeigen diese Schaltungsfunktion wie sie im Schema auf S.10 ausgeführt ist. Der digitale Output schaltet das Relais 10K1, welches anschliessend die Leistung auf das Elektroventil überträgt.

Die digitalen Outputs sorgen ebenfalls für die Schaltung der Entleerungspumpe im Reservoir 1.

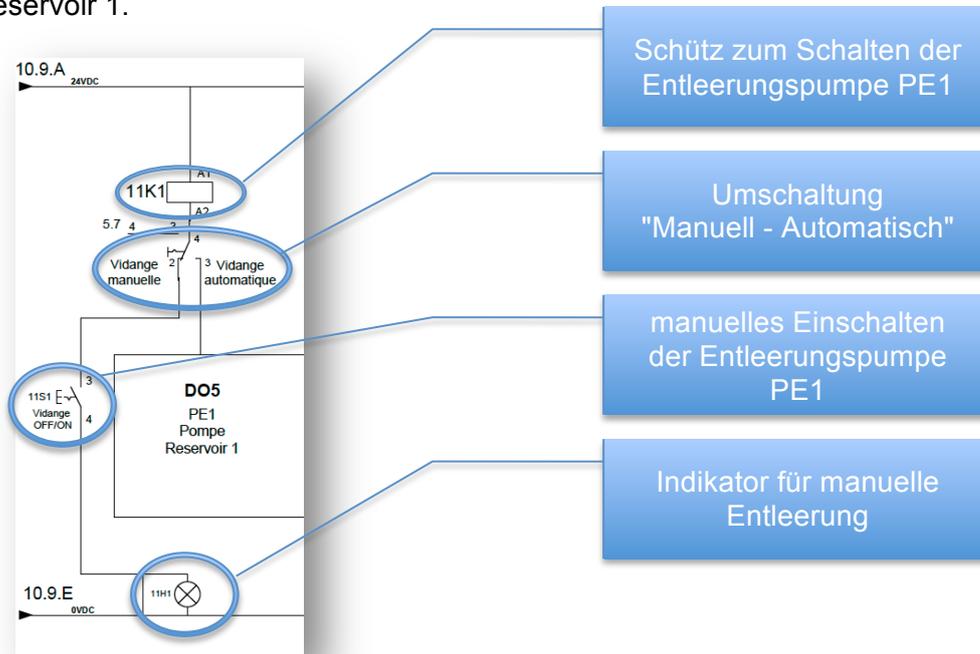


Abbildung 24 Karte 2 / man.-autom. Entleerung

Wie auf dem Bild ersichtlich kann die Entleerung durch die Pumpe auch manuell eingeleitet werden, was im Reinigungsfall oder im Störfall der Automationshardware genutzt werden kann. Die Realisierung der Umschaltung wird durch den Schalter 11S2 realisiert. Die Pumpe kann anschliessend durch den rastenden Schalter 11S1 aktiviert werden. Die Signallampe 11H1 zeigt den Betrieb im manuellen Modus an.

### 3. Karte / Analog Inputs

#### Beschreibung der Karte

Die analoge Eingangskarte NI 9203 dient zum Einlesen von Werten. Im Gegensatz zu digitalen Karten, welche nur "EIN/AUS" lesen können, bietet die analoge Karte die Möglichkeit Signale von 4-20mA auszuwerten.

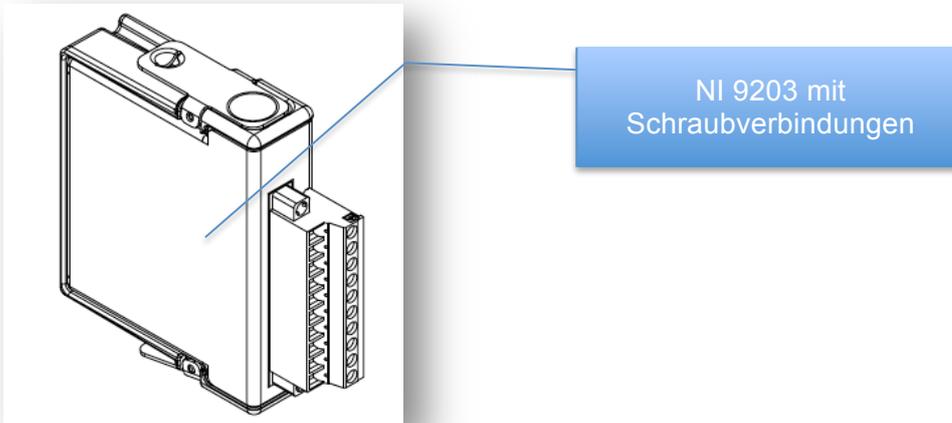


Abbildung 25 Karte 3 / NI 9203

Analoge Eingang- & Ausgangskarten haben gewöhnlich eine kleinere Anzahl Eingänge als die digitalen Karten. Sie können jedoch auch dieselbe Aufgabe wie digitale Karten übernehmen. Durch die geringere Anzahl Aus / Eingänge, sind die analogen Karten bedeutend teurer als die digitalen.

Analoge Eingangskarten werden daher nur zum Einlesen von analogen Signalen verwendet.

Ein schneller Austausch der Karte ist durch Abnehmen der gesamten Schraubverbindungseinheit problemlos möglich.

Der interne Aufbau einer analogen Eingangskarte ist bedeutend komplexer als diejenige einer digitalen Eingangskarte (vgl. 1.Karte / digital Inputs).

Die Eingangssignale werden mit Hilfe eines Multiplexers in den Buffer geschrieben, welcher anschliessen von einem "Analog-Digital Converter" gewandelt wird.

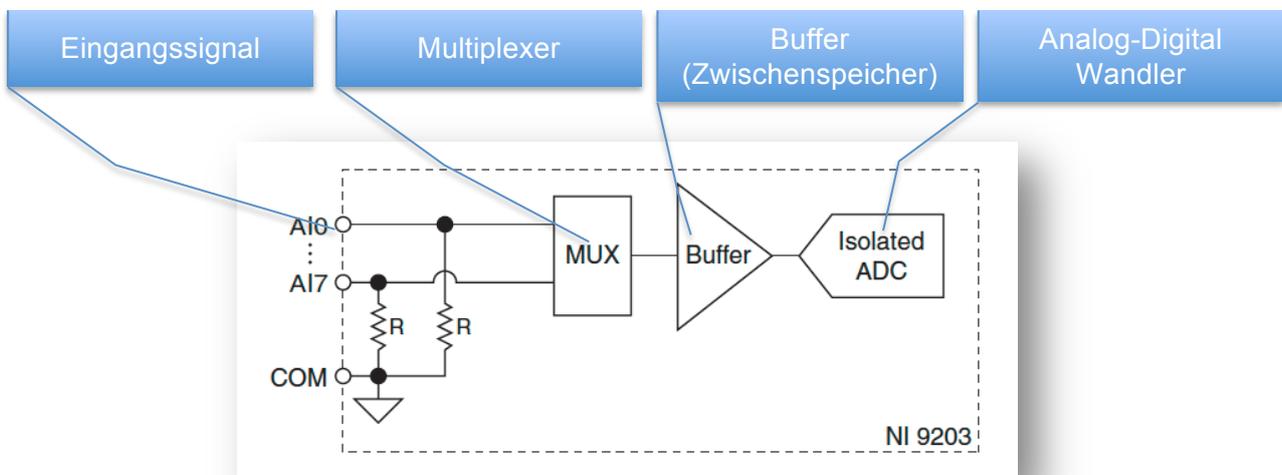


Abbildung 26 Karte 3 / interner Schaltungsaufbau Karte

Grundlegende Eigenschaften der Karte:

- 8 digitale Eingänge
- 16 Bit Auflösung
- Updaterate von 200kHz
- Konversionszeit 5 us

Die analoge Eingangskarte misst:

- den Durchfluss
- den Druck vor sowie nach der Prüfeinheit
- den Druck im Behälter 2
- die Wassertemperatur
- den Differenzdruck über der Prüfeinheit

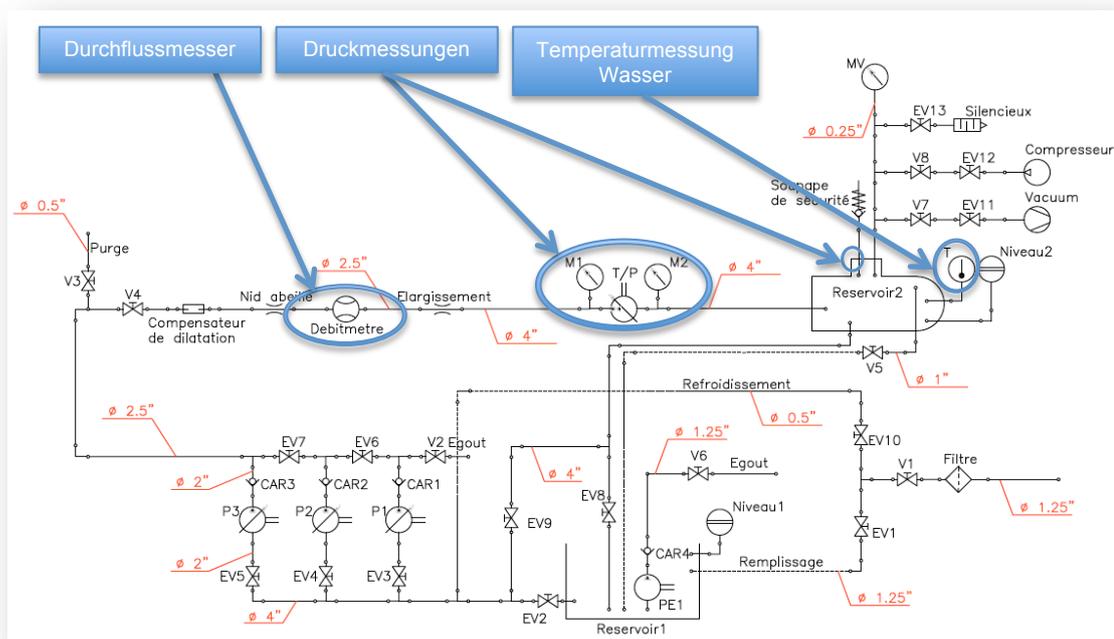


Abbildung 27 Karte 3 / Schema - angeschlossene Elemente

### Durchflussmesser

Der Anschluss des Durchflussmessers wurde aufgrund eines vorhandenen Geräts geplant. Da jedoch der Test des Durchflussmessers ein negatives Resultat in Bezug auf die Verwendung in diesem Projekt ergab, ist noch kein Ersatzgerät bekannt. Da sich jedoch der Anschluss eines Durchflussmessers nicht stark ändert, wurde die Darstellung im Schema beibehalten.

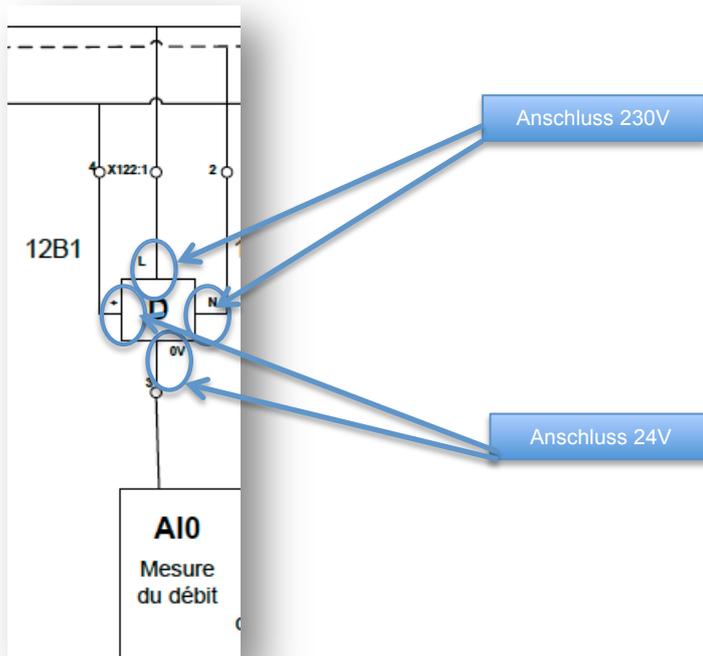


Abbildung 28 Karte 3 / Schema - Anschluss Durchflussmesser

### Drucksensoren

Es werden 3 Drucksensoren im System eingesetzt.

1. Vor der Testeinheit
2. Nach der Testeinheit
3. im Reservoir 2

Die Sensoren sind vom Typ Cerabar PMC 131 von Endress & Hauser.



Cerabar PMC 131

Abbildung 29 Karte 3 / Drucksensor Cerabar PMC 131

Dieser Sensor basiert auf dem Prinzip, dass eine keramische Membran aufgrund des Drucks geringfügig auslenkt. Die druckproportionale Kapazitätsänderung wird an den Elektroden des Keramiksensors gemessen. Aufgrund der Tatsache dass der Drucksensor keine Flüssigkeit für die Druckübertragung benötigt, ist der Sensor voll

vakuumtauglich. Da im Reservoir 2 durchaus ein Unterdruck erzeugt wird, ist diese Art von Sensor in unserer Anlage sinnvoll.

Kenngrößen des Sensors :

- Stromausgang 4-20 mA
- Eingangswiderstand der SPS : <math><2\text{k}\Omega</math> (die NI 9203 hat einen Eingangswiderstand von 138 Ohm)
- Genauigkeit <math><0.6\%</math>

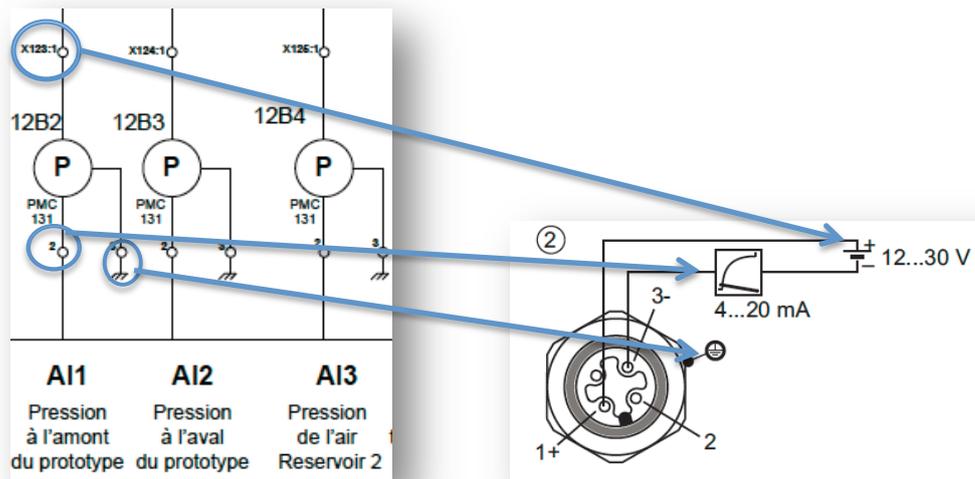


Abbildung 30 Karte 3 / Anschluss Drucksensor

Der Drucksensor PMC 131 wird mit 24 VDC angeschlossen. Das Ausgangssignal des Sensors wird direkt mit dem Automaten verbunden. Das Gehäuse wird geerdet. "P" ist der Kennbuchstaben für Druckmessgeräte.

Im Bild wurde jeweils der Anschluss nur anhand von einem einzelnen Sensor illustriert. Die weiteren Sensoren werden identisch angeschlossen.

## Temperaturmesssensor Reservoir 2

Um die Wassertemperatur zu messen, wird im Reservoir 2 ein Kompaktthermometer vom Typ Easytemp TMR31 eingesetzt.

Dieser Thermometer funktioniert mit einem PT100 und kann optional mit einem integrierten 4-20mA Messumformer geordert werden.

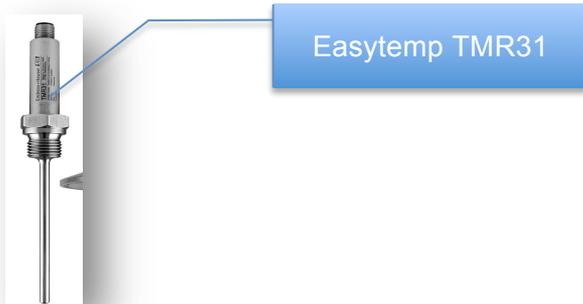


Abbildung 31 Karte 3 / Temperatursensor

Kenngrossen des Sensors :

- Messtemperaturen von -50 bis 130 °C
- Ausgangssignal Wahlweise 4-20mA oder PT100 Klasse4, 4Leiter Anschluss
- Genauigkeit :  $0.25K + 0.002 * [Zahlenwert\ der\ Temp.\ in\ ^\circ C]$
- Antwortzeit von 3 Sekunden

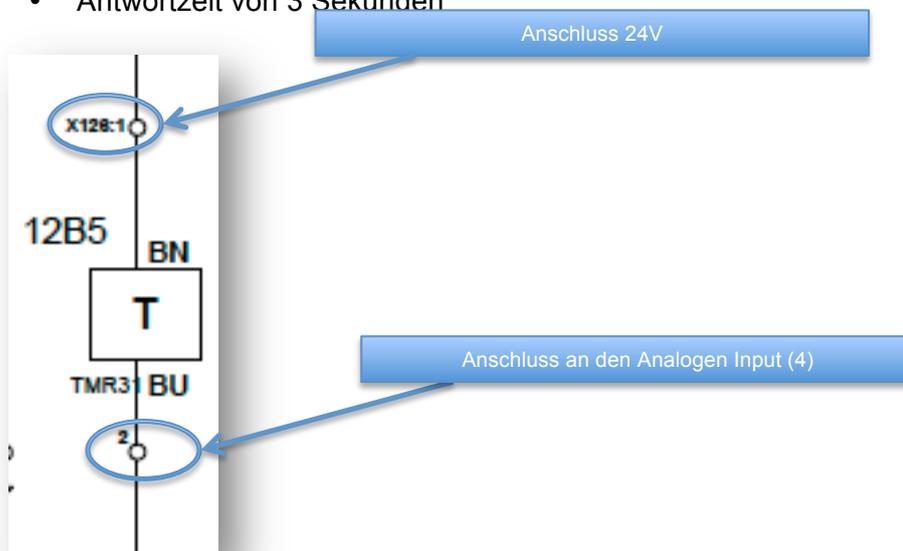


Abbildung 32 Karte 3 / Anschluss Temperatursensor

Die Temperatursonde wird mit dem optionalen 4-20mA Messumformer eingesetzt und kann daher direkt mit BN an 24VDC sowie mit BU an den Analogen Input "AI4 "angeschlossen werden.

Bei der Auswahl des Temperatursensors musste darauf geachtet werden, dass der Sensor möglichst direkt 4-20mA als Ausgabegrösse bietet. Der Anschluss eines PT100 ist an unsere analoge Karte nicht möglich, da sie nicht für solche Messungen ausgelegt ist.

### Differenzdrucksensor

Der Differenzdrucksensor über der zu prüfenden Einheit ist vom Typ Deltabar M PMD55 von Endress & Hauser.

Die Wahl eines zusätzlichen Druckmessers bei der Prüfeinheit, hat den Sinn die Genauigkeit zu erhöhen. Einfache, relative Drucksensoren geben einen guten Anhaltspunkt über den Druck am jeweiligen Ort. Der Differenzdruckmesser gibt

jedoch sehr präzise den Druckunterschied zwischen Ein-/Ausgang des zu prüfenden Objekts. Aufgrund dieser Angabe kann bereits der Druckabfall über dem jeweiligen Element ermittelt werden, ohne bereits 2 Genauigkeitsfehler in Kauf nehmen zu müssen.

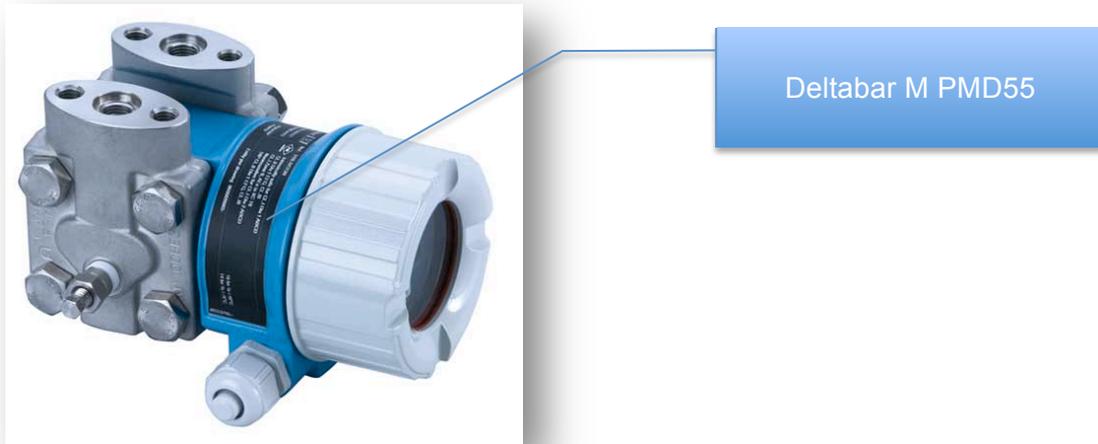


Abbildung 33 Karte 3 / Differenzdruckmesser Deltabar PMD55

Kenngrößen des Sensors:

Ausgangssignal von 4-20mA

Messgenauigkeit von 0,1%



Abbildung 34 Karte 3 / Anschluss Differenzdruckmesser

Der Deltabar PMD55 wird mit 24V angeschlossen und der 4-20mA Ausgang wird mit dem analogen Input verbunden. Zusätzlich muss der Sensor geerdet werden.

## 4. Karte / Analog Outputs

### Beschreibung der Karte

Die analoge Ausgangskarte NI 9265 dient in unserem Falle dazu, die Geschwindigkeit der Pumpen auszugeben. Jede Pumpe benötigt ein 4-20mA Signal, durch welches die Geschwindigkeit definiert wird. Die analogen Ausgänge können auch die Aufgabe von digitalen Ausgängen übernehmen, was jedoch viel teurer ist, da eine analoge Karte viel weniger Ausgänge bietet.

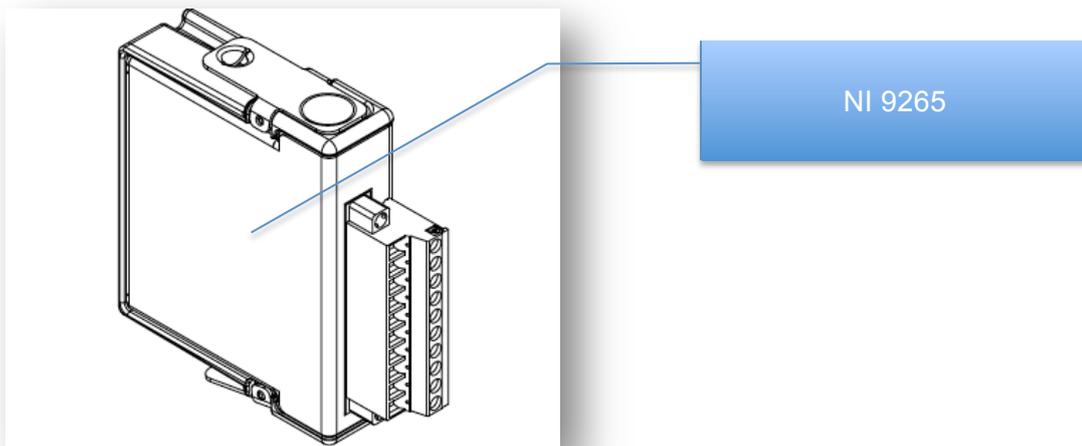


Abbildung 35 Karte 4 / NI 9265

Jeder Kanal besitzt einen digital-analog Konverter (DAC), welcher das Stromsignal erzeugt.

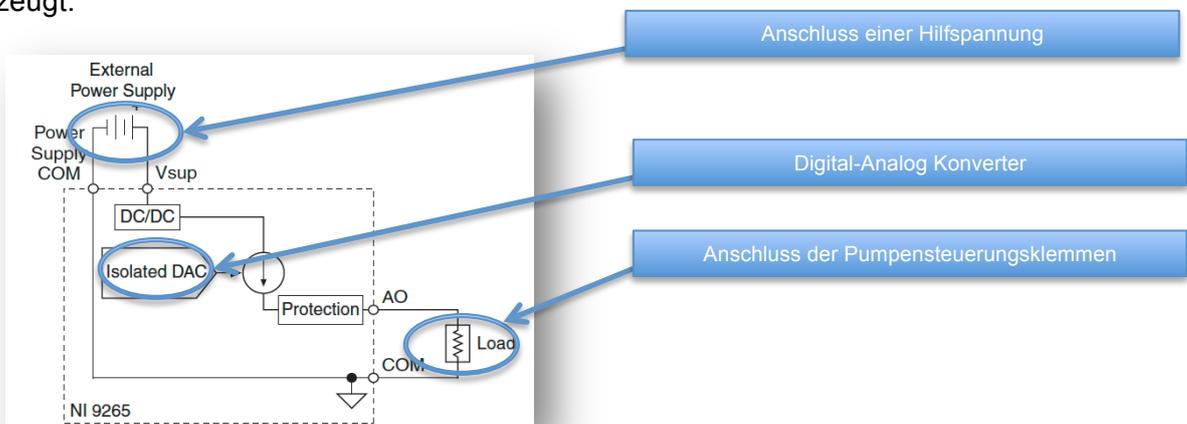


Abbildung 36 Karte 4/ interner Aufbau Karte

Um die angeschlossenen Geräte mit genügend Leistung versorgen zu können, benötigt diese Karte einen zusätzlichen Anschluss einer Hilfsquelle. Die Last wird zwischen dem Analog Output und dem Com Anschluss angeschlossen.

Grundlegende Eigenschaften der Karte :

- 4 analoge Ausgänge
- 16 Bit Auflösung (DAC)
- Vsup von 9-36VDC
- Updatezeit bei 3 Kanälen von 7.5us

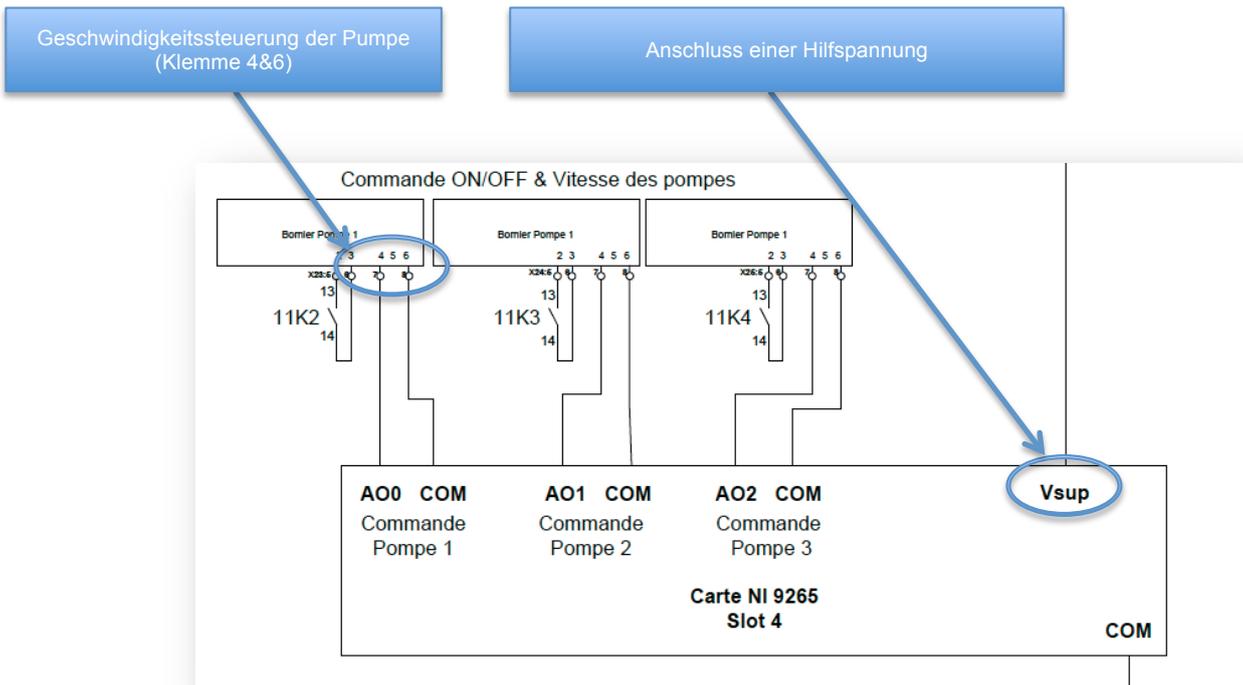


Abbildung 37 Karte 4 / Anschluss Geschwindigkeitssteuerung

Die Pumpen haben neben dem Eingang für das Signal der Geschwindigkeitsvorgabe und Leistungsanschluss einen Kontakt welcher das grundsätzliche Ein/Ausschalten der Pumpe kontrolliert.

Dieser Kontakt kommt von einem Relais welches auf Seite 11 (digitale Outputs) stammt. Die Logik des Programms kann somit die Pumpe durch aktivieren eines Ausgangs direkt ausschalten, ohne der Geschwindigkeitsregelung Beachtung zu schenken.

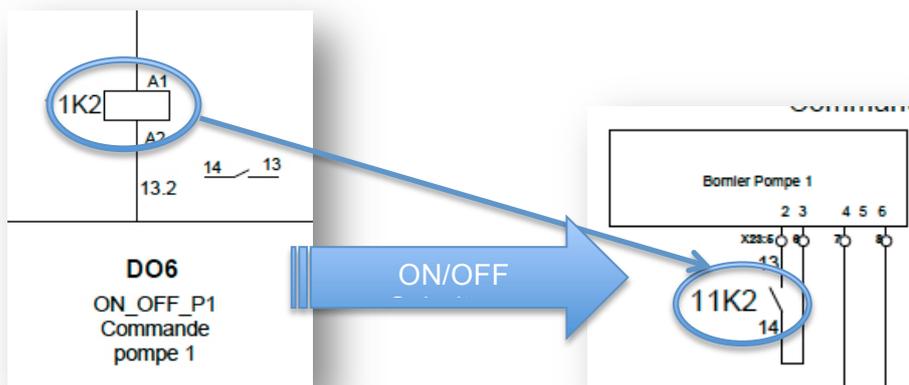


Abbildung 38 Karte 4 / ON-OFF Schaltung an Pumpenklemme

Wie auch bei den bisherigen Illustrationen wird nur jeweils auf 1 Element hingewiesen, wobei die weiteren identischen Elemente auf die gleiche Weise angeschlossen werden.

## 5.Karte / schnelle digital Inputs

### Beschreibung der Karte

Die digitale Eingangskarte NI 9411 ist sehr schnell. Sie erreicht rund die 14mal höhere Abtastgeschwindigkeit als die NI 9225.

Da das Erreichen einer solchen Geschwindigkeit mehr Elektronik benötigt, besitzt die Karte nur 6 Eingänge

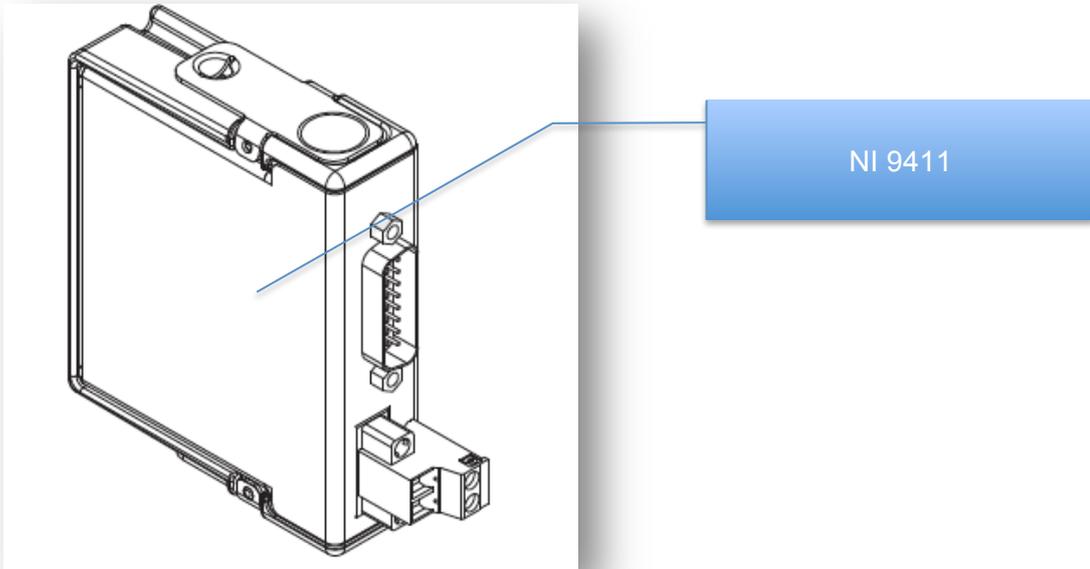


Abbildung 39 Karte 5 / NI 9411

Die Karte kann auf differentieller Basis arbeiten bspw. mit differentiellen Encodern. In unserem Fall wird jedoch der "Singel-Ended" Betrieb benutzt.

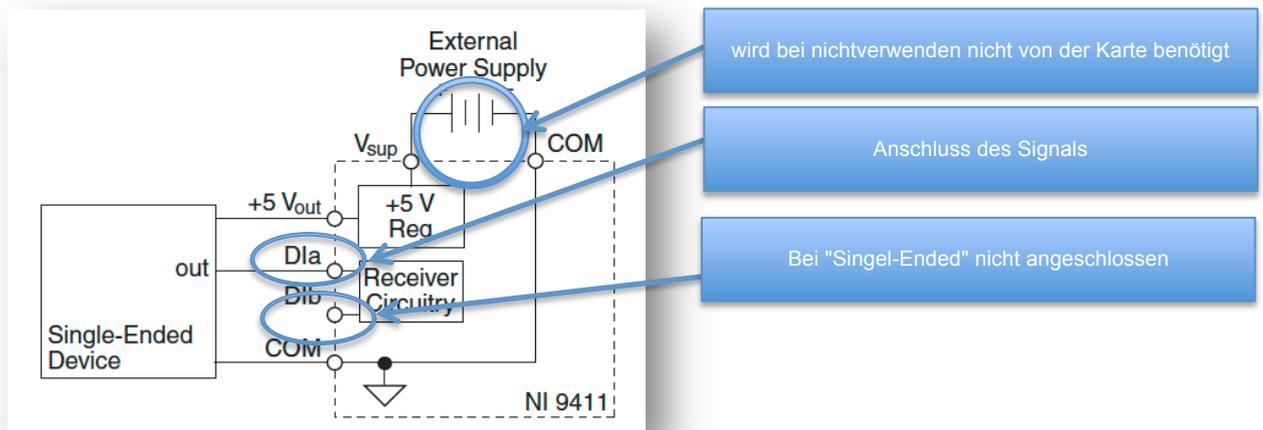


Abbildung 40 Karte 5 / Beschaltung Karte

Grundlegende Eigenschaften der Karte :

- 6 digitale Eingänge
- Input high Range von 2-24VDC

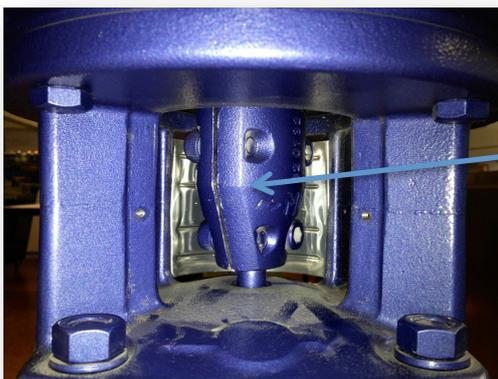
- Input low Range von 0-0.8VDC
- Eingangsverzögerungszeit von 500ns

Muss die Karte die Sensoren nicht selbst mit 5VDC versorgen, benötigt sie den Anschluss einer externen Speisung nicht.

Diese Karte dient der Geschwindigkeitserfassung durch Impulse des Geschwindigkeitssensors der Pumpen. Ein einzelner Messumformer, welcher aus Impulsen direkt 4-20mA wandelt, kostet deutlich mehr als eine, hier verwendete Karte welche erlaubt, bis zu 6 solcher Signale zu messen.

Um die Geschwindigkeit der Pumpen festzustellen, wurde die Methode mit einer einseitigen Lichtschranke gewählt. Die induktive Möglichkeit fiel aufgrund der Nähe zu Motor und weiterem Metall aus. Ebenfalls müsste ein Stück Metall auf die Welle geschweisst werden, um detektiert zu werden. Wird dies nicht exakt auf der Gegenseite wiederholt, gibt es Unwuchten an der Welle, was auf Dauer die Lager schädigt.

Der verwendete Sensor wird an dem Übergangspunkt von Motorwelle zu Pumpenwelle eingesetzt. Dieser Ort wurde aufgrund der leichten Zugänglichkeit gewählt.



Einsatzort des optischen Sensors zur Messung der Umdrehungsgeschwindigkeit

Abbildung 41 Karte 5 / Einsatzort Geschwindigkeitssensor

## Einweglichtschranke



V180-2 Photoelectric Sensor

Abbildung 42 Karte 5 / Geschwindigkeitssensor V180-2 Photoelectric

Die Einweglichtschranke V180-2 kann Signalfolgen von bis zu 1 kHz erfassen. Dies bedeutet 1000 Umdrehungen pro Sekunde. Da der verwendete Motor lediglich bis ca. 3000 Umdr./min erreicht, wird die Kapazität des Sensors nicht gefordert.

Kenngrößen des Sensors:

- Betriebsspannung 10-30VDC
- Schaltfrequenz 1000Hz
- Erleichterte Ausrichtung durch extrem hellen Lichtfleck
- Signalspannung von 0-Versorgungsspannung-1.8V

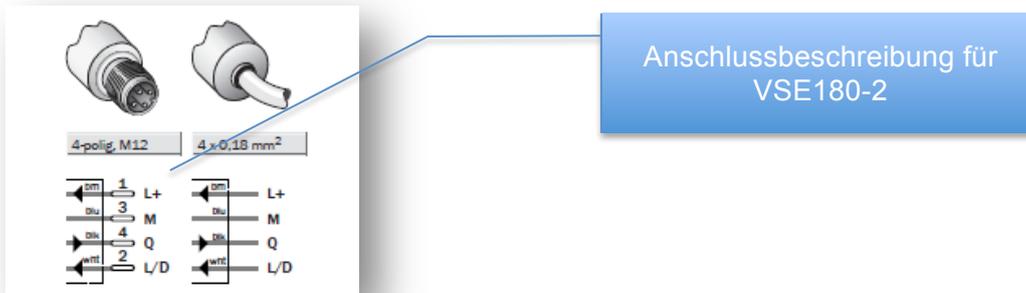


Abbildung 43 Karte 5 / Anschlussbeschreibung VSE 180-2

Aufgrund dieser Anschlussbeschreibung ist

- L+ die 24V Versorgung
- M die Masse (0V)
- Q das Ausgangssignal welches auf den Automaten geleitet wird
- L/D die Steuerleitung

Da Hellschaltend gewählt wurde, wird an die L/D Steuerleitung ein 24V Signal gelegt. (L/D = + UV: hellschaltend L.ON | L/D = 0 V: dunkelschaltend D.ON)

Anschlüsse des optischen Sensors.

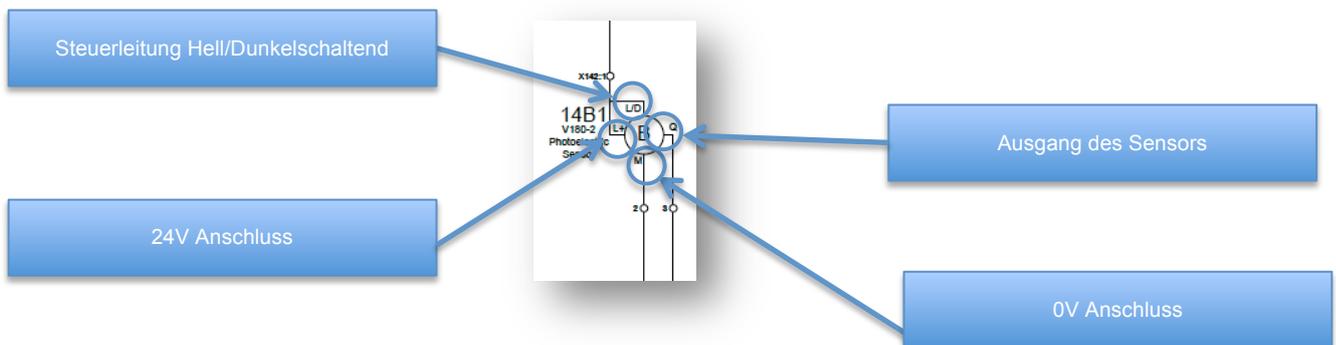


Abbildung 44 Karte 5 / Schema - Anschluss Geschwindigkeitssensor

## Automation / Labview

### Allgemeine Informationen

---

#### Weshalb Labview?

Labview unterscheidet sich in 2 wesentlichen Punkten von anderen gebräuchlichen Programmiersprachen.

Labview wird mit der Programmiersprache G programmiert, was eine graphische Programmierung durch Verbinden von Funktionsblöcken darstellt. Trotzdem sind die üblichen Konzepte der Programmierung vorhanden, wie etwa While-, For-Schleifen sowie die bekannten Datentypen aus anderen Programmiersprachen.

Der weitere Unterschied besteht in der Tatsache das Labview nicht sequentiell Befehle abarbeitet, sondern dem Prinzip des Datenflusses folgt. Demzufolge wird eine Funktion ausgeführt, sobald alle nötigen Daten für die Ausführung der Funktion vorhanden sind.

Die Vorteile dieser graphischen Programmierung sind die einfache Veränderung der Funktionen durch einen anderen Programmierer. Ein in bspw. C geschriebenes Programm ist von anderen Programmierern meist nur schwer zu interpretieren und benötigt mehr Zeit, als die Analyse eines Programmes in Labview. Aufgrund dieser intuitiven Programmierung, wird Labview häufig in Labors und Testeinrichtungen verwendet, da häufig der Code verändert werden muss.

Neben den praktischen Vorteilen von Labview war der didaktische Hintergrund für Diplomanden / Studenten ein Grund, diese andere Art des Programmierens zu wählen.

## Einführung in die Programmierung mit Labview

### Grundlegender Aufbau von Labview

Labview arbeitet mit sogenannten VI's. VI bedeutet "virtuelles Instrument". Diese VI besitzen jeweils ein "Frontpanel" sowie das "Blockdiagramm".

Das Frontpanel dient zur Darstellung von Ein- / Ausgabestellen, das Blockdiagramm bildet die Logik im Hintergrund.

Ein einfaches Beispiel :



Abbildung 45 Bsp. Frontpanel

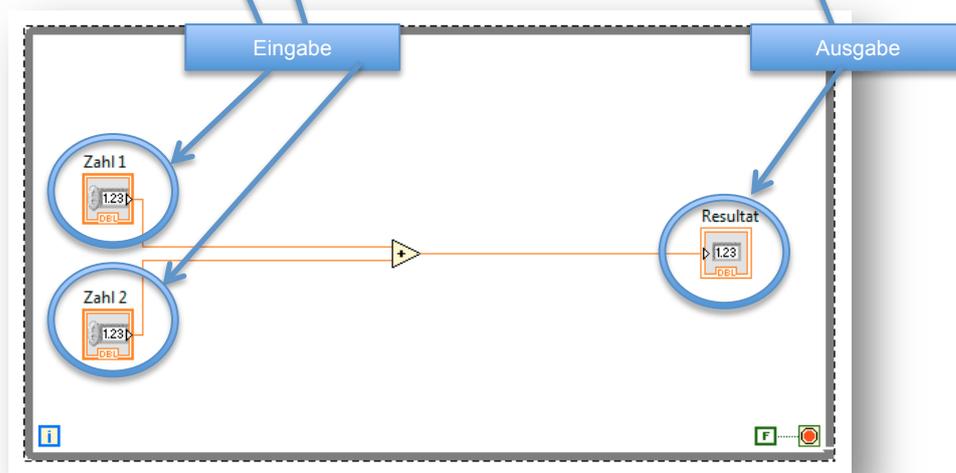


Abbildung 46 Bsp. Blockdiagramm

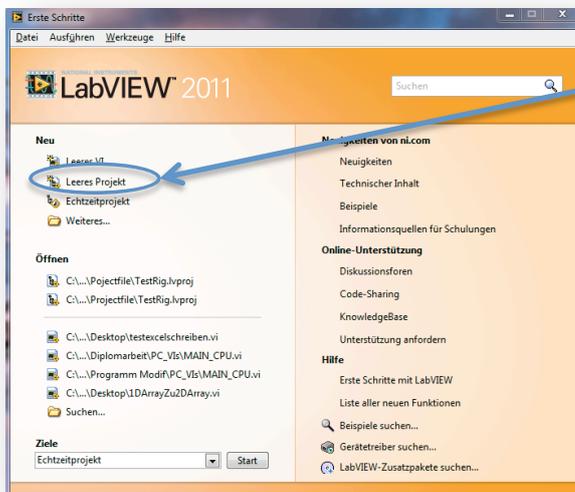
Die Zahl 1 & 2 können auf dem Frontpanel eingegeben werden. Die Logik im Hintergrund, welche in einer Endlosschleife ohne Geschwindigkeitsbegrenzung läuft, berechnet das Resultat und gibt die Ausgabe auf dem Frontpanel aus.

Es können beliebig viele verschiedene VI's erstellt werden. Komplexe Berechnungen können in einem VI programmiert werden und anschliessend in andere VI's implantiert werden. Ein implantiertes VI wird dadurch zu einem Sub-VI.

Je nach Verwendungsart der Sub-VI's, muss der Speichernutzung sowie dem Einsetzen von Schleifen Beachtung geschenkt werden, da sie ansonsten den Datenfluss unterbrechen können, oder in die gleichen Speicherblöcke schreiben.

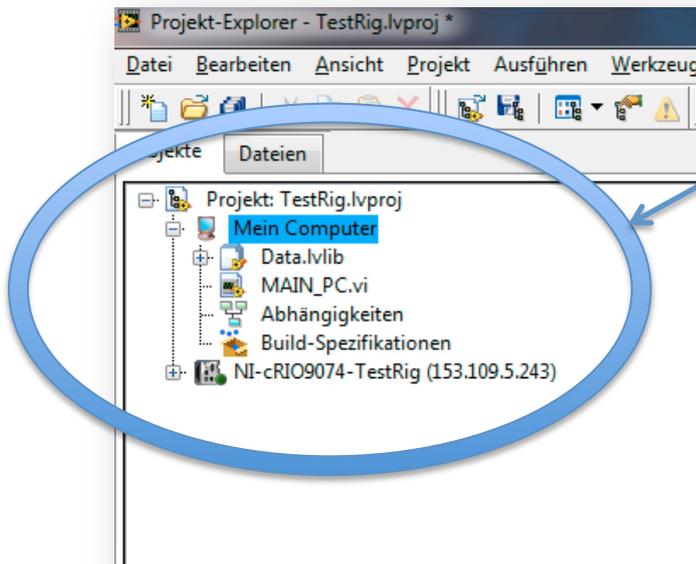
Sub-VI werden meist für komplexere Abläufe, welche mehrmals an verschiedenen Arten eingesetzt werden, sinnvoll.

Die VI's stellen jedoch nur ein Teil eines Programms dar. Zu einem Programm gehört noch der Automat, sowie die geteilten Variablen (s. nächster Abschnitt). Diese Elemente können im Rahmen eines Projektes miteinander verbunden werden.



leeres Projekt

Abbildung 47 Projektauswahl



VI's / Daten / Automat nach erfolgter Einfügung in ein Projekt

Abbildung 48 Projektexplorer

Im oben Dargestellten Projekt sind die Daten / VI's sowie der Automat bereits importiert worden. Mehr dazu in den folgenden Abschnitten.

### Variablen & Datenlogging mit Citadel

Labview kennt 3 Arten von Variablen. Die lokalen Variablen sind nur für das jeweilige VI verfügbar. Eine lokale Variable kann durch einen Rechtsklick auf das

Anzeigeelement -> Erstellen -> lokale Variable erstellt werden.

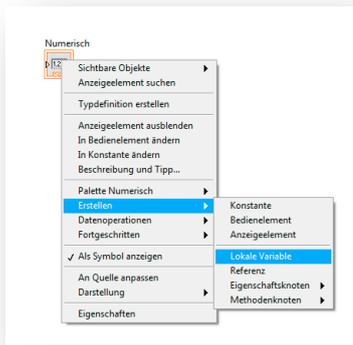


Abbildung 49 lokale Variable erstellen

Weiter existieren die globalen Variablen. Da diese Variablen von mehreren VI's gelesen werden können, jedoch kein Datenaustausch zwischen verschiedenen Geräten zulassen, wurden sie im Rahmen des Projekts nicht verwendet. Damit die CPU des Automaten, welche ein eignes VI besitzt, mit dem VI des Computers kommunizieren kann, ist der Weg durch "Shared Variables" gegeben. Diese Variablen können von beiden Geräten beschrieben oder gelesen werden. Wird die erste Variable erstellt, wird ebenfalls eine Library erstellt. Innerhalb dieser Datenbank können anschliessend gewünscht viele Variablen erstellt werden:

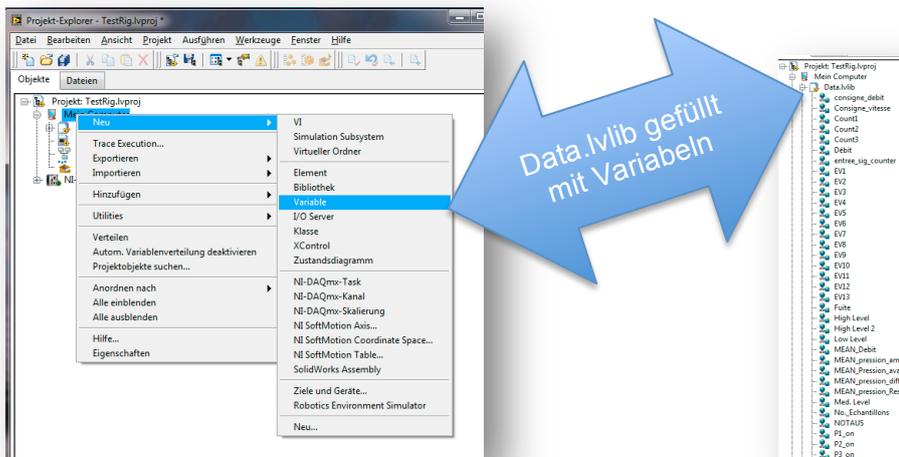


Abbildung 50 Variablen & Library

Eine Variable hat viele Einstellungsmöglichkeiten. Damit eine Variable von anderen Geräten gelesen werden kann, muss der Variablentyp auf Netzwerk eingestellt sein. Der Datentyp ist nach Verwendung zu wählen.

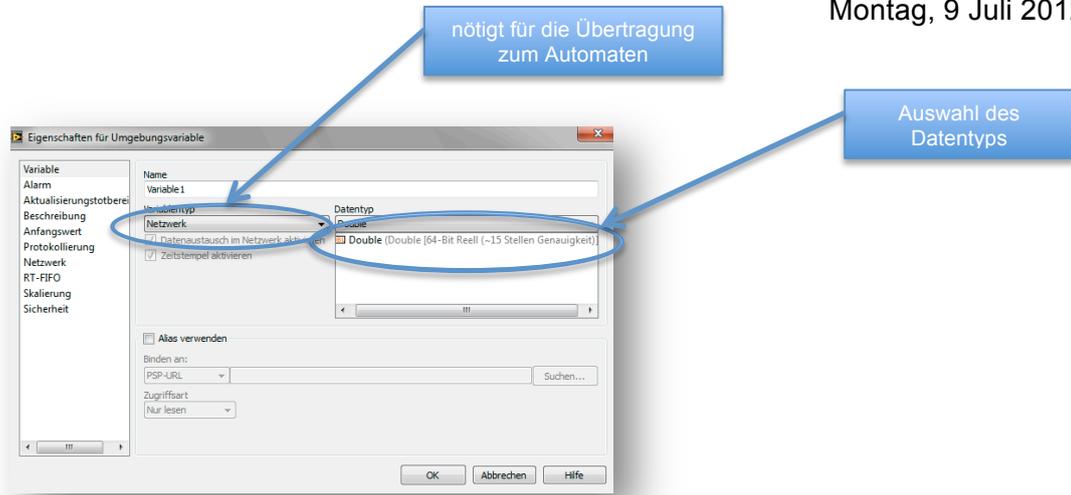


Abbildung 51 Optionen Variablen

Umgebungsvariablen (shared variables) können ebenfalls von Labview geloggt werden. Dies wird häufig bei Testbänken angewandt, da die erstellten Messlisten von Labview nicht manuell geändert (gefälscht) werden können. Die Datenbank für die geloggt Daten müssen mit dem Tool "Measurement und Automation" erstellt werden und sollten direkt am finalen Speicherort platziert werden, damit die Verbindungen nicht durch ein Kopieren zerstört werden.

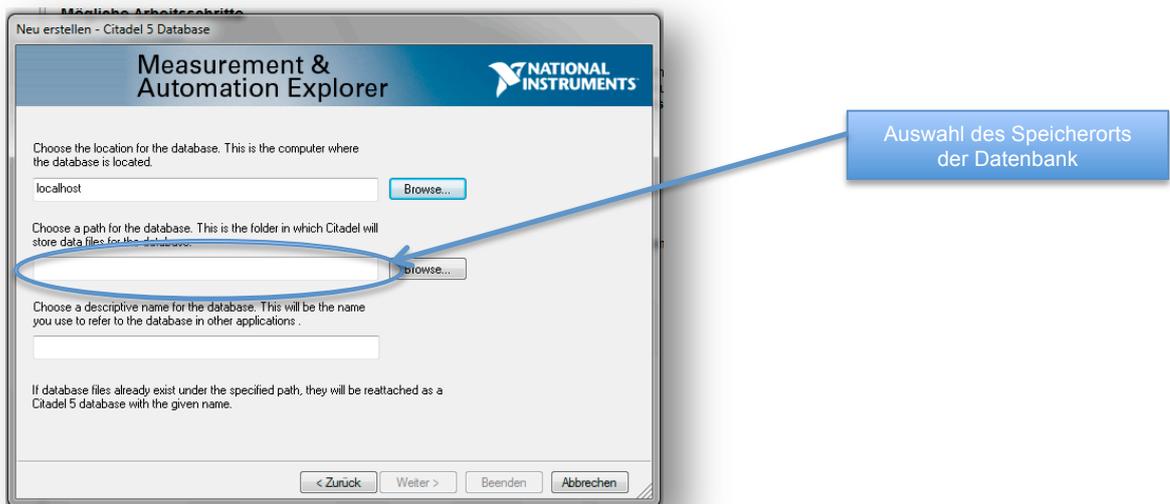


Abbildung 52 Citadel Speicherposition

Diese Datenbank trägt den Namen, der seinem Pfad entspricht. Die Datenbank in Labview muss mit der neu erstellten Datenbank gekoppelt sein. Folgend nun die einzelnen Schritte zur Anbindung der Datenbank.

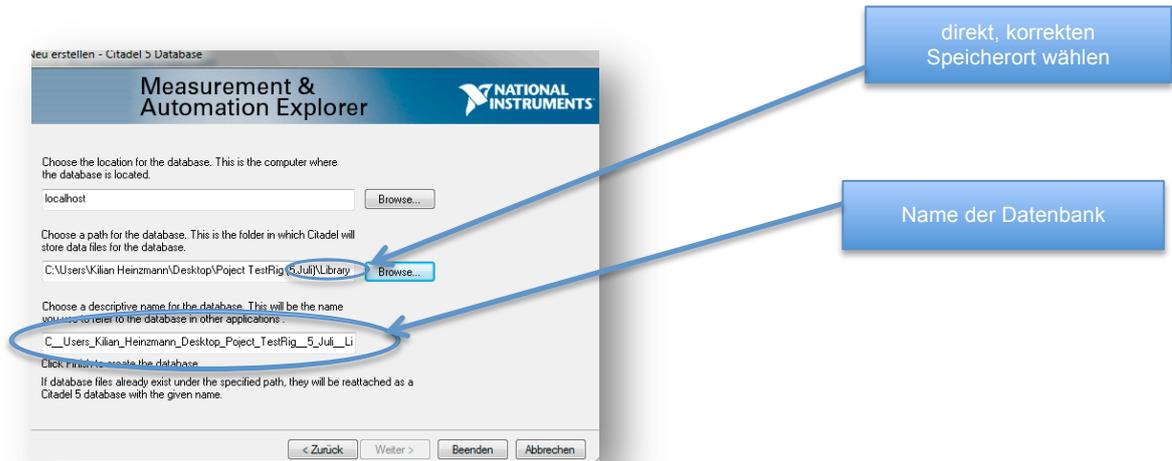


Abbildung 53 Citadel Speichernamen Datenbank

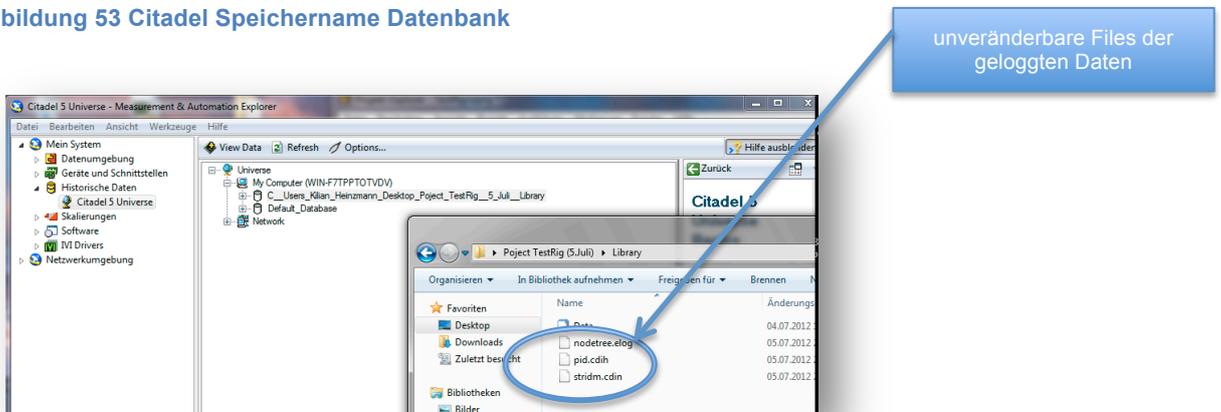


Abbildung 54 Dateien von Citadel

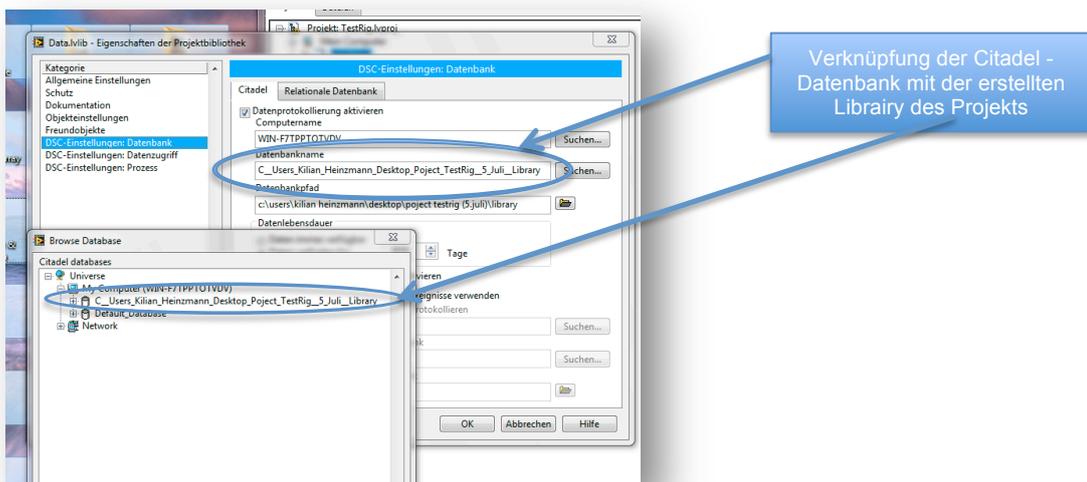


Abbildung 55 hinzufügen von Variablen-Datenbank zu Citadel Datenbank

Neben der Abspeicherung von Messwerten können ebenfalls Alarmer ausgelöst werden.

Die Variablen die zu protokollieren gewünscht sind, müssen nun in ihren Eigenschaften zur Protokollierung freigegeben werden.



Abbildung 56 Protokollierung Variablen

Die geloggtten Werte können nur noch via Labview ausgelesen werden, ein verändern ist ausgeschlossen, damit keine Messwertfälschung betrieben werden kann.

Die geloggtten Daten können durch die Angabe von Zeit und Datum anschliessend jederzeit wieder in Labview gelesen oder gar in ein Excel-File exportiert werden.

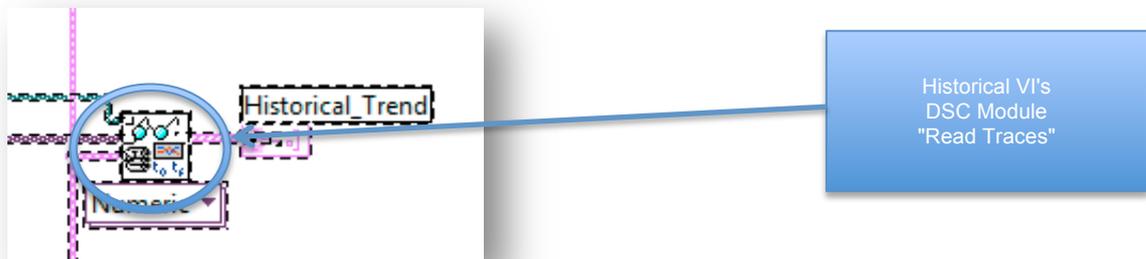


Abbildung 57 Auslesen der Citadel - Protokollierungen

Ein Beispiel zur Verwendung findet sich im Programm " Echangeur.lvproj" welches von Herrn Vacari Aldo programmiert wurde.

### Geschwindigkeit von Umgebungsvariablen

Beim Einsatz von sehr vielen Umgebungsvariablen ist es sinnvoll den die Zugriffsart durch einen Rechtsklick auf die im Blockdiagramm befindliche Variable auf " durch programmatischen Zugriff ersetzen" zu setzen.

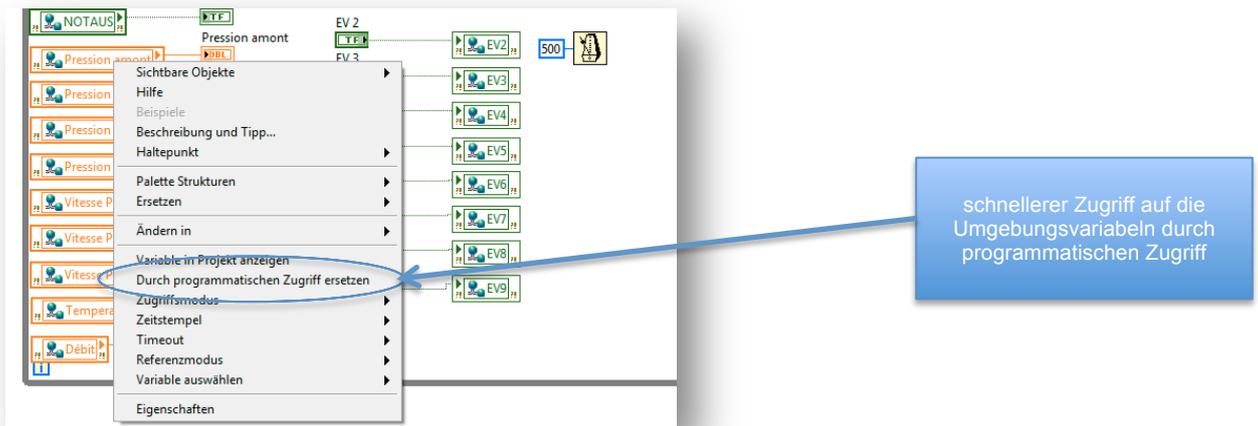


Abbildung 58 programmatischer Zugriff (Variablen)

## Strukturierung des Programms

Die Strukturierung und Aufbau eines Programms gehört mit zu den wichtigsten Punkten eines soliden Programms.

Die Struktur eines Programms besteht aus der Anordnung der Ordner in einem geeigneten Register, sowie die eigentliche Programmstruktur auf den 3 Ebenen FPGA / CPU / PC.

Die Ordnerstruktur wurde auf die Übersichtlichkeit und schnelle Auffindbarkeit von Elementen optimiert.

Die Aufteilung der Ordnerstruktur nach Programmstruktur ergibt die praktischen Vorteile, zusammenpassende VI's gruppieren zu können.

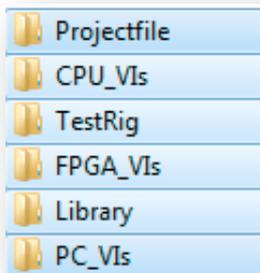
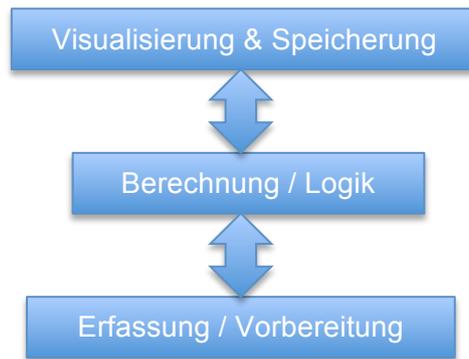


Abbildung 59 Ordnerstruktur

Der Ordner "Projectfile" enthält die Projektdatei, welche die anderen VI's beinhaltet. Die CPU VI's enthalten alle virtuellen Instrumente welche auf der CPU installiert werden. Aktuell befindet sich jeweils nur 1 VI in den jeweiligen Ordnern, da keine speziellen Berechnungen durchgeführt werden, welche eine eigene VI benötigen hätten.

Unter TestRig befindet sich die Datenbank von Citadel. Das Haupt-VI der FPGA ist ebenfalls in einem eigenen Ordner FPGA\_VIs abgelegt. Der Library-Ordner enthielt eine Testliste von Variablen, welche während der Programmierung temporär benutzt werden konnten.

Das Programm wurde grob in 3 Kategorien unterteilt.



Die unterste Schicht erfasst die wichtigen Daten. Da die Rohdaten von Sensoren teilweise sehr schnell verarbeitet werden müssen, wie bspw. das Pulssignal der Lichtschranke zur Geschwindigkeitsmessung, werden diese direkt nach Empfang, für die Berechnungen aufbereitet.

Die Ebene "Berechnungen" beinhaltet die nötigen Berechnungen um aus gemessenen Werte wie Druck, Durchfluss, Temperatur etc. die wichtigen Parameter wie den Wirkungsgrad auszugeben. Neben den üblichen Berechnungen beinhaltet diese Ebene ebenfalls die gesamte Logik der Anlage.

Da die berechnenden Aufgaben alle in der 2ten Ebene erledigt wird, bleibt der Dritten Ebene nur noch die Speicherung gewünschter Daten sowie die Visualisierung übrig. Diese Unterteilung in 3 Ebenen wurde im Hinblick auf die Schichtstruktur welche in unserem Automationssystem herrscht, erstellt.

Die hypothetisch erstellte Struktur wird folgendermassen auf dem Automationssystem implementiert:

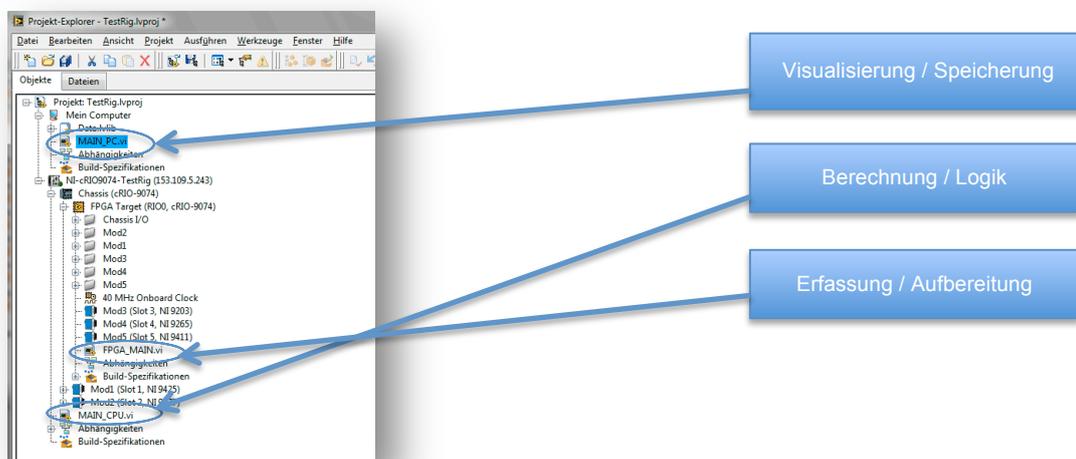


Abbildung 60 Programmstruktur

Die Visualisierung und Speicherung der Daten befindet sich direkt auf dem Computer. Dies ist naheliegend, da die Visualisierung auf dem PC geschieht und somit auch direkt die Programmierung der Visualisierung hier gespeichert wird. Die Speicherung wurde aus demselben Grund auf diese Ebene verschoben. Würde in der CPU abgespeichert werden, müssten die Daten anschliessend via FTP auf den PC verschoben werden.

Die Berechnungen sowie die Logik werden auf der CPU programmiert. Die CPU eines Automaten arbeitet zuverlässiger als ein PC, daher wird die Anlage hauptsächlich von der CPU gesteuert.

In der FPGA werden nur die nötigsten Umwandlungen, welche zudem noch sehr schnell erledigt werden müssen, programmiert. Die FPGA ist das schnellste Element in der Kette, jedoch benötigt das Aufspielen des Programmes schnell 20min, was für die Entwicklung sehr hinderlich / umständlich ist.

## Inbetriebnahme des compactRio

Die Lieferung des Automats erfolgte in Woche 6-7 der Diplomarbeit.

Zum testen der Funktionsfähigkeit und zum Erlernen der internen Struktur wurde der Automat mit einer 24V Speisung anhand Bedienungsanleitung angeschlossen. Weiter wurde zum Test der Kartenkommunikation ein Drucksensor sowie ein Funktionsgenerator beschafft.

Bereits bei der Installation der Treiber auf den cRio erschien die erste Fehlermeldung:

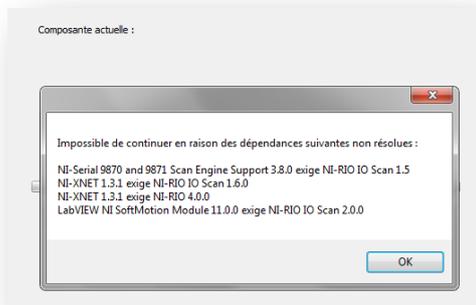


Abbildung 61 Fehlermeldung cRIO

Diese Fehlermeldung konnte durch das Installieren der neuesten Treiber beseitigt werden.

Die Eingliederung des cRio erfolgt mit folgenden Schritten :

- 1) Öffnen des Projekts
- 2) Rechtsklick auf den Projektname -> NEW -> Devices & Targets

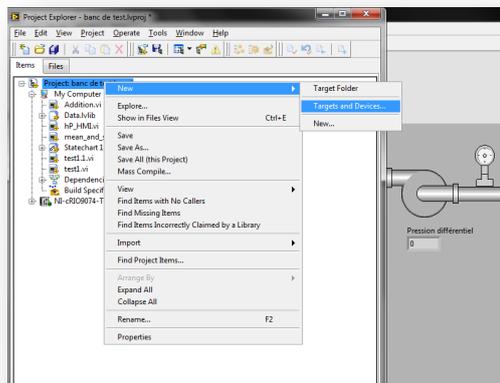


Abbildung 62 Einbindung cRio in Projekt

Ein neues Gerät / Ziel wird zum Projekt hinzugefügt. Auf diese Weise können auch weitere Elemente / Hardware an das Projekt angebunden werden.

### 3) Auswahl von "Real Time CompactRIO"

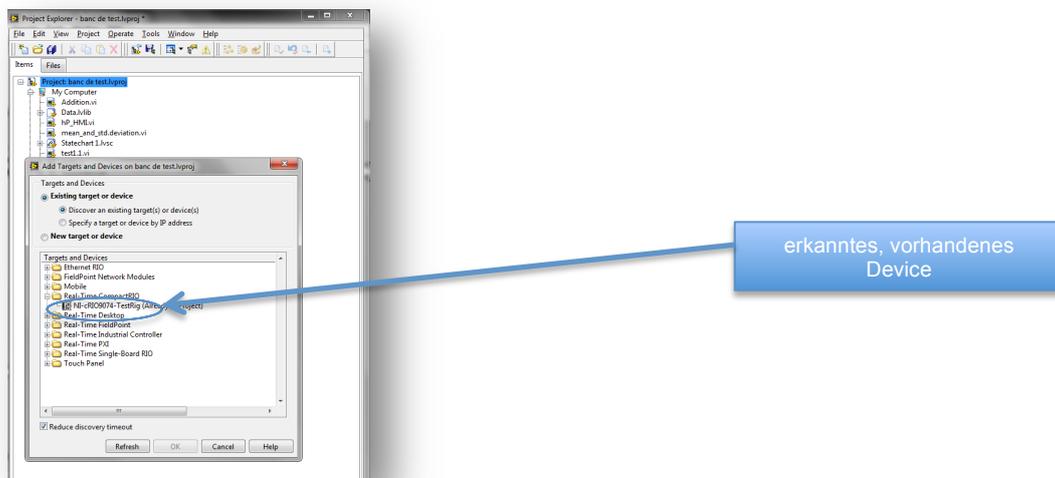


Abbildung 63 Hinzufügen des cRio zu Projekt

Wird das Gerät hier nicht angezeigt, muss in der Zusatzsoftware "Measurement & Automation" die Gerätekonfiguration / Netzwerkkonfiguration überprüft werden.

### 4) Auswahl des NI cRIO 9074

### 5) Auswahl zwischen Scan-Interface / FPGA-Interface

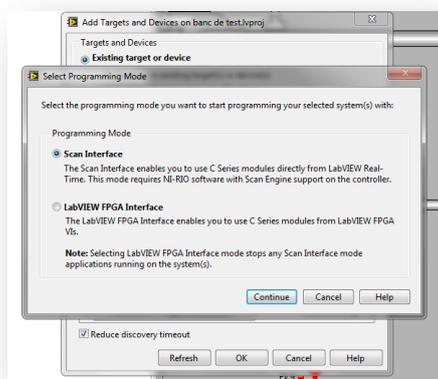


Abbildung 64 Auswahl - Scan o. FPGA Interface

Das Scan-Interface wurde von NI hauptsächlich entwickelt, damit die cRio's auch von Labview-Versionen ohne installiertes FPGA-Pack programmiert werden können. Durch das Scan-Interface verliert der Programmierer die Möglichkeit, die schnelle FPGA Einheit zu programmieren. Die Signale werden durch die FPGA direkt an die CPU durchgeschleust.

Da für das TestRig-Programm die schnelle FPGA Einheit für die Erfassung des Geschwindigkeitssignals der Pumpen benötigt wird, kommt das LabVIEW FPGA Interface zum Einsatz.

Ein geeignetes Kabel wurde im Rahmen der Tests gelötet. Die Fotos dokumentieren die Anschlussbelegung.

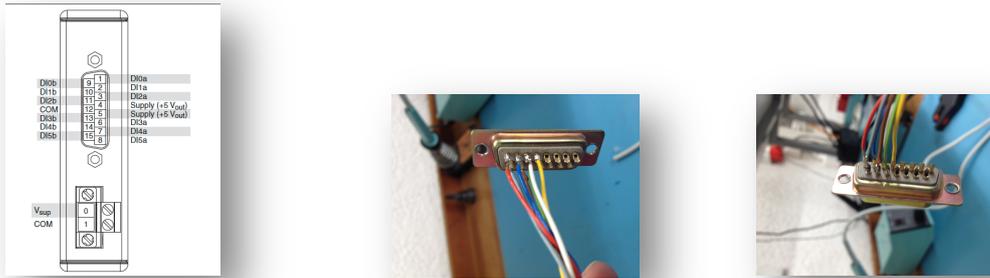


Abbildung 65 Erstellung von Testkabeln

## Realisierte Programme

---

### Einführung

Die hier aufgeführten Programmteile dienen als Basis für die Weiterentwicklung des Programms. Diese Programme dienen als Grundbaustein und sind beliebig erweiterbar. Das ganze Programm ist modular aufgebaut und basiert auf einfachen Strukturen.

Diese Programme haben keinen Anspruch auf Vollständigkeit, da sie im ersten Schritt zur Weiterentwicklung realisiert worden sind.

Bei einem Projekt in diesem Ausmass ist eine vollständige Programmierung der Software unmöglich, ohne praktischen Erfahrungen, Messungen und Tests auf der tatsächlichen Anlage. Ebenfalls können sich weitere Anforderungen an die Software stellen, welche mit der Zeit integriert und fortlaufend verbessert werden müssen.

### HMI

Das HMI (Human Machine Interface) dient dem Benutzer zur Kontrolle der Installation. Diese Ansicht dient ausschliesslich der Visualisierung und dem Empfang von Befehlen des Anwenders.

Das HMI wurde auf dem Computer programmiert, da das HMI auch auf diesem angezeigt wird.

Das HMI erhielt in unserem Fall auch die Aufgabe, den Speicherort für die Messungen festzulegen und dient im Hintergrund zur Speicherung der gewünschten Daten auf dem Computer.

Das Frontpanel besteht aus einer Ansicht von Registern, welche im ersten Schritt die Anlage darstellen inkl. aller Messpunkte. Nahe bei den Messpunkten wurde intuitiv die Anzeige des aktuellen Wertes platziert. Bestimmte Elemente können durch ihr anklicken aktiviert / deaktiviert werden. So konnte an Übersicht gewonnen werden.

Manuell gesteuerte Elemente sind bspw. schwarz markiert, was bedeutet, dass deren Zustand nicht visualisiert werden kann. Die Elektroventile welche ebenfalls von Hand gesteuert werden, können Ihren Zustand an den Automaten weiterleiten, so dass Ihr Zustand angezeigt werden kann.

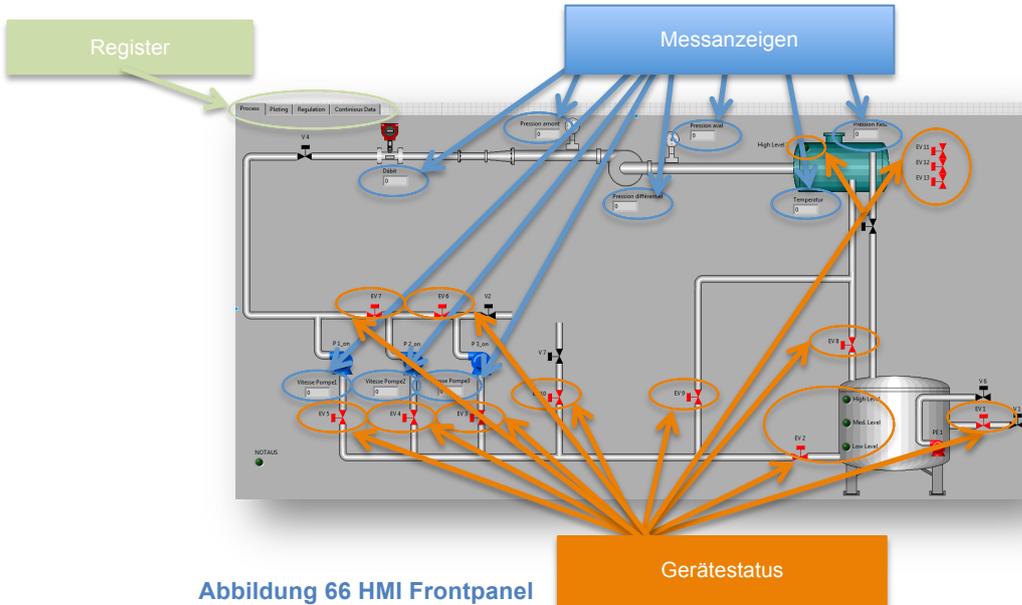


Abbildung 66 HMI Frontpanel

Die Ventile EV2 bis und mit EV9 gehören zu der Sorte Ventile, welche trotz manueller Steuerung durch die Rückführung eines Signals visualisiert werden können.

Das Blockdiagramm der HMI ist sehr simpel gehalten. Das Ziel dieses Aufbau's ist, einen so direkten Weg wie möglich durch das System zur Hardware herzustellen. Bei direkten Tests auf der Anlage, können dadurch kleine Fehler schneller identifiziert werden.

Es befinden sich 2 Schleifen im Blockdiagramm. Die erste Schleife dient lediglich zum empfangen der Werte der CPU sowie dem Senden der Benutzereingaben. Die zweite Schleife dient der Datenerfassung.

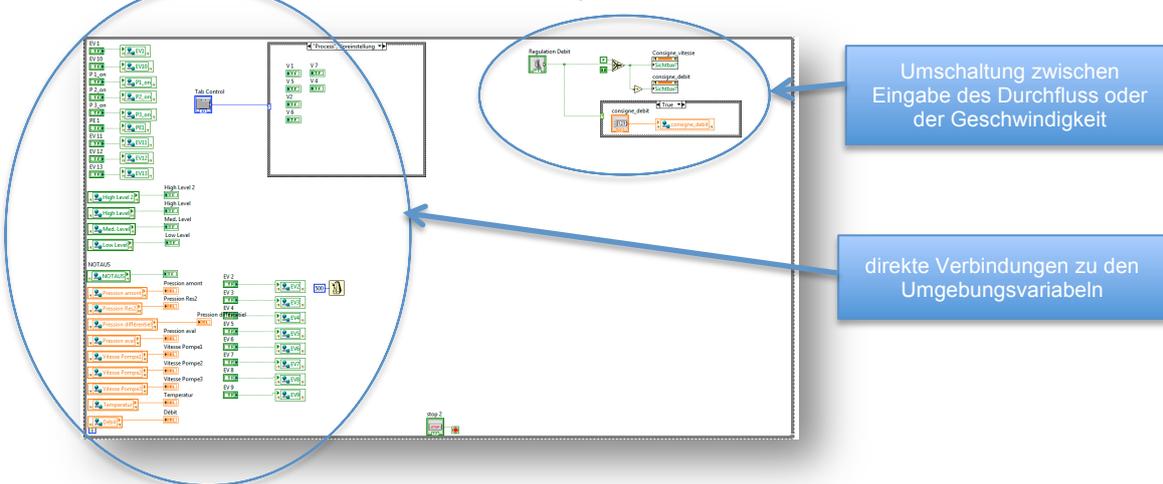


Abbildung 67 HMI Blockdiagramm

Wie auf dem Bild zu erkennen, werden die Elemente auf dem Frontpanel vorerst ohne gegenseitige Sicherung direkt zu den Umgebungsvariablen verbunden. Oben rechts ist eine Schaltungsmöglichkeit gegeben, um je nach Auswahl, das Eingabefenster der Geschwindigkeit oder des Durchflusses freizugeben. Auf dem Frontpanel sieht das folgendermassen aus:

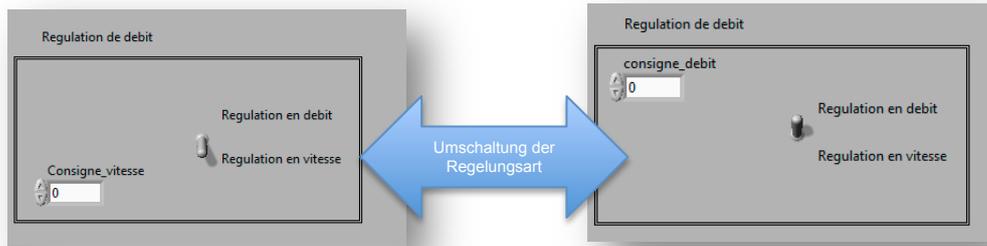


Abbildung 68 Regelungswahlschalter

Die 2te Schlaufe gestaltet sich in Form einer Sequenz. Beim Start des Programms werden 3 Zustände durchlaufen und anschliessend wird im 4ten Zustand in einer While-Schleife verweilt.

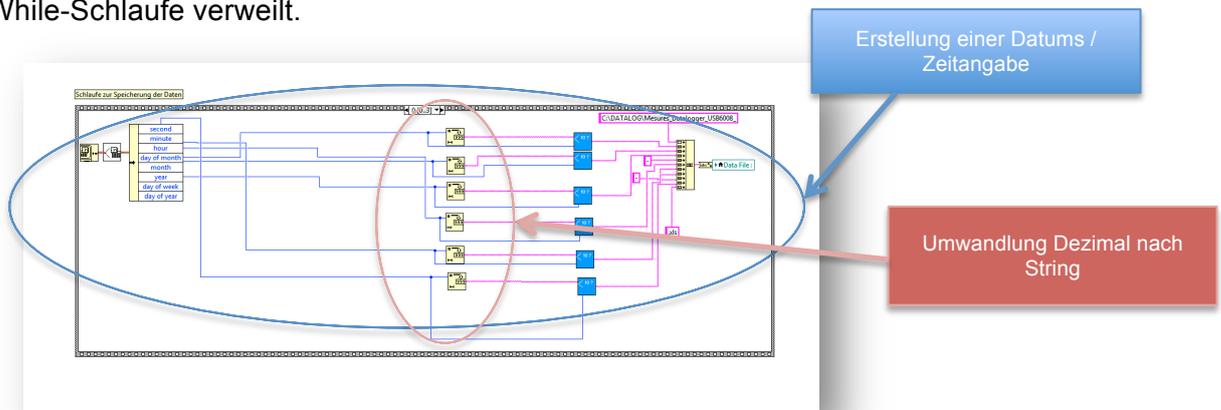


Abbildung 69 Datumserstellung

Die erste Schlaufe dient zur Erstellung einer Datumsanzeige welche später an den Namen des File's angefügt wird. Die ausgelesenen Daten werden erst in String's umgewandelt, und anschliessen zusammengefügt, so dass eine Zeile Text entsteht

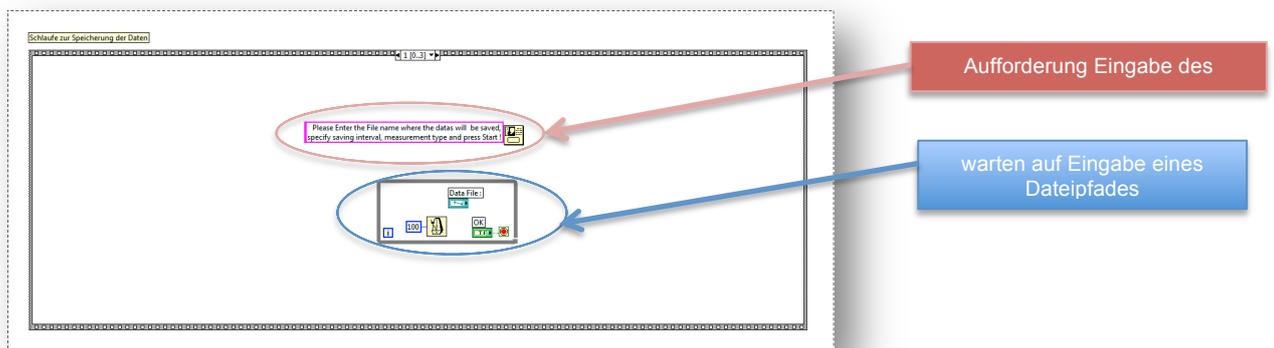


Abbildung 70 Eingabeaufforderung

Die 2te Sequenz fragt den Benutzer nach dem Speicherort des Logging-file's. Anschliessend wartet dieses Sequenz bis der Benutzer einen Speicherort ausgewählt hat und mit OK dessen Richtigkeit bestätigt.

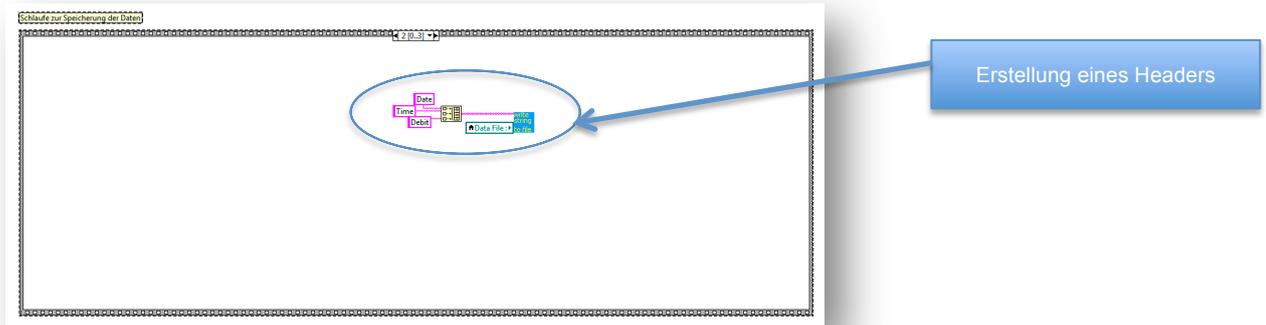


Abbildung 71 Erstellung Header

Damit die Daten bei einem erneuten Programmstart nicht direkt hintereinander angefügt werden, erstellt diese Sequenz bei jedem Programmstart eine neue Linie welche das vorhin erstellte Datum, die Zeit sowie den Namen des gemessenen Wertes in die erste Reihe schreibt. Diese erste Linie stellt den "Header" des Log-file's dar.

Die letzte Sequenz dient der eigentlichen Datenspeicherung. Sie enthält eine Endlos-Schleife welche durchgehend prüft ob Daten gespeichert werden sollen.

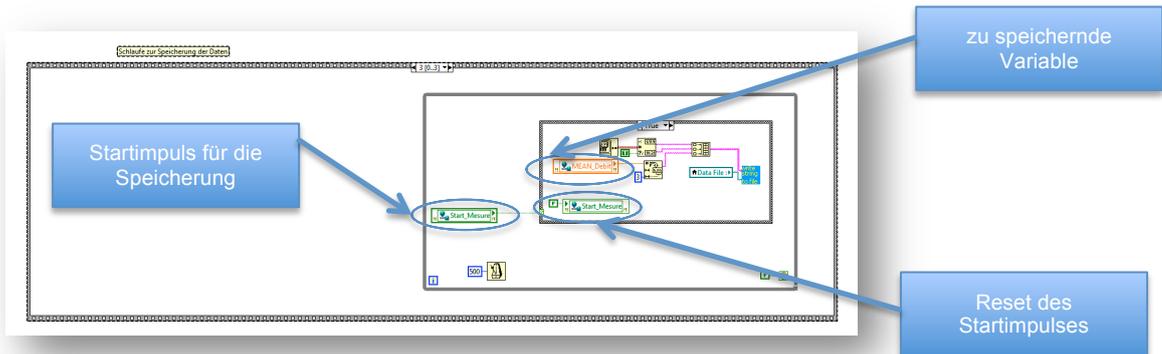


Abbildung 72 Speicherung von Daten

Damit nicht ständig alle Werte gespeichert werden, kann durch die Variable "Start\_Measure" eine Speicherung gestartet werden. Innerhalb der speichernden Schleife wird dieser Wert wieder zurückgesetzt damit die Schleife nur 1x ausgeführt wird.

### CPU

Im VI der CPU befinden sich 3 Schleifen. Diese Schleifen bestimmen die Funktionsweise und die Logik nach welcher die Anlage zu funktionieren hat. Der Code kann schnell geändert werden und ist sehr maschinennah.

Die erste Schleife holt alle Inputs / Outputs in die Umgebungsvariablen damit sie universell auf allen Ebenen zur Verfügung stehen. Zudem wird mit derselben Geschwindigkeit eine Case-Struktur welche auf Knopfdruck den Mittelwert aus einer beliebigen Anzahl Werten generiert und nach Wunsch abspeichern lässt.

Eine weitere Schleife kümmert sich um die Regelung. Aktuell wurde die Geschwindigkeitsregelung sowie eine übergeordnete Durchflussregelung implementiert. Die letzte Schleife bekommt die Anzahl Flanken der

Geschwindigkeitsmessung übermittelt und berechnet daraus die Geschwindigkeit in Hz und tr/min.

Das Frontpanel dient hauptsächlich dem Entwickler Aus diesem Grund findet man hier auch die nötigen Einstellungsmöglichkeiten für die Regler.

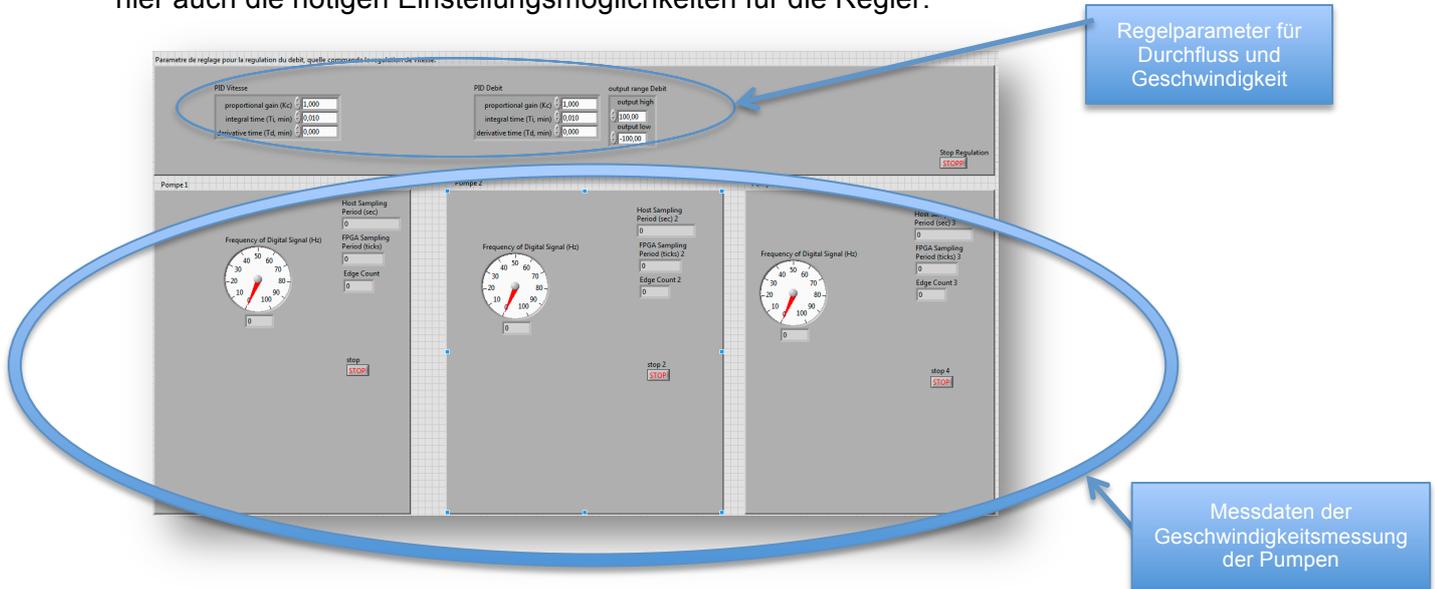


Abbildung 73 CPU Frontpanel

Die Darstellung dient zur Entwicklung, dem Testbetrieb und Kontrolle der Schaltung und deren Funktionsweise. Aus diesem Grund werden alle nötigen Informationen direkt angezeigt. Die PID Parameter werden aus praktischen Gründen ebenfalls auf dem Frontpanel angezeigt. Es bietet sich an, diese Ansicht bei den Tests zu nutzen.

Folgend, die Darstellung der ersten Schlaufe welche die I/O's je nach Herkunft, aus der FPGA oder direkt in die Umgebungsvariablen schreibt. Anzumerken ist, dass das mehrmalige Öffnen des Ports von CPU -> FPGA die Geschwindigkeit verringert. Aus diesem Grund wurde der Port hier nur 1 mal geöffnet, um alle Daten aus der FPGA in einem Zyklus zu holen.

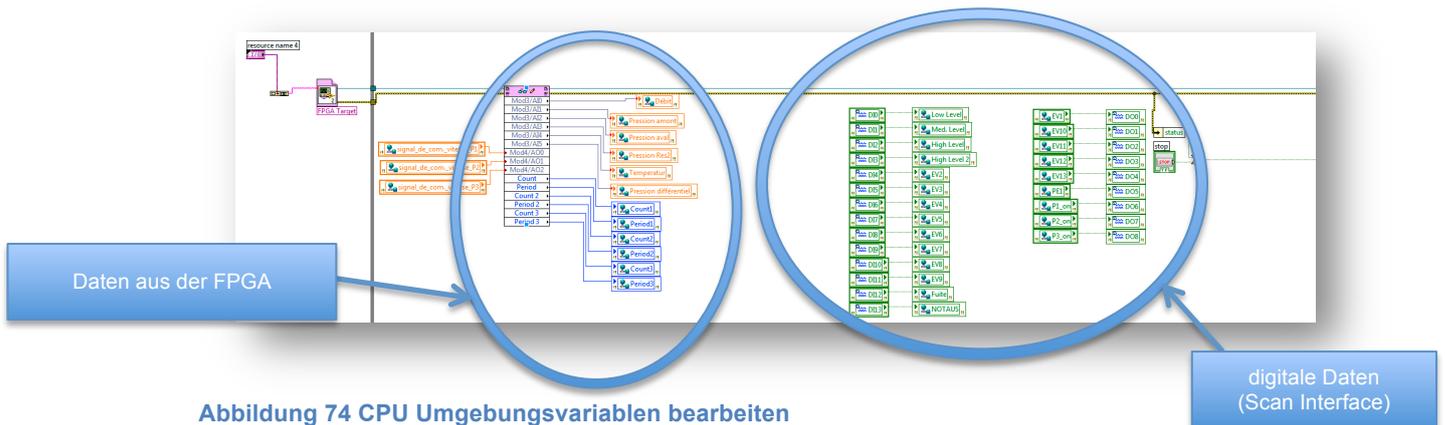


Abbildung 74 CPU Umgebungsvariablen bearbeiten

Die analogen Daten werden durch die FPGA geholt. Die digitalen Daten werden in einer Art Scanmodus betrieben. Der Grund für diese Separation ist, die analogen Daten benötigen eher eine Durchschnittsberechnung, welche je nach Geschwindigkeit der Daten, sehr schnell

erfolgen muss. Dies ist nur in der FPGA möglich. Die digitalen Daten sind meist einfache Ein-/Aus Angaben / Befehle, welche keine schnelle Postverarbeitung benötigen.

In derselben Schleife wird ebenfalls die Möglichkeit zur Erfassung der Daten - Mittelwertberechnung und nach Wunsch Weitergabe zur Speicherung, gegeben. Diese zwei Elemente (Übertragung der Werte in die Umgebungsvariablen sowie die Sammlung und Durchschnittsberechnung) in derselben Schleife zu programmieren hat den Vorteil dass die Werte Synchron übertragen werden. 2 einzelne Schleifen laufen unabhängig von einander und funktionieren asynchron. Die synchrone Datenübergabe zwischen 2 Schleifen benötigen die Implementation von Queues. Dies ist durchaus möglich, doch wurde für die Grundlagenprogrammierung auf den Einsatz von solchen Mitteln möglichst verzichtet werden.

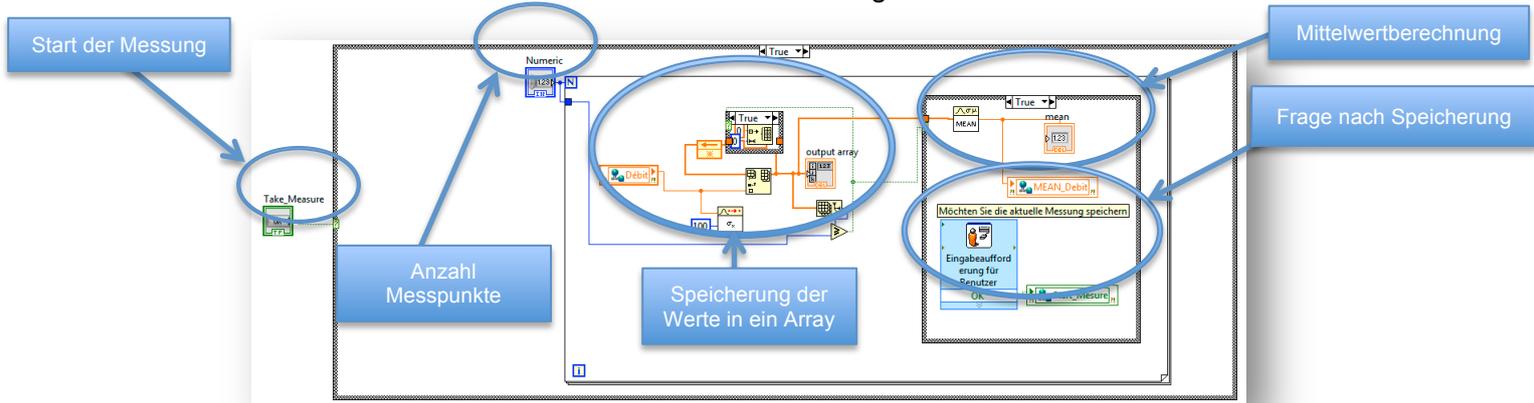


Abbildung 75 CPU Durchschnittsberechnung

Durch den Schalter "Take\_Measure" kann eine Sequenz von Werten, dessen Anzahl durch das Eingabefeld Numeric eingegeben werden kann, in eine Tabelle gespeichert werden. Diese Tabelle wird bei Erreichen der gewünschten Anzahl Elementen, an einen Funktionsblock der den Mittelwert berechnet (MEAN) weitergegeben. Das Case in welchem sich der "MEAN" -Block befindet, dient dazu, dass jeweils nur ein fertig berechneter Mittelwert zu berechnen ist. Ohne diese Case-Struktur würde der Mittelwert der Tabelle andauernd berechnet werden.

Der Mittelwert wird anschliessend direkt durch die Umgebungsvariable "MEAN\_Debit" anderen Stufen zur Verfügung gestellt. Während dieses Prozesses wird der Benutzer gefragt ob eine Speicherung dieses Wertes erwünscht ist. Die Bestätigung zur Speicherung der Variable durch den Speicheralgorithmus im VI des Computers "HMI", erfolgt durch die Umgebungsvariable "Start\_measure".

Die Schleife der Regelung besteht aus 2 Regelsystemen.

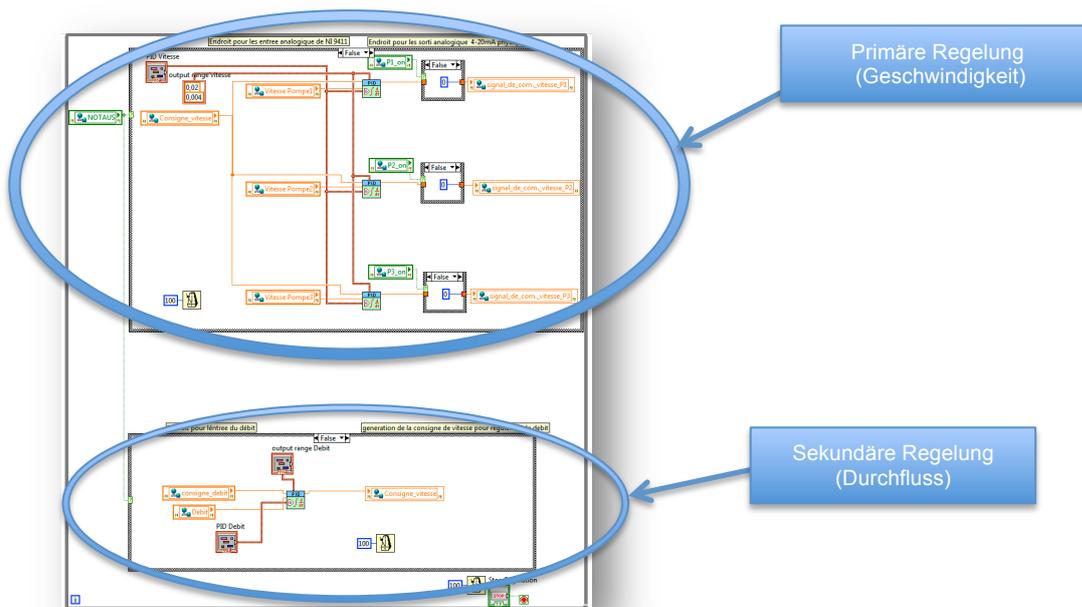


Abbildung 76 Regelung

Primär wird die Geschwindigkeit geregelt. Die Geschwindigkeit bietet die Basis aller übergeordneten Regelungen, da die Geschwindigkeit, das Einzige ist, was von jeder Pumpe gemessen werden kann.

Aufbauend auf der Geschwindigkeitsregelung befindet sich die Durchflussregelung. Der Regelausgang der Durchflussregelung ist der Eingang der Geschwindigkeitsregelung.

Für die Programmierung einer Regelung stellt Labview die nötigen Bausteine zur Verfügung. So wurde in diesem Fall für jede Pumpe ein PID-Glied verwendet. Die Eingänge des PID-Glieds bilden: die Vorgabe der Geschwindigkeit "Consigne\_vitesse", der Messwerts der Pumpengeschwindigkeit "Vitesse\_PompeX" sowie den Regelparametern und einer Ausgangsbegrenzung.

Eine andere Möglichkeit als die Verwendung eine Ausgangsbegrenzung, stellt die Verwendung von Prozentblöcken dar. So kann der Eingang sowie der Ausgang in Prozent umgewandelt werden, was eine Ausgangsbeschränkung unnötig macht.

Alle 3 Pumpen besitzen derzeit dieselben Parameter für die Regelung des PID-Reglers sowie dieselbe Geschwindigkeitsvorgaben. Die PID-Regler-Einstellungen werden auf allen Pumpen dieselben sein. Falls die Pumpen bei identischer Geschwindigkeit nicht gleich viel Wasser fördern, muss dies mit einer Messung pro Pumpe kontrolliert und allenfalls eine Korrektur der "Consigne\_vitesse" eingesetzt werden.

Eine weitere Schleife läuft parallel zu den 2 anderen mit. Sie berechnet aus der Zeit zwischen 2 Messpunkten, die gewünschte Geschwindigkeit.

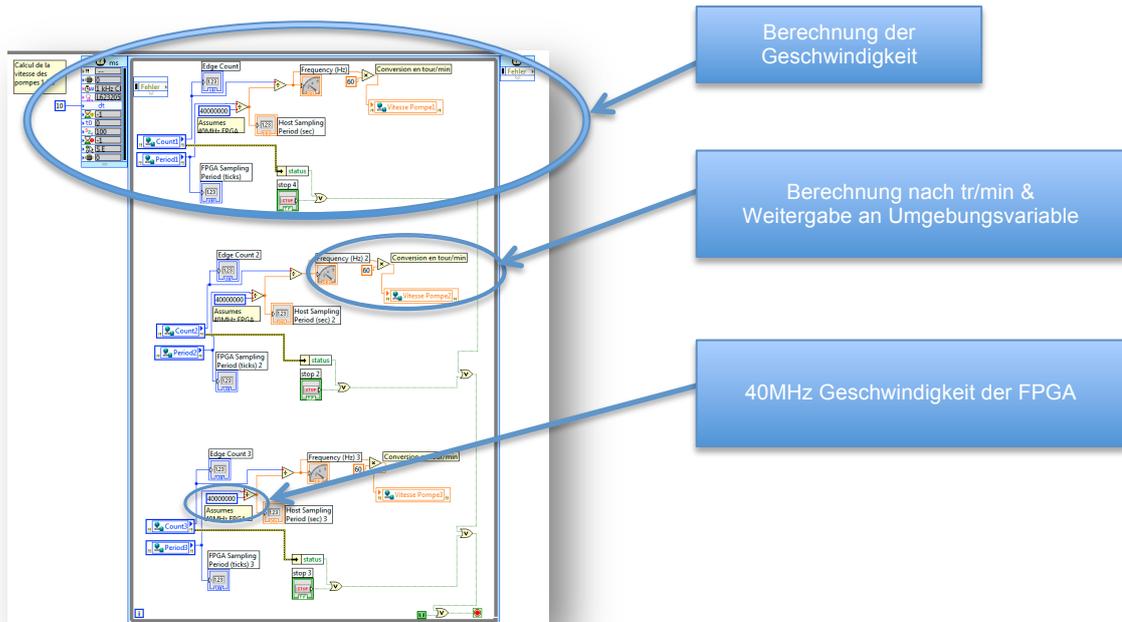


Abbildung 77 Geschwindigkeitssignalumwandlung

Anhand der Anzahl erfassten Flanken in "Count", der bekannten Geschwindigkeit der FPGA und der verstrichenen Zeit zwischen den 2 Counts, kann anhand dieses Algorithmus die Frequenz bestimmt werden.

Die Umwandlung der Frequenz in Umdrehungen pro Minute, kann durch eine Multiplikation mit 60 bewerkstelligt werden. Diese Rechnung funktioniert nur solange wie exakt 1 Punkt pro Umdrehung gemessen wird, daher darf auf der Welle nur 1 Reflektionspunkt angebracht werden.

### FPGA

Die FPGA stellt den schnellsten Teil der Systemkette dar. Sie arbeitet in einem Bereich von 0.0000001s, und damit ca. 1000 mal schneller als die CPU. Schnelle Tätigkeiten wie das präzise Erfassen von schnell aufeinanderfolgenden Punkten welche zu 100% erfasst werden müssen, erfordern den Einsatz der FPGA.

Das Frontpanel der FPGA dient lediglich zur schnellen Kontrolle der Funktion der FPGA.

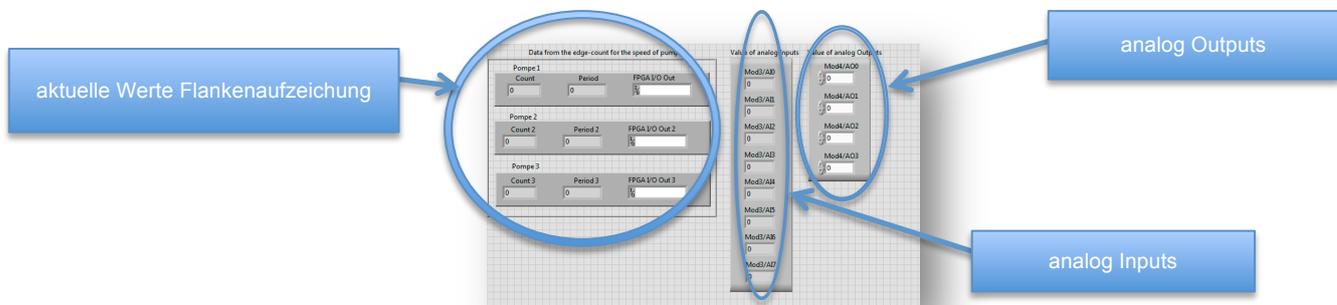


Abbildung 78 FPGA Frontpanel

Der aufwändigste Teil der FPGA, stellt die Erfassung der Flanken dar.

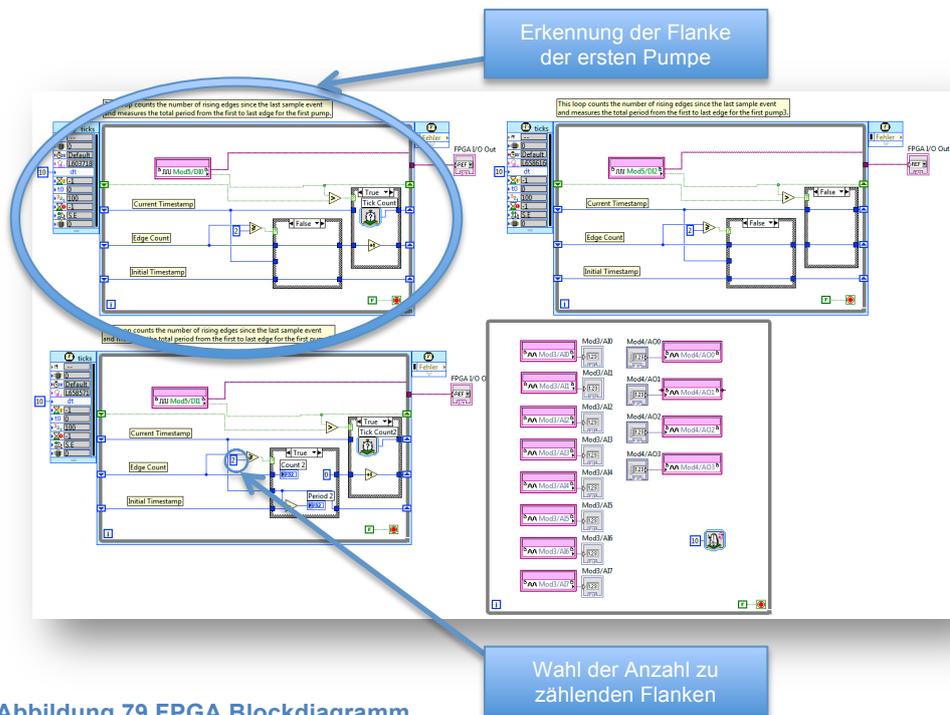


Abbildung 79 FPGA Blockdiagramm

Dieses Counterprogramm war ursprünglich zur Messung mit einer sich selbst anpassenden Anzahl von Flanken ausgerüstet. Da jedoch bei tiefen Umdrehungen die Anzahl Flanken bis auf eine einzelne reduziert wurde, ist das Programm so umgeändert worden, dass immer 2 Flanken gezählt werden müssen. Dadurch kann die Präzision bis auf min. 10Hz gewährleistet werden. Durch diese Änderung nimmt man in Kauf, dass bei ganz schnellen Taktfolgen, der Counter nicht mehr funktioniert. Dieser Fall wird bei der Messung der Pumpenumdrehungen (max 50-60Hz nicht passieren, da das Limit bei 2MHz liegt. (s. Tests)  
 Der Counter funktioniert mittels mehrerer Schieberegister, welche den "alten" Wert, bei der nächsten Iteration wieder am Anfang der Schleife zur Verfügung stellt.

## Tests

### Test der Geschwindigkeitsmessung

Aufbau:

Das Sensorsignal wird durch den Frequenzgenerator simuliert.  
 Kontrolliert wird das Signal ebenfalls durch das Oszilloskop Tectronicx DPO 2024.  
 Messergebnisse:

Frequenz : 10 Hz  
 Amplitude : 4V  
 Duty Cycle : 20%  
 Alle 3 Eingänge mit identischem Signal.

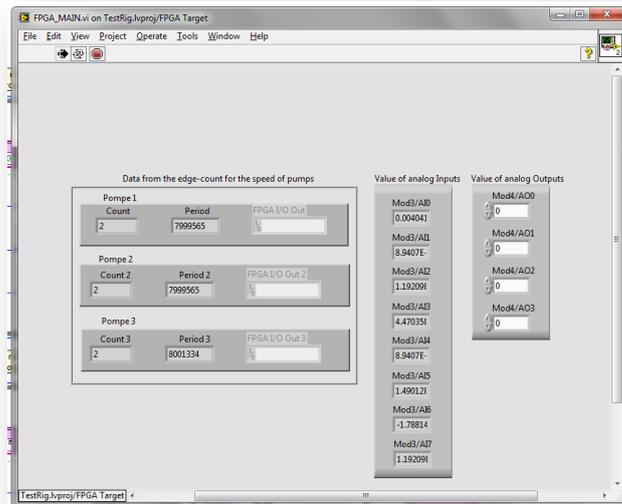


Abbildung 80 FPGA Funktionskontrolle

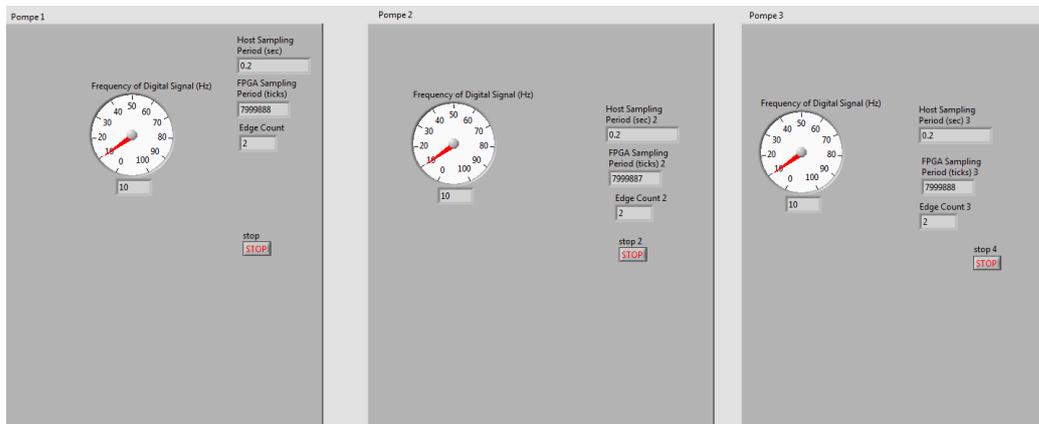


Abbildung 81 Test Geschwindigkeitsmessung 10Hz

Durch die Begrenzung auf 2 Counts, ist eher das obere Limit beschränkt. Daher ein max. Test, um sicherzustellen dass unsere Pumpe im funktionierenden Bereich des Counters funktioniert.

Das nächste Bild dokumentiert die korrekte Weiterleitung von  $10 \text{ Hz} * 60 = 600 \text{ Rpm}$ .

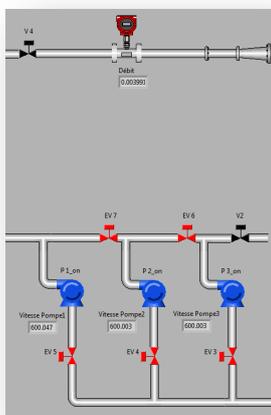


Abbildung 82 Test Fontpanel

Frequenz : von 0 - max. Anzeigefolge  
 Amplitude : 4V  
 Alle 3 Eingänge mit identischem Signal.

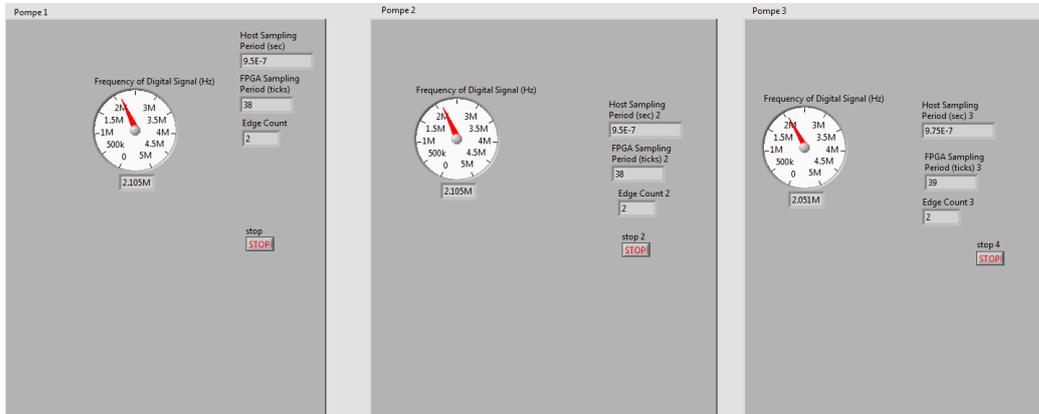


Abbildung 83 Test Geschwindigkeitsmessung 2.105 Mhz

Bei diesem Test wurde eine maximale Geschwindigkeit von ca. 2 Mhz realisiert. Dies ist ein sehr guter Wert und zeigt, dass die Pumpen keine Belastung für den Zähler darstellen.

### Test der Speicherefunktion in eine Exceldatei

Aufbau:

Verwendung eines stabilen Signals am Eingang, um den Messwert zu stabilisieren. Anschliessend wird ein Messdurchgang gestartet und das Excel File mit der abgespeicherten Variable verglichen.

Testergebnisse :

Die Funktion der Speicherung in eine Tabelle verlief tadellos. Für diesen Test wurde ein Excel File für den Bericht angefertigt, damit die Einfügung des Headers in die Tabelle angezeigt werden kann.

zu speichernder Messwert

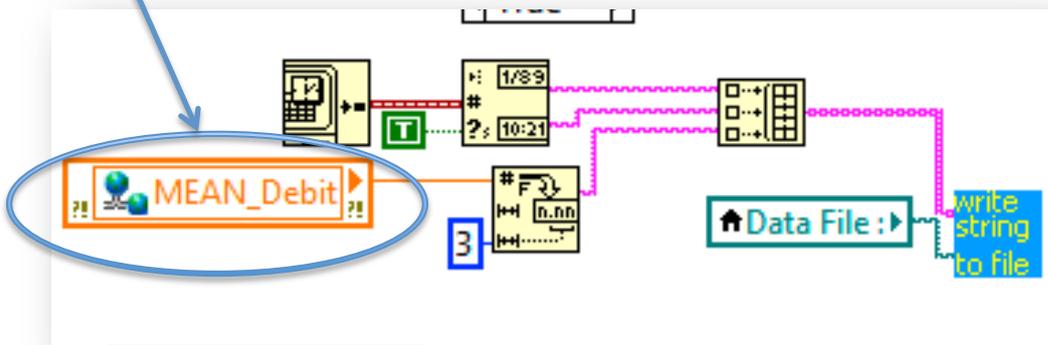


Abbildung 84 Speicherung in Excel-File

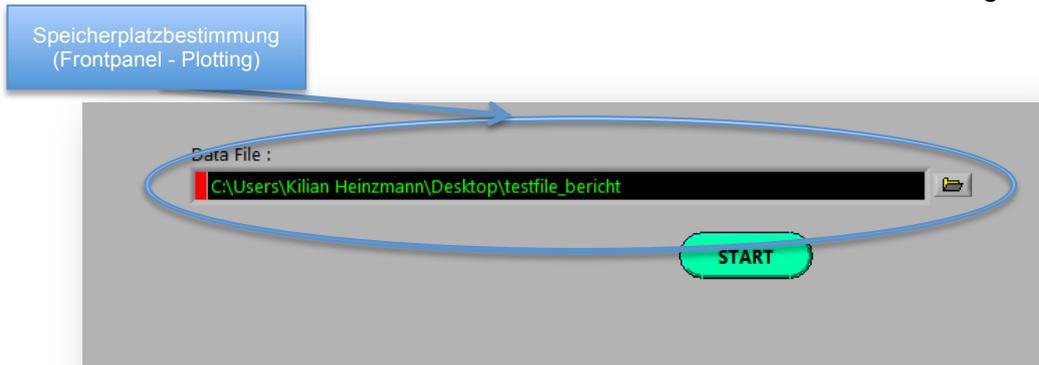


Abbildung 85 Frontpanel - Dateipfad

Das erstellte File wurde anschliessen mit Microsoft Excel geöffnet :

	A	B	C
	Date	Time	Debit
	08.07.12	18:50:56	0,004
	Date	Time	Debit
	08.07.12	18:51:11	0,004

Abbildung 86 beschriebenes Excel-File

Das Excel-File zeigt den Exakten Wert von 4mA, welche durch den Drucksensor an den Eingang abgegeben worden sind.

Die einzelnen Schaltungen wurden während des Aufbaus immer wieder mit simulierten Eingängen getestet, so dass die Funktion nach der Einfügung ins Projekt gegeben ist.

## Verbesserungen

Eine Verbesserungsmöglichkeit an der Grundlagenprogrammierung wäre der Einsatz eines "FIFO's" (First Input - First Output) was eine bedeutend höhere Übertragungsrate zwischen FPGA und CPU zulassen würde.

Designtechnisch wäre ebenfalls ein passenderes "Tanklayout" auf dem Frontpanel angebracht.

## Schlussfolgerung

Die beginnende Sensorauswahl, welche durch mechanische Befestigungsmöglichkeiten, elektrische Anschlussmöglichkeiten und von Datasheet-Vergleichen geführt wurde, zeigte die aufwändige Planung die ein solches Projekt selbst bei der grundlegenden Auswahl der Komponenten nötig ist. Sehr viele verschiedenen Elemente beeinflussen die Sensorauswahl, bspw. der Prozessdruck spielt bei verschiedenen Messsystemen eine Rolle.

Die Semesterarbeit war eine gute Vorbereitung auf die Diplomarbeit, trotzdem gab es viele kleine Änderungen bezüglich der Materialauswahl aufgrund von neuem Wissen. Die Semesterarbeit ist im Anhang angefügt.

Die anschliessende Auswahl des Automationssystems verlief ähnlich wie die Auswahl der Sensoren. Die Geschwindigkeiten, Anschlussmöglichkeiten, Anzahl Ein- / Ausgänge und der Preis waren die hauptsächlichlichen Vergleichspunkte unter den Karten.

Die Planung der Geschwindigkeitsmessung, bei welcher bspw. kein induktiver Näherungssensor verwendet werden konnte aufgrund des sich sehr nah befindenden Motors, wie auch der schwierigen Montage einer Erhebung auf der Welle.

Überdies müsste auf beiden Seiten der Welle, exakt symmetrisch eine Erhebung angebracht werden, um Unwuchten zu verhindern. Aus diesem Grund wurde ein optischer Sensor ausgewählt. Der optische Sensor liefert jedoch ein Ein-/Aus Signal, für welches ein spezielles Konvertierungstool gekauft werden müsste. Jedoch kostet eine schnelle digitale Automationskarte weniger als 1 Konvertierungstool, und kann zudem bis zu 6 Quellen erfassen.

Während der Diplomarbeit konnte ich mein fachliches Wissen im Bereich der Planung und Elektroschema zeichnen, vertiefen und weiterentwickeln. Das Erlernen und Einarbeiten in die Hard-/Software von National Instruments (Labview) war eine grosse Bereicherung für meine berufliche Zukunft. Die Möglichkeit ein Softwareprojekt von den erforderlichen Elementen bis hin zur Programmierung auf 3 verschiedenen Hardwareebenen mitentwickeln zu können, waren eine lehrreiche Erfahrung im technischen Bereich, wie auch im Bereich der Teamarbeit.

## Danksagung

Ich möchte mich an dieser Stelle bei Frau Dr. Münch Cécile und Herrn Ing. Salamin Joseph bedanken, die mich bei meiner Diplomarbeit unterstützt, mir mit wertvollen Hinweisen und Tipps zur Seite gestanden und als Gutachter meiner Arbeit fungiert haben. Mein Dank geht auch an Ing. Meier Bastien und Dr. Hasmatuchi Vlad, die mir durch ihr Fachwissen und ihre Anregungen sehr oft weitergeholfen haben.

Ebenfalls bedanke möchte ich mich bei Herr Vacari Aldo, welcher mir als Spezialist im Bereich Automation (Labview) bei allen Fragen im Bereich der Labviewprogrammierung zur Verfügung stand.

Zum Schluss möchte ich mich bei Sozialpädagogin Andrea Jaeger bedanken, die sich bereit erklärte meine Diplomarbeit gegenzulesen und sich nicht scheute Kritik zu üben.

## Datum & Unterschrift

9. Juli 2012

Kilian Heinzmann

---

---

## Anhänge

## **Elektroschema**

---

**Liste der Ein/Ausgänge**

---

## **Semesterarbeit**

---