

**Dokumentation IPA**  
 EOL Kapazitiver Tankfüllstandsgeber

Auszubildender Informatiker Fachrichtung Applikationsentwicklung Hallauer Joel

# Inhaltsverzeichnis

[1. Inhaltsverzeichnis 1](#_Toc36477195)

[2. Glossar 3](#_Toc36477196)

[3. Vorwort 4](#_Toc36477197)

[3.1 Zweck des Dokuments 4](#_Toc36477198)

[3.2 Allgemeines 4](#_Toc36477199)

[Verwendete Tools 4](#_Toc36477200)

[Vorgehensmodell 4](#_Toc36477201)

[4. Informieren 5](#_Toc36477202)

[4.1 Ausgangslage 5](#_Toc36477203)

[Ist-Stand 5](#_Toc36477204)

[Soll-Stand 5](#_Toc36477205)

[Anforderungen 5](#_Toc36477206)

[4.2 Ziele 6](#_Toc36477207)

[Muss-Ziele 6](#_Toc36477208)

[Kann-Ziele 6](#_Toc36477209)

[5. Planen 7](#_Toc36477210)

[5.1 Terminplan 7](#_Toc36477211)

[5.2 Meilensteine 8](#_Toc36477212)

[5.3 Ablauf EOL 9](#_Toc36477213)

[5.4 Vordefinierte Testfälle 10](#_Toc36477214)

[Kommunikation 10](#_Toc36477215)

[EOL 10](#_Toc36477216)

[5.5 Prüfplan 11](#_Toc36477217)

[Testing aller Tests 11](#_Toc36477218)

[Einzeltest 11](#_Toc36477219)

[6. Entscheiden 12](#_Toc36477220)

[6.1 Ablaufdiagram 12](#_Toc36477221)

[7. Realisieren 13](#_Toc36477222)

[7.1 Coding 13](#_Toc36477223)

[1.1.1 Log-Datei 13](#_Toc36477224)

[8. Kontrollieren 14](#_Toc36477225)

[8.1 Prüffälle 14](#_Toc36477226)

[T001 - Auf Prüfdatei warten 14](#_Toc36477227)

[9. Auswerten 15](#_Toc36477228)

[9.1 Reflexion 15](#_Toc36477229)

[Allgemein 15](#_Toc36477230)

[10. Abbildungsverzeichnis 16](#_Toc36477231)

# Glossar

|  |  |
| --- | --- |
| Fachwort | Erklärung |
| DUT | Device Under Test: Prüfling |
| EOL | „End of Line“ Endkontroll Software, welche ein Gerät auf seine Korrektheit in jeder Hinsicht prüft. |
| \*.gui Datei | Userinterface Datei welche von der verwendeten Entwicklungsumgebung CVI verwendet wird. |
| NMEA2000 | NMEA2000 ist ein Netzwerktyp, welcher von der veratron zur Datenübertragung in verschiedenen Boots-komponenten eingesetzt wird. |
|  |  |

# Vorwort

## Zweck des Dokuments

Dieses Dokument dient dazu, die Schritte in der praktischen Arbeit von Joel Hallauer nachvollziehbar darzulegen. Es werden alle Arbeitsschritte sowie Gedankengänge zur Arbeit «EOL für Kapazitiver Tankfüllstandsgeber» dokumentiert und beschrieben.

## Allgemeines

### Verwendete Tools

|  |  |
| --- | --- |
| Tool | Verwendung |
| SlickEdit Pro (v 19.0.11) | Codeing, Codegestaltung |
| CVI Labwindows (v 15.0.1) | Kompilieren des Codes, GUI Gestaltung |
| Notepad++ (v 7.5.9) | Temporäre Notizen |
| GanttProject (v 2.8.5) | Terminplan |
| Microsoft Word | Dokumentation und Aufgabenstellung |

### Vorgehensmodell

Für die Arbeit zur EOL für einen Kapazitiven Tankfüllstandsgeber wird das Vorgehensmodell IPERKA verwendet.

**Informieren:**Was für Ziele wurden definiert?  
Wie sieht die Ausgangslage aus?  
Welche Mittel stehen zur Verfügung?

**Planen:**Wann sollen die Arbeiten erledigt werden (Terminplan/Meilensteine)?  
Was muss getestet werden?

**Entscheiden:**Ablauf der EOL bestimmen

**Realisieren:**Programmierung der KommunikationProgrammierung der EOL

**Kontrollieren:**Testen der Kommunikation  
Testen der EOL

**Auswerten:**Reflexion

# Informieren

## Ausgangslage

### Ist-Stand

Aktuell besteht bereits eine EOL-Vorrichtung für analoge Tankfüllstandgeber. Jedoch ist diese Vorrichtung Steuerungs- und Softwaremässig veraltet. Für die Steuerung sind bereits keine Ersatzteile mehr erhältlich. Auf der EOL-Vorrichtung können lediglich zwei analoge Tankfüllstandsgeber getestet werden:

* **Tankfüllstandsgeber Source-Mode:**  
  liefert je nach Füllstand des Tankes, eine Spannung, welche vom Anzeigeinstrument ausgewertet wird und dann entsprechend anzeigt
* A close up of a blue wall

  Description automatically generatedA picture containing indoor, table, desk, sitting

  Description automatically generated**Tankfüllstandsgeber Sink-Mode:**  
  verbraucht je nach Füllstand des Tankes, einen Strom. Dieser Stromverbrauch wird vom Anzeigeinstrument ausgewertet und entsprechend angezeigt

Abbildung 1 kapazitiver Tankfüllstandsgeber

Abbildung 2 Vorrichtung EOL kapazitiver Tankfüllstandsgeber

### Soll-Stand

Die bestehende EOL-Vorrichung soll Hard- und Softwaremässig auf einen aktuellen Stand gebracht werden. Neu soll neben den analogen Tankfüllstandgebern auch ein digitaler Tankfüllstandgeber getestet werden können. Dieser Kommuniziert neu über NMEA2000 wodurch diese neu implementiert werden muss. Die Überarbeitung der Hardware ist nicht Bestandteil dieser IPA.

### Anforderungen

* Die Steuersoftware für die Endkontrolle „kapazitiver Tankfüllstandsgeber“ muss neu mit aktuellerer Technik (Windows, C, Ethernet, ...) erstellt werden.
* Der Ablauf der neuen Software muss gemäss der Prüfspezifikation (Dokument 10371354.pdf) erstellt werden.
* Die NMEA2000 Kommunikation muss für das Produkt angepasst werden (TPL\_PTS\_Z10006660\_AA.pdf).

## Ziele

### Muss-Ziele

|  |  |
| --- | --- |
| Ziel | Kriterium |
| Vorrichtung über GUI steuerbar | Ablauf muss über GUI gestartet werden können.  Programmzustand muss über GUI ersichtlich sein. |
| Die Artikeldefinitionen aus Ini-Datei laden | Alle Artikel aus bestehender Ini-Datei fehlerfrei einlesen |
| Zu prüfender Artikel aus Liste auswählbar | Alle Artikel aus Ini-Datei werden zur Auswahl aufgelistet |
| Analoge kapazitive Tankfüllstandsgeber prüfbar | Analoge kapazitive Tankfüllstandsgeber müssen geprüft werden können, gemäss Pruefvorschr\_98601240\_.pdf (Ablauf) Kapitel 4.3 |

### Kann-Ziele

|  |  |
| --- | --- |
| Ziel | Kriterium |
| Diagnosepanel Inputs und Outputs der Vorrichtung | Alle Ein-/Ausgänge beobachtbar/steuerbar |
| Digitale kapazitive Tankfüllstandsgeber prüfbar | Digitale kapazitive Tankfüllstandsgeber müssen geprüft werden können, gemäss Pruefvorschr\_98601240\_.pdf (Ablauf) Kapitel 4.3 und TPL\_PTS\_Z10006660\_AA.pdf (Kommunikation) |

# Planen

## Terminplan



Abbildung 3 Terminplan Text

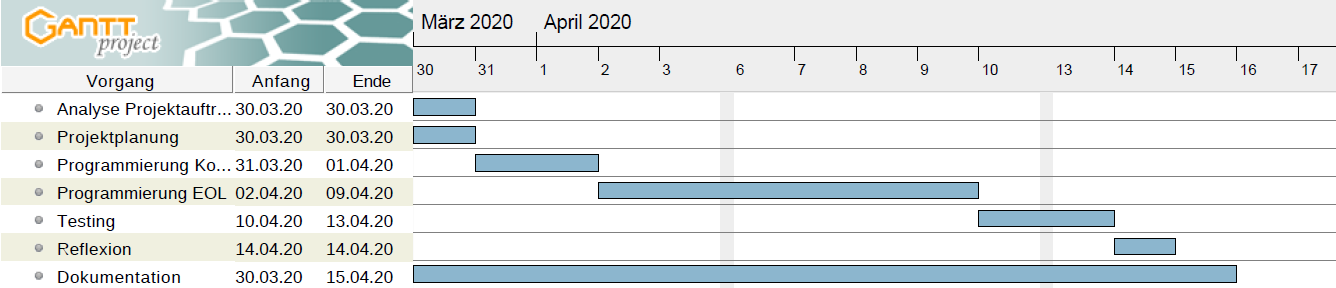


Abbildung 4 Terminplan Grafisch

## Meilensteine

|  |  |  |
| --- | --- | --- |
| Datum | Meilenstein | Zuweisung |
| 30.03.2013 | Projektstart | HaJ |
| 30.03.2013 | Projektplan | HaJ |
| 02.04.2013 | Kommunikation implementiert | HaJ |
| 10.04.2013 | EOL Implementiert | HaJ |
| 14.04.2013 | Getestet | HaJ |
| 15.04.2013 | Reflektiert und kontrolliert | HaJ |
| 16.04.2013 | Projektende | HaJ |

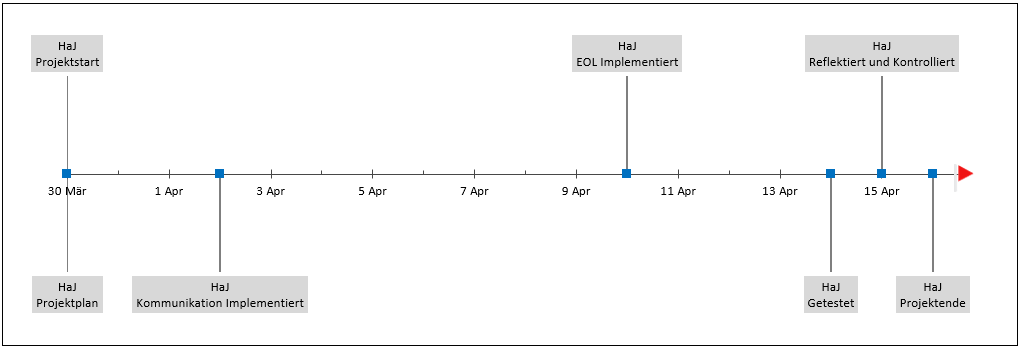


Abbildung 5 Meilenstein Planung

## Ablauf EOL

Die Prüfung eines Tankfüllstandgebers durch die EOL soll wie folgt ablaufen:

1. Nachdem der DUT an die Vorrichtung angeschlossen wurde, sendet die EOL einen Login Command an den Prüfling, um mit ihm kommunizieren zu können.
2. Nun wird der Wasserstand in den Tanks so lange erhöht, bis er laut dem Referenzgerät 100 % erreicht hat.
3. Somit kann nun der Wert vom Prüfling abgelesen werden.
4. Die Ausgelesenen Werte können nun miteinander verglichen werden. Stimmen die Werte mit der Beachtung der Toleranz überein, so kann der Test weitergeführt werden. Andernfalls wird eine Fehlermeldung angezeigt und die Prüfung abgebrochen, da der Prüfling fehlerhaft ist.
5. Nun senkt man den Wasserstand wieder, bis das Referenzgerät 50 % anzeigt.
6. Ist der Wasserstand nun bei 50 % kann der Wert beim Prüfling ausgelesen werden.
7. Wenn dir Wert des DUT’s und des Referenzgerätes übereinstimmen, so wird der geprüfte Tankfüllstandsgeber als «in Ordnung» angesehen und die Prüfung wird abgeschlossen. Stimmen die Werte jedoch nicht überein, wird eine Ausgabe erfolgen, welche über den Fehler berichtet und die Prüfung wird abgebrochen.

Auf der Vorrichtung ist genug Platz, um bis zu Zehn Prüflinge zur selben Zeit zu anzuschliessen. Jedoch kann nur das Signal eines Prüflings zur gleichen Zeit ausgelesen werden.

Mit einem Potentiometer an der Vorrichtung ist es jedoch möglich, den DUT auszuwählen, von welchem man aktuell die Daten auslesen möchte.

Um Zeit zu sparen, wird nun nachdem man den Wasserspiegel erhöht oder gesenkt hat, alle Zehn Prüflinge prüfen indem man mit dem Potentiometer das Signal umschaltet. Erst wenn von allen Prüflingen der Wert eingelesen und überprüft wurde, fährt die EOL im Test weiter.

## Vordefinierte Testfälle

Die verschiedenen Testfälle werden im Vorhinein definiert. Die einzelnen Tests werden nach dem ebenfalls im Vorhinein ausgearbeiteten Prüfplan durchgeführt. Sobald alle Tests durchlaufen wurden, werden wenn nötig Massnahmen definiert und ergriffen.

### Kommunikation

|  |  |  |
| --- | --- | --- |
| ID | Test | Erwartetes Resultat |
| TK01 | Lesen der Analogen Inputs | Die Analogen Inputs können korrekt vom Tankfüllstandsgeber eingelesen werden. |
| TK02 | Lesen der Digitalen Inputs | Die Digitalen Inputs können korrekt vom Tankfüllstandsgeber eingelesen werden. |
| TK03 | Schreibend er Digitalen Outpus | Die Digitalen Outputs können korrekt am Tankfüllstandsgeber gesetzt werden. |

### EOL

|  |  |  |
| --- | --- | --- |
| ID | Test | Erwartetes Resultat |
| TE01 | Wasser füllen | Durch das Schalten eines Ventiles über da EA-Modul wird Luft in den Wassertank gepumpt. Durch den Steigenden Druck im Tank steigt das Wasser die dafür vorgesehenen Rohre hinauf. |
| TE02 | Wasser halten | Durch das Schalten eines weiteren Ventils über das EA-Moduls wird das Wasser in den Rohren gehalten, selbst wenn der Luftdruck im Wassertank nachlässt. |
| TE03 | Zu wenig / Zu viel Wasser | Ist zu wenig oder zu viel Wasser in die Rohre gepumpt worden, so soll dies vom Referenzgeber erkannt werden. Entsprechen soll auch eine Benachrichtigung an den Benutzer stattfinden. |
| TE04 | Entfernen des DUT | Entfernt man einen DUT bevor die Prüfung abgeschlossen ist, so soll dies von der EOL erkannt und die Prüfung gestoppt werden. |
| TE05 | Korrektes Darstellen der ausgelesenen Werte | Die von den Tankfüllstandsgebern ausgelesenen Werte sollen korrekt dargestellt werden. Es soll auf einen Blick erkennbar sein, ob sich der gelesene Wert im Toleranzbereich befindet. |

## Prüfplan

### Testing aller Tests

Um sicherzustellen, dass alle Tests erfolgreich sind, wurde ein Testkonzept entwickelt, damit am Ende die komplette Applikation funktioniert. Die Folgende Grafik soll das Testkonzept grafisch darstellen:



Abbildung 8 Prüfplan gesamt

##### Beschreibung

Begonnen wird mit dem ersten Testfall. Falls dieser ohne Probleme und Bugs erfüllt ist, geht man zum nächsten Testfall und testet diesen. Dies macht man so lange, bis man einen Fehler findet. In diesem Fall löst man das Problem sofort und fokussiert sich darauf, den Bug zu fixen. Sobald auch dieser Testfall erfolgreich ist, fängt man wieder mit dem ersten Testfall an und geht nochmal alle genau durch. Dies hat den Vorteil, dass falls die Änderung im Programm einen Einfluss auf früheren Test hat, man diese findet und ebenfalls beheben kann. Man macht dieses Spiel nun so lange, bis alle Bugs behoben wurden und man sagen kann das alle Tests erfolgreich erfüllt wurden.

### Einzeltest



Abbildung 9 Prüfplan einzeln

Die einzelnen Anforderungen werden nach dem Blackbox Konzept getestet. Ein Black Box Verfahren ist ein spezifikationsorientiertes Testentwurfsverfahren. Dabei werden die Testfälle ausschließlich aus der Spezifikation des zu testenden Objekts abgeleitet, ohne dabei dessen innere Struktur, den Code, zu berücksichtigen. Es wird also nur das von aussen sichtbaren Verhalten des Testobjektes beobachtet.

Bei jedem Test werden die erwarteten Resultate mit dem tatsächlichen Resultat vergleichen, Unterschiede festgehalten und gegebenenfalls Massnahmen definiert.

# Entscheiden

## Ablaufdiagram

# Realisieren

## Coding

# Kontrollieren

## Prüffälle

Beim Prüfen wird strikt nach dem definierten Prüfplan vorgegangen und es werden die im Voraus definierten Testfälle abgehandelt.

### T001 - Auf Prüfdatei warten

**Beschreibung**

**Erwartetes Ergebnis**

**Test**

|  |  |  |
| --- | --- | --- |
| Test | Ergebnis | Symbol |
|  |  |  |
|  |  |  |

**Auswertung / Massnahmen**

# Auswerten

## Reflexion

### Allgemein

##### Gelerntes

###### Dokumentation

###### Funktionen

###### Reibungspunkte

# Abbildungsverzeichnis

[Abbildung 11 Code, Log-Datei Benennung 2](#_Toc36452916)

[Abbildung 12 Code, Log-Datei Inhalt 2](#_Toc36452917)

[Abbildung 13 Prüfplan gesamt 3](#_Toc36452918)

[Abbildung 14 Prüfplan einzeln 3](#_Toc36452919)