**Disposition zur Masterarbeit**

Verwaltung von Ventiltests

Student: Andreas Stucki

Betreuer: Marc-André Bumann

Datum: 24.02.2017

Inhaltsverzeichnis

[1 Projektbeschreibung / Ausgangslage 3](#_Toc475293055)

[1.1 Situation 3](#_Toc475293056)

[1.2 Problemstellung 3](#_Toc475293057)

[1.3 Ist-Zustand 3](#_Toc475293058)

[1.3.1 Stärken 5](#_Toc475293059)

[1.3.2 Schwächen 5](#_Toc475293060)

[2 Zielsetzung 5](#_Toc475293061)

[2.1 Quantitative Ziele 5](#_Toc475293062)

[2.2 Qualitative Ziele 5](#_Toc475293063)

[3 Aufgabenabgrenzung 5](#_Toc475293064)

[4 Vorgehenssystematik 6](#_Toc475293065)

[4.1 Milestones 6](#_Toc475293066)

[4.2 Zeitplan 6](#_Toc475293067)

[5 Resultate 6](#_Toc475293068)

[6 Glossar 6](#_Toc475293069)

[7 Abbildungsverzeichnis 6](#_Toc475293070)

# Projektbeschreibung / Ausgangslage

## Situation

Die Firma VAT stellt Vakuumventile her. Der Controller übernimmt die Steuerfunktion dieser Vakuumventile. (siehe Abb. 1) Dieser ist modular aufgebaut und besteht grob gesagt aus drei Komponenten. Die zentralen Elemente sind auf dem Masterboard bestückt, welches unabdingbar aber angepasst an die jeweilige Ventilhardware ist. Das Herzstück des Controllers ist der Mikrokontroller, für welche VAT eine eigene Firmware entwickelt. Weiter ist oder sind Motorbausteine nötig, welche eine weitere Firmware von unserem Lieferanten benötigt.

Das Interface Board ist nach Kundenwunsch angefertigt. Arbeitet der Kunde mit einem Feldbus System, so wird eine Interface Software nötig. Die dritte Komponente ist die Option Unit, welche Zusatzfunktionen beinhaltet.



Abbildung 1: Basiskonzept Ventil Controller (Marugg, 2010)

## Problemstellung

Ich wurde von der Firma VAT eingestellt und beauftragt die Qualität des Controllers mit Hilfe einer Testumgebung zu überprüfen. Mit dem Programm TTIC2 entwickelte ich eine Testoberfläche, welche einzelne Tests in eine Kollektion zusammenfasst und nacheinander ausführt.

Aktuell muss nach Ausführung der Testkollektion das Report File nach fehlerhaften Testdurchläufe durchsucht werden. Das Report File enthält alle erfolgreichen Testschritte und erreicht eine sehr grosse Datenmenge. Aus diesem Grund wird meistens nur der letzte Report vor einer Ventil Firmware Freigabe im SVN abgelegt. Weiter werden auch Zertifizierungsfiles abgelegt. Dieses enthält das Datum des zuletzt erfolgreichen Durchlaufs des Tests.

## Ist-Zustand

Das Tool TTIC2, welches die Grundlage dieser Arbeit bietet, wurde eingeführt um Ventil Firmwaren zu qualifizieren. Mit Hilfe der Testoberfläche des TTIC2 werden einzelne Ventiltests zu einer Kollektion zusammengefasst und ausgeführt (siehe Abb. 1). Die Ventiltests sind in CVI geschrieben. Der Grund liegt in der einfachen ansprechen der Hardware über die National Instruments Komponenten.

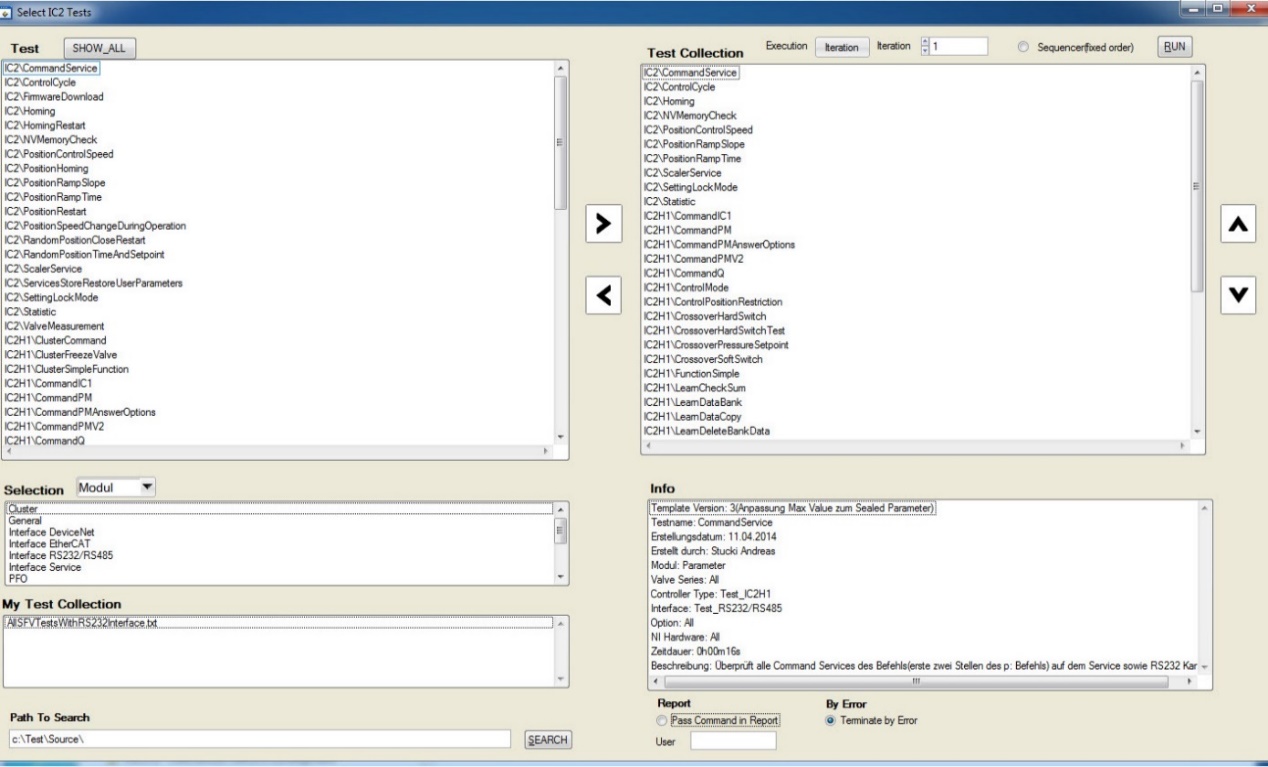


Abbildung 2: Ansicht der TTIC2 Oberfläche für die Auswahl der Testkollektion

Während die Testkollektion abläuft zeigt das Reportfenster (siehe Abb. 2), den aktuellen Test sowie welche die bereits ausgeführt worden sind sowie noch werden. Die Testergebnisse werden laufend notiert. Der User kann die Testkollektion jederzeit abbrechen.

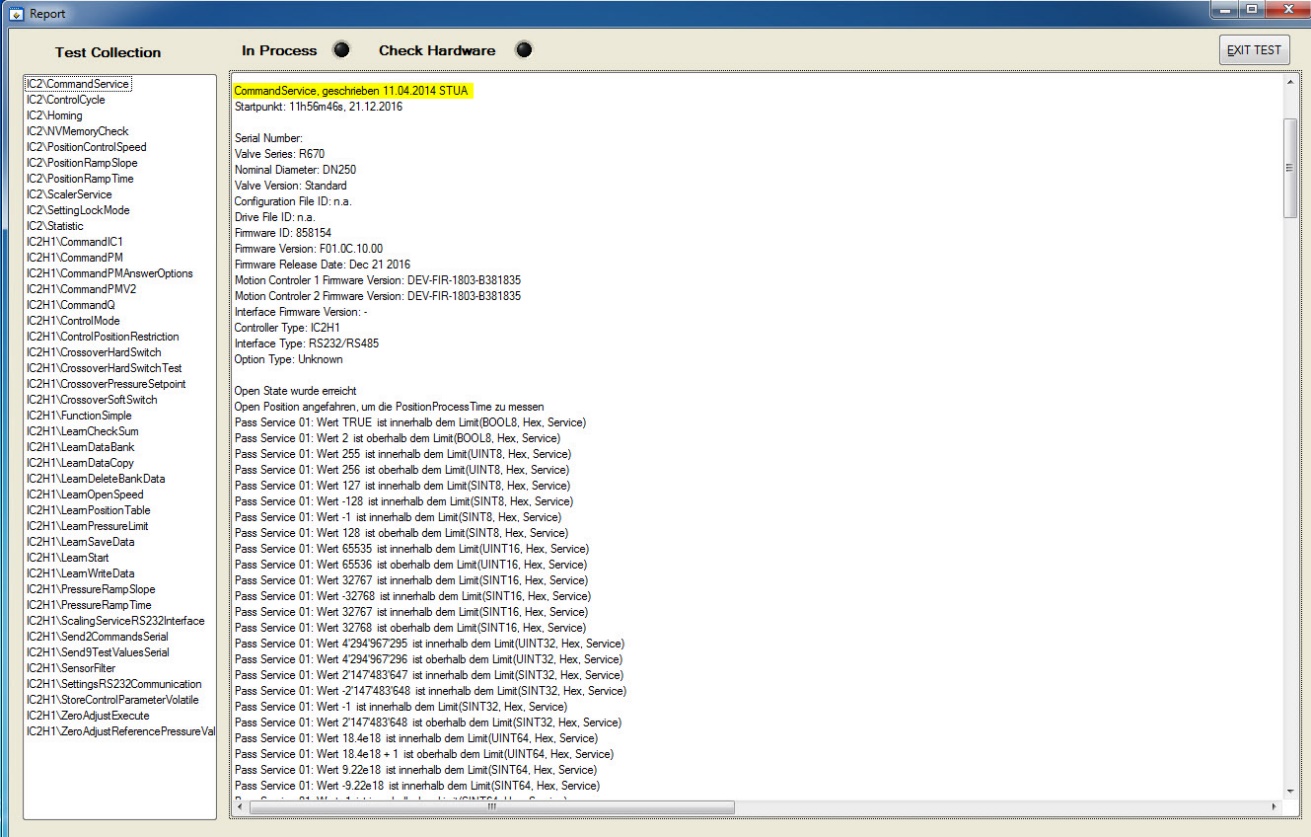


Abbildung 3: Report Ansicht währendem die Tests ausgeführt werden

Nach Ausführung der Testkollektion kann der entstandene Report an einem gewünschten Ort abgespeichert werden. Aus Gründen der grossen Datenmenge wird der Report nicht automatisch abgespeichert. Weiter entstehen Evaluierungsdokumente bei jedem erfolgreichen Test ein Zertifizierungsfile und bei fehlgeschlagenen ein Diagnostikfile.

### Stärken

* Auf einem Blick sind alle Funktionalitäten der Oberfläche ersichtlich (keine Verschachtelungen)
* Die verfügbaren Tests können nach Hardware Eigenschaften gefiltert werden
* Wird ein einzelner Test angewählt, so wird eine Beschreibung des Tests sowie die Hardware Anforderungen angezeigt
* Es können Testkollektionen abgespeichert werden
* Fortlaufender Report wird auf der Oberfläche angezeigt und im Hintergrund in einem Textfile hinterlegt
* Automatische Generierung von Zertifizierungsfile im fehlerfreien Ablauf eines Tests sowie bei auftretenden Fehler eines Diagnostikfiles

### Schwächen

* **Das Programm wird auf mehreren Rechnern ausgeführt**
  + Erschwerte Auswertung der Tests
  + Die abgespeicherten Testkollektionen sind nur auf dem jeweiligen Rechner sichtbar
* **Keinen Verlauf der Testergebnisse ersichtlich über verschiedene Ventil Firmwaren**
  + Letzter Report File wird im Subversion abgelegt
  + Keine schnelle Suche ob der Testfehler schon einmal aufgetreten ist
  + Fehlermeldung nur im Reportfile ersichtlich, welches eine enormen Datenmenge enthält
* Keine Sicherstellung des Grundzustandes
  + Ventil Firmware, Motion Controller Firmware, Interface Firmware und Ventileinstellungen bei Bedarf verändern
* Auslesung der Testergebnisse geschieht im Report File (manuelle Suche nach Fehlern)

# Zielsetzung



Abbildung 4 Konzept Masterarbeit

Die Testergebnisse sollen in der SoftwareVersionsDatabase (SQL, Structured Query Language) hinterlegt werden.

Diese Datenbank besteht bereits und hat die einzelnen Firmware Inhalte abgespeichert.

Erweitert soll sie mit:

* Name der Testkollektion
* Datum der Ausführung
* Anzahl Fehlerhafte Tests
* Resultat der einzelnen Tests
* Bei einem fehlerhaften Test eine kurze Fehlermeldung sowie ein Diagnostikfile

Weiter wird mit folgenden Informationen der Grundzustand des Testablaufs festgelegt:

* Ventil Firmware
* Motion Controller Firmware
* Interface Firmware
* Antriebsfile
* Konfigurationsfile

Es soll das Programm ETIC2 erstellt werden, welches eine Übersicht über die Ergebnisse der einzelnen Testkollektionen gibt. Das ETIC2 listet die einzelnen Testkollektionen unter einem definierten Grundzustand auf. Das RTIC2 soll zudem noch eine Report Funktion beinhalten, welche zum definiertem Grundzustand alle Ergebnisse rapportiert.

## Quantitative Ziele

* Die Datenbank muss Schreibanfragen von vier gleichzeitigen Benutzern bearbeiten können.
* Jede einzelne Testkollektion muss im ETIC2 zu einem Grundzustand zugeordnet worden sein.
* Der Grundzustand kann nur mit bereits vorhandenen Einträgen in der SoftwareVersionsDatabase definiert werden.

## Qualitative Ziele

* Unter Angabe des Namens des Grundzustandes muss auf Knopfdruck im ETIC2 eine Auswertung über alle ausgeführten Testkollektionen mit den dazugehörigen Resultaten aufgelistet werden.
* Das ETIC2 soll eine ausgeprägte Suchfunktion enthalten, welche eine schnelle Suche nach Fehlermeldungen erlaubt.
* Mit dem ETIC2 soll das Resultat der ausgeführten Testkollektion schnell und einfach ersichtlich sein.

# Aufgabenabgrenzung

* Die Weiterentwicklung der TTIC2 Applikation ist nicht Teil der Masterarbeit.
  + Ausgenommen ist die Anbindung der Datenbank
* Keine Anpassungen an den einzelnen Tests

# Vorgehenssystematik

Die Arbeitsdokumente sowie der Code wird auf Github abgelegt. Der Grund liegt in dem verteilten Zugriff der Dokumente von verschiedenen Geräten sowie die Versionierung des Codes.

|  |  |  |
| --- | --- | --- |
| **Projektplan** | **Was** | **Wie** |
| SoftwareVersionsDatabase Modellierung | Datenbanktabellen und ihre Attribute definieren | Sql Skript erstellen und in SQL Management Studio ausführen |
| TTIC2 Zugriff auf SoftwareVersionsDatabase | Speicherung der Testresultate + Auswahl Grundzustand | Anbindung der SoftwareVersionsDatabase mit Entity Framework |
| Erstellung ETIC2 |  |  |
| * Design View Model | Auslegung der ETIC2 Oberfläche | Präsentationlogik in WPF erstellen |
| * Codierung nach MVVM | Programmierung der ETIC2 Logik | Mit dem Entwurfskonzept MVVM in C# |
| * Anbindung SoftwareVersionsDatabase | Auslesung der Testresultate | Auslesung der SoftwareVersionsDatabase mit Entity Framework |
| * Ausgabe Bericht | Erstellung Testreport in PDF Format | Noch nicht klar |
| * Unit Test | Validierung des ETIC2 | Mit Hilfe des Unit Test Framework |
| * Reserve | - | - |
| Schriftliche Arbeit | Beschreibung der Arbeitsschritte | Wird in Word geschrieben |
| Fertigstellen der schriftlichen Arbeit | Dokumentation der Resultate | Wird in Word geschrieben |

Tabelle 1 Vorgehenssystematik

## Wirtschaftliche Aspekte

In zwei Bereichen können durch die Erstellung des ETIC2 kontinuierlich Zeit eingespart werden. Erstens ist das Auslesen der einzelnen Testresultate nach Ablauf einer Testkollektion auf einen Blick möglich. Zudem ist die Erstellung eines Testreports auf einen Knopfdruck beschränkt.

Weiter kann über die Historisierung der Testresultate besser einen auftretenden Fehler eingegrenzt und analysiert werden.

## Zeitplan

|  |  |  |  |
| --- | --- | --- | --- |
| **Projektplan** | **Start** | **Ende** | **Aufwand [h]** |
| Einreichen der Disposition | 28.02.2017 | 28.02.2017 |  |
| Start Masterarbeit | 01.05.2017 | 01.05.2017 |  |
| SoftwareVersionsDatabase Modellierung | 01.05.2017 | 05.05.2017 | 15 |
| TTIC2 Zugriff auf SoftwareVersionsDatabase | 08.05.2017 | 21.05.2017 | 30 |
| Erstellung ETIC2 | 22.05.2017 | 13.08.2017 | 155 |
| * Design View Model | 22.05.2017 | 25.05.2017 | 10 |
| * Codierung nach MVVM | 26.05.2017 | 09.07.2017 | 70 |
| * Anbindung SoftwareVersionsDatabase | 26.05.2017 | 09.07.2017 | 5 |
| * Ausgabe Bericht | 10.07.2017 | 16.07.2017 | 20 |
| * Unit Test | 17.07.2017 | 30.07.2017 | 30 |
| * Reserve | 31.07.2017 | 13.08.2017 | 20 |
| Schriftliche Arbeit | 28.02.2017 | 07.08.2017 | 80 |
| Fertigstellen der schriftlichen Arbeit | 14.08.2017 | 31.08.2017 | 40 |
| Abgabe der Masterarbeit | 31.08.2017 |  |  |
|  |  |  |  |
| Projektarbeit | 28.02.2017 | 31.08.2017 | 320 |

Tabelle 2 Zeitplan

# Resultate

# Glossar

|  |  |
| --- | --- |
| TTIC2 | Test Tool Integrierter Controller Generation 2: Testoberfläche für alle integrierten Ventilcontroller der 2ten Generation |
| ETIC2 | Evaluation Tool Integrierter Controller Generation 2. Auswertungsoberfläche für die Testkollektionen |
| IC | Integrierter Controller: Der Controller, die Steuereinheit, befindet sich direkt beim Vakuumventil. |
| CVI | Ist eine ereignisorientierte Programmiersprache, welche auf C basiert und von National Instruments entwickelt wurde. . |
| Diagnostik File | Enthält alle Ventilparameter mit ihren aktuellen Werten. Zur genaueren Auswertung eines Fehlers. |
| SVN | Apache Subversion, freie Software zur Versionsverwaltung. |
| Konfigurations-  file | Enthält alle Abweichungen der Software gegenüber den Standard Ventil Firmware Einstellungen, welche in der Firmware hinterlegt sind. |
| Antriebsfile | Enthält alle Ventilhardware spezifische Abweichungen gegenüber den Standard Einstellungen, welche in der Firmware hinterlegt sind. |
| Grundzustand | Der Grundzustand setzt sich aus der Angabe der Ventil Firmware, Motion Controller Firmware sowie Optional der Interface Firmware, Antriebsfile sowie Konfigurationsfiles zusammen. Jeder Grundzustand erhält einen eindeutigen Namen. |
| WPF | Windows Presentation Foundation |
| MVVM | Mode View ViewModel |

# Abbildungsverzeichnis und Tabellenverzeichnis

[Abbildung 1: Basiskonzept Ventil Controller (Marugg, 2010) 3](#_Toc475804485)

[Abbildung 2: Ansicht der TTIC2 Oberfläche für die Auswahl der Testkollektion 4](#_Toc475804486)

[Abbildung 3: Report Ansicht währendem die Tests ausgeführt werden 4](#_Toc475804487)

[Abbildung 4 Konzept Masterarbeit 5](#_Toc475804488)

[Tabelle 1 Vorgehenssystematik 6](#_Toc475804517)

[Tabelle 2 Zeitplan 7](#_Toc475804518)

# Literaturverzeichnis

Marugg, L. (2010). PG\_Info\_Hardware. VAT Interne Präsentation. Haag