Pflichtenheft

HiL-Interface Abwassersteuergerät Dismessa

Summary	Anforderungen an das HiL-Interface des Dismessa HiL-Prüfstandes, das im Rahmen der IPA von Yannick Burkhalter entwickelt wird.
Verfasser	Lukas Röthlisberger (LRO); Simon Grossenbacher (SGR)
Approved by	Bruno Christen (BCH)

Version

Date / Initials	Description	Version	Planned Review
2018-01-08/ LRO	Dokument erstellt – V00.00.01	V00.00.01	N
2018-01-26/ LRO	Erste Version zur Freigabe	V01.00.00	DR

Review column: **DR**=Design Review; **SR**=Status Review; **N**=No formal review

Freigabe

Leiter Entwicklung Elektronik

Datum: _____ Visum: _____

Inhaltsverzeichnis

1 Einleitung

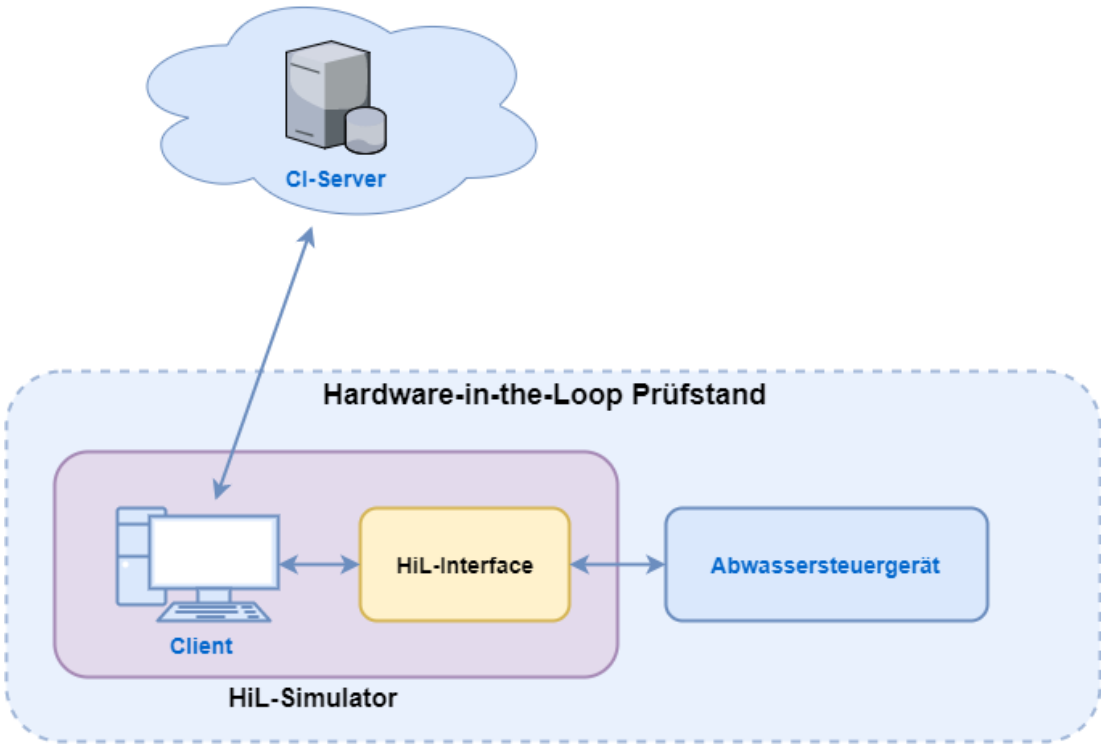
Die Anforderungen an das Testing eines Produkts in der Entwicklungsphase nehmen stetig zu. Weiter wird die Intervall-Zeit zwischen den einzelnen Revisionen immer kürzer. Diese beiden Tendenzen führen dazu, dass das Testing zusehends automatisiert werden muss. Dies geschieht unter anderem durch das Einsetzen von HiL-Simulatoren (Hardware in the Loop), die es ermöglichen, ein komplettes System in einer definierten Umgebung automatisch zu testen. Im Zuge dieser IPA soll ein HiL-Interface zu einem, sich aktuell in der Entwicklung befindendem, Abwassersteuergerät realisiert werden (Projektname Dismessa).

Dieses Dokument definiert die Anforderungen und Rahmenbedingung für die Umsetzung der IPA von Yannick Burkhalter (YBU).

2 Kontext

Die Firma Biral AG mit Sitz in Münsingen ist der führende Hersteller modernster Pumpen für die Haustechnik, den kommunalen Bereich und die Industrie in der Schweiz. Für eine Neuentwicklung im Produktportfolio der Abwasserhebeanlagen wird ein automatisiertes Testsystem in Form eines Hardware-in-the-Loop-Prüfstands entwickelt. Im Rahmen der IPA soll ein spezifisches HiL-Interface entwickelt werden, das die Ein- und Ausgänge eines Abwassersteuergeräts stimuliert und auswertet.

Die Integration des HiL-Interface in das automatisierte Testsystem sieht wie folgt aus:



3 Aufgabe

YBU hat sein Praktikum im 4. Lehrjahr bei der Biral AG absolviert. Während dieser Zeit wurde unter anderem auch ein rudimentärer Simulator für die Abwassersteuerung entwickelt. Die Aufgabe der IPA umfasst die Weiterentwicklung des bestehenden Designs hin zum produktiv einsetzbaren HiL-Interface, das alle Ein- und Ausgänge des Abwassersteuergeräts abdeckt, Fehlerinjektionen ermöglicht und über eine Schnittstelle mit dem Client des Testsystems kommunizieren kann.

3.1 Anforderungen an die Grundfunktion

Die Anforderungen an die Grundfunktion haben zum Ziel, eine funktionsfähige Baugruppe zu entwickeln, die sich in den Hardware-in-the-Loop-Prüfstand intergieren lässt.

3.1.1 Mechanischer Aufbau

Die elektronischen Komponenten des Hil-Interfaces müssen in ein geeignetes Gehäuse eingebaut werden. Dabei gilt die Anforderung, dass die gleiche Schutzart zu erzielen ist, wie beim Abwassersteuergerät (IP54). Bei Stromführenden Komponenten ist darauf zu achten, dass entweder der Berührungsschutz nach Schutzart IP2X gewährleistet ist oder mit Sicherheitskleinspannung (SELV) gearbeitet wird.

3.1.2 Stromversorgung

Die Stromversorgung für das HiL-Interface ist auf 24V festgelegt. Das Speisungskonzept des Interfaces gilt es für den Dauerbetrieb (24/7) auszulegen.

3.1.3 Schnittstellen

Das HiL-Interface muss alle «Systemschnittstellen» abdecken. Ausgenommen davon sind spezifische Serviceschnittstellen, Programmierschnittstellen und systeminterne Schnittstellen (z.B. Stromwandler).

Systemschnittstellen der Abwassersteuerung:

Bezeichnung	Typ	Funktion	Bemerkung
AUS	Digital Input (2 Kanal)	Auswertung Schwimmerschalter	Auswertung Wechselkontakt -> Überwachung von Kontaktfehler resp. Kabelbruch
EIN1	Digital Input (2 Kanal)	Auswertung Schwimmerschalter	Auswertung Wechselkontakt -> Überwachung von Kontaktfehler resp. Kabelbruch
EIN2	Digital Input (2 Kanal)	Auswertung Schwimmerschalter	Auswertung Wechselkontakt -> Überwachung von Kontaktfehler resp. Kabelbruch
HOCH	Digital Input (2 Kanal)	Auswertung Schwimmerschalter	Auswertung Wechselkontakt -> Überwachung von Kontaktfehler resp. Kabelbruch
Ext OFF	Digital Input (1 Kanal)	Externe Sperrung der Steuerung	
WSK1	Digital Input (1 Kanal)	Wicklungsschutzkontakt Motor1	
WSK2	Digital Input (1 Kanal)	Wicklungsschutzkontakt Motor2	
OSK	Digital Input (1 Kanal)	Ölsperrkammer Motor1 und 2	Parallelverdrahtung
ACI 4-20mA	Analog Input	Analog Control Input	Überwachung auf Unter- oder Überschreitung des Intervalls [4 mA 20mA]
Relais1	Digital Output	BM (Betriebsmeldung)	Potentialfreier Wechselkontakt, nur ein Kontakt auswerten.
Relais2	Digital Output	SAM (Sammelalarmmeldung)	Potentialfreier Wechselkontakt, nur ein Kontakt auswerten.
Relais3	Digital Output	Custom1	Potentialfreier Wechselkontakt, nur ein Kontakt auswerten.
Relais4	Digital Output	Custom2	Potentialfreier Wechselkontakt, nur ein Kontakt auswerten.
Alarmhorn	230 VAC Schaltkontakt	Schaltkontakt Alarmhorn	Achtung galvanisch getrennte Auswertung notwendig!
Motor1	400 VAC Schaltkontakt	Schaltkontakt Motor Pumpe1	Achtung galvanisch getrennte Auswertung notwendig!
Motor2	400 VAC Schaltkontakt	Schaltkontakt Motor Pumpe2	Achtung galvanisch getrennte Auswertung notwendig!

Anmerkung:
Die digitalen Eingänge müssen potentialfrei geschalten werden. An den Klemmen liegt eine Spannung von 24 Volt an. Bei geschlossenem Kontakt fliesst ein Strom von 5 mA (Frittstrom, engl. wetting current).

Zusätzliche Schnittstellen:

Damit die Abwassersteuerung aus einem definierten Systemzustand gestartet werden kann, soll die Stromversorgung der Abwassersteuerung über das HiL-Interface geschalten werden können. Dazu muss ein geeigneter Schaltmechanismus (Leistungsrelais, Halbleiterrelais, Schütz etc.) evaluiert werden.

3.1.4 Fehlerinjektion

Die Abwassersteuerung ist in der Lage, fehlerhafte Zustände der externen Systemkomponenten zu erkennen. Das HiL-Interface muss demzufolge in der Lage sein, bei allen überwachten Systemkomponenten resp. Schnittstellen gezielt Fehler zu injizieren.

3.1.5 Business-Logic

Das HiL-Interface bildet nur eine Schnittstelle zwischen DUT[1] und dem Client des Prüfstandes. Die Testprozedur wird über den Client definiert. Es sollen keine logischen «Testabläufe» im Interface abgebildet werden oder irgendwelche Zustände des Systems interpretiert werden.

3.2 Optionale Erweiterungen

3.2.1 Firmware

Die Firmware hat zur Aufgabe die Befehle des Clients (siehe Abschnitt 3.2.1.1) auszuwerten und entsprechend die Schnittstelle zum Abwassersteuergerät (gemäss Abschnitt 3.1.3) zu bedienen. Das Design der Firmware ist so zu gestalten, dass eine Erweiterung der Schnittstelle zum Client nachträglich realisierbar ist. Weiter ist die Betriebszeit 24/7 zu berücksichtigen[2].

3.2.1.1 Schnittstelle zum Client

Die Kommunikation zwischen Client und HiL-Interface soll über die UART des µC erfolgen. Die Anbindung zum Client ist mit einem FT232R USB zu UART Interface zu realisieren. Als Alternativlösung müssen die Pins der UART auf eine Stiftleiste geführt werden, vorzugsweise mit dem Pinout gängiger USB zu UART Adapterkabel.

Für die Kommunikation zwischen Client und HiL-Interface ist ein einfaches Protokoll zu implementieren, das über getter und setter Kommandos einen Datenaustausch ermöglicht.

Technische Anforderungen:

- Physical Layer UART, 3.3V-Pegel, 115200 baud (8D, 1S, 0P)
- Request – Response Pattern
- Client = Master, HiL-Interface = Slave
- Keine verbindungsorientierte Kommunikation
- Keine Redundanzprüfung (CRC)
- Response Timeout <1s
- Plain text protocol -> human readable ASCII Format für den Aufbau der getter und setter Funktionen mit Zeilenumbruch als Abschluss (Windows Variante \r\n)
- Status CODES (Response) als binary protocol für eine einfache Verarbeitung der Antwort

[1] DUT – Device under Test
[2] Ein Watchdog ist zu implementieren

Übersicht der Kommandos:

CMD	Datatype	Response	Call example	Bemerkung
help	-	Liste mit allen CMDs	help\r\n	
version	-	VXX.YY.ZZ	version\r\n	
reset	-	0\r\n	reset\r\n	Set all parameters to default
set power	0:= OFF, 1:= ON	0\r\n	set power 0\r\n	Switch DUT main power
set wsk0	0:= inaktiv, 1:= aktiv	0\r\n	set wsk0 1\r\n	
set wsk1	0:= inaktiv, 1:= aktiv	0\r\n	set wsk1 1\r\n	
set osk	0:= inaktiv, 1:= aktiv	0\r\n	set osk 1\r\n	
set aci_current	0:1:25, [aci_current] = mA	0\r\n	set aci_current 10\r\n	
set ext_off	0:= inaktiv, 1:= aktiv	0\r\n	set ext_off 1\r\n	
set level_switch0	0:= inaktiv 1:= aktiv 2:= error_cable 3:= error_switch	0\r\n	set level_switch0 1\r\n	
set level_switch1	0:= inaktiv 1:= aktiv 2:= error_cable 3:= error_switch	0\r\n	set level_switch1 1\r\n	
set level_switch2	0:= inaktiv 1:= aktiv 2:= error_cable 3:= error_switch	0\r\n	set level_switch2 1\r\n	
set level_switch3	0:= inaktiv 1:= aktiv 2:= error_cable 3:= error_switch	0\r\n	set level_switch3 1\r\n	
get relay_alarm	-	0\r\n:= inaktiv, 1\r\n:= aktiv	get relay_alarm\r\n	
get relay_operation	-	0:= inaktiv, 1:= aktiv	get relay_operation\r\n	
get relay_custom0	-	0:= inaktiv, 1:= aktiv	get relay_custom0\r\n	
get relay_custom1	-	0:= inaktiv, 1:= aktiv	get relay_custom1\r\n	
get horn_alarm	-	0:= inaktiv, 1:= aktiv	get horn_alarm\r\n	

get motor0	-	0:= inaktiv, 1:= aktiv	get motor0\r\n	
get motor1	-	0:= inaktiv, 1:= aktiv	get motor1\r\n	
get wsk0	-	-> setter datatype	get wsk0\r\n	Set-Parameter abfragen
get wsk1	-	-> setter datatype	get wsk1\r\n	Set-Parameter abfragen
get osk	-	-> setter datatype	get osk\r\n	Set-Parameter abfragen
get aci_current	-	-> setter datatype	get aci_current\r\n	Set-Parameter abfragen
get ext_off	-	-> setter datatype	get ext_off\r\n	Set-Parameter abfragen
get level_switch0	-	-> setter datatype	get level_switch0\r\n	Set-Parameter abfragen
get level_switch1	-	-> setter datatype	get level_switch1\r\n	Set-Parameter abfragen
get level_switch2	-	-> setter datatype	get level_switch2\r\n	Set-Parameter abfragen
get level_switch3	-	-> setter datatype	get level_switch3\r\n	Set-Parameter abfragen

Anmerkung: Leerzeichen sind als ASCII-Zeichen 32 (space) zu verstehen

Error Codes:

Response value	Bedeutung
>= 0	CMD valid/response data
-1	CMD not valid
-2	Buffer overflow

3.2.2 Client-Software

Die Client Software muss die Schnittstelle zum HiL-Interface ansteuern können. Es wird zwischen zwei Einsatz-Szenarien unterschieden.

3.2.2.1 Demo-Applikation

Die Demo-Applikation soll lediglich die Möglichkeiten den HiL-Interface aufzeigen und eine einfache Bedienung via einer GUI ermöglichen. Weiter ist ein Messe-Modus denkbar, der zum Präsentieren des Abwassersteuergeräts auf einer Messe dient[3].

3.2.2.2 Unit-Test Library

Die Library soll eine API zur der Client-Schnittstelle[4] bieten, um eine einfache Integration in ein Unit-Test-Framework zu ermöglichen. Die genauen Anforderungen an eine solche Library sind noch nicht definiert und sind Teil eines parallellaufenden Projekts.

4 Randbedingungen

Folgend sind Randbedingungen des HiL-Interfaces, respektive des HiL-Simulators definiert.

4.1 Einsatzgebiet

Der HiL-Simulator wird unter Laborbedingungen eingesetzt und nur durch geschultes Fachpersonal bedient.

4.2 Aufbau HiL-Simulator

Wie bereits im Abschnitt 3.1.1 definiert, verfügt der Simulator über ein passendes Gehäuse, in das ebenfalls das HiL-Interface integriert wird. Alle mechanischen Komponenten sind nicht Teil der vorliegenden IPA und werden im Vorfeld evaluiert und umgesetzt.

4.3 Entwicklungsumgebung

Es gilt die Firma-internen Werkzeuge zu berücksichtigen.

- Schema/Layout: Mentor PADS
- IDE: Atollic TRUEStudio
- Programmiersprache C für die Firmware-Entwicklung (mindestens C99)

4.4 Betreuung

Die Betreuung während der IPA wird durch Mitarbeiter der Elektro-Entwicklung wahrgenommen.

Auftraggeber	Bruno Christen (BCH)
Applikationswissen	Bruno Christen (BCH); Tim Blaser (TBL)
Software-Engineering	Michael Burkhalter (MBU); Simon Grossenbacher (SGR)
Hardware-Engineering	Lukas Röthlisberger (LRO)

[3] z.B. mit virtuellen Wassertank
[4] Siehe Abschnitt 3.2.1.1

5 Planung

Damit Abweichungen von den angestrebten Zielen frühzeitig erkannt und entsprechende Massnahmen eingeleitet werden können, ist eine Planung zu unterhalten.

5.1 Zeitrahmen

Für die Durchführung der Arbeiten steht der Zeitrahmen vom 12. Februar 2018 bis 03. April 2018 zur Verfügung (exklusive eines 2-wöchigen Unterbruchs).

5.2 Prioritäten

In erster Priorität soll die Realisierung der Grundfunktionen angegangen werden. Optionen (Firmware, sowie Applikations-Software) haben zweitrangige Bedeutung.

6 Dokumentation

Neben der üblichen Projektdokumentation (Funktionsbeschreibung, Versionsverwaltung, usw.) ist ein tägliches Journal zu führen. Das Journal dokumentiert den Projektfortschritt sowie alle getroffenen Entscheide und Ergebnisse in einer reproduzierbaren Form.

Das Projekt sowie die erarbeiteten Ergebnisse und Erkenntnisse sind zudem am Schluss der IPA anlässlich einer Präsentation vorzustellen.