

Kerstin WALLNER, BSc

---

MASTERARBEIT

---

AUFBAU UND EVALUATION EINER AUTOMATISCHEN  
VERIFIKATIONSPLATTFORM FÜR MEDIZINISCHE  
ELEKTRONIKKOMPONENTEN

---

ausgeführt zum Zwecke der Erlangung des akademischen Grades einer  
Diplom-Ingenieurin der Studienrichtung Biomedical Engineering

eingereicht an der Technischen Universität Graz

Institut für Medizintechnik  
Technische Universität Graz  
Stremayrgasse 16, 8010 Graz

Universitärer Betreuer und Begutachter:

ao.Univ.-Prof.Hermann Scharfetter

Betrieblicher Betreuer und Begutachter:

Dipl.-Ing. Armin Rieß

Graz, August 2016

---

# EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

---

Graz, am

---

(Unterschrift)

# STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

---

date

---

(signature)

---

# DANKSAGUNG

An dieser Stelle möchte ich mich bei allen bedanken, die mich während der Erstellung meiner Master-Thesis unterstützt haben.

Ich möchte mich bei dem gesamten Team der Abteilung R&D der Firma B. Braun Avitum AG für die freundliche und hilfsbereite Arbeitsatmosphäre bedanken. In besonderem möchte ich mich bei Herrn Dipl.-Ing. Armin Rieß bedanken, der mich im Rahmen der Anfertigung meiner Arbeit betreute. Mein Dank geht an alle Kollegen, die mir wertvolle Anregungen und Hinweise gaben, welche zum Gelingen meiner Arbeit beitrugen. Erwähnen möchte ich an dieser Stelle Dirk Möller, Andreas Laudert und Jan Eric Herr.

Nicht zuletzt gebührt ao. Univ.-Prof. Hermann Scharfetter ein großer Dank für die universitäre Betreuung und Begutachtung meiner Arbeit.

Abschließend bedanke ich mich bei meiner Familie und meinen Freunden, die mir mein Studium durch ihre Unterstützung ermöglicht haben und mir starken Rückhalt über die Dauer meines gesamten Studiums gegeben haben.

Kerstin Wallner

Melsungen, 16. August 2016

---

## SPERRVERMERK

Die vorgelegte Master-Thesis mit dem Titel "Aufbau und Evaluation einer automatischen Verifikationsplattform für medizinische Elektronikkomponenten" beinhaltet vertrauliche Informationen und Daten des Unternehmens B. Braun Avitum AG. Diese Arbeit darf nur vom Erst- und Zweitgutachter, sowie berechtigten Mitgliedern des Prüfungsausschusses eingesehen werden. Eine Vervielfältigung und Veröffentlichung der Master-Thesis ist auch auszugsweise nicht erlaubt. Dritten darf diese Arbeit nur mit der ausdrücklichen Genehmigung des Verfassers und des Unternehmens B. Braun Melsungen AG zugänglich gemacht werden. Die Gültigkeit des Sperrvermerks ist, in Absprache mit der B. Braun Melsungen AG, auf drei Jahre festgesetzt.

---

# Aufbau und Evaluation einer automatischen Verifikationsplattform für medizinische Elektronikkomponenten

## Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung und Evaluation einer automatischen Verifikationsplattform für medizinische Elektronikkomponenten. Nach der Analyse der Anforderungen an die Prüfumgebung kann nach dem V-Modell eine Prüfumgebung erstellt werden, die in weiterer Folge der Verifikation unterzogen wird. Die Verifikation wird dabei anhand des Monitor Interface Boards (MIB) durchgeführt. Bei dem MIB handelt es sich um eine Leiterplatte, welche zum Informationsaustausch zwischen Anwender und der Dialysemaschine NDS dient. Im Anschluss an die Verifikation erfolgt die Validierung der Prüfumgebung, um sicherzustellen, dass die an die Prüfumgebung gestellten Anforderungen erfüllt worden sind.

Das Ergebnis dieser Arbeit ist ein Hardware-in-the-Loop (HiL) Software Framework (IBM RQM, NI TestStand, NI LabVIEW), das dazu dient funktionelle Spezifikationen anhand der Anforderungen an den Prüfling zu untersuchen. Das HiL Software Framework bindet reale Hardwarekomponenten in eine virtuell-simulierte Umgebung ein und erreicht eine nahtlose Traceability, von den Anforderungen bis hin zum implementierten LabVIEW Testfall. Für die Erstellung dieses HiL-Testsystems ist die Kenntnis aller Eingangs- und Ausgangssignale des Prüflings notwendig, um die realen Systemumgebungen bestmöglich zu simulieren. Durch die Verwendung dieses HiL-Testsystems können bereits funktionelle Tests auf Komponentenebene durchgeführt werden, ohne die Komponente in das vollständige System der Dialysemaschine integrieren zu müssen.

## Schlüsselwörter

Dialyse, Hardware-in-the-Loop, Verifikation, Automatisierung, V-Modell, Traceability, LabVIEW, TestStand

---

# Development and evaluation of an automatic verification platform for medical electronic components

## Abstract

The present thesis deals with the development and evaluation of an automatic verification platform for medical electronic components. After the analysis of the requirements a test environment can be created according to the V-model, which will subsequently be verified. This verification process will be conducted by the Monitor Interface Board (MIB). The MIB is a circuit board that exchanges information between users and the dialysis machine NDS. Following the verification, the validation of the test environment is conducted to ensure that the stated requirements have been met.

The result of this thesis is a Hardware-in-the-Loop (HiL) software framework (IBM RQM, NI TestStand, NI LabVIEW), which examines functional specifications, based on the demands / requirements of the DUT (device under test). The HiL Software Framework incorporates real hardware components in a virtually simulated environment and achieves a seamless/consistent traceability, from the requirements to the implemented LabVIEW test case. In order to simulate the real system environments in the best possible way, the knowledge of all the input and output signals of the DUT is necessary for the preparation of this HiL test system. By using this HiL test system, functional testing of components can be performed without having to integrate the component into the complete system of the dialysis machine.

## Keywords

dialysis, hardware-in-the-loop, verification, automation, V-Model, traceability, LabVIEW, TestStand



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>		<b>xii</b>
<b>Tabellenverzeichnis</b>		<b>xv</b>
<b>Abkürzungsverzeichnis</b>		<b>xvii</b>
<b>1 Einleitung</b>		<b>1</b>
1.1	B. Braun Group . . . . .	1
1.2	Aufgabenstellung . . . . .	2
<b>2 Grundlagen</b>		<b>3</b>
2.1	Dialyse . . . . .	3
2.2	V-Modell . . . . .	5
2.3	Hardware-in-the-Loop-Testsysteme . . . . .	6
2.4	Software . . . . .	7
2.4.1	IBM - Rational DOORS . . . . .	7
2.4.2	IBM - Rational Quality Manager . . . . .	8
2.4.3	National Instruments - TestStand . . . . .	8
2.4.4	National Instruments - LabVIEW . . . . .	9
2.5	Dialysemaschine NDS . . . . .	9
2.6	NDS Komponente - Monitor Interface Board . . . . .	11
2.6.1	Operation State Display . . . . .	13
2.6.2	Touchpanel . . . . .	14
2.6.3	Lautsprecher . . . . .	15
2.6.4	User Interface Board . . . . .	16
2.6.5	Kommunikation MIB - TLC . . . . .	17
2.6.5.1	Funktionscode . . . . .	18
2.6.5.2	Registeradresse . . . . .	19
2.6.5.3	Registerwert . . . . .	20
2.6.5.4	Zyklische Redundanzprüfung . . . . .	21
2.6.6	Kommunikation MIB - LLS . . . . .	21
2.6.7	Kommunikation MIB-Touch - TLC . . . . .	25



<b>3</b>	<b>Entwurf und Implementierung</b>	<b>29</b>
3.1	Signalanalyse und Aufstellung des Konzepts für die automatische Verifikationsplattform . . . . .	30
3.1.1	Filterung der Anforderungen in DOORS . . . . .	30
3.1.1.1	Anforderungen aus der IDS . . . . .	31
3.1.1.2	Anforderungen aus der CRS . . . . .	36
3.1.2	Erstellung des Testplans im RQM . . . . .	38
3.1.2.1	Testsuite MCF-TLC . . . . .	39
3.1.2.2	Testsuite MSF-LLS . . . . .	49
3.1.2.3	Testsuite TCF-TLC . . . . .	54
3.1.3	Anforderungen an das Messsystem . . . . .	58
3.1.3.1	Schnittstellenkonvertierung . . . . .	59
3.1.3.2	Signalanpassung: Pegelwandler . . . . .	59
3.1.3.3	Analog-Digital-Wandler/Digital-Analog-Wandler . . . . .	60
3.2	Implementation der Verifikationsplattform . . . . .	61
3.2.1	Aufbau des HiL-Testsystems . . . . .	61
3.2.2	Schnittstellen der Hardwarekomponenten des HiL-Testsystems	62
3.2.3	Entwicklung der Testumgebung in TestStand . . . . .	63
3.2.4	Entwicklung der LabVIEW Softwareumgebung . . . . .	67
<b>4</b>	<b>Nachweis der Wirksamkeit des HiL-Testsystems</b>	<b>75</b>
<b>5</b>	<b>Diskussion und Ausblick</b>	<b>81</b>
<b>6</b>	<b>Zusammenfassung</b>	<b>85</b>
<b>7</b>	<b>Literaturverzeichnis</b>	<b>87</b>
<b>8</b>	<b>Anhang</b>	<b>91</b>
8.1	Detaillierter Inhalt der Anforderungen der IDS . . . . .	92
8.1.1	Detaillierte Beschreibung der Register zur Kommunikation MIB-TLC . . . . .	92
8.1.1.1	MIB-TLC Hardware Variant Identifier (Reg-Addr: 0x00) IDS-3184 [11] . . . . .	92
8.1.1.2	MIB-TLC Hardware Version (Reg-Addr: 0x01) IDS-3186 [11] . . . . .	92
8.1.1.3	MIB-TLC Firmware Version (Reg-Addr: 0x02) IDS-3188 [11] . . . . .	92
8.1.1.4	MIB-TLC Hardware Version (Reg-Addr: 0x03) IDS-3190 [11] . . . . .	93

8.1.1.5	MIB-TLC Controller Status (Reg-Addr: 0x04) IDS-3192 [11] . . . . .	93
8.1.1.6	MIB-TLC Keyboard Hard Keys Status (Reg-Addr: 0x05) IDS-3196 [11] . . . . .	93
8.1.1.7	MIB-TLC Power-ON-OFF Hard Key Status (Reg-Addr: 0x06) IDS-3198 [11] . . . . .	94
8.1.1.8	MIB-TLC Detailed Firmware Version Information (Reg-Addr: 0x07) IDS-3200 [11] . . . . .	94
8.1.1.9	MIB-TLC Reset Control (Reg-Addr: 0x20) IDS-3204 [11]	94
8.1.1.10	MIB-TLC Keyboard Hard Keys LED Control (Reg-Addr: 0x21) IDS-3210 [11] . . . . .	94
8.1.1.11	MIB-TLC Loudspeaker Amplifier Sound Level (Reg-Addr: 0x22) IDS-3212 [11] . . . . .	95
8.1.1.12	MIB-TLC OSD RGB Control (Reg-Addr: 0x23) IDS-3214 [11] . . . . .	95
8.1.1.13	MIB-TLC OSD Brightness Control (Reg-Addr: 0x24) IDS-3216 [11] . . . . .	95
8.1.1.14	MIB-TLC OSD Test Sequence Control (Reg-Addr: 0x25) IDS-3219 [11] . . . . .	96
8.1.1.15	MIB-TLC OSD Alarm Environment Configuration Control (Reg-Addr: 0x26) IDS-3222 [11] . . . . .	96
8.1.1.16	MIB-TLC FPD Backlight-Brightness-Control (Reg-Addr: 0x27) IDS-3225 [11] . . . . .	96
8.1.1.17	MIB-TLC OSD RGB Individual Test Control (Reg-Addr: 0x28) IDS-3277 [11] . . . . .	97
8.1.1.18	MIB-TLC Loudspeaker Amplifier Mute (Reg-Addr: 0x29) IDS-3230 [11] . . . . .	97
8.1.2	Pinbelegung des MIBs aus der IDS . . . . .	97
8.2	Detaillierter Inhalt der Anforderungen der CRS . . . . .	103
8.3	Pinbelegung: Schnittstellen der Hardwarekomponenten des HiL Testsystems . . . . .	105
8.3.1	Pinbelegung der D-Sub Steckerverbindung (Prüfadapter ↔ Prüfling - MIB) . . . . .	105
8.3.2	Pinbelegung der USB Steckerverbindungen . . . . .	106
8.3.3	Pinbelegung der VHDCI Steckerverbindungen (Prüfadapter ↔ NI-Hardware) . . . . .	107



# Abbildungsverzeichnis

2.1	Schematische Darstellung des Stofftransports durch die semipermeable Membran . . . . .	4
2.2	Ablauf der Hämodialyse . . . . .	5
2.3	V-Modell . . . . .	6
2.4	Dialysemaschine NDS [9] . . . . .	10
2.5	Bestückungsplan MIB [23] . . . . .	11
2.6	Entwicklungsmuster MIB V5 (mit Debug-Steckern) . . . . .	11
2.7	Schnittstellen des Monitor Interface Boards . . . . .	12
2.8	Darstellung der MIB Komponenten und Interfaces . . . . .	13
2.9	Aufbau und Funktion eines 5-Draht-Touchscreens . . . . .	15
2.10	Bestimmung der Touch-Position in X-Richtung und in Y-Richtung [22] . . . . .	15
2.11	Hardkeys und LEDs des UIB . . . . .	16
2.12	Datenfluss TLC-Signale . . . . .	17
2.13	Datenfluss LLS-Signale . . . . .	21
2.14	Datenfluss Touchscreen-Signale . . . . .	26
3.1	Ablaufplan der Methodik . . . . .	29
3.2	Schematischer Aufbau des SW-Frameworks bestehend aus IBM DOORS, IBM RQM, NI TestStand und NI LabVIEW . . . . .	38
3.3	Screenshot aus RQM: Aufbau des Testplans . . . . .	39
3.4	Screenshot aus RQM: Festfälle der Testsuite MCF-TLC . . . . .	40
3.5	Testfall 1: Hardware Identifier auslesen . . . . .	41
3.6	Testfall 7: Hardkeys Status auslesen (alle Hardkeys nicht gedrückt) . . . . .	42
3.7	Steuerung der Hintergrundbeleuchtung . . . . .	47
3.8	Testdesign: Testfall 99 . . . . .	50
3.9	Screenshot aus RQM: Anforderungen aus DOORS für Testfall 99 in RQM integriert . . . . .	50
3.10	Testdesign: Schematische Darstellung Testfall 119 . . . . .	55
3.11	Oszillogramm: Anschluss "P15 Touch Panel" . . . . .	56
3.12	Screenshot aus RQM: Anforderungen aus DOORS für Testfall 119 in RQM integriert . . . . .	58
3.13	Blockschaltbild: Funktion der Adapterleiterplatte . . . . .	59

3.14	Aufbau des SW-Frameworks bestehend aus IBM DOORS, IBM RQM, NI TestStand und NI LabVIEW . . . . .	61
3.15	Schnittstellen der Hardwarekomponenten des HiL-Testsystems . . . . .	63
3.16	TestStand: Arten von Testschritten . . . . .	64
3.17	TestStand: Aufbau der Sequenzdatei . . . . .	65
3.18	TestStand: Testschritte von Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) . . . . .	66
3.19	TestStand: Testschritte von Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) . . . . .	67
3.20	LabVIEW VI zur Kommunikation mit dem LLS . . . . .	69
3.21	Architektur FPGA-Programmierung . . . . .	70
3.22	Darstellung der Grundstruktur eines Host-VIs . . . . .	71
3.23	Simulation einer Touch-Position über die 5-Wire-Sense- und (X)-Leitung	73

# Tabellenverzeichnis

2.1	CD-OSDHW-1053: Alarm Prioritäten und Signalfarben . . . . .	14
2.2	IDS-3171: TLC-MIB Kommunikationsprotokoll . . . . .	18
2.3	IDS-3165 - Funktionscode . . . . .	19
2.4	Lese- und Schreibregister des MIB-TLCs . . . . .	20
2.5	IDS-3288: LLS-MIB Kommunikationsprotokoll . . . . .	22
2.6	IDS-3295: MIB-LLS Receive Register 0 . . . . .	22
2.7	IDS-3298: MIB-LLS Receive Register 1 . . . . .	23
2.8	IDS-3304: MIB-LLS Transmit Register 0 . . . . .	24
2.9	IDS-3307: MIB-LLS Transmit Register 1 . . . . .	24
2.10	IDS-3310: MIB-LLS Transmit Register 2 . . . . .	25
2.11	IDS-3313: MIB-LLS Transmit Register 3 . . . . .	25
2.12	IDS-3381: MIB-Touch zu TLC: Touch-Data-Protokoll . . . . .	26
2.13	IDS-3369: MIB-Touch Firmware Version . . . . .	26
2.14	IDS-3391: TLC → MIB-Touch Kommunikation . . . . .	27
2.15	IDS-3393: MIB-Touch → TLC Kommunikation . . . . .	27
3.1	Zusammenfassung der Anforderungen aus der IDS . . . . .	32
3.2	Anforderungen an die Messhardware abgeleitet aus der Signalanalyse .	33
3.3	Signalanalyse: Kommunikation zwischen LLS und MSF - IDS-2477 . .	34
3.4	Signalanalyse: Touchpanel - IDS-2474 . . . . .	34
3.5	IDS-3282: MIB-LLS Firmware Version Format . . . . .	35
3.6	IDS-3381: MIB-Touch zu TLC: Touch-Data-Protokoll . . . . .	36
3.7	Zusammenfassung der Anforderungen aus der CRS . . . . .	36
3.8	Akzeptanzkriterien der OSD LED Spannungen . . . . .	45
3.9	Akzeptanzkriterien der OSD Blinkfrequenz . . . . .	46
3.10	Akzeptanzkriterien der Einschaltzeit (tEin) des pulsbreitenmodulierten Signals bei unterschiedlicher Helligkeit . . . . .	47
3.11	Toleranzbereich der Spannungsversorgung 5VTLC, 12VTLC und 5V_ STBY . . . . .	48
3.12	Referenzbestimmung der Positionsdaten bei konstanter Touch-Position in X- und in Y-Richtung . . . . .	57
3.13	Überblick der LabVIEW Implementierungstypen: Host-VIs . . . . .	67

3.14	Überblick der LabVIEW Implementierungstypen: Host-VIs & FPGA-VIs	68
4.1	Auflistung der Prüfmuster für den Nachweis der Wirksamkeit des HiL-Testsystems . . . . .	76
8.1	MIB-TLC Hardware Variant Identifier (Reg-Addr: 0x00) . . . . .	92
8.2	MIB-TLC Hardware Version (Reg-Addr: 0x01) . . . . .	92
8.3	MIB-TLC Firmware Version (Reg-Addr: 0x02) . . . . .	92
8.4	MIB-TLC Controller Status (Reg-Addr: 0x04) . . . . .	93
8.5	MIB-TLC Keyboard Hard Keys Status (Reg-Addr: 0x05) . . . . .	93
8.6	MIB-TLC Reset Control (Reg-Addr: 0x20) . . . . .	94
8.7	MIB-TLC Keyboard Hard Keys LED Control (Reg-Addr: 0x21) . . . . .	94
8.8	MIB-TLC Loudspeaker Amplifier Sound Level (Reg-Addr: 0x22) . . . . .	95
8.9	MIB-TLC OSD RGB Control (Reg-Addr: 0x23) . . . . .	95
8.10	MIB-TLC OSD Brightness Control (Reg-Addr: 0x24) . . . . .	95
8.11	MIB-TLC OSD Test Sequence Control (Reg-Addr: 0x25) . . . . .	96
8.12	MIB-TLC OSD Alarm Environment Configuration Control (Reg-Addr: 0x26) . . . . .	96
8.13	MIB-TLC FPD Backlight-Brightness-Control (Reg-Addr: 0x27) . . . . .	96
8.14	MIB-TLC Loudspeaker Amplifier Mute (Reg-Addr: 0x29) [11] . . . . .	97
8.15	Signalanalyse: Backlight Interface - BLI - IDS-2480 [11] . . . . .	97
8.16	Signalanalyse: Operation State Display - OSD - IDS-2479 [11] . . . . .	98
8.17	Signalanalyse: Audiosignal - IDS-2472 [11] . . . . .	98
8.18	Signalanalyse: Lautsprecher - IDS-2473 [11] . . . . .	99
8.19	Signalanalyse: Kommunikation zwischen TLC und MCF - IDS-2476 [11]	99
8.20	Signalanalyse: User Interface Board UIB - IDS-2478 [11] . . . . .	99
8.21	Signalanalyse: Kommunikation zwischen TLC und TCF - IDS-2475 [11]	100
8.22	Signalanalyse: Power Out - IDS-2470 [11] . . . . .	101
8.23	Signalanalyse: Touchpanel - IDS-2474 [11] . . . . .	101
8.24	Signalanalyse: Power Supply Control - IDS-2471 [11] . . . . .	101
8.25	Signalanalyse: Spannungsversorgung - Power In - IDS-2470 [11] . . . . .	102
8.26	Signalanalyse: Kommunikation zwischen LLS und MSF - IDS-2477 [11]	102
8.27	Anforderungen an die MIBHW (CRS-MIBHW) [6] . . . . .	103
8.28	Anforderungen an den MCF (CRS-MCF) [5] . . . . .	104
8.29	Anforderungen an den MSF (CRS-MSF) [7] . . . . .	104
8.30	Anforderungen an den TCF (CRS-TCF) [8] . . . . .	105
8.31	Pinbelegung: D-Sub Steckerverbindung (Prüfadapter ↔ Prüfling - MIB)	105
8.32	Pinbelegung: USB Steckerverbindungen . . . . .	106
8.33	Pinbelegung: NI PXI-7813R Digital-RIO-Modul Connector 0 . . . . .	107

8.34 Pinbelegung: NI PXI-7813R Digital-RIO-Modul Connector 1 . . . . . 108





# Abkürzungsverzeichnis

A/D-Wandler .	Analog-Digital-Wandler
BLI .....	Backlight Interface
BP .....	Blutpumpe
CDS .....	Component Design Specification
CDTR .....	Component Design Test Report
CDTS .....	Component Design Test Specification
CRC .....	Cyclic Redundancy Check
CRS .....	Component Requirement Specification
CVEP .....	Component Verification Plan
CVER .....	Component Verification Report
D/A-Wandler .	Digital-Analog-Wandler
DOORS .....	Dynamic Object Oriented Requirements System
DS .....	System Design Specification
EMV .....	Elektromagnetische Verträglichkeit
ENA .....	Backlight Interface Enable
EOM .....	End of Message
FPD .....	Front Panel Display
FPGA .....	Field-Programmable Gate Array
HiL .....	Hardware-in-the-Loop
IEC .....	Internationale Elektrotechnische Kommission
IT- Systeme ...	Informationstechnische Systeme
LabVIEW ....	Laboratory Virtual Instrumentation Engineering Workbench
LLC .....	Low Level Controller
LLD .....	Low Level Digital Board
LLS .....	Low Level Supervisor
LSB .....	Least Significant Bit
MCF .....	MIB-TLC-Mikrocontroller
MIB .....	Monitor Interface Board
MIBHW .....	Monitor Interface Board Hardware
MSB .....	Most Significant Bit
MSF .....	MIB-LLS-Mikrocontroller

NC	.....	Not Connected
NDS	.....	New Dialysis System
OPM	.....	Out Patient Market
OSD	.....	Operation State Display
OSDHW	.....	Operation State Display Hardware
PDC	.....	Protocol Data Consistency
PRB	.....	Product Requirement Profile
PS-Interface	..	Power Supply Interface
PWR	.....	Power Supply
RQM	.....	Rational Quality Manager
RS	.....	System Requirement Specification
SOM	.....	Start of Message
TCF	.....	Touch-Mikrocontroller
TLC	.....	Top Level Controller
UIB	.....	User Interface Board
V-Modell	.....	Vorgehensmodell.
VAP	.....	Validation Plan
VAR	.....	Validation Report
VEP	.....	System Verification Plan
VER	.....	System Verification Report
VI	.....	Virtuelles Instrument
VISA	.....	Virtual Instrument Software Architecture

---

# 1 Einleitung

## 1.1 B. Braun Group

Das Unternehmen wurde 1839 von Julius Wilhelm Braun durch den Erwerb der Rosen-Apotheke in Melsungen gegründet. B. Braun hat sich seit 1839 zu einem weltweit agierenden Konzern mit 55.719 Mitarbeitern in 64 Ländern entwickelt. Weltweit versorgt das Unternehmen den Gesundheitsmarkt mit Produkten für Anästhesie, Intensivmedizin, Kardiologie, extrakorporale Blutbehandlung oder Chirurgie sowie mit Dienstleistungen für Kliniken, niedergelassene Ärzte und den Homecare-Bereich.

Das Unternehmen teilt sich in vier Sparten:

- Hospital Care versorgt Krankenhäuser mit Infusions- und Injektionslösungen sowie allen Produkten der medizinischen Einmalversorgung.
- Aesculap ist Weltmarktführer für handgehaltene chirurgische Instrumente. Aesculap legt den Fokus auf Produkte und Dienstleistungen für alle chirurgischen und interventionellen Kernprozesse.
- Out Patient Market kümmert sich um die Patientenversorgung außerhalb des Krankenhausbetriebs und von chronisch Kranken bzw. Langzeitpatienten.
- B. Braun Avitum umfasst Produktlieferung und medizinische Dienstleistung zur extrakorporalen Blutbehandlung (Dialyse). Als Komplettanbieter versorgt B. Braun Avitum seine Kunden zusätzlich mit sämtlichen, während der Behandlung eines Patienten, benötigten Verbrauchsmaterialien (Dialysier- und Spüllösungen, Blutschlauchsysteme, Dialysatoren und Gefäßzugänge)[12].

Diese Masterarbeit wurde in Zusammenarbeit mit der Sparte B. Braun Avitum organisiert und durchgeführt.

## 1.2 Aufgabenstellung

Zum jetzigen Zeitpunkt erfolgt ein vollständiger Systemtest der Dialysemaschine am Ende des Entwicklungsprozesses. Erst dabei wird das Zusammenspiel aller Komponenten getestet. Um die Mitarbeiter- und Kosteneffizienz zu steigern, führt man bereits vor dem vollständigen Systemtest Komponententests durch. Diese Komponententests sollen Fehler der Komponenten im frühest möglichen Stadium erkennen, um Kosten, die durch Änderungen verursacht werden, zu minimieren. Diese Komponententests werden oftmals manuell durchgeführt. Um manuelle Tests zu ersetzen, müssen solche Tests automatisiert werden. Nur durch die Automatisierung kann die Reproduzierbarkeit verbessert und eine Zeitersparnis, bei gleicher Testtiefe, erreicht werden. Zusätzlich sollen zur Verminderung des dokumentarischen Aufwands automatisch auditsichere Testreports erzeugt werden. Eine Verknüpfung der Tests mit den Anforderungen erhöht zudem die Traceability<sup>1</sup>. Ziel dieser Arbeit ist es, diese automatische Testplattform für Elektronikkomponenten aufzusetzen, um so den Testaufwand weiter zu minimieren.

Im folgenden Abschnitt wird auf die Aufgabenstellung für diese Arbeit eingegangen. Inhalt dieser interdisziplinären Arbeit ist es, eine automatische Testplattform, die nach dem Hardware-in-the-Loop (HiL) Prinzip realisiert wird, aufzusetzen und zu evaluieren. Die Verifikation wird dabei anhand des Monitor Interface Boards (MIB) durchgeführt.

Es ergeben sich folgende zu bearbeitenden Punkte:

- Analyse der Systemanforderungen
- Konzeption eines Lösungskandidaten
- Erstellung eines Verifikationskatalogs
- Aufsetzen eines SW-Frameworks (IBM RQM, NI TestStand, NI LabVIEW)
  - Anbindung einer Elektronik und der Testplattform an das SW-Framework
  - Implementierung der Verifikation mittels NI LabVIEW/TestStand
- Nachweis der Wirksamkeit der Implementierung
- Evaluierung der Ergebnisse bezogen auf die Aufgabenstellung

---

<sup>1</sup> Rückverfolgbarkeit

---

## 2 Grundlagen

### 2.1 Dialyse

Bei der Niereninsuffizienz sind die Nieren nicht mehr dazu in der Lage, das Blut ausreichend zu filtern und die harnpflichtigen Substanzen auszuscheiden. Dadurch steigt die Konzentration dieser Stoffe im Blut, wodurch Organe und Vorgänge im Körper negativ beeinflusst werden. Als Konsequenz der schlechteren Filterfunktion kann es auch dazu kommen, dass weniger Wasser und Elektrolyte ausgeschieden werden, was zu Wassereinlagerungen und Bluthochdruck führen kann. Eine verminderte Produktion von Erythropoetin kann zu einer Anämie (Blutarmut) führen. Um den Symptomen entgegen zu wirken, wird die Dialyse eingesetzt [2, 27]. Es gibt verschiedene Verfahren, die angewendet werden können.

#### **Ablauf der Hämodialyse**

Die Dialyse ist ein lebensnotwendiges Verfahren für Patienten mit einer Niereninsuffizienz, welche die Möglichkeit der extrakorporalen Reinigung des Blutes von Schadstoffen bietet. Hierbei wird dem Patienten über einen Shunt<sup>2</sup> unreines Blut über die arterielle Punktionsstelle entzogen und über die venöse Punktionsstelle das gereinigte Blut zurückgegeben.

Über die arterielle Punktionsstelle gelangt das Blut in den Dialysator, welcher die künstliche Niere darstellt (siehe Abbildung 2.2). Dialysatoren bestehen aus einer großen Anzahl von hohlen Mikrofilamenten. In den Filamenten wird das Blut befördert und von Dialysat umspült. Die Wände der Mikrofilamente sind semipermeabel (halbdurchlässig), wodurch Wasser und Abfallstoffe die Membran passieren können, Blutkörperchen und große Makromoleküle jedoch nicht (siehe Abbildung 2.1).

---

<sup>2</sup> arterio-venöse Verbindung

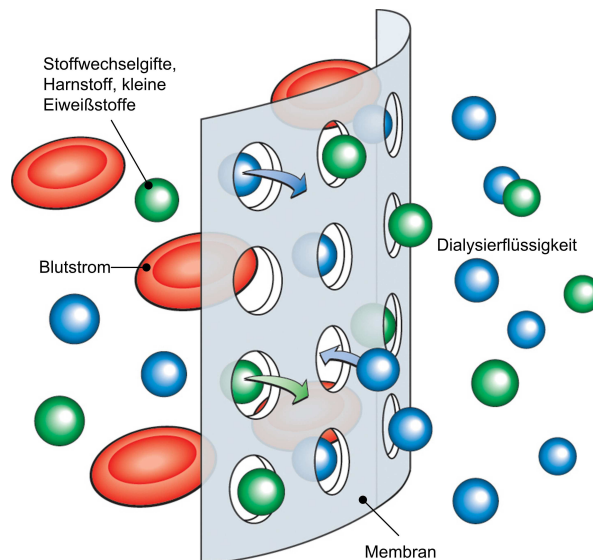


Abbildung 2.1: Schematische Darstellung des Stofftransports durch die semipermeable Membran [4].

Durch Diffusion, Filtration und Konvektion gehen stickstoffhaltige Abbauprodukte und andere Abfallsubstanzen durch die semipermeablen Filamente in die Dialysatflüssigkeit über. Durch die Zusammensetzung des Dialysats ist es möglich, extreme Elektrolytverluste zu vermeiden und dennoch das Blut von Schadstoffen zu reinigen. Aufgrund eines Überdrucks auf der Blutseite und eines Unterdrucks auf der Dialysatseite kommt es zur Ultrafiltration. Dieser Begriff bezeichnet den Durchtritt des Blutplasmas in das Dialysat. Dadurch kann einer Überwässerung des Patienten entgegengewirkt werden. Das überschüssige Wasser des Körpers sammelt sich zum Großteil nicht im leicht erreichbaren Blut, sondern im umliegenden intrazellulären Gewebe und im Interstitium an. Da mithilfe der Dialyse aber nur die Entwässerung über das Blut erfolgen kann, muss die Ultrafiltrationsrate so gewählt werden, dass ausreichend Flüssigkeit aus den Zellgeweben in das Blut nachfließen kann. Diesen Vorgang bezeichnet man Refilling<sup>3</sup>. Es gibt zwei Hauptmerkmale des Patienten, die dazu herangezogen werden können. Dies sind einerseits der Blutdruck und andererseits das Blutvolumen. Beide Parameter geben Aufschluss über das Wohlbefinden und den Wasserhaushalt des Patienten. Zu hohe Ultrafiltrationsraten führen zu niedrigem Blutdruck, weil das Refilling nicht schnell genug erfolgt. Schließlich fließt das gereinigte Blut wieder über die venöse Punktionsstelle in den Blutkreislauf des Patienten [2, 14, 27].

<sup>3</sup> Wiederauffüllen

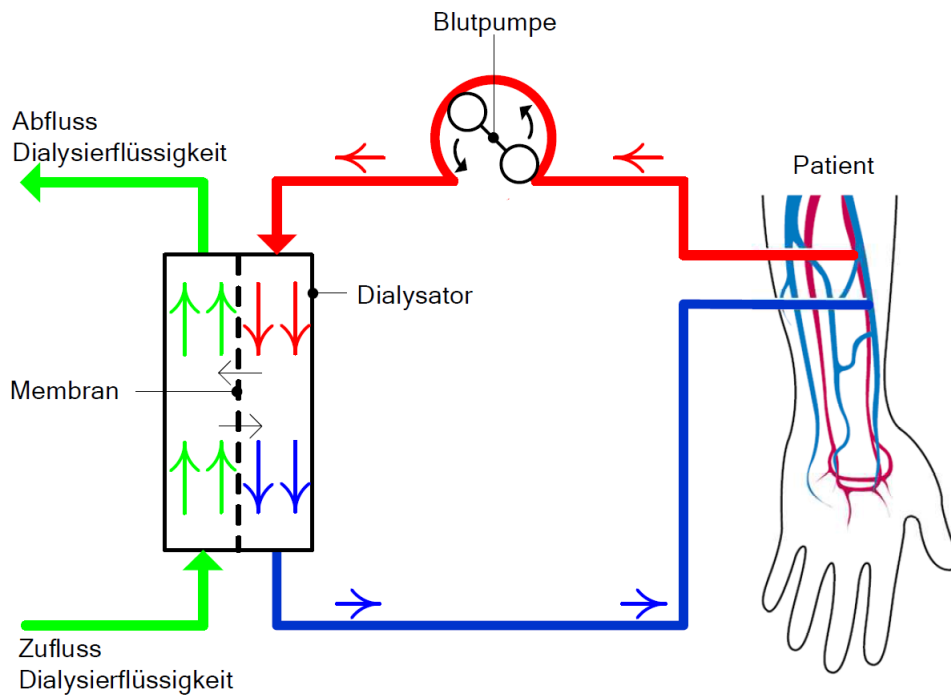


Abbildung 2.2: Ablauf der Hämodialyse: Die rote Linie kennzeichnet die arterielle Blutabnahme und die grüne Linie kennzeichnet die venöse Blutrückführung. Die blaue Linien stellen das Dialysat dar.

Neben der Blutreinigung wird während der Dialyse auch der Elektrolythaushalt (Natrium, Kalium, Calcium, Magnesium) sowie das Säure-Base-Gleichgewicht des Blutes reguliert [27].

## 2.2 V-Modell

Das V-Modell ist ein anerkannter Entwicklungsstandard für Informationstechnische Systeme (IT Systeme) und wurde entwickelt, um die Planung von Projekten zu systematisieren und dadurch die Qualität der Ergebnisse zu steigern. Veröffentlicht wurde das V-Modell erstmals im August 1992 [19]. Im Juni 1997 [18] und im Februar 2005 [13] kam es zu einer Aktualisierung an den jeweiligen Stand der Technik.

Es handelt sich bei dem V-Modell um ein Vorgehensmodell, mit dem nicht nur die Qualität, sondern auch die Nachverfolgbarkeit der Systementwicklung deutlich verbessert werden kann. Die Systementwicklung und auch die Entwicklung von Unterstützungssystemen, welche zum Beispiel zum Testen und Warten verwendet werden, erfolgt durch hierarchische Zerlegung in immer kleinere Einheiten (Produkt  $\rightarrow$  System  $\rightarrow$  Komponente). Die Realisierung und Integration von Systemen wird gemäß den Spezifikationen durchgeführt [28].



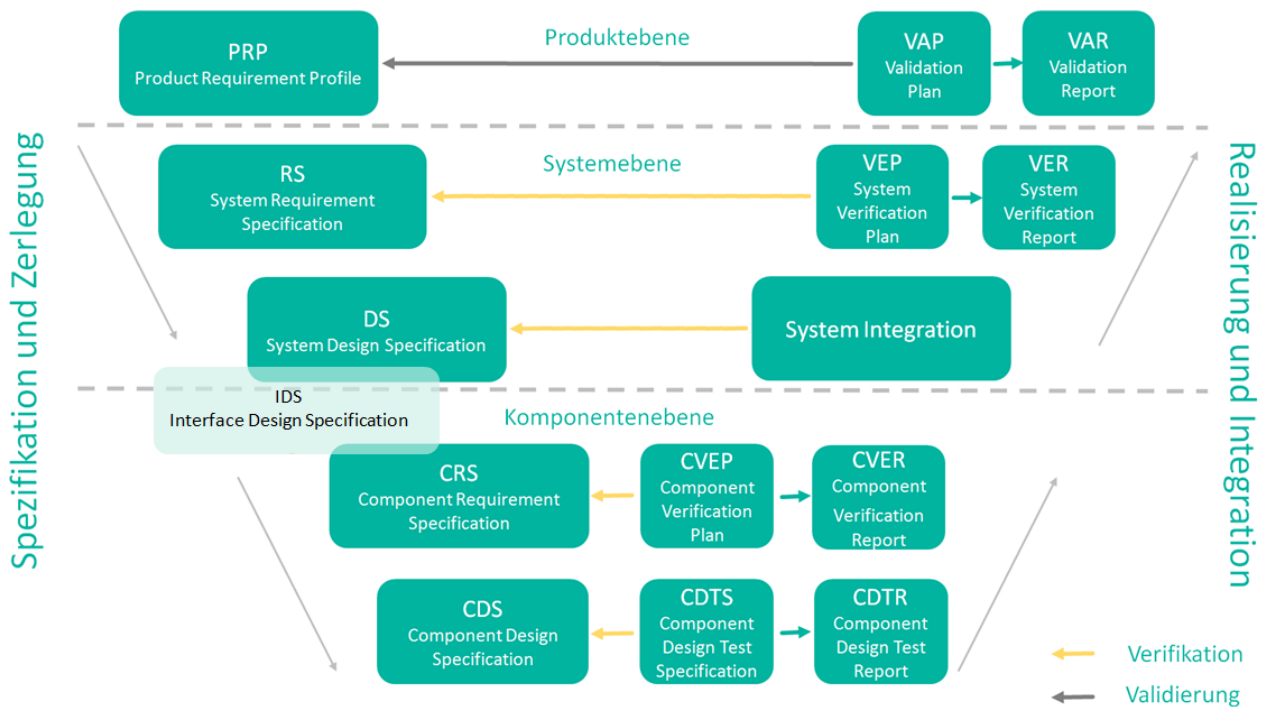


Abbildung 2.3: V-Modell: Produkt Verifikations- & Validierungsstrategie

In Abbildung 2.3 ist das V-Modell eines Produktes, wie zum Beispiel der NDS, dargestellt. Die Komponentenebene bildet die Basis dieser Masterarbeit, denn die Component Requirement Specification (CRS) beinhaltet die Anforderungen an Komponenten und dazugehörige Software. Mithilfe des Component Designs (CD) muss die Component Requirement Specification (CRS) umgesetzt werden. An der Schnittstelle zwischen System- und Komponentenebene befindet sich die Interface Design Specification (IDS), welche neben der CRS, Kernbestandteil dieser Arbeit ist. Nachdem das Design entwickelt wurde, muss getestet werden, ob das Design auch der Spezifikation entspricht. Dies geschieht durch die Test Spezifikation und den Test Report.

## 2.3 Hardware-in-the-Loop-Testsysteme

Tests bieten die Möglichkeit, die Zuverlässigkeit nachzuweisen und Fehlfunktionen vorzubeugen. Hardware-in-the-Loop (HiL) Testsysteme dienen der Qualitätssicherung. Reale Hardwarekomponenten werden dabei in eine simulierte Umgebung eingebunden, die dazu dient die System-Umgebung nachzubilden. Fehler, die durch Umwelt, Temperatur oder EMV-Einflüsse hervorgerufen werden, werden im Rahmen des HiL-Tests nicht betrachtet. Vor allem sollen diese Tests funktionelle Spezifikationen abdecken und untersuchen. Von Relevanz kann auch die Untersuchung von fehlerhaften Zuständen oder nicht plausiblen Zuständen sein. Diese können mit einem HiL-Testsystem einfacher in der Simulation als

in Realität überprüft werden. Mit einem HiL-Testsystem werden die Funktionen direkt an den Pins der Hardwarekomponente überprüft. Über einen PC werden die Testprogramme gesteuert. HiL Testsysteme können als manuelle Tests oder auch als vollautomatische Tests entwickelt werden. Bei voll automatischen Tests ist oft auch die Generierung von Protokollen aller Testfälle implementiert. Dies bietet einen Vorteil für die Person, die mit diesem Testsystem arbeitet, da sich der dokumentarische Aufwand von Einzelergebnissen vermindert. Der Test kann ein Gesamtergebnis liefern, in dem eine Gesamtbewertung inbegriffen ist, die das Ergebnis auf *Test in Ordnung* oder *Test fehlgeschlagen* reduziert. Einmal entwickelte Tests können ohne großen Aufwand immer wieder durchgeführt und modifiziert werden. Dies führt zu einer hohen Wirtschaftlichkeit der Tests. Wichtig für den Entwickler eines HiL-Testsystems ist ein fundiertes Wissen über Modellbildung und Simulation. Die Qualität des Tests ist davon abhängig, wie gut die Testfälle spezifiziert und ausgeführt wurden. Nur so können Fehler und Lücken aufgedeckt werden. Die Softwarebasierte Simulation hat den großen Nachteil, dass die genauen Betriebsbedingungen nicht nachgebildet werden können. HiL bietet die Möglichkeit, die Lücke zwischen reiner Simulation und Normalbetrieb zu überbrücken und ermöglicht das Testen unter realen Bedingungen. So kann das Risiko, Fehler erst im Systemtest zu erkennen, reduziert werden. HiL-Testsysteme unterscheiden sich von der Simulation durch die Integration einer realen Komponente, wie beispielsweise einem MIB, aus einer Dialysemaschine [1, 21, 20].

## 2.4 Software

### 2.4.1 IBM - Rational DOORS

DOORS steht für Dynamic Object Oriented Requirements System und dient dem Anforderungsmanagement. Diese Software bietet die Möglichkeit, Anforderungen an ein Produkt zusammenzutragen, um sie so den Mitarbeitern besser verständlich zu machen. Neben Erfassung, Verfolgung, Analyse und dem Änderungsmanagement von Informationen, dient DOORS dem Nachweis, dass Anforderungen und Standards erfüllt werden. Mitarbeiter können Anforderungen somit besser bearbeiten und prüfen. Die Anforderungen können mit Attributen versehen werden, sodass der User leicht nach den gewünschten Anforderungen filtern kann. Jede Anforderung wird mit einer Identifikationsnummer eindeutig gekennzeichnet. Einzelne Anforderungen können auch miteinander oder mit Testplänen oder Testfällen verknüpft werden, um so die Traceability zu ermöglichen [15].

### 2.4.2 IBM - Rational Quality Manager

Rational Quality Manager (RQM) ist eine Test-Management-Tool, welches seine Anwendung im Bereich des Qualitätsmanagements findet. Es dient der Testplanung, Testkonstruktion und Testausführung. RQM bietet die Möglichkeit, dass mit reduzierten Betriebs- und Verwaltungskosten, verbesserte produktübergreifende Transparenz und Berichterstattung erreicht werden kann. In RQM sind Tests in Testpläne eingeteilt. Jeder Testplan enthält bestimmte Anforderungen, die getestet werden sollen. Entsprechend der Anforderungen werden Testfälle und Testscripte erstellt. Bei einem manuellen Test zeigt RQM jeden Schritt an, der getestet werden muss. Der Tester führt die Schritte durch und fügt die Ergebnisse in RQM ein. Bei einem automatischen Test ruft RQM das entsprechende Programm auf und führt den Test durch. Das Ergebnis wird dann an RQM zurückgegeben und in RQM gespeichert [3, 16].

Neben der übersichtlichen Verwaltung von Tests, besitzt das RQM-Tool die Möglichkeit einer nahtlosen Anbindung zu IBM DOORS und NI TestStand.

### 2.4.3 National Instruments - TestStand

TestStand ist eine Testmanagementsoftware, welche dazu eingesetzt wird, automatisierte Tests schneller zu entwickeln. Mit einem Sequenzeditor können Testsequenzen erstellt werden, welche Codemodule (zum Beispiel: LabVIEW, C, C++, .NET) einbinden können. Zusätzlich können Abarbeitungsreihenfolgen, Berichterstellung und die Datenbankanbindung festgelegt werden. Mittels TestStand können mehrere Geräte gleichzeitig getestet werden und auch an einem Gerät mehrere Tests ausgeführt werden. TestStand bietet die Möglichkeit, automatische Berichte für jede Prüfung in verschiedenen Standardformaten (XML, HTML, TXT) zu erstellen. Durch die genannten Merkmale und Vorteile können der Prüfdurchsatz gesteigert und Prüfsysteme schneller entwickelt werden [26].

Durch einen NI Test Integrations Adapter kann eine Verbindung zwischen TestStand und RQM hergestellt werden. Dadurch können TestStand Anwendungen und Parameter mit RQM Testfällen verknüpft werden und so die Traceability in Projekten betreffend der Anforderungen, des Qualitätsplan und im Bereich des Änderungsmanagement verbessert werden. Ausführungsstatus und Testergebnisse von TestStand können über RQM, lokal oder über das Netzwerk, aufgerufen werden[25].

### 2.4.4 National Instruments - LabVIEW

LabVIEW steht für Laboratory Virtual Instrumentation Engineering Workbench und ist eine grafische Programmierumgebung entwickelt von National Instruments.

Einsatzgebiete von LabVIEW:

- Datenerfassung und Analyse
- Steuerung von Messgeräten
- Automatisierungstechnik

LabVIEW bietet eine Reihe von Eigenschaften für den Entwickler, unter anderem Reduzierung von Entwicklungszeiten durch einfache Integration verschiedener Datenerfassungsgeräte und Sensoren, einfache Automatisierung, paralleler Abläufe durch Multithreading und einfache Entwicklung von Benutzeroberflächen. Außerdem besitzt es eine sehr große Palette an standardisierten Funktionen, die ein breites Applikationsspektrum abdecken. Dieses reicht von der industriellen Bildverarbeitung, über Feldbussysteme, embedded Sensorsignalverarbeitung bis hin zu echtzeitfähigen Systemen. Funktionsblöcke werden in LabVIEW Virtuelle Instrumente (VIs) genannt und bestehen aus drei Hauptbestandteilen: Dem Frontpanel, welches die Benutzerschnittstelle enthält, dem Blockdiagramm, welches den grafischen Programmcode enthält und dem Symbol/ Anschlussfeld, welches die Ein- und Ausgänge eines VIs anzeigt [24].

## 2.5 Dialysemaschine NDS

Die Dialysemaschine NDS wurde am 22. Mai 2016 von B. Braun unter dem Namen Dialog iQ in den Markt eingeführt. NDS steht für New Dialysis System. In der vorliegenden Arbeit wird die Maschine als NDS bezeichnet.



Abbildung 2.4: Dialysemaschine NDS [9]

Die NDS wird über drei Mikroprozessorsysteme gesteuert, darunter der Top-Level-Controller (TLC) der Low-Level-Controller (LLC) und der Low-Level-Supervisor (LLS). Der Aufbau der NDS ist nach funktionellen Einheiten gegliedert [10]:

- **Top Level System**

Die Bedienung der NDS erfolgt über einen Touchscreen. Im Monitorgehäuse befindet sich das MIB, das User-Interface-Board (UIB), die Operation-State-Displays (OSDs) und der TLC. Der TLC bildet die Schnittstelle zwischen dem Benutzer und der Steuerung der Dialyse.

- **Electronic Sub-Rack**

Dieser Einschub wird weiter unterteilt in Motor Online Board, Power-Board-Valves, Low-Level-Digital-Board (LLD), Analog-Board und das Basic-Board. Auf dem LLD befindet sich der LLC und der LLS. Der LLC steuert die gesamte Aktorik und Sensorik und wertet zusätzlich die Sensorsignale aus. Der LLS ist Teil des Sicherheitskonzepts der NDS. Er überwacht die Aktorik und Sensorik komplett unabhängig vom Steuerungskanal, um sicherzustellen, dass der LLC diese korrekt ansteuert.

- **UF Sub-Rack**

In dieser Ebene befinden sich die meisten Komponenten die mit der Ultrafiltration in Zusammenhang gebracht werden. Dazu gehören verschiedene Ventile, Positionssensoren, der Blutleckdetektor und die Bilanzierungskammer.

- **DF Sub-Rack**

Dieses Rack beinhaltet verschiedene Pumpen, darunter die Konzentratpumpe, die Pumpe für die Beförderung von Bicarbonat und die Ultrafiltrationspumpe. Zudem sind auf diesem Einschub Temperatursensoren, Leitfähigkeitssensoren und Ventile.

- **Water Sub-Rack**

Auf dieser Ebene erfolgt die Temperierung und Entgasung des verwendeten Wassers. Sie umfasst einen Heizstab, verschiedene Ventile, einen Vorlaufbehälter, Temperatur und Drucksensoren, Fluss- und Entgasungspumpen.

## 2.6 NDS Komponente - Monitor Interface Board

Das Monitor Interface Board (MIB) (siehe Abbildung 2.6), welches sich im Top Level System der NDS befindet, wird nach dem Bestückungsplan (siehe Abbildung 2.5) assembliert.

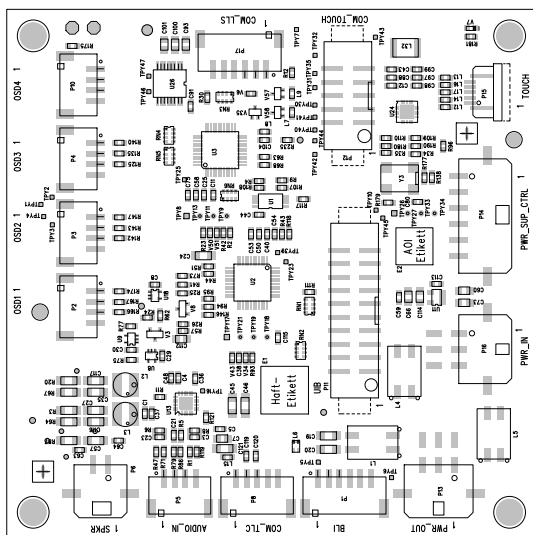


Abbildung 2.5: Bestückungsplan MIB [23]

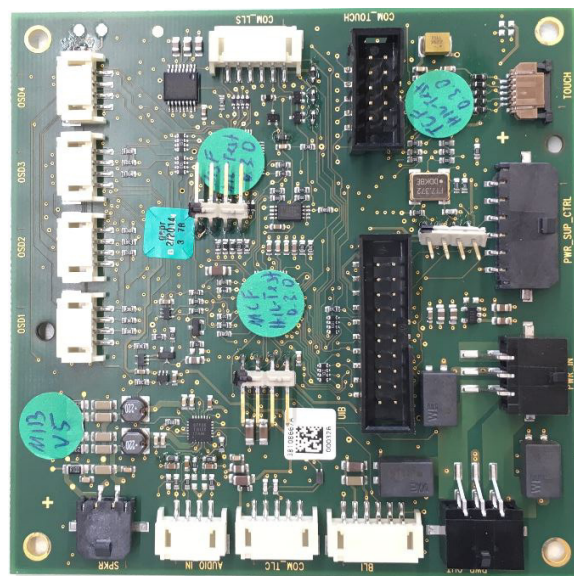


Abbildung 2.6: Entwicklungsmuster MIB V5 (mit Debug-Steckern)

Es dient dem Informationsaustausch zwischen Anwender und Maschine und realisiert folgende Funktionen:

- Auslesen der Hardkeys-Statusinformationen
- Steuerung und Statusabfrage der Hardkey LEDs
- Abfrage der Touch-Positionsdaten

- Steuerung der Lautstärke des Lautsprechers sowie der Ausgabe eines Audiosignals
- Steuerung der OSDs
- Steuerung der Helligkeit der Display-Hintergrundbeleuchtung
- Auslesen und Einlesen von Parametern (z. B. Firmware-Version, Hardware-Version)

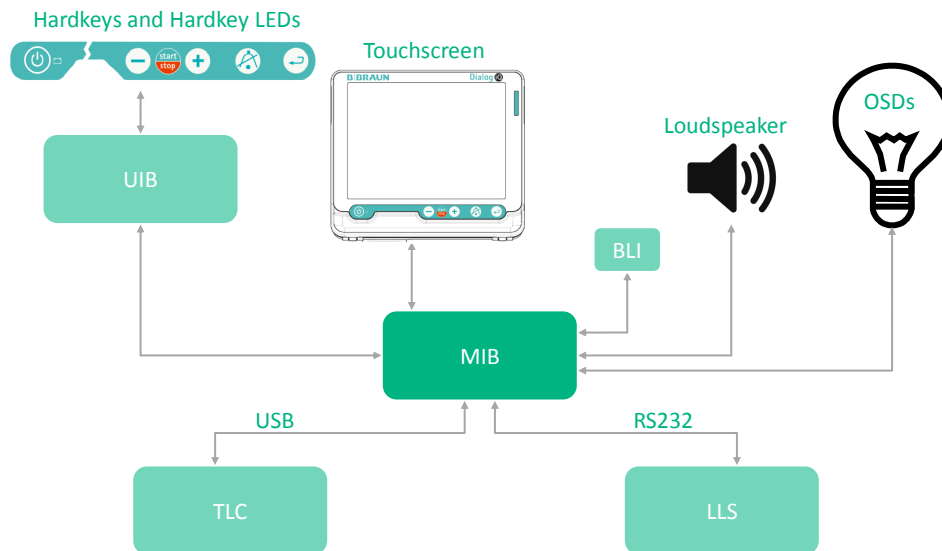


Abbildung 2.7: Schnittstellen des Monitor Interface Boards

Der Bediener steuert das Dialysegerät durch Berührung des Touchscreens sowie durch Drücken der Hardkeys auf dem UIB. Diese Informationen werden sowohl an den LLS als auch den TLC über unabhängige Kommunikationskanäle weitergegeben. Durch den zweikanaligen Aufbau kann die Steuerung durch den TLC über den LLS überwacht werden und gegebenenfalls greift der LLS in den Steuerprozess des TLC ein, wodurch die Patientensicherheit sichergestellt wird. Die Schnittstellen des MIB werden in Abbildung 2.7 vereinfacht dargestellt.

Das MIB wird über drei Mikroprozessorsysteme gesteuert. Darunter der MIB-TLC- Mikrocontroller, der MIB-LLS-Mikrocontroller und der Touch-Mikrocontroller. Mit Hilfe dieser drei Mikrocontroller wird die Kommunikation mit dem TLC und dem LLS realisiert.

Über das Operation-State-Display (OSD) werden dem Anwender über LEDs verschiedene Betriebszustände signalisiert. Zusätzlich verstärkt das MIB Audiosignale vom TLC

und gibt sie über einen Lautsprecher aus. Auf die einzelnen Komponenten wird in den folgenden Absätzen genauer eingegangen.

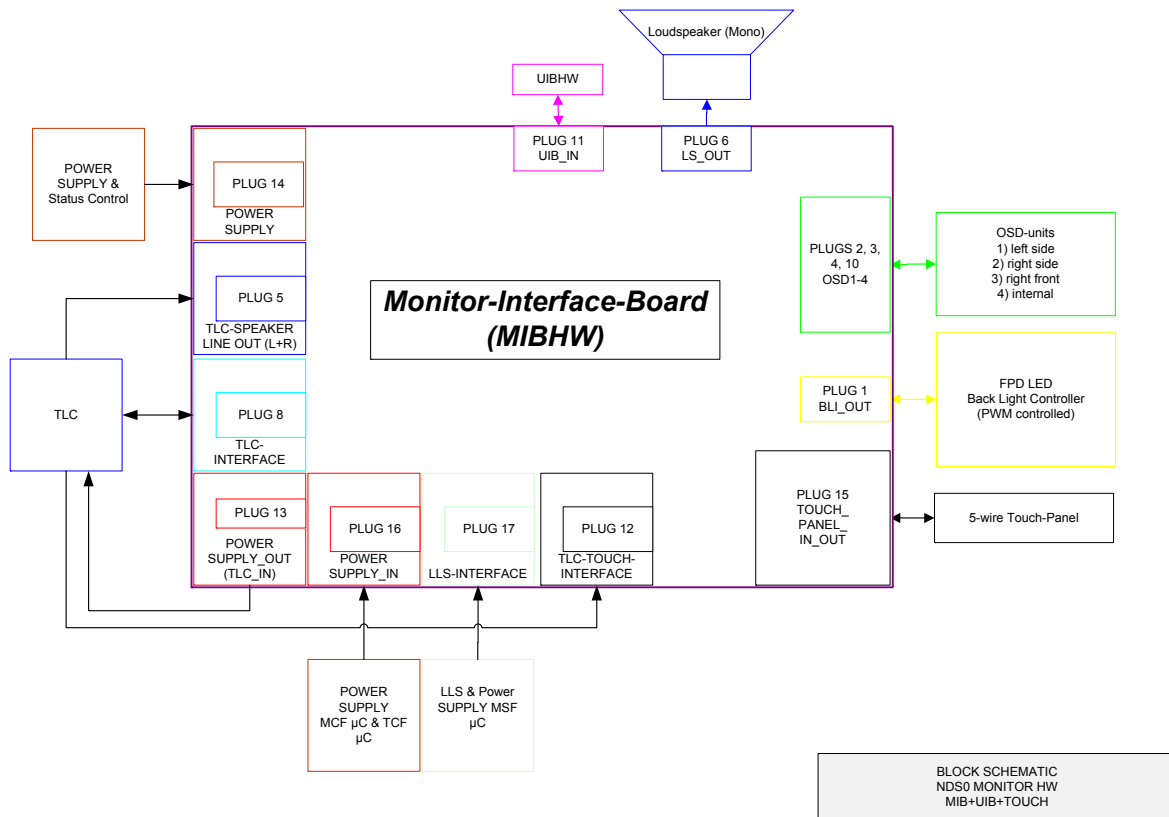


Abbildung 2.8: Darstellung der MIB Komponenten und Interfaces

### 2.6.1 Operation State Display

Über das Operation State Display (OSD) werden die Betriebszustände des Dialysegerätes signalisiert, wobei für die NDS nur die Signalfarben rot, gelb und grün verwendet werden (siehe Tabelle 2.1). Jede der OSDs besteht aus drei LEDs (Rot, Grün, Blau). Die OSDs werden durch den TLC über den MIB-TLC-Mikrocontroller (MCF) gesteuert. Außerdem hat der MIB-LLS-Mikrocontroller (MSF) zusätzlich die Möglichkeit, alle roten OSDs zu aktivieren. Diese Funktion ist völlig unabhängig von der MCF Steuerung und hat eine höhere Priorität. Der MSF ermittelt kontinuierlich den aktuellen OSD-Hardware Aktivierungszustand und sendet diese Information auf LLS-Anfrage an diesen.



Tabelle 2.1: CD-OSDHW-1053: Alarm Prioritäten und Signalfarben

Priority-Index	Alarm Priorität	Signal Farbe	R	G	B	Funktion
3	Alarm mit höchster Priorität	Rot	1	0	0	langsam blinkend (0.6 Hz) schnell blinkend (2.4 Hz)
2	Alarm mit mittlerer oder niedriger Priorität	Gelb	1	1	0	langsam (0.6 Hz) schnell blinkend (2.4 Hz)
1	Betriebsbereit	Grün	0	1	0	langsam blinkend (0.6 Hz) schnell blinkend (2.4 Hz) Dimmfunktion

Alle OSDs haben die Möglichkeit, mit einer bestimmten Frequenz zu blinken. Die Frequenzen mit der die LEDs blinken können sind in der IEC 60601-1-8 §6.3.2.2 Tabelle 2 [17] verankert. Die von der Internationalen Elektrotechnischen Kommission (IEC) herausgegebene Norm IEC 60601-1-8 definiert die Charakteristiken und Anforderungen an Alarmsysteme für Medizingeräte. Die Blinkfrequenz entspricht bei mittlerer Alarm Priorität 0.6 Hz und bei hoher Alarmpriorität 2.4 Hz.

## 2.6.2 Touchpanel

Bei dem Touchscreenmonitor der Dialysemaschine handelt es sich um einen 5-Draht resistiven druckempfindlichen Touchscreen. Das in der Monitorfront integrierte Touchpanel verwendet zwei Indiumzinnoxid (ITO) Schichten. Die obere flexible Schicht liegt dabei über der unteren starren Schicht und ist von dieser mit Hilfe von Abstandshaltern, sogenannten Spacern, getrennt.

An den Eckpunkten der unteren Schicht befindet sich jeweils eine Elektrode. Um die Touch-Position zu bestimmen, werden mit Hilfe eines Touch-Panel-Mikrocontrollers jeweils die beiden benachbarten Elektroden, über vom Touch-Panel-Controller angesteuerte Treiberkanäle, elektrisch verbunden und die Spannung +5V beziehungsweise Masse angelegt (siehe Abbildung 2.10). Anschließend erfolgt dieselbe Operation für das zweite Elektrodenpaar. Die obere Schicht dient als Messelement.

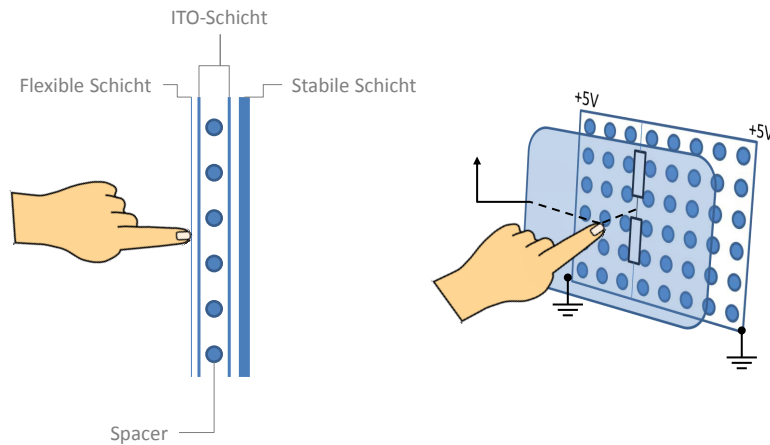


Abbildung 2.9: Aufbau und Funktion eines 5-Draht-Touchscreens

Durch Anlegen der Spannung (+5V) und Masse an den jeweiligen Elektrodenpaaren entsteht ein Spannungsgradient entlang der Schicht. Bei der Berührung der oberen Schicht werden die leitfähigen ITO Schichten in zwei Widerstandsbereiche geteilt, die einen elektrischen Spannungsteiler ausbilden und proportional zur Touch-Position sind. Über den fünften Draht wird diese Spannung der unteren stabilen Schicht weitergeleitet. Durch das Ergebnis in X-Richtung und das Ergebnis in Y-Richtung kann die Touch-Position bestimmt werden [22].

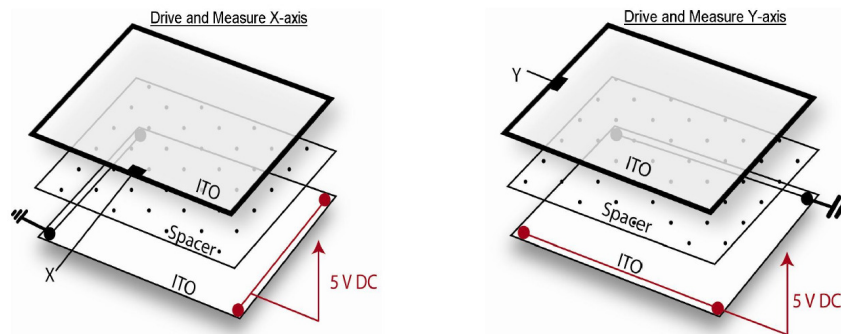


Abbildung 2.10: Bestimmung der Touch-Position in X-Richtung und in Y-Richtung [22]

### 2.6.3 Lautsprecher

Das vom TLC erzeugte Audiosignal wird auf dem MIB verstärkt (Bauteil: MAX9768) und über einen Lautsprecher ausgegeben. Der verwendete Klasse-D-Verstärker verfügt über eine Lautstärkeregelung sowie eine Mute-Funktion. Die Lautstärkeregelung kann analog oder mit Hilfe eines  $I^2C^4$ -Bus umgesetzt werden. Auf dem MIB ist der  $I^2C$  Modus realisiert, wodurch über ein Befehlsregister die Lautstärke eingestellt wird.

<sup>4</sup> Inter-Integrated Circuit

## 2.6.4 User Interface Board

Das User Interface Board (UIB) bietet eine Eingabemöglichkeit des Menschen an die Maschine. Die Hauptbestandteile sind Hardkeys-Taster und LEDs, die der Hintergrundbeleuchtung der Hardkeys dienen. Lediglich die Power ON/OFF Taste, die Blutpumpe (BP) START / STOP Taste, die AQ Taste und die EQ Taste bieten eine Hinterleuchtungsfunktion zur Benutzerführung am Gerät. Die BP+ und die BP- Taste werden nicht beleuchtet. Die Steuerung der Hintergrundbeleuchtung für die Tasten erfolgt über den MCF.

Alle Tasten sind mit dem MCF und dem MSF verbunden, um im Fall einer Fehlfunktion eines Mikrocontrollers oder der Versorgungsspannung eine unabhängige Bestimmung der Hardkey-Tasten ermöglichen zu können. Die Hintergrundbeleuchtung der Tasten wird nicht überwacht.

UIB Hardkeys :

- POWER ON/OFF Taste (Dialysegerät ein- und ausschalten)
- Blutpumpe + Taste (Blutfluss erhöhen)
- Blutpumpe – Taste (Blutfluss verringern)
- Blutpumpe START / STOP Taste (Blutpumpe starten /Blutpumpe stoppen)
- AQ Taste (Alarm Mute / Alarm quittieren)
- EQ Taste (Eingegebene Daten bestätigen)

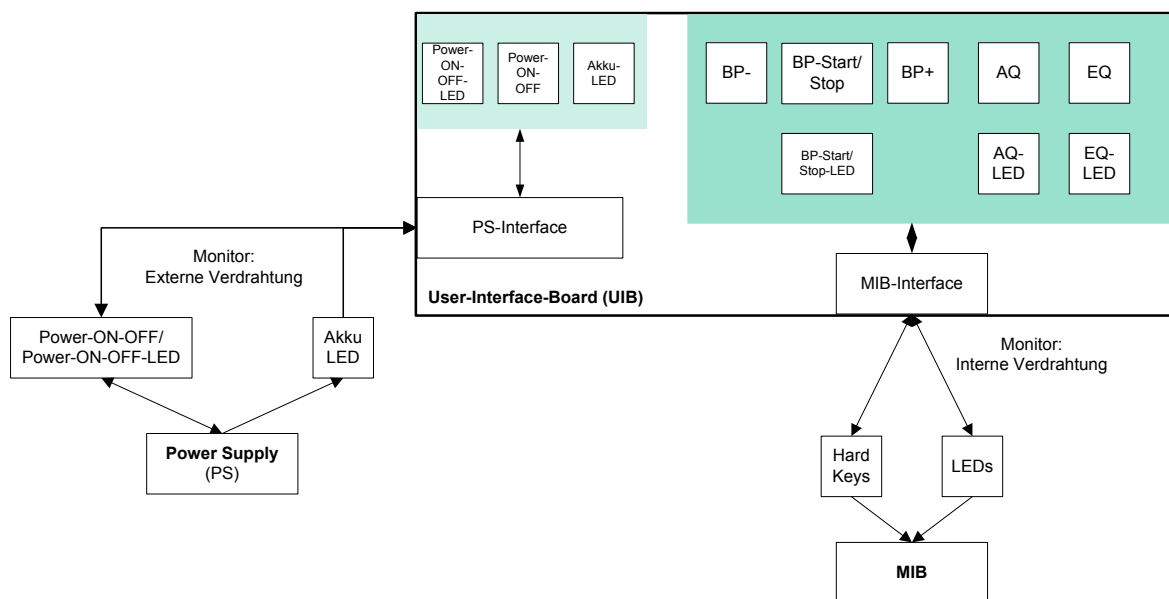


Abbildung 2.11: Hardkeys und LEDs des UIB

Zusätzlich besitzt das UIB die Akku-LED, welche dazu dient, den Ladestatus, der im Dialysegerät integrierten Batterie, anzuzeigen. Die Steuerung der Akku-LED obliegt dabei allein dem Netzteil des Dialysegeräts. Da diese Ansteuerung funktionell nicht ausgewertet wird, ist sie nicht Teil dieser Arbeit.

### 2.6.5 Kommunikation MIB - TLC

Die Kommunikation zwischen dem MIB und dem TLC findet über den MIB-TLC-Mikrocontroller statt. Es handelt sich hierbei um den Mikrocontroller MB95F394 der Firma Fujitsu. Dessen RS232-Schnittstelle wird über einen RS232-USB Konverter (FTD232 der Firma Future Technology Devices International) zu einer USB Schnittstelle konvertiert. Die Kommunikation zwischen dem MIB und dem TLC basiert auf einem an das Modbus-Protokoll angelehntes proprietäres serielles Kommunikationsprotokoll und nutzt hexadezimale ASCII Zeichen. Registerorientiert können Nachrichten geschrieben und gelesen werden. Eine Nachricht vom TLC besteht aus sieben Bytes. Das MIB empfängt die Nachricht, verarbeitet sie und sendet eine Antwort an den TLC zurück. Tabelle 2.2 zeigt das Kommunikationsformat zwischen dem TLC und dem MIB.

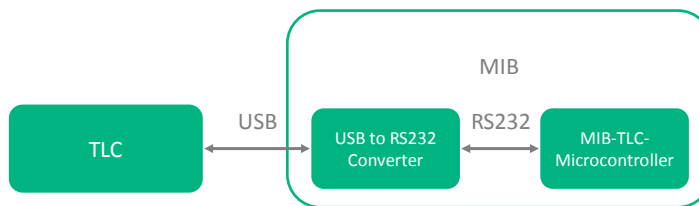


Abbildung 2.12: Datenfluss TLC-Signale

Parameter der Datenübertragung (IDS-3157):

- Baudrate: 19200 Bits/s
- Datenbits: 8
- Bit-order: Das niederwertigste Bit (LSB) wird zuerst übertragen
- Parität: gerade
- Stoppbits: 1

Tabelle 2.2: IDS-3171: TLC-MIB Kommunikationsprotokoll

Beschreibung	SOM	Funktionscode: Read/Write	Registeradresse	Registerwert	CRC		EOM
# Bytes	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Befehl Bytes	0x3A	0x03 0x06	0xXX	0xYY	0xZZ (low-byte)	0xZZ (high-byte)	0x0D

**Legende zur Tabelle 2.2:**

- Start-of-Message (SOM): Zeigt den Beginn des Befehls
- Funktionscode Read/Write: Der Funktionscode definiert, ob geschrieben oder gelesen wird
- Registeradresse: Adresse des Registers
- Registerwert: Wert des Registers
- CRC: Es wird die CRC16 aus Funktionscode, Registeradresse und Registerwert ermittelt. Der Sollwert wird später mit dem Istwert verglichen, um Übertragungsfehler zu erkennen.
- End-of-Message (EOM): Zeigt das Ende des Befehls

**2.6.5.1 Funktionscode**

Ein MIB-Datenwort (Bits7-0) besteht aus 8 Bits, wobei jeweils 4 Bits zu einem Nibble zusammengefasst werden können. Damit enthält ein MIB-Datenwort ein niederwertiges (Bits 3-0) und höherwertiges Nibble (Bits 7-4). Im niederwertigen Nibble des Funktionscode wird definiert, ob ein Register gelesen oder in ein Register geschrieben werden soll. Außerdem können auftretende Fehler in den Funktionscode integriert werden. Fehler werden über das höherwertigen Nibbel im Funktionscode zurückgegeben (siehe Tabelle 2.3). Die in Tabelle 2.3 nicht verwendeten Prioritäten sind für zukünftige Erweiterungen vorgesehen. Im Falle mehrerer Fehler wird der Fehlercode mit der höchsten Priorität zurückgegeben. Bei Transmissionsfehler oder Fehlern in der CRC können unplausible oder fehlerhafte Befehle entstehen. In diesen Fällen werden Standardwerte zurückgegeben, um sichtbar zu machen, dass der Befehl fehlerhaft war. Der Standardwert für den Funktionscode ist 0x0F. Der Standardwert für die Registeradresse und den Registerwert ist 0xFF.

Tabelle 2.3: IDS-3165 - Funktionscode

Priorität	Funktionscode Bit 7-0	Beschreibung
0x01	0000011b(0x03)	Read-Modus
0x03	0000110b(0x06)	Write-Modus
0x05	<b>0001XXXXb</b> (0x1Y)	Funktionscode hat keinen Effekt
0x06	<b>0010XXXXb</b> (0x2Y)	Register Value ist ungültig (zum Beispiel ein Soundlevel >100%)
0x07	<b>0011XXXXb</b> (0x3Y)	Register Adresse ist ungültig (zum Beispiel Zugriff auf das Register 0xFF)
0x0C	<b>1000XXXXb</b> (0x8Y)	Ungültige CRC
0x0D	<b>1001XXXXb</b> (0x9Y)	Transmissions oder Framing Fehler (zum Beispiel: Fehler in der SOM oder EOM)
0x10	<b>1100XXXXb</b> (0xCY)	MIB-TLC-Mikrocontroller Reset

### 2.6.5.2 Registeradresse

Bei Registeradresse 0x00 bis 0x07 handelt es sich um Register, die ausschließlich gelesen werden können. Die Registeradressen 0x20 bis 0x29 können gelesen und beschrieben werden (siehe Tabelle 2.4).

Eine genaue Beschreibung der Register würde den Rahmen der Grundlagen sprengen, daher enthält Tabelle 2.4 nur eine kurze Beschreibung der einzelnen Register. Eine detaillierte Beschreibung ist dem Anhang im Kapitel 8.1.1 zu entnehmen.

Tabelle 2.4: Lese- und Schreibregister des MIB-TLCs

IDS	Registeradresse [hex]	Zugriff	Registerfunktion	Beschreibung
3184	0x00	R	Hardware Variant Identifier	Bestimmung des verwendeten Hardware Moduls ("M"= 0x4D für MIB)
3186	0x01	R	Hardware Version	Auslesen der Hardware Version
3188	0x02	R	Firmware Version	Auslesen der Firmware Version
3190	0x03	N/A	Hardware Version	ungültige Registeradresse - nur für interne Entwicklungszwecke reserviert
3192	0x04	R	Controller Status	Auslesen des Controller-Reset-Status
3196	0x05	R	Hard Keys Status	Auslesen der Statusinformationen der Hard Keys
3198	0x06	N/A	Power-ON-OFF Hard Key Status	ungültige Registeradresse - nur für interne Entwicklungszwecke reserviert
3200	0x07	N/A	Detailed Firmware Version Information	ungültige Registeradresse - nur für interne Entwicklungszwecke reserviert
3204	0x20	R/W	Reset Control	Durchführung des MIB-TLC Resets
3210	0x21	R/W	Hardkeys LED Control	Steuerung der LEDs des UIBs
3212	0x22	R/W	Loudspeaker Amplifier Sound Level	Lesen und Setzen der Lautstärke (minimale Lautstärke 0%. Defaultwert 100%)
3214	0x23	R/W	OSD RGB Control	Steuerung der Farbe und der Blinkfunktion der OSDs
3216	0x24	R/W	OSD Brightness Control	Steuerung der Helligkeit der OSDs
3219	0x25	R/W	OSD Test Sequence Control	Gezieltes Deaktivieren der OSD LEDs (Simulation von Fehlern)
3222	0x26	R/W	OSD Alarm Environment Configuration Control	OSD-Signalfarbe nach dem unmittelbaren Einschalten des Dialysegerätes bis zur Betriebsbereitschaft
3225	0x27	R/W	FPD Backlight Brightness Control	Steuerung der Hintergrundbeleuchtung des TFT-Displays
3227	0x28	N/A	OSD RGB Individual Test Control	ungültige Registeradresse - nur für interne Entwicklungszwecke reserviert
3230	0x29	R/W	Loudspeaker Amplifier Mute	Steuerung der Mute-Funktion des Lautsprechers

### 2.6.5.3 Registerwert

Über den Registerwert kann der Inhalt eines Registers verändert werden. Wenn man zum Beispiel über den Funktionscode definiert, dass im nächsten Befehl gelesen wird (Funktions-

onscode: 0x03), wird der vom TLC übertragene Registerwert ignoriert. Als Antwort erhält man eine Nachricht, die den Inhalt der gewünschten Registeradresse zurückliefert.

Bei einem Schreibbefehl (Funktionscode: 0x06) wird der Registerwert an der gewünschten Registeradresse auf den vom TLC übertragenen Wert gesetzt. Die MIB-Antwort enthält den zuvor erhaltenen Registerwert.

### 2.6.5.4 Zyklische Redundanzprüfung

Um Fehler während der Übertragung zu erkennen, wird das Verfahren der zyklischen Redundanzprüfung (CRC) angewendet. Die CRC wird vom Sender (TLC) berechnet und in die Nachricht eingebunden. Der Empfänger (MIB) berechnet die CRC der empfangenen Nutzdaten und vergleicht die berechnete CRC mit der übertragenen CRC. Wenn diese beiden Werte nicht übereinstimmen, kommt es zu einem Fehler, der über den Funktionscode zurückgegeben wird.

### 2.6.6 Kommunikation MIB - LLS

Die Kommunikation zwischen MIB und LLS findet über den MIB-LLS-Mikrocontroller statt. Es handelt sich hierbei, wie auch beim MIB-TLC-Mikrocontroller, um einen MB95F394 der Firma Fujitsu. Der Datenaustausch erfolgt über eine RS232-Schnittstelle. Eine Nachricht vom LLS besteht aus einem Byte, welche ebenfalls wie beim MCF mit einer CRC (hier allerdings nur 8 Bit) gesichert ist. Das MIB empfängt diese Nachricht verarbeitet sie und sendet eine Antwort an den LLS zurück.

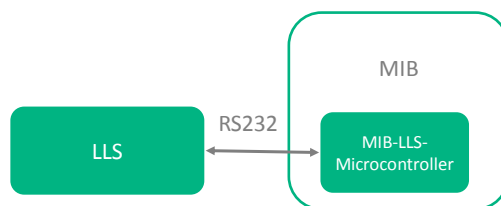


Abbildung 2.13: Datenfluss LLS-Signale

Parameter der Datenübertragung (IDS-3286):

- Baudrate: 19200 Bits/s
- Datenbits: 8



- Bit-order: Das niederwertigste Bit (LSB) wird zuerst übertragen
- Parität: gerade
- Stoppbits: 1

Tabelle 2.5: IDS-3288: LLS-MIB Kommunikationsprotokoll

	Start Bit	Bit0 (lsb)	Bit1	Bits2-6	Bit7 (msb)	Parity	Stop Bit	Start Bit	CRC	Parity	Stop Bit
#Bits	(1)	(1)	(1)	(5)	(1)	(1)	(1)	(1)	(8)	(1)	(1)
Data	0	Data-Bit[0]	Data-Bit[1]	Data-Bits[2:6]	Data-Bit[7]	EP1	0	0	<CRC>	EP2	0

Die logische Aufteilung des übertragenen 8-Bit-Datenworts ist dabei so gewählt, dass zwei Empfangs- und maximal 4-Senderegister adressiert werden können. Es gibt zwei MIB-LLS Receive Register (MIB-LLS Receive Register 0, MIB-LLS Receive Register 1) mit denen Befehle vom LLS an den MSF gesendet werden können. Über die vier MIB-LLS Transmit Register (MIB-LLS Transmit Register 0, MIB-LLS Transmit Register 1, MIB-LLS Transmit Register 2, MIB-LLS Transmit Register 3) können verschiedene Zustände und Informationen vom MIB-LLS-Mikrocontroller an den LLS zurückgesendet werden.

Tabelle 2.6: IDS-3295: MIB-LLS Receive Register 0

Bits	Name	Beschreibung	Default
7-6	Auswahl des Transmit-Registers	Bits[76]=00 → Register 0 Bits[76]=01 → Register 1 Bits[76]=10 → Register 2 Bits[76]=11 → Register 3	00
5	n/a	n/a	0
4	LLS Re-Transmissions Steuerung	0= Re-Transmission deaktiviert 1= Re-Transmission aktiviert	0
3	Auswahl der Blinkfrequenz der roten OSD LED	0=Blinkfrequenz 0.6Hz 1=Blinkfrequenz 2.4Hz	0
2	Blinkfunktion der roten OSD LED aktivieren/deaktivieren	0=Blinken deaktiviert 1=Blinken aktiviert	0
1	Rote OSD LED steuern	0=Rote OSD LED aus 1=Rote OSD LED an	0
0	Receive Register	Auswahl des Receive Registers 0=Receive Register 0 1=Receive Register 1	0

Einige Informationen im MIB-LLS werden bis zum Auslesen durch den LLS zwischengespeichert (Latchfunktion) und erst nach der Übertragung an diesen bis zum nächsten Auslesen gelöscht. Der Status der Hard Keys und des Lautsprechers können nach einer Transmission durch Setzen des Bits 4 des Receive Registers 0 ein weiteres Mal abgefragt werden. Diese Funktion wurde für den Fall eines seriellen Übertragungsfehlers implementiert.

Tabelle 2.7: IDS-3298: MIB-LLS Receive Register 1

Bits	Name	Beschreibung	Default
7-6	Auswahl des Transmitter Registers	Bits[76]=00 → Register 0 Bits[76]=01 → Register 1 Bits[76]=10 → Register 2 Bits[76]=11 → Register 3	00
5	n/a	n/a	0
4	Reset	MIB-LLS Reset Sequenz	0
3	Version Information	0=detaillierte Version Information wird nicht gesendet 1=detaillierte Version Information wird gesendet	0
2	TLC Mute durch LLS	0=TLC Muting aktiviert 1=TLC Muting deaktiviert	0
1	AQ Hardkey LED	0=AQ LED wird durch TLC gesteuert 1=AQ LED wird zusätzlich durch LLS beleuchtet	0
0	Receive Register	Auswahl des Receive Registers 0=Receive Register 0 1=Receive Register 1	0

Der LLS hat als vom TLC unabhängige Überwachungsfunktion aufgrund normativer Anforderungen die Möglichkeit, eine vom TLC unabhängige optische Alarmierung zu generieren. Zu diesem Zweck kann der LLS über den MIB-LLS eine gegenüber dem TLC höher priorisierte optische Alarmierung generieren und kann die Ansteuerung der roten OSD-LEDs (siehe Tabelle 2.6), der AQ LED und der Mute Funktion (siehe Tabelle 2.7) übernehmen.

Tabelle 2.8: IDS-3304: MIB-LLS Transmit Register 0

Bits	Name	Beschreibung	Default
7	Receive Error	Fehler im übertragenen Byte 0= Kein Fehler 1= Fehler	0
6	n/a	n/a	0
5	n/a	n/a	0
4	EQ Hardkey	0= nicht gedrückt 1= gedrückt	0
3	AQ Hardkey	0= nicht gedrückt 1= gedrückt	0
2	BP+ Hardkey	0= nicht gedrückt 1= gedrückt	0
1	BP Start/BP Stop Hardkey	0= nicht gedrückt 1= gedrückt	0
0	BP- Hardkey	0= nicht gedrückt 1= gedrückt	0

Der LLS nimmt eine überwachende Rolle ein. Informationen über den Hardkey-Status der EQ, AQ, BP+, BP- und BP Start/Stop Taste können abgefragt werden (siehe Tabelle 2.8). Außerdem kann der Status der OSD LEDs (siehe Tabelle 2.9) und des Lautsprecher-Outputs (siehe Tabelle 2.9) abgefragt werden. Das Transmit Register 2 enthält die MIB-LLS Firmware Version (siehe Tabelle 2.10) und das Transmit Register 3 bietet Auskunft darüber, ob die Lautsprecher Muting-Funktion aktiviert oder deaktiviert ist und ob ein Lautsprechersignal detektiert wurde oder nicht (siehe Tabelle 2.11).

Tabelle 2.9: IDS-3307: MIB-LLS Transmit Register 1

Bits	Name	Beschreibung	Default
7	Receive Error	Fehler im übertragenen Byte 0= kein Fehler 1= Fehler	0
6	n/a	n/a	0
5	OSD LED Error Status	0=alle OSD LEDs haben den gleichen Status 1=nicht alle OSD LEDs haben den gleichen Status	0
4	Status der blauen OSD LED	0=blaue OSD LED nicht aktiviert 1=blaue OSD LED aktiviert	0
3	Status der grünen OSD LED	0=grüne OSD LED nicht aktiviert 1= grüne OSD LED aktiviert	0
2	Status der roten OSD LED	0=rote OSD nicht aktiviert 1=rote OSD aktiviert	0
1	n/a	n/a	0
0	n/a	n/a	0

Tabelle 2.10: IDS-3310: MIB-LLS Transmit Register 2

Bits	Name	Beschreibung	Default
7	Receive Error	Fehler im übertragenen Byte 0= kein Fehler 1= Fehler	0
6 - 0	MIB-LLS Firmware Version	n/a	0000000

Tabelle 2.11: IDS-3313: MIB-LLS Transmit Register 3

Bits	Name	Beschreibung	Default
7	Receive Error	Fehler im übertragenen Byte 0= kein Fehler 1= Fehler	0
6 - 4	MIB-LLS Reset	Zeigt den MIB-LLS Reset Status	111
3	n/a	n/a	0
2	n/a	n/a	0
1	Lautsprecher Muting Status	0= Lautsprecher wird nicht gemu- tet 1= Lautsprecher ist gemutet	0
0	Lautsprecher Output Status	0= es wurde kein Lautsprechersi- gnal detektiert 1= es wurde ein Lautsprechersi- gnal detektiert	0

Zur Erkennung von seriellen Übertragungsfehlern zwischen MSF und LLS wird das Verfahren der zyklischen Redundanzprüfung (CRC) angewendet. Bei Übertragungsfehlern signalisiert der MSF dem LLS dies über das Bit 7 in seinem zurückgesendeten Transmit Registers.

### 2.6.7 Kommunikation MIB-Touch - TLC

Zur Bestimmung der Touch-Position kommuniziert der MIB-Touch-Host-Mikrocontroller MB95F332 von Fujitsu mit dem Touch-Panel-Controller Mircochip AR1020. Die intern weiter verarbeitete Touch-Position wird dann zyklisch über eine RS232-Schnittstelle zum TLC übertragen (siehe Abbildung 2.14).

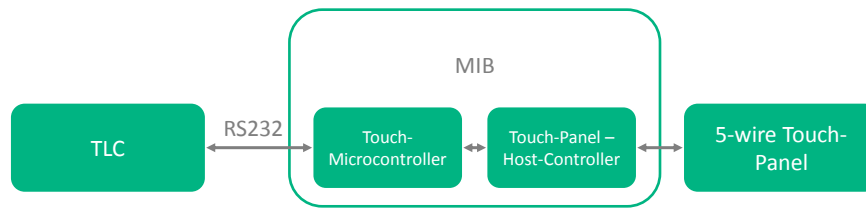


Abbildung 2.14: Datenfluss Touchscreen-Signale

Parameter der Datenübertragung (IDS-3375):

- Baudrate: 9600 Bits/s
- Daten Bits: 8
- Bit-order: Das niederwertigste Bit (LSB) wird zuerst übertragen
- Parität: keine
- Stopp Bits: 1

Die MIB-Touch-Daten enthalten die Touch-Position, welche nach dem TSHARC RS232 Protokoll übermittelt wird (siehe Tabelle 2.12). Die MIB-Touch Daten werden durchschnittlich 25 bis 35 mal pro Sekunde vom MIB-Touch Mikrocontroller an den TLC gesendet. Außerdem ist es möglich, die Firmware Version des MIB-Touch-Mikrocontrollers auszulesen (siehe Tabelle 2.13).

Tabelle 2.12: IDS-3381: MIB-Touch zu TLC: Touch-Data-Protokoll

TSHARC RS232 Protokoll	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
# Bits	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Touch-Position Datenbyte 0	1	P	X11	X10	X9	Y11	Y10	Y9
Touch-Position Datenbyte 1	0	X8	X7	X6	X5	X4	X3	X2
Touch-Position Datenbyte 2	0	Y8	Y7	Y6	Y5	Y4	Y3	Y2
Touch-Position Datenbyte 3	0	0	0	0	X1	X0	Y1	Y0

Tabelle 2.13: IDS-3369: MIB-Touch Firmware Version

MIB-Touch Firmware Version		
FW release state	Major Version	Minor Version
<b>Bit 7</b>	<b>Bit 6-4</b>	<b>Bit 3-0</b>
0: development version 1: released for production	(000 <sub>b</sub> -111 <sub>b</sub> )	(0 <sub>hex</sub> -F <sub>hex</sub> )

Um die Firmware-Version des MIB-Touch-Mikrocontrollers korrekt auslesen zu können, muss ein Befehl entsprechend der Tabelle 2.14 gesendet werden. Dieser Befehl besteht aus

der Start-of-Message dem Registercode und der End-of-Message. Die Antwort des MIBs an den TLC (siehe Tabelle 2.15) besteht aus mindestens 4 Bytes: Start-of-Message-Byte, Registercode, Requestcode, Registerwert und End-of-Message-Byte.

Tabelle 2.14: IDS-3391: TLC → MIB-Touch Kommunikation

Befehl	SOM	Registercode	EOM	Funktion wird unterstützt
# Bytes	(1)	(1)	(1)	N/A
MIB-Touch Firmware Version auslesen	0x3A	0x46	0x3F	Ja
MIB-Touch detaillierte Firmware Version auslesen	0x3A	0x56	0x3F	nur in Entwicklungsversionen
MIB-Touch Reset durchführen	0x3A	0x52	0x21	Ja

Tabelle 2.15: IDS-3393: MIB-Touch → TLC Kommunikation

Befehl	SOM	Registercode	Requestcode	Registerwert	EOM
# Bytes	(1)	(1)	(1)	(≥ 0)	(1)
MIB-Touch Firmware Version	0x3A	0x46	0x3F	<aktuelle MIB-Touch Firmware Version>	0x0D
MIB-Touch detaillierte Firmware Version	0x3A	0x56	0x3F	<detaillierte Firmware Version>	0x0D
MIB-Touch Reset	0x3A	0x52	0x21	N/A	0x0D
MIB-Touch Error	0x3A	0x45	0x52	0x52	0x0D



# 3 Entwurf und Implementierung

Dieser Abschnitt gliedert sich in zwei Hauptabschnitte. Zum besseren Verständnis über das Zusammenwirken von Kapitel 3 und 4 siehe Abbildung 3.1.

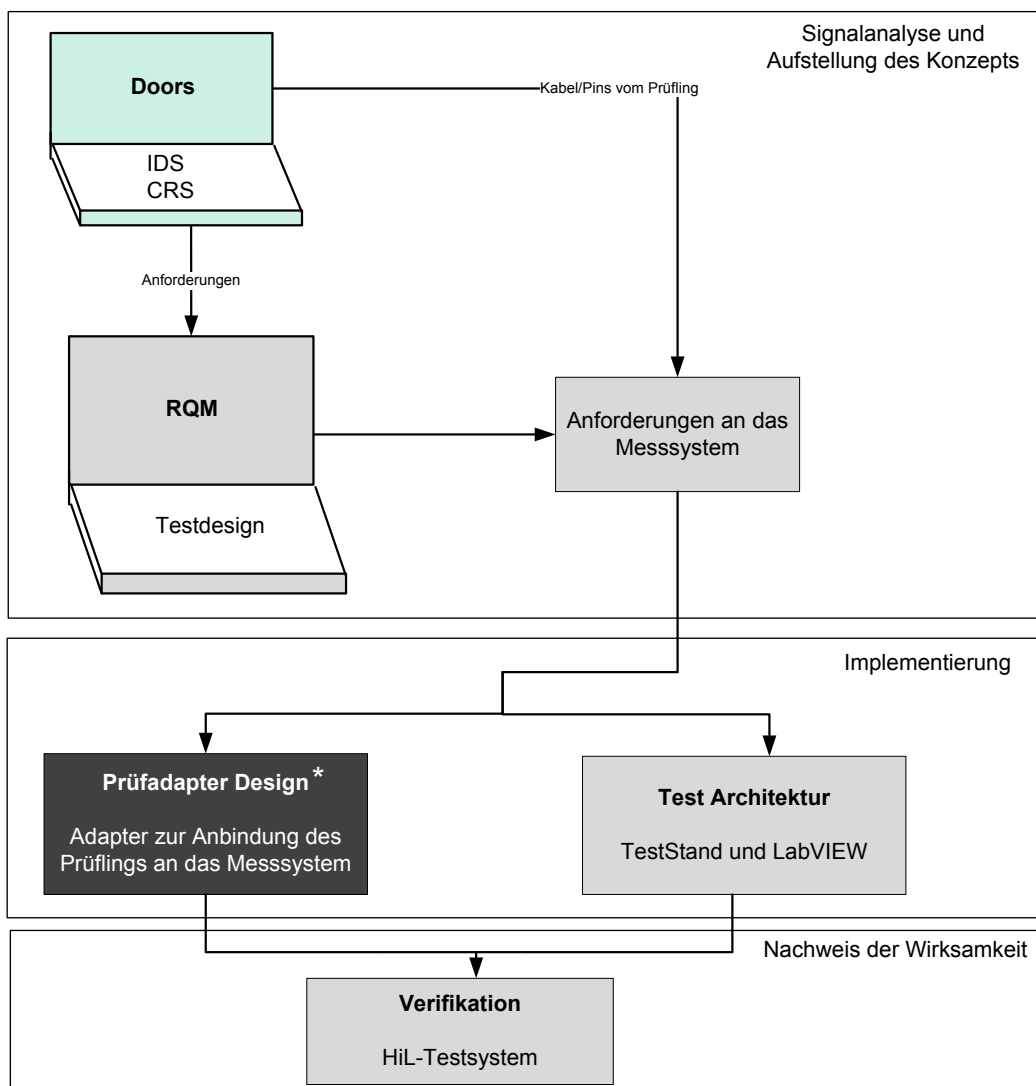


Abbildung 3.1: Ablaufplan der Methodik



Zuerst erfolgte eine Signalanalyse basierend auf den Informationen, die aus DOORS gewonnen werden können. Im Anschluss wurde das Konzept erstellt, indem Testfälle definiert und Anforderungen an das Messsystem ermittelt wurden. In der Phase der Implementierung wurde von B. Braun ein Adapter für die Anbindung des Prüflings an das Messsystem entwickelt. Außerdem kam es in der Phase der Implementierung zum Aufbau der Test Architektur in TestStand und LabVIEW. Im Anschluss an Entwurf und Entwicklung kam es zum Nachweis der Wirksamkeit der oben genannten Implementierung. Dazu wurde eine Verifikation des HiL-Testsystems durchgeführt.

Dieses Kapitel wird von zwei Testfällen begleitet, um das Vorgehen der Erstellung und Durchführung der Testfälle sowie den Aufbau der Verifikationsplattform exemplarisch darzustellen. Die vollständigen Testfälle wurden dem Betreuer zur Verfügung gestellt [30]. Es handelt sich dabei um *Testfall 99*, bei dem es um das Auslesen der FW-Version des MSF geht und um *Testfall 119*, bei dem die Touch-Position vom TCF an den TLC übermittelt werden soll.

## 3.1 Signalanalyse und Aufstellung des Konzepts für die automatische Verifikationsplattform

Um Testfälle definieren zu können, muss man alle Anforderungen an den Prüfling betrachten. Diese Anforderungen sind bei B. Braun im Anforderungsmanagementsystem DOORS verankert.

### 3.1.1 Filterung der Anforderungen in DOORS

DOORS umfasst die Anforderungen an die Dialysemaschine und deren Komponenten. Für diese Master-These sind nur die Anforderungen relevant, die die Komponente MIB betreffen und im Rahmen des Interfacetests funktionell testbar sind. Da DOORS mehrere 10.000 Anforderungen enthält, wurden die funktionell testbaren Anforderungen der IDS und der CRS mit dem Attribut *Testable* versehen, um im Anschluss nach diesem Attribut zu filtern und so eine eingeschränkte Sicht (*MIB-Verification*) auf die relevanten Anforderungen zu haben. Diese Anforderungen werden im Anschluss über Anforderungslinks mit Testfällen im RQM verknüpft. Diese Filterung hat den Vorteil, dass zudem in der Zukunft Coverage-Analysen selektiv für jede Anforderung und die Testfälle des MIBs durchgeführt werden können.

### **3.1.1.1 Anforderungen aus der IDS**

In der IDS sind die Schnittstellenanforderungen enthalten. Diese lassen sich weiter unterteilen in Anforderungen an die Physikalischen-Interfaces und die Anforderungen an die Software-Software-Interfaces. Da die IDS sehr umfangreich ist, wurden die funktionell testbaren Anforderungen, die aus der IDS gefiltert worden sind, in einer Tabelle zusammengefasst (siehe Tabelle 3.1).

Tabelle 3.1: Zusammenfassung der Anforderungen aus der IDS

Datenbank	ID	Kommentar	Anforderungstyp	
IDS	3157	MCF-TLC Datenübertragungsparameter (RS232)	Software-Software-Interfaces	
	3165	MCF Funktionscode		
	3171	MCF-TLC Modbus-Protokoll Format		
	3184	MCF Hardware Variant Identifier		
	3186	MCF MCF Hardware Version		
	3188	MCF Firmware Version		
	3190	MCF Hardware Version		
	3192	MCF Controller Status		
	3196	MCF Hard Keys Status		
	3198	MCF Power-ON-OFF Hard Key Status		
	3200	MCF Detailed Firmware Version Information		MCF-TLC
	3204	MCF Reset Control		
	3210	MCF Hardkeys LED Control		
	3212	MCF Loudspeaker Amplifier Sound Level		
	3214	MCF OSD RGB Control		
	3216	MCF OSD Brightness Control		
	3219	MCF OSD Test Sequence Control		
	3222	MCF OSD Alarm Environment Configuration Control		
	3225	MCF FPD Backlight Brightness Control		
	3227	MCF OSD RGB Individual Test Control		
	3230	MCF Loudspeaker Amplifier Mute		
	3241	MCF Power-ON Default Einstellungen		
	3282	MSF Firmware Version Format	Software-Software-Interfaces	
	3286	MSF-LLS Datenübertragungsparameter (RS232)		
	3288	MSF-LLS Kommunikationsprotokoll		
	3295	MSF Receive Register 0 (rote OSD LEDs, Re-Transmissions Funktion)		
	3298	MSF Receive Register 1 (Mute Status, AQ Hardkey LED)		
	3304	MSF Transmit Register 0 (Hardkey Status)		
	3307	MSF Transmit Register 1 (OSD LED Status)		
	3310	MSF Transmit Register 2 (Firmware Version)		
	3313	MSF Transmit Register 3 (Lautsprecher Muting- und Output Auslesen)		
	3333	MSF Power-ON Default Einstellungen		
	3369	TCF Firmware Versions Format	Software-Software-Interfaces	
	3375	TCF-TLC Datenübertragungsparameter (RS232)		
	3381	TCF zu TLC Protokoll (Touch-Daten)		
	3391	TLC zu TCF Protokoll Format		TCF-TLC
3393	TCF zu TLC Protokoll Format (Firmware, Reset)			
3396	TCF zu TLC Datenrate			
2480	Elektrische Signale: BLI	Physikalische-Interfaces		
2479	Elektrische Signale: OSD			
2472	Elektrische Signale: Audiosignal			
2473	Elektrische Signale: Lautsprecher			
2476	Elektrische Signale: Kommunikation zwischen TLC und MCF			
2478	Elektrische Signale: UIB			
2475	Elektrische Signale: Kommunikation zwischen TLC und TCF			
2470	Elektrische Signale: Power Out			
2474	Elektrische Signale: Touchpanel			
2471	Elektrische Signale: Power Supply Control			
2470	Elektrische Signale: Power In			
2477	Elektrische Signale: Kommunikation zwischen LLS und MSF			

Genauere Informationen zu den einzelnen Anforderungen würden an dieser Stelle über den Rahmen der Arbeit hinausgehen und sind daher im Anhang in Kapitel 8.1.1 beigefügt.

### Signalanalyse/Physikalische-Interfaces

In der Untergruppe Physikalische-Interfaces sind die Anforderungen für die elektrische Verdrahtung des MIBs enthalten. Jeder einzelne Stecker und jeder Pin wird darin aufgelistet und spezifiziert. Diese Informationen über alle Ein- und Ausgangssignale des MIBs wurden für die Signalanalyse herangezogen, da nur so eine geeignete Prüfumgebung erstellt werden kann.

Im Kapitel 8.1.2 des Anhangs befinden sich die aus der IDS gewonnenen Informationen über Ein- und Ausgangssignale in Form von Tabellen (siehe Tabelle 8.15, 8.16, 8.17, 8.18, 8.19, 8.20, 8.21, 8.22, 8.24). Die letzte Spalte in jeder dieser Tabellen beinhaltet jeweils die ausgearbeiteten Anforderungen an die Messhardware. Über diese Spalte wird ausgehend von dem Signal des jeweiligen Pins bestimmt, welche Anforderungen sich dadurch für das Messsystem ergeben. Eine Zusammenfassung der Signalanalyse und den daraus abgeleiteten Anforderungen an das Messsystem kann Tabelle 3.2 entnommen werden.

Tabelle 3.2: Anforderungen an die Messhardware abgeleitet aus der Signalanalyse

[A]nalog/ [D]igital [P]ower	[I]nput/ [O]utput	Anzahl	Kommentar
A	I	17	n/a
A	O	3	n/a
D	I	12	n/a
D	O	10	n/a
P	O	4	n/a
n/a	n/a	1	USB- Schnittstelle
n/a	n/a	2	RS232- Schnittstellen

- *Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 1:*  
Die IDS-2477 (siehe Tabelle 3.3) enthält die Anforderungen an den Anschluss "P17 COM LLS", welcher zur Kommunikation zwischen LLS und MSF dient. Zusätzlich kann über Pin 2 ein externer Reset aller Mikrocontroller des MIBs ausgelöst werden. Die letzte Spalte der Tabelle enthält die "Anforderungen an die Messhardware", neben einer RS232 Schnittstelle ist ein digitaler Ausgang für den externen Reset notwendig und eine Spannungsversorgung für den MSF.

Tabelle 3.3: Signalanalyse: Kommunikation zwischen LLS und MSF - IDS-2477

Ziel	Pin	Signal Name	Beschreibung	[I]nput/ [O]utput	[A]nalog / [D]igital / [P]ower	Low- Level [V]	High- Level [V]	Logical Level	Anforderung an die Messhard- ware
P17 COM LLS	1	5V LLS	+5V DC	I	P	n/a	n/a	n/a	Power O
	2	Reset-S	Reset from LLS	I	D	0V	5V	0: LLS Reset	DO
	3	TXD-S	Transmit Data from MIB to LLS	O	D	-12V	+12V	RS232	RS232
	4	RXD-S	Receive Data from LLS to MIB	O	D	-12V	+12V	RS232	RS232
	5	DGND	Digital Ground	O	P	n/a	n/a	n/a	-
	6	DGND	Digital Ground and Shielding	O	P	n/a	n/a	n/a	-

- *Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 1:*

Tabelle 3.4 zeigt die Spezifikation von dem Anschluss "P15 Touch Panel" am MIB. Der letzten Spalte der Tabelle ("Anforderungen an die Messhardware") kann entnommen werden, welche Pins für die Messhardware als analoge oder digitale Ein- oder Ausgänge definiert werden. Die Pins X+, Y+, 5 Wire Sense, Y- und X- entsprechen den fünf Leitungen des 5-Draht resistiven Touchpanels.

Tabelle 3.4: Signalanalyse: Touchpanel - IDS-2474

Ziel	Pin	Signal Name	Beschreibung	[I]nput/ [O]utput	[A]nalog / [D]igital / [P]ower	Low- Level [V]	High- Level [V]	Logical Level	Anforderung an die Messhard- ware
P15 Touch Panel	1	X+	top right	I/O	A	n/a	n/a	n/a	DI
	2	Y+	bottom right	I/O	A	n/a	n/a	n/a	DI
	3	5 Wire Sense	5 Wire analog sen- sing	I	A	n/a	n/a	n/a	AO
	4	Y-	top left	I/O	A	n/a	n/a	n/a	DI
	5	X-	bottom left	I/O	A	n/a	n/a	n/a	DI, DO

Analog zu diesem Anschluss wurden auch alle anderen Anschlüsse des MIBs analysiert.

### Software-Software-Interfaces

Neben den elektrisch-physikalischen Anforderungen gibt es zusätzlich Software-Software Anforderungen. Diese Anforderungen, welche in der Untergruppe der Software-Software-Interfaces definiert sind, bestimmen die Kommunikation der Mikroprozessorsysteme der NDS (LLS, TLC) mit dem Mikroprozessorsystem des MIBs (MCF, MSF, TCF).

Eine Einteilung in drei Gruppen ist hier sinnvoll:

- MCF - TLC
- MSF - LLS
- TCF - TLC

Die Kapitel 2.6.5 bis 2.6.7 beschreiben detailliert die Kommunikation zwischen Mikroprozessorsystemen.

- *Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 2:*  
Zusätzlich zu den Anforderungen an den Anschluss gibt es Software-Software-Interface-Anforderungen, die zur Erstellung dieses Testfalls berücksichtigt wurden. Dazu zählt:
  - *IDS-3286:*  
Beinhaltet Parameter der RS232 Datenübertragung (8 Datenbits, 1 Stopbit, gerade Parität, Baudrate 19200 Bits/s, beginnt mit dem niederwertigsten (LSB) Bit) (siehe auch Kapitel 2.6.6)
  - *IDS-3288:*  
Beinhaltet Informationen über das LLS-MIB Kommunikationsprotokoll (siehe Tabelle 2.5). Aufbau des MIB-LLS Kommunikationsframes: 2 Nachrichten mit jeweils 8 Nutzdatenbits (8 Databits, CRC8), Parität, Start- und Stop-Bit.
  - *IDS-3310:*  
Beschreibt den Aufbau des Transmit Registers 2, über welches die MSF-Firmware-Version (Bit 6-0) ausgelesen werden kann (siehe Tabelle 2.10)
  - *IDS-3282:*  
Beinhaltet Informationen über das Firmware Version Format (siehe Tabelle 3.5)

*Tabelle 3.5: IDS-3282: MIB-LLS Firmware Version Format*

<b>MIB-Touch Firmware Version</b>			
Reiceive Error	FW release state	Major Version	Minor Version
<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5-3</b>	<b>Bit 2-0</b>
0 = no error 1 = error	0: development version 1: released for production	(000 <sub>b</sub> -111 <sub>b</sub> )	(000 <sub>b</sub> -111 <sub>b</sub> )

- *Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 2:*  
Neben den Anforderungen an die elektrischen Signale des Touchpanels sind für die Erstellung eines Testfalls auch die Software-Software-Interface-Anforderungen aus

der IDS zu berücksichtigen. Aus der Anforderung IDS-3381 kann entnommen werden, dass Touch-Positionen nach dem Touch-Data-Protokoll in Tabelle 3.6 übermittelt werden müssen.

Tabelle 3.6: IDS-3381: MIB-Touch zu TLC: Touch-Data-Protokoll

TSHARC RS232 Protokoll	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
# Bits	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Touch-Position Datenbyte 0	1	P	X11	X10	X9	Y11	Y10	Y9
Touch-Position Datenbyte 1	0	X8	X7	X6	X5	X4	X3	X2
Touch-Position Datenbyte 2	0	Y8	Y7	Y6	Y5	Y4	Y3	Y2
Touch-Position Datenbyte 3	0	0	0	0	X1	X0	Y1	Y0

Die X- und Y-Touch-Positionen werden als 12 Bit Werte über 4 Datenbytes verteilt zyklisch vom TCF zum TLC gesendet. Zudem wird der Status (gedrückt / nicht gedrückt) im Pen up/down Bit kodiert.

### 3.1.1.2 Anforderungen aus der CRS

Zusätzlich zu den Interface-Anforderungen (elektronisch/physikalisch, Software-Software) der IDS besitzt die Komponente MIB, eigene Komponentenanforderungen, die in separaten DOORS Modulen gelistet sind. Die CRS zur Komponente MIB teilt sich in vier Subgruppen:

- CRS-MIBHW
- CRS-MCF
- CRS-MSF
- CRS-TCF

Die funktionell testbaren Anforderungen der CRS sind in Tabelle 3.7 zusammengefasst. Der genaue Inhalt der einzelnen Anforderungen befindet sich in den Tabellen 8.27 bis 8.30 im Anhang in Kapitel 8.2.

Tabelle 3.7: Zusammenfassung der Anforderungen aus der CRS

Datenbank	Anzahl	IDs
CRS-MIBHW	13	3461, 3562, 3604, 3605, 3606, 3607, 3475, 3564, 3563, 3565, 3603, 3608, 3609
CRS-MCF	19	501, 647, 712, 714, 713, 653, 717, 718, 715, 720, 716, 719, 682, 683, 711, 687, 752, 753, 759
CRS-MSF	9	355, 366, 367, 368, 404, 369, 442, 443, 449
CRS-TCF	4	298, 313, 317, 377

Die erste Gruppe CRS-MIBHW definiert, in welcher Weise die Hardware des MIBs (MIBHW) realisiert werden muss. In Abhängigkeit der Mikrocontroller werden Funktionen aufgelistet, die die MIBHW ermöglichen muss. Zum Beispiel muss die MIBHW dem TCF ermöglichen, eine Touch-Position zu bestimmen oder dem MCF ermöglichen, Alarme mit mittlerer Priorität in Gelb über die OSD-Hardware (OSDHW) anzuzeigen.

Die zweite Subgruppe, CRS-MCF, definiert die Anforderungen an den auf der MIBHW realisierten MCF. Neben den Funktionen, dass der TLC über den MCF die interne Firmware-Versionsnummer ermitteln können muss oder die Steuerung der Tastenhintergrundbeleuchtung des UIBs ermöglichen muss, werden hier Anforderungen an die Steuerung der OSDs und viele andere Anforderungen aufgelistet.

Die dritte Gruppe umfasst die Anforderungen an den MSF, der hauptsächlich überwachende Funktionen übernimmt. Unter anderem soll der MSF dem LLS ermöglichen, die Hardkey-Statusinformationen auszulesen, die OSD-Statusinformationen auszulesen oder zum Beispiel auch Alarme mit hoher Priorität über die roten OSD LEDs anzuzeigen.

Die letzte Untergruppe nennt sich CRS-TCF. In dieser Gruppe befinden sich jene Funktionen, die der TCF gegenüber dem TLC ermöglichen muss. Dazu gehört zum Beispiel die Funktion, dass der TLC vom TCF eine Softwareversionsnummer bestimmen können muss oder, dass an den TLC eine Touch-Position gesendet werden können muss, um die grafische Benutzeroberfläche der NDS Dialysemaschine bedienen zu können.

- *Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 3:*  
Nach der inhaltlichen IDS-Filterung wird die CRS auf weitere Anforderungen gefiltert, die für funktionelle Prüfungen herangezogen werden können. In diesem Beispiel wurde die *CRS-MSF-355* identifiziert, die fordert, dass der MSF dem LLS die Ermittlung der internen Firmware-Versionsnummer ermöglichen muss.
  
- *Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 3:*  
Nachdem alle Spezifikationen, welche für den Touch relevant sind aus der IDS extrahiert wurden, wurde die CRS auf weitere Anforderungen, welche den Touch betreffen, untersucht. Hierbei sind drei Anforderungen von Bedeutung:
  - *CRS-MIBHW 3461:*  
MIBHW muss dem TCF ermöglichen die Touch-Position zu ermitteln
  - *CRS-MIBHW 3562:*  
MIBHW muss dem TCF ermöglichen die Touch-Position an den TLC zu senden



- *CRS-TCF-313*:

TCF muss dem TLC die Bestimmung der Touch-Position ermöglichen

### 3.1.2 Erstellung des Testplans im RQM

Mithilfe der in DOORS erfassten Anforderungen (IDS, CRS) kann ein Testplan (TP-MIB) in RQM entwickelt werden. Die Vorgehensweise zur Erarbeitung des Testplans ist in Abbildung 3.2 schematisch dargestellt.

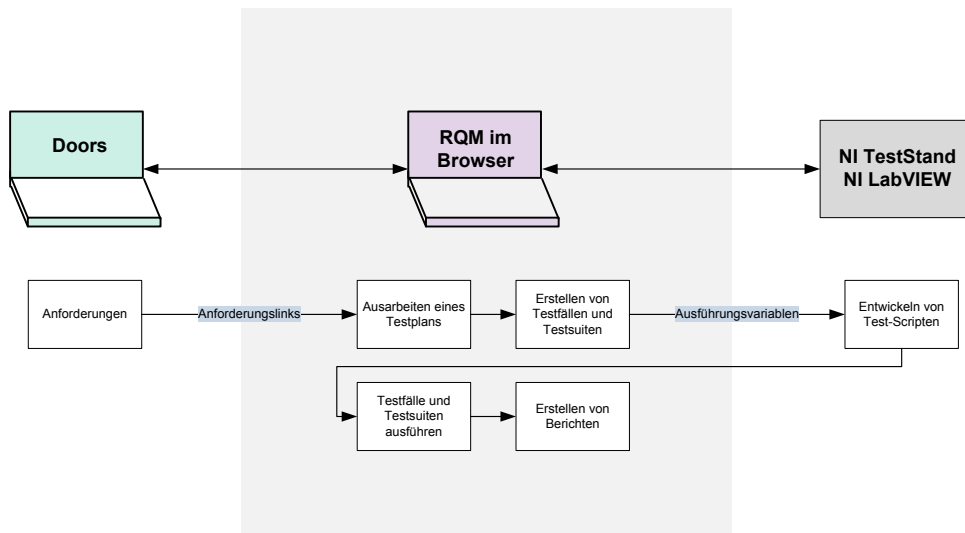


Abbildung 3.2: Schematischer Aufbau des SW-Frameworks bestehend aus IBM DOORS, IBM RQM, NI TestStand und NI LabVIEW. Nachdem die Anforderungen aus Doors extrahiert wurden, kann in RQM (hellgrau hinterlegt) das Testdesign entwickelt werden.

Die Anforderungen aus DOORS bilden das Grundgerüst, welches notwendig ist, um den Testplan im RQM-Tool aufzubauen. Der Testplan umfasst verschiedene Testfälle, die anhand der Anforderungen entwickelt werden. Die wesentlichen Features im RQM zur Definition des Gesamtdesigns eines Testfalls sind:

**Testfalldesign**, in dem einzelne Testschritte aufgelistet werden,

**Erwartete Ergebnisse**, worin Akzeptanzkriterien eines Tests definiert werden,

**Anforderungslinks**, welche die Verbindung zu DOORS darstellen und somit die Traceability in Richtung DOORS sicherstellen. Falls Anforderungen geändert oder gelöscht werden, können die zugehörigen Testfälle in RQM ermittelt werden und so schnell und einfach die Testpläne oder Testfälle aktualisiert werden.

Jeder einzelne Testfall wird, nachdem er fertig designet wurde, mit einem Test-Script verknüpft. Allgemein gibt es automatische und manuelle Test-Scripte, die Maßnahmen zum Implementieren eines Testfalls enthalten. Da das Ziel dieser Arbeit darin besteht, eine automatische Verifikationsplattform aufzubauen, wird ein automatisches Test-Script verwendet. Bevor automatische Test-Scripte jedoch verwendet werden können, ist es notwendig, einen Ausführungsadapter zu konfigurieren. Als Ausführungsadapter dient der NI TestStand Adapter, da mit NI TestStand Scripten, sogenannten Sequenzdateien, gearbeitet wird. Über eine Ausführungsvariable kann somit die lückenlose Tracability weiter zu den NI Softwarekomponenten sichergestellt werden.

Für eine bessere Übersicht über die Testfälle wurden die Testfälle, wie auch die Anforderungen, in Übergruppen, sogenannten Testsuiten, zusammengefasst (siehe Abbildung 3.3). Diese Testsuiten beinhalten die für den jeweiligen Mikrocontroller (MCF, MSF, TCF) relevanten Testfälle (TestSuite MCF-TLC, MSF-LLS, TCF-TLC). Auf die einzelnen Test-Suiten wird in den Kapiteln 3.1.2.1, 3.1.2.2 und 3.1.2.3 eingegangen.

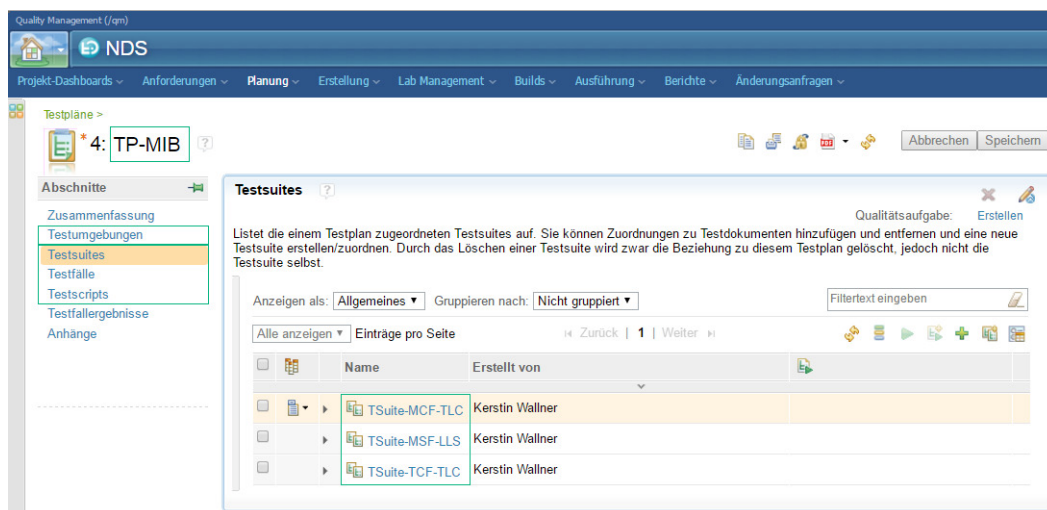


Abbildung 3.3: Screenshot aus RQM: Aufbau des Testplans. Der Testplan (TP-MIB) besteht aus drei Testsuiten, welche funktionell zusammengehörige Testfälle umfassen. Testfälle rufen Test-Scripte auf, die mithilfe eines Ausführungsadapter in einer Testumgebung durchgeführt werden.

#### 3.1.2.1 Testsuite MCF-TLC

Diese Testsuite enthält alle Testfälle, die mit dem MCF in funktionell-testbaren Zusammenhang stehen. Ein Ausschnitt der in RQM designeten Testfälle wird in Abbildung 3.4 dargestellt. Insgesamt umfasst diese Testsuite 81 Testfälle, auf die in den nachfolgenden Abschnitten eingegangen wird. Explizite Details zu jedem einzelnen Testfall wurden dem Betreuer zur Verfügung gestellt [30].

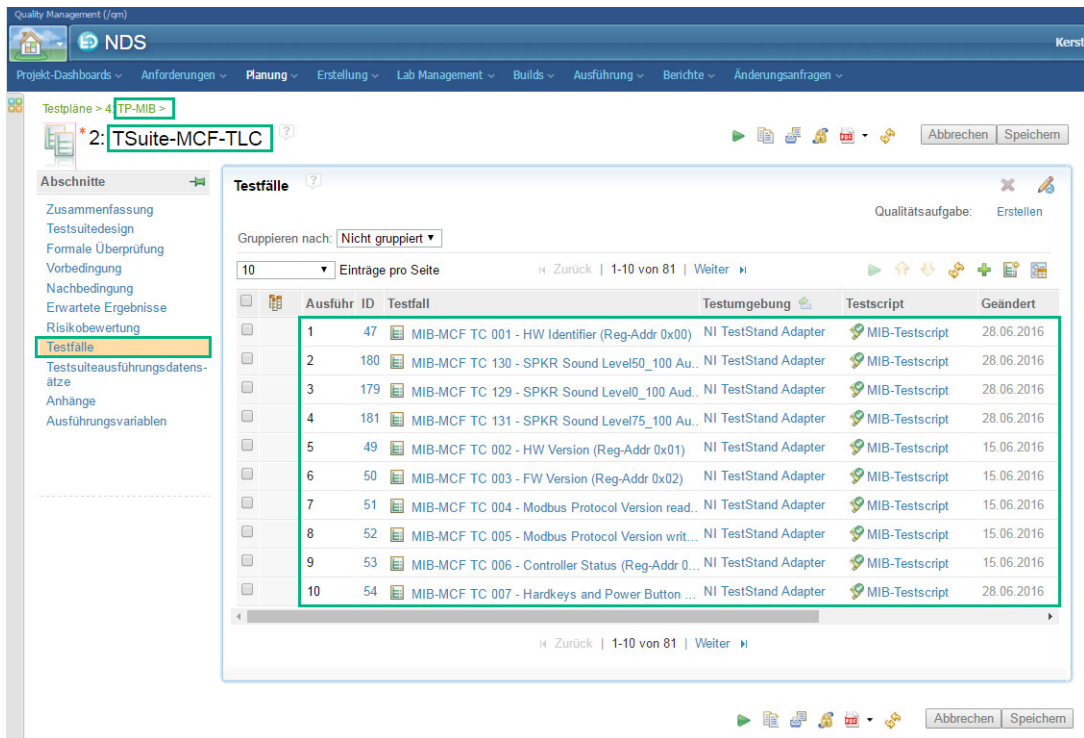


Abbildung 3.4: Screenshot aus RQM: Testfälle der Testsuite MCF-TLC

### Auslesen von voreingestellten Parameter des MCF

Über die ersten fünf Registeradressen (0x00 bis 0x04) können verschiedene Informationen ausgelesen werden. Um die Informationen des MCFs auszulesen, muss der entsprechende Befehl gesendet werden. Die Abbildung 3.5 zeigt, wie die Vorgehensweise zur Prüfung des Hardware Variant Identifiers aussieht. Alle Testfälle der TestSuite MCF-TLC sind im wesentlichen nach dieser Architektur aufgebaut. Man benötigt für den Test einen definierten Ausgangspunkt. Hierfür wird die Versorgungsspannung zunächst deaktiviert, um die angeschlossene Peripherie im nächsten Schritt zurücksetzen zu können. Zu Beginn des Tests wird die Versorgungsspannung aktiviert und ein Dummy-Lesezugriff ausgeführt. Dieser ist notwendig, um die Reset-Information im Funktionscode zu löschen. Im Anschluss erfolgt der Ablauf gemäß Abbildung 3.5.

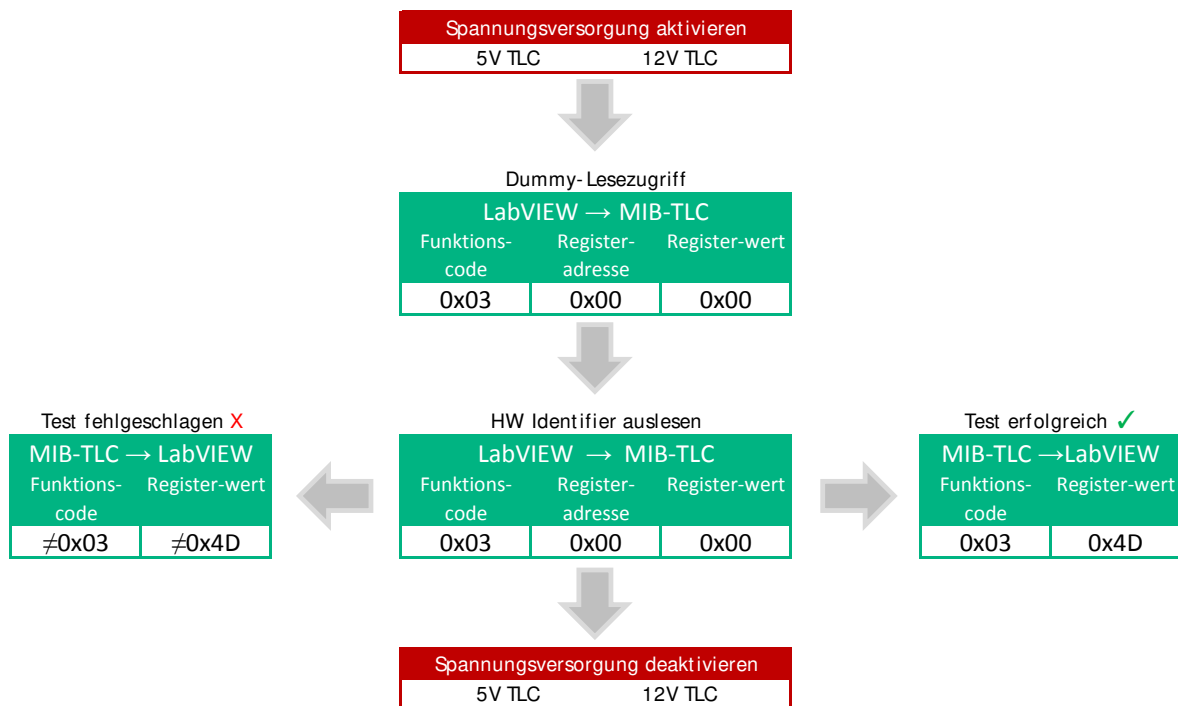


Abbildung 3.5: Testfall 1: Hardware Identifier auslesen

Analog dazu erfolgt das Auslesen der Informationen zur Hardware Version (Registeradresse: 0x01), Firmware Version (Registeradresse: 0x02), Modbus Protocol Version (Registeradresse: 0x03) und zum Controller Status (Registeradresse: 0x04).

### Überprüfung der Hardkeys

In den Testfällen, die zur Überprüfung der Status Informationen der Hardkeys dienen, wird ein Tastendruck simuliert und über den MCF-Mikrocontroller der Status aller Hardkeys ausgelesen. Diese Testfälle basieren auf der IDS-3196, in der definiert wird, wie die Statusinformationen der Hardkeys ausgelesen werden können. Zusätzlich beinhaltet die CRS-MIBHW-3475, CRS-MIBHW-3563 und CRS-MCF-647, Informationen darüber, dass es dem TLC ermöglicht werden muss, die Statusinformationen der Hardkeys zu ermitteln.

Da die am UIB realisierten Taster eine low-aktive Logik verwenden, wird ein Tastendruck durch das auf Masse ziehen der Leitung erreicht. Um für die simulierten Tastendrucke immer von einem definierten Ausgangspunkt auszugehen, werden zu Beginn alle Tasten als "nicht gedrückt" durch Setzen des Pegels auf +5V simuliert. Im Anschluss werden die ein-

zelen Tasten für die Tastensimulation auf Low-Pegel gesetzt und die Statusinformationen aus Register 0x05 ausgelesen. In Abbildung 3.6 wird der Testfall 7 dargestellt. In diesem Testfall wird keine der Tasten betätigt und der Status über den MCF-Mikrocontroller ausgelesen. Um zu untersuchen, ob sich bei einer Betätigung der Tasten der Status ändert, wurde für jede Taste ein weiterer Testfall erstellt. Falls in einem der Tests der Registerwert oder der Funktionscode einen Wert ungleich dem Erwartungswert liefert, schlägt der Test fehl.

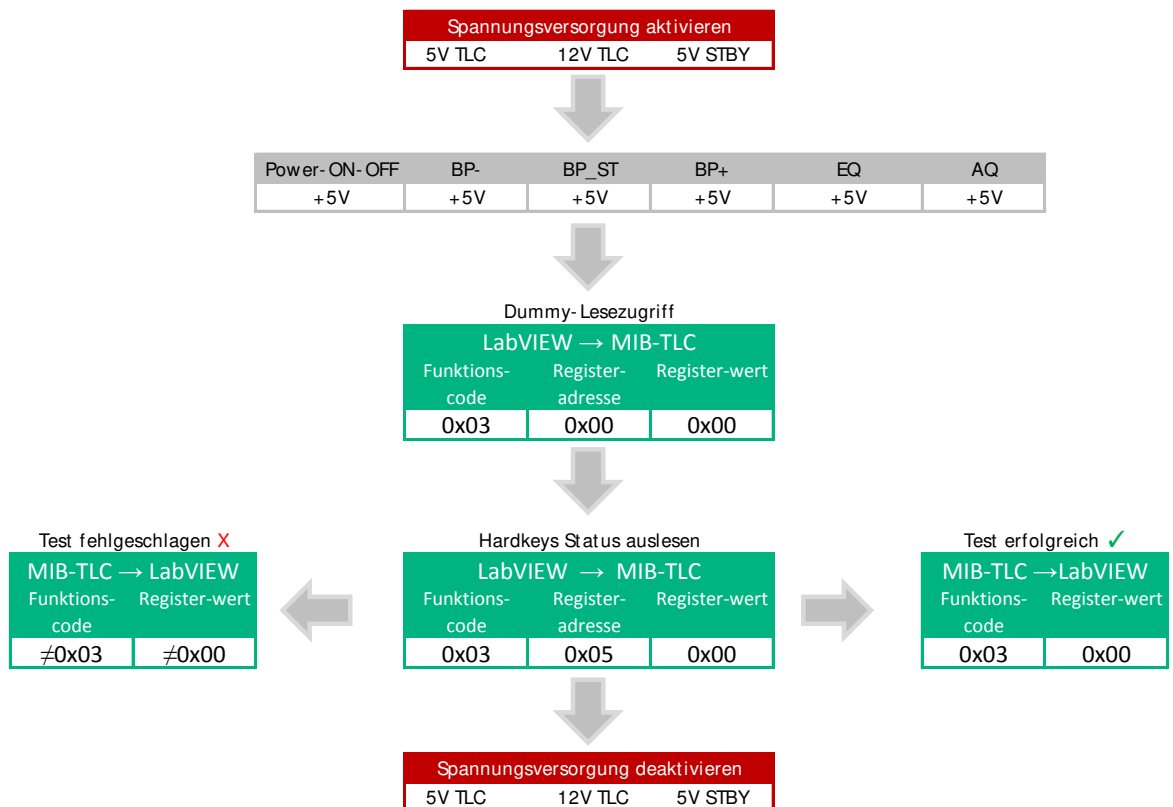


Abbildung 3.6: Testfall 7: Hardkeys Status auslesen (alle Hardkeys nicht gedrückt)

Neben den bisher genannten Registern (0x01 bis 0x05) gibt es noch zwei weitere reine Leseregister (Registeradresse 0x06 und 0x07). Bei Verwendung eines freigebenden Prüflings ist jeder Zugriff auf diese Register jedoch ungültig. Dies wird überprüft, indem ein Lese- und ein Schreibbefehl an den MCF-Mikrocontroller gesendet wird. In jedem Fall muss der Funktionscode hier 0x3X liefern.

### Überprüfung der Hardkey-LEDs

Diese Testfälle wurden entwickelt, um die Funktion der Hardkey-LEDs zu überprüfen. Diese Testfälle basieren auf der IDS-3210, in der definiert wird, wie die LEDs angesteuert werden können. Außerdem basieren diese auf der CRS-MCF-712, CRS-MCF-713 und CRS-MCF-714, welche beinhalten, dass es dem TLC ermöglicht werden muss, über den MCF die Tasten-Hintergrundbeleuchtung zu steuern. Lediglich die Blutpumpe (BP) START / STOP Taste, die AQ Taste und die EQ Taste werden durch eine LED beleuchtet. Die BP+ und die BP- Taste können nicht hinterleuchtet werden. Die Anode der LED liegt am UIB konstant an einem +5V Pegel und die Kathode kann über den MCF am MIB gesteuert werden. Wenn die Kathode der LEDs auf Masse gezogen wird, wird die Leuchtdiode aktiv und beginnt zu leuchten. Man sendet, je nachdem welche LED man aktivieren möchte, den entsprechenden Befehl an den MCF und misst die Pegel am Ausgang "P11 UIB", um zu überprüfen, ob der MCF die entsprechende Leuchtdiode aktiviert.

### Überprüfung des Audiosignals am Lautsprecher

Über das MIB werden die Audiosignale des TLCs verstärkt und über einen Lautsprecher ausgegeben. Dazu wurden folgende Anforderungen aus DOORS identifiziert:

- IDS-3212: Beinhaltet den Aufbau von Registeradresse 0x22. Über dieses Register kann die Lautstärke verändert werden.
- CRS-MIBHW-3609: Die MIBHW muss so realisiert sein, dass der MCF die Lautstärke verändern kann.
- CRS-MCF-716: TLC muss über den MCF Lautstärke und Mute-Funktion steuern können.
- CRS-MCF-687: Der Lautsprecher muss ein hörbares Signal ausgeben.
- IDS-3241: Beinhaltet die Power-ON Einstellungen des MCFs

Um den in der MIBHW integrierten Verstärker zu überprüfen, muss dem MIB am Anschluss "P5 Audio IN" ein definiertes Eingangssignal zur Verfügung gestellt werden. Der Test wird mit zwei unterschiedlichen Eingangssignalen durchgeführt. Die einzige Bedingung an das Signal, die aus der CRS hervorgeht ist, dass es hörbar sein muss. Nähere Informationen zur Signalcharakteristik beinhaltet die Design-Spezifikation, in der das Audiosignal wie folgt definiert ist:

- CD-MIBHW-2014: Das Eingangssignal des Verstärkers "Audio\_IN-Line-L und Audio\_IN-Line-R" ist definiert durch ein Sinussignal mit einer Spitzen-Spitzen-Spannung von 2.0 Vpp. Das Eingangssignal "Audio-IN-PC-Beep" ist eine Rechteckschwingung

mit einer Amplitude von 3.6 V<sub>pp</sub> and 50% Duty-Cycle<sup>5</sup>. Die Frequenz des Audiosignals ist variabel 200 Hz bis 5 kHz.

Basierend auf dieser Anforderung wurde eine Sinusschwingung mit einer Spitzen-Spitzen-Spannung von 2V und einer Frequenz von 440 Hz definiert sowie eine Rechteckschwingung mit einer Amplitude von 3.6V und ebenso einer Frequenz von 440 Hz. Nachdem man die Spannungsversorgung des MIBs aktiviert und ein Audiosignal am Anschluss "P5 Audio IN" angelegt hat, muss über das Register 0x29 die Mute-Funktion, welche standardmäßig nach dem Einschalten des Dialysegeräts aktiviert wird, deaktiviert werden, um am Ausgang "P6 SKPR" des MIBs die Spannung und Frequenz des Ausgangssignals zu messen.

Durch eine Veränderung der Lautstärke über das Register 0x22 kommt es zu einer Veränderung der Amplitude am Ausgang. Da keine Anforderungen über die Höhe der Amplitude in DOORS verankert sind, reicht ein relativer Vergleich der Amplituden bei unterschiedlichen Lautstärken. Die Amplituden bei unterschiedlicher Lautstärke (0%, 50% 75%) werden relativ zu der Amplitude bei 100% Lautstärke verglichen. Eine niedrige Lautstärke bedingt eine niedrigere Amplitude am Ausgang.

Zusätzlich zur Überprüfung der Amplituden wird die Frequenz überprüft. Wenn das MIB ohne einen Fehler verstärkt, dann sollte die am Eingang des MIB anliegende Frequenz auch am Ausgang auftreten. Als Akzeptanzkriterium wurde die angelegte Frequenz  $\pm 10$  Hz also 430 Hz bis 450 Hz definiert. Eine Spektralanalyse des verstärkten Signals wird nicht durchgeführt, daher bleiben die durch Oberschwingungen verursachten Verzerrungen während des Tests unberücksichtigt.

#### Überprüfung der OSDs

Für die Überprüfung der OSDs wurden folgende Anforderungen aus DOORS herausgefiltert:

- IDS-3214: Aufbau von Registeradresse 0x23. Über dieses Register kann die Farbe und die Blinkfunktion der OSDs gesteuert werden.
- IDS-3216: Aufbau von Registeradresse 0x24. Über dieses Register kann die Helligkeit der OSDs gesteuert werden.
- IDS-3219: Aufbau von Registeradresse 0x25. Mit diesem Register können einzelne LEDs deaktiviert werden um einen Fehler zu erzeugen.
- IDS-3222: Aufbau von Registeradresse 0x26. Über dieses Register kann das OSD-Power-ON-Verhalten eingestellt werden.

---

<sup>5</sup> Tastverhältnis

- IDS-3241: Power-ON Einstellungen des MCFs
- CRS-MCF-653: MCF muss dem TLC die Steuerung der OSD Anzeige (OSD Farbe, OSD Frequenz, OSD Helligkeit) ermöglichen
- CRS-MCF-682, CRS-MCF-683, CRS-MCF-711: Zuordnung der Alarmprioritäten und der OSD-Blinkfrequenzen zu den OSD LEDs
- CRS-MIBHW-3607: MIBHW muss dem MCF ermöglichen den Gerätestatus "Betriebsbereit" mit grünen LEDs auf der OSDHW anzuzeigen
- CRS-MIBHW-3606: MIBHW muss dem MCF ermöglichen den Gerätestatus "Alarme mit niedriger Priorität" mit gelben LEDs auf der OSDHW anzuzeigen
- CRS-MIBHW-3604: MIBHW muss dem MCF ermöglichen den Gerätestatus Alarme mit hoher Priorität mit roten LEDs auf der OSDHW anzuzeigen
- CRS-MCF-717: MCF muss dem TLC ein einstellbares OSD-Power-ON-Verhalten ermöglichen
- CRS-MCF-718 MCF muss den OSD-Status beim Einschalten auf Alarm mit mittlerer Priorität mit Signalanzeige gelb setzen

Basierend auf dieser Liste von Anforderungen wurden die Testfälle 30 bis 50 realisiert. Hauptsächlich geht es darum, dass die OSDs über den MCF ansteuerbar sein müssen. Um zu überprüfen, ob die OSDs nach einer Aktivierung durch den MCF auch wirklich aktiviert worden sind, ist es notwendig, eine Spannungsmessung am Anschluss P2, P3, P4 und P10 durchzuführen. Zur Bewertung, ob die gemessene Spannung an den OSDs dem jeweiligen Zustand entspricht, wurde eine Referenzspannungsmessung, an einem vom Entwickler funktionstüchtig bewerteten MIB durchgeführt. Das Ergebnis dieser Messung befindet sich in Tabelle 3.8.

Tabelle 3.8: Akzeptanzkriterien der OSD LED Spannungen

OSD LEDs	LED ist an	LED ist aus
Rot	$2.3V < U < 3.5V$	$4.0V < U < 5.2V$
Grün	$1.2V < U < 2.7V$	$4.0V < U < 5.2V$
Blau	$1.2V < U < 2.4V$	$4.0V < U < 5.2V$

Neben der Überprüfung, ob eine bestimmte Farbe durch den MCF aktiviert werden kann, soll sichergestellt werden, dass es dem MCF auch möglich ist, schnelles und langsames Blinken der OSDs zu aktivieren. Dazu ist eine Frequenzmessung an den OSDs notwendig. Die Blinkfrequenz soll bei mittlerer Alarm Priorität 0.6 Hz und bei hoher Alarmpriorität 2.4 Hz entsprechen. Als Akzeptanzkriterium wird ein Toleranzbereich basierend auf der IEC 60601-1-8 §6.3.2.2 Tabelle 2 [17] definiert.



Tabelle 3.9: Akzeptanzkriterien der OSD Blinkfrequenz

	Blinkfrequenz
schnell blinkend	$1.4\text{Hz} < f < 2.8\text{Hz}$
lanksam blinkend	$0.4\text{Hz} < f < 0.8\text{Hz}$

Des weiteren wird überprüft, ob die Helligkeit der grünen OSD variabel ist. Prinzipiell ist es möglich, die Helligkeit aller OSDs einzustellen. In Realität wird aber nur die grüne OSD gedimmt. Diese Dimmfunktion wurde realisiert, um die Nachtdialyse für den Patienten angenehmer zu gestalten. Alarme sollen hingegen auch während der Nachtdialyse in maximaler Helligkeit signalisiert werden. Überprüft wird die Dimmbarkeit der grünen OSDs über einen relativen Vergleich der Spannungsamplituden bei unterschiedlichen Helligkeiten der grünen OSDs. Bei maximaler Helligkeit ist die gemessene Spannung minimal bei minimaler Helligkeit ist die gemessene Spannung maximal. In jedem Fall befindet sich die Spannung innerhalb der in Tabelle 3.8 aufgezeigten Grenzen.

### Überprüfung der Hintergrundbeleuchtung

Die Hintergrundbeleuchtung des Monitors der NDS kann über das MIB gesteuert werden. Folgende fünf Anforderungen aus DOORS stehen mit der Hintergrundbeleuchtung in Zusammenhang.

- IDS-3225: Aufbau von Register 0x27. Über dieses Register kann die Hintergrundbeleuchtung des Monitors eingestellt werden.
- IDS-3241: Power-ON Einstellungen des MCFs
- CRS-MIBHW-3608: MIBHW muss dem MCF ermöglichen, die Helligkeit der Hintergrundbeleuchtung einzustellen
- CRS-MCF-715: MFC muss dem TLC die Steuerung der Hintergrundbeleuchtung des Displays ermöglichen
- CRS-MCF-720: MCF muss die Helligkeit beim Einschalten auf 100% setzen

Um zu überprüfen, ob das MIB die Helligkeit verändert, muss über das Register 0x27 der entsprechende Befehl gesendet werden. Nachdem der Befehl gesendet wurde, gibt das MIB ein pulsbreitenmoduliertes Signal mit einer Grundfrequenz von 288Hz am Anschluss "P1 BLI" Pin 4 aus. Je nach Tastgrad des pulsbreitenmodulierten Signals (PWM), verändert sich die Helligkeit. Die Helligkeit lässt sich von 0% bis 100% einstellen, wobei 0% der minimalen und 100% der maximalen TFT-Helligkeit entspricht. Hierbei entspricht 0% Helligkeit einer minimalen PWM von 20%.

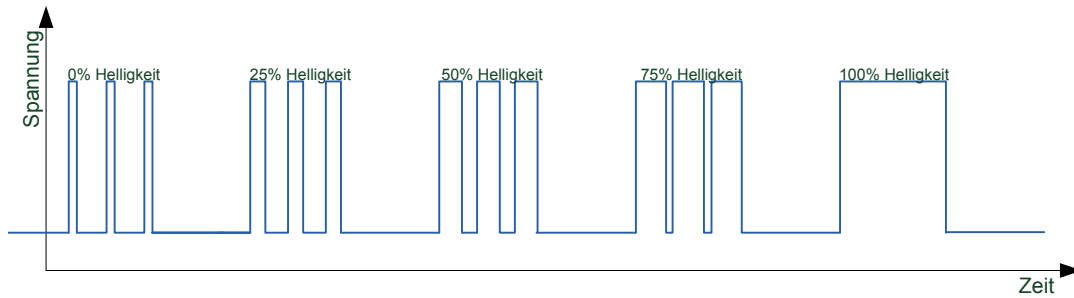


Abbildung 3.7: Steuerung der Hintergrundbeleuchtung. Über den Tastgrad eines pulsbreitenmodulierten Signals wird die Helligkeit des Monitors der NDS eingestellt.

Getestet wird das Signal bei einer Helligkeit von 0, 25, 50, 75 und 100 Prozent. Nachdem die jeweilige Helligkeit mit dem MCF eingestellt worden ist, wird die Einschaltzeit des pulsbreitenmodulierten Signals bestimmt und mit den in Tabelle 3.10 erfassten Referenzwerten verglichen.

Tabelle 3.10: Akzeptanzkriterien der Einschaltzeit ( $t_{Ein}$ ) des pulsbreitenmodulierten Signals bei unterschiedlicher Helligkeit

Helligkeit in %	Toleranzbereich Einschaltzeit in ms
0	$0.687 < t_{Ein} < 0.701$
25	$1.376 < t_{Ein} < 1.404$
50	$2.059 < t_{Ein} < 2.101$
75	$2.752 < t_{Ein} < 2.808$

Da bei einer Helligkeit von 100% eine konstante Gleichspannung anliegt, wurde in diesem Testfall nicht die Einschaltzeit, sondern die Amplitude des pulsbreitenmodulierten Signals gemessen und mit dem Referenzwert verglichen. Der Referenzwert kann der IDS-2480 entnommen werden. In der IDS-2480 steht, dass das pulsbreitenmodulierte Signal ein High-Level von 3.3V besitzt. Darum wird ein Toleranzbereich von 3.1V bis 3.5V als Akzeptanzkriterium festgelegt.

### Überprüfung des MCF-Verhaltens bei Abweichung der Spannungsversorgung von der Nennspannung

Üblicherweise werden die zur Testsuite MCF-TLC gehörenden Testfälle mit einer Nennspannung von +12V (Signalname: 12VTLC) und +5.2V (Signalname: 5VTLC) über den Anschluss "P16 PWR\_IN" versorgt. Da die Spannungen auch einen gewissen Toleranzbereich besitzen, muss überprüft werden, ob bei einer vom Nennwert abweichenden Spannung (siehe Tabelle 3.11) der MCF noch bestimmungsgemäß funktioniert.

Zusätzlich gibt es noch eine vom Netzteil kommende Spannung (5V\_STBY), die über den Anschluss "P14 PWR\_SUP\_CTRL" eingespeist wird und für das Auslesen der Statusinformation der POWER ON/OFF Taste notwendig ist. Die Überprüfung wird über das Steuern der OSDs und das Auslesen der Statusinformation der Hardkeys am UIB durchgeführt.

Tabelle 3.11: Toleranzbereich der Spannungsversorgung 5VTLC, 12VTLC und 5V\_STBY

Signal	Nennspannung	Toleranzbereich
5VTLC	5.2V	$4.75V < 5VTLC < 5.25V$ (5V±5%)
12VTLC	12V	$11.4V < 12VTLC < 12.6V$ (12V±5%)
5V_STBY	5.2V	$4.75V < 5V\_STBY < 5.25V$ (5V±5%)

### Überprüfung anhand des Funktionscodes

In den Testfällen mit der ID 67 bis 74 werden bewusst Protokollfehler vom TLC zum MCF gesendet, um zu überprüfen, ob der MCF den Fehler erkennt und über den Funktionscode zurückgibt. Genaue Informationen zum Funktionscode können dem Kapitel 2.6.5.1 entnommen werden.

Folgende Fehler werden überprüft:

- Funktionscode-Fehler:  
Fehler kann hervorgerufen werden, indem man in ein Register, welches ausschließlich ausgelesen werden kann, schreiben möchte.
- Registerwert-Fehler:  
Fehler wird hervorgerufen, wenn man einen ungültigen Registerwert verwendet, wie zum Beispiel eine Lautstärke oder Helligkeit über 100%.
- Registeradressen-Fehler:  
Fehler wird hervorgerufen, indem eine Registeradresse angesprochen wird, die nicht unterstützt wird. Das kann zum Beispiel auch eine Registeradresse sein, die für interne Entwicklungszwecke genutzt werden kann, aber in einer freigegebenen Version nicht zugänglich ist.
- CRC-Fehler:  
Fehler tritt bei einer fehlerhaften CRC auf.
- Übertragungsfehler:  
Fehler tritt auf, wenn man eine falsche SOM oder EOM sendet.

Zusätzlich zu den oben genannten Fehlern gibt es eine Anforderung aus der CRS (CRS-MCF-759), aus der hervorgeht, dass eine Unterspannungserkennung implementiert sein

mus. Wenn der Fall eintreten sollte, dass die Versorgungsspannung (5VTLC) zu niedrig wird, soll der Mikroprozessor solange im definierten Reset bleiben, bis die Spannung wieder die Nennspannung erreicht hat. Die Information, ob ein Reset nach einem Spannungseinbruch erfolgt ist, wird in diesem Testfall über den Funktionscode überprüft.

### 3.1.2.2 Testsuite MSF-LLS

Der LLS überwacht die Aktorik und Sensorik der NDS Dialysemaschine. In der Testsuite MSF-LLS befinden sich 40 Testfälle, die in funktionellem Zusammenhang mit dem LLS und dem MSF stehen. Neben seiner überwachenden Tätigkeit hat er zusätzlich die Fähigkeit, unabhängig vom TLC die Steuerung der roten OSDs, der AQ-LED und der Mute-Funktion zu übernehmen.

#### Überprüfung über das Auslesen von Registern

Die Überprüfung von Registern bildet das Fundament dieser Testsuite. Der Befehl muss entsprechend dem Kommunikationsprotokoll (siehe Kapitel 2.6.6) gesendet werden und im Anschluss muss die vom MIB erhaltene Antwort ausgewertet werden. Anhand von Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) wird gezeigt, wie eine derartige Überprüfung stattfindet und wie die aus DOORS gefilterten Anforderungen in RQM verlinkt werden können.

- *Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 4:*  
Basierend auf den Anforderungen, welche in Schritt 1, 2 und 3 identifiziert wurden, kann ein Testfall entwickelt werden. In diesem Beispiel soll geprüft werden, ob es dem LLS möglich ist, die Firmware-Versionsnummer auszulesen und ob es sich um die vom Entwickler voreingestellten Parameter handelt. Dazu wurde der Test wie in Abbildung 3.8 aufgebaut. Zuerst wird die Spannungsversorgung (Anschluss P16, P17) aktiviert. Danach wird der Befehl 0x80 0x2F an den MSF gesendet, der dazu dient die Firmware-Versionsnummer auszulesen. In Abhängigkeit von der Antwort des MSF, wird der Test als bestanden oder fehlgeschlagen beurteilt.

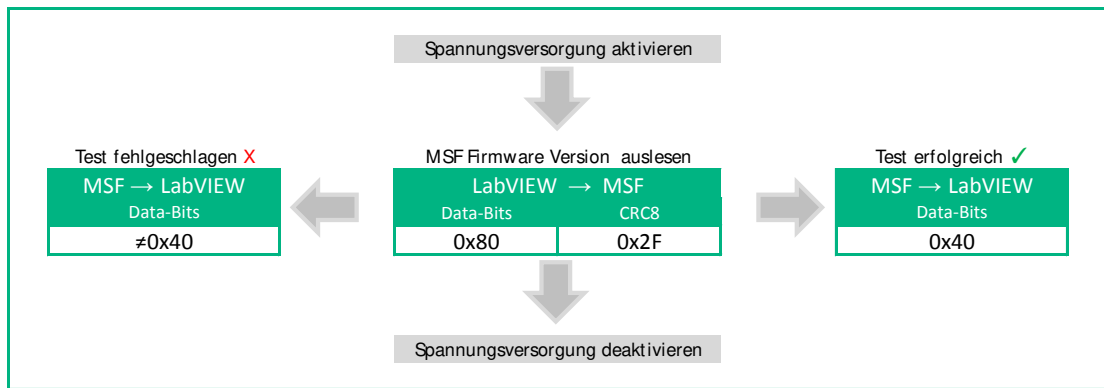


Abbildung 3.8: Testdesign: Testfall 99. Da manche Anforderungen eine Tabelle enthalten (IDS-3310, IDS-3282, IDS-3288) kann in der Zusammenfassung der Anforderungslinks in RQM nur die eindeutige ID, aber kein zusätzlicher Text gezeigt werden. Wenn eine Anforderung in DOORS jedoch als Text ausformuliert wurde, ist dieser bereits in der Zusammenfassung der Anforderungslinks ersichtlich (CRS-MSF-355, IDS-3286).

Nachdem das Testdesign und das Akzeptanzkriterium in RQM definiert worden ist, können die Anforderungen aus DOORS mit dem Testfall verbunden werden. In Abbildung 3.9 wird gezeigt, dass dieser Test mit den fünf identifizierten Anforderungen verknüpft ist.

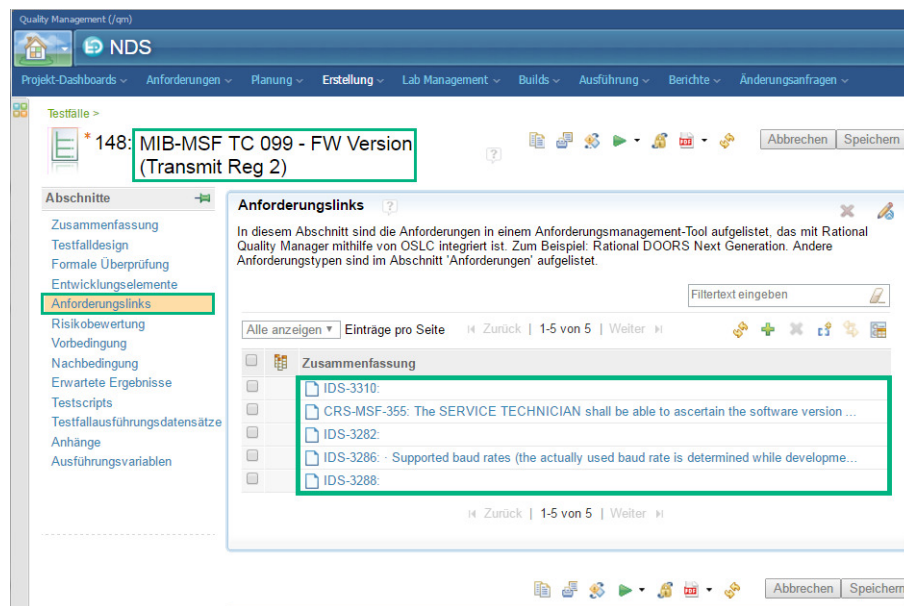


Abbildung 3.9: Screenshot aus RQM: Anforderungen aus DOORS für Testfall 99 in RQM integriert

## Überprüfung der Steuerung der roten OSDs

Der LLS kann mit Hilfe des MSF über das Receive Register 0 die roten OSDs aktivieren. Außerdem kann man über dieses Register auch die Blinkfrequenz der roten OSDs einstellen. Folgende Anforderungen wurden in diesem Zusammenhang identifiziert:

- IDS-3295: Aufbau von Receive Register 0. Damit können die roten OSD LEDs angesteuert werden.
- CRS-MIBHW-3603: MIBHW muss dem LLS ermöglichen, Alarme mit höherer (oder gleich, wenn bereits höchste Priorität erreicht) Priorität als der Alarm-Priorität des TLC anzuzeigen.
- CRS-MSF-404: MSF muss dem LLS ermöglichen den Betriebszustand "Alarm mit hoher Priorität" mit roter Signalanzeige anzuzeigen.

Zur Sicherstellung der Ansteuerung der roten OSD LEDs durch den LLS wird zunächst durch TLC über den MCF eine OSD-Signalfarbe ungleich Rot eingestellt. In den Testfällen 78 bis 80 wurden zuerst die grünen OSDs über den TLC angesteuert und danach eine Ansteuerung der roten OSDs über den LLS getestet. Die Akzeptanzkriterien der Spannungs- und Frequenzmessung unterschieden sich nicht von den in der Testsuite MCF-TLC genannten. Diese Werte können Tabelle 3.8 und 3.9 entnommen werden.

### **Überprüfung des Audiosignals am Lautsprecher**

Der LLS hat die Möglichkeit, die Mute-Aktivierung des TLCs zu deaktivieren und so ein Alarmsignal über den Lautsprecher auszugeben. Außerdem kann der LLS überprüfen, ob ein Audiosignal ausgegeben wird und ob der Lautsprecher gemutet wird. Folgende Anforderungen wurden in diesem Zusammenhang in DOORS identifiziert:

- IDS-3298: Aufbau von Receive Register 1. Über dieses Register kann die Mute-Funktion des TLCs aufgehoben werden.
- IDS-3313: Aufbau von Transmit Register 3. Über dieses Register kann überprüft werden, ob der Lautsprecher gemutet wird und ob ein Lautsprechersignal detektiert wurde.
- CRS-MSF-369: MSF muss dem LLS eine Überwachung der Lautsprecherausgabe (Lautsprecher aktiv/ Lautsprecher nicht aktiv) ermöglichen.

Für diese Testfälle ist es notwendig, am Anschluss "P5 Audio In" ein Audiosignal anzulegen. Gewählt wurde eine Sinusschwingung mit einer Amplitude von 2Vpp und einer Frequenz von 440Hz. Nach Anlegen des Audiosignals wird zum Beispiel überprüft, ob der LLS über den MSF die Mute-Aktivierung des TLCs deaktivieren kann. Die Überprüfung erfolgt über eine Frequenzmessung und das Auslesen von Transmit Register 3, welches in Bit 0 und Bit 1 die Lautsprecherinformation (Lautsprecher-Aktiv-Status, Muting-Status) enthält. Wenn die gemessene Frequenz am Anschluss "P6 SPKR" der Frequenz des Eingangssignals entspricht, wird das Signal ohne Fehler vom MIB übertragen.

### Überprüfung der Steuerung der AQ-LED

Der LLS kann parallel zum TLC die Hinterleuchtung der AQ-Taste aktivieren, allerdings ist eine Deaktivierung funktional nicht möglich. Daher wurden auch Testfälle implementiert, die dieses Szenario abdecken. Folgende Anforderungen wurden aus DOORS identifiziert:

- IDS-3298: Aufbau von Receive Register 1. Über dieses Register kann eine zusätzliche Ansteuerung der AQ-LED erfolgen.
- CRS-MSF-367: MSF muss dem LLS die Aktivierung der Tasten-Hintergrund-Beleuchtung (AQ-Taste) unabhängig vom TLC ermöglichen.

Um herauszufinden, ob eine Ansteuerung durch den LLS möglich ist, werden zuerst alle Hardkey LEDs über den TLC deaktiviert. Diese Maßnahme dient zur Sicherstellung, dass die AQ-LED durch den LLS aktiviert werden kann. Im Anschluss wird der Befehl, der für die Aktivierung der AQ-LED notwendig ist, über die LLS-Schnittstelle an das MIB gesendet. Über die Messung des Spannungspegels am Ausgang "P11 UIB" erfolgt die Beurteilung des Tests.

### Überprüfung der Statusinformation der Hardkeys

Der LLS kann die Statusinformation der Hardkeys abfragen. In DOORS gibt es dazu folgende Anforderungen:

- IDS-3304: Aufbau von Transmit Register 0. Über dieses Register kann die Statusinformation der Hardkeys ausgelesen werden.
- CRS-MIBHW-3564: MIBHW muss dem MSF das Ermitteln der Hardkey-Status-Informationen der UIBHW ermöglichen.
- CRS-MIBHW-3565: MIBHW muss dem MSF das Senden der Hardkey-Status-Informationen zum LLS ermöglichen.
- CRS-MSF-366: MSF muss dem LLS die Ermittlung der Hardkey-Status-Informationen unabhängig vom TLC ermöglichen.

Für diese Tests ist es notwendig, die Betätigung einer Taste (BP+ Taste, BP- Taste, BP-Start Taste, EQ Taste, AQ Taste) zu simulieren und anschließend über das Transmit Register 0 die Statusinformation der Hardkeys auszulesen. Je nachdem, welche Taste betätigt wird, sind die entsprechenden Bits des Transmit Register 0 gesetzt. Die gesetzten Bits werden mit dem Eingangssignal verglichen, um zu beurteilen, ob der richtige Status zurückgegeben wird.

### Überprüfung der Statusinformation der OSDs

Der LLS kann den Aktivierungsstatus der OSD LEDs abfragen. Es kann über das Transmit Register 1 ermittelt werden, ob die blauen, roten oder grünen OSDs aktiv sind und es kann auch ermittelt werden, ob alle vier OSDs die selbe Farbe besitzen. In DOORS ist folgendes festgehalten:

- IDS-3307: Aufbau von Transmit Register 1. Über dieses Register kann ermittelt werden, welche OSDs aktiv sind und ob Aktivierungsfehler vorliegen.
- CRS-MSF-368: MSF muss dem LLS die Ermittlung der OSD Statusinformationen ermöglichen

Für diese Tests werden die unterschiedlichen OSDs über den TLC aktiviert und danach über das Transmit Register 1 die Statusinformation der OSDs ausgelesen. Im Anschluss wird der empfangene Wert des Transmit Registers 1 mit dem vom TLC gesetzten Wert verglichen und beurteilt.

### Überprüfung des MSF-Verhaltens bei Abweichung der Spannungsversorgung von der Nennspannung

Wie bei der Testsuite MCF-TLC muss auch bei der Testsuite MSF-LLS das Verhalten bei Abweichung der Spannungsversorgung von der Nennspannung untersucht werden. Der MSF wird von der 5VLLS Spannung versorgt, welche eine Nennspannung von 5.2V besitzt. Jedoch ist eine Spannung von 4.75V bis 5.25V noch im Toleranzbereich. Ob der MSF auch bei einer Spannung über oder unter der Nennspannung funktioniert, wird anhand eines Audio-Testfalls getestet. Hierzu wird wieder ein Audiosignal (Sinusschwingung: Amplitude 2Vpp, Frequenz 440Hz) am Anschluss "P5 Audio IN" angelegt und über das Transmit Register 3 (genaue Informationen siehe Tabelle 2.11) ausgelesen, ob der MSF ein Audiosignal detektiert.

### Überprüfung der Fehlererkennung des MSF

Der MSF signalisiert in jedem zurückgesendeten Datenbyte (d.h. Transmit Register 0 bis 3 ) mit Bit 7 seinen Fehlerstatus. Darum wird in diesen Testfällen überprüft, ob bei einer falschen CRC oder bei einer falsch ausgewählten Parität ein Fehler in diesem Bit angezeigt wird. Diese Testfälle basieren auf den folgenden Anforderungen:

- IDS-3304: Aufbau von MSF Transmit Register 0
- IDS-3307: Aufbau von MSF Transmit Register 1
- IDS-3310: Aufbau von MSF Transmit Register 2
- IDS-3313: Aufbau von MSF Transmit Register 3



- IDS-3286: MSF-LLS Datenübertragungsparameter (RS232)
- CRS-MSF-442: Zyklische Prüfung der Kommunikationsdaten vom MSF zum LLS
- CRS-MSF-443: Zyklische Prüfung der Kommunikationsdaten vom MSF zum LLS

Nachdem eine fehlerhafte CRC gesendet oder eine falsche Parität eingestellt worden ist, wird in diesen Testfällen Bit 7 des jeweiligen Transmit Registers überprüft.

### 3.1.2.3 Testsuite TCF-TLC

Inhalt dieser Testsuite sind 13 Testfälle, die mit dem TCF in einem funktionalen Zusammenhang stehen. Unter anderem wird abgeprüft, ob die Firmware-Versionsnummer einem voreingestellten Wert entspricht, ob ein Reset des TCFs durchgeführt werden kann, ob die Datenübertragungsrate bei Touch-Daten-Transfer den Anforderungen entspricht und, ob Touch-Daten bei Simulation eines Touch-Signals vom MIB an den TLC übertragen werden. Zusätzlich zu diesen Tests muss überprüft werden, ob der TCF Kommunikationsfehler erkennen und eine Fehlernachricht zurück sendet. Anhand von Testfall 119 wird gezeigt, wie ein Test für eine funktionelle Überprüfung der Richtigkeit der Auswertung der Touch-Daten aussieht und wie die aus DOORS gefilterten Anforderungen in RQM verlinkt werden können.

- *Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 4:*

Über das MIB erfolgt die Verarbeitung der Benutzereingabe am Touchscreen. Für die Überprüfung, ob die Verarbeitung korrekt funktioniert, könnte ein externer Touchpanel angeschlossen und die Touch-Position manuell durch Drücken ausgeübt werden. Damit der Test jedoch automatisch ausgeführt werden kann, werden die MIB Eingangssignale des Touch-Panels am Anschluss "P15 Touch Panel" simuliert. Der Ablauf von Testfall 110 wird schematisch in Abbildung 3.10 dargestellt.

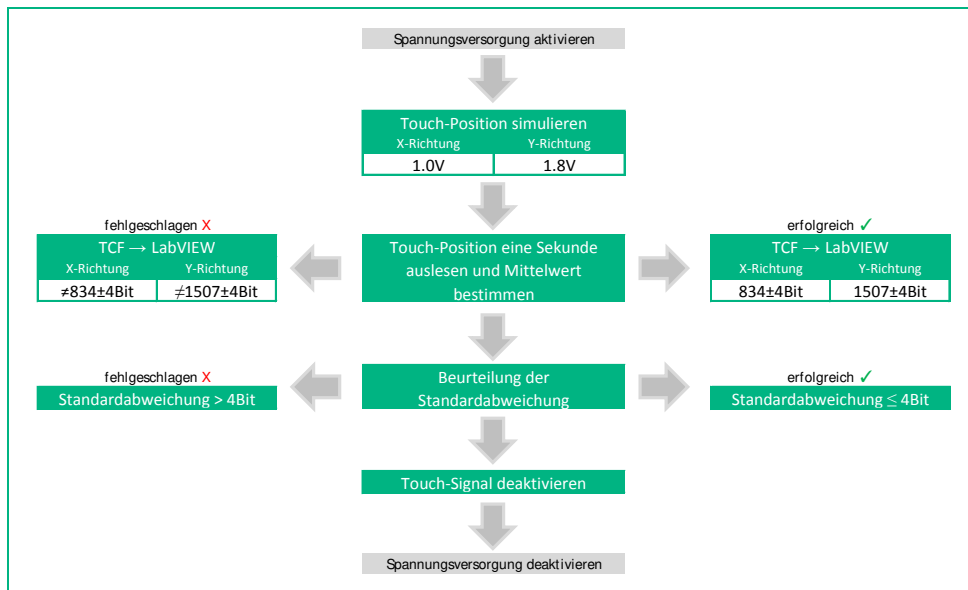


Abbildung 3.10: Testdesign: Schematische Darstellung von Testfall 119 zur Überprüfung, ob Touch-Signale korrekt über das MIB verarbeitet werden.

In Testfall 119 wird die richtige Verarbeitung des MIB über eine konstante Touch-Position (X-Richtung: 1V, Y-Richtung: 1.8V) überprüft. Wie in Kapitel 2.6.2 bereits beschrieben, führen 5 Leitungen vom MIB zum 5-Draht resistiven Touchpanel. Zwei Leitungen (X-, X+) für die Bestimmung in X-Richtung, zwei Leitungen (Y-, Y+) für die Bestimmung in Y-Richtung und eine Leitung, welche als Messelement (5-Wire-Sense) dient.

Das simulierte Signal wird über den Anschluss "P15 Touch Panel" eingespeist, welches der Touch-Panel-Host-Controller und in weiterer Folge der TCF verarbeiten muss. Mit Hilfe des Touch-Panel-Controllers werden die angelegten Spannungen in X- und Y-Richtung in digitale Positionswerte umgerechnet und über den TCF an den TLC übermittelt. Das Senden der Position an die 5-Wire-Sense Leitung, über das Anlegen von Spannungen, stellt ein zeitkritisches Ereignis im Mikrosekunden-Bereich dar. Aus diesem Grund ist es wichtig, ein korrektes Timing zu verwenden.

Im Gegensatz zu den bisherigen Eingangssignalen gestaltet sich die Simulation der Touch-Position etwas schwieriger. Folgende Schritte müssen bei der Übergabe einer Touch-Position an das MIB berücksichtigt werden:

1. Zuerst muss das "Erwachen" (Touch Wake-up/Touch Detection) des Touch Controllers ausgelöst werden. Dies geschieht über einen  $200\mu\text{s}$  langen Pull-

down der (X-)-Leitung (digitaler Ausgang). Der Touch-Controller erkennt den Touch als "gedrückt".

2. Danach wird die Spannung in Y-Richtung an 5-Wire-Sense (P15-3) gelegt. Dies geschieht für mindestens  $300 \mu\text{s}$  oder solange, bis die (Y-)- Leitung (digitaler Eingang) eine steigende Flanke erkennt.
3. Im Anschluss wird die Spannung in X-Richtung über 5-Wire-Sense solange ausgegeben, bis X- (Digitaler Eingang) eine fallende Flanke erkennt oder  $300 \mu\text{s}$  vergangen sind.

Für ein besseres Verständnis über den sequenziellen Ablauf der Simulation der Touch-Position dient das Oszillogramm in Abbildung 3.11.

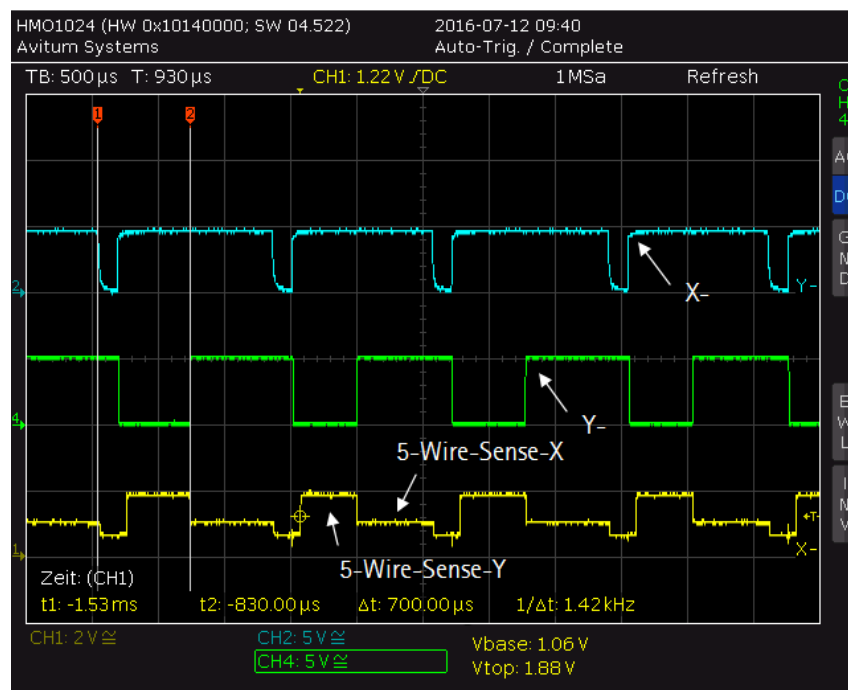


Abbildung 3.11: Oszillogramm: Anschluss "P15 Touch Panel". Darstellung des Signalverlaufs von 5-Wire-Sense (Ch1, gelb) in Abhängigkeit von der X- (Ch2, blau) und Y- (CH4, grün) Leitung.

Die Touch-Position muss vom MIB erkannt werden und die richtigen Positionsdaten über den TCF ausgegeben. Um zu beurteilen, ob es sich um die richtige Touchposition handelt, wurde über eine Referenzmessung die Position bestimmt, die sich durch Anlegen von  $1.0 \text{ V}$  in X- und  $1.8 \text{ V}$  in Y-Richtung ergeben sollte. Tabelle 3.12 enthält die Ergebnisse der Referenzmessung und die davon abgeleiteten Toleranzen.

Tabelle 3.12: Referenzbestimmung der Positionsdaten bei konstanter Touch-Position in X- und in Y-Richtung

Richtung	Spannung	Referenzbestimmung	Toleranzbereich
X-Richtung	1.0V	834	834±4Bit
Y-Richtung	1.8V	1507	1507±4Bit

Für die Auswertung der Positionsdaten werden die Daten, welche vom Touch-Panel-Host-Controller an den TCF gesendet werden, eine Sekunde lang aufgezeichnet und im Anschluss der Mittelwert und die Standardabweichung bestimmt.

Die im Touch Panel Controller verwendete interne Auflösung beträgt 12 Bit, dies entspricht 4096 (0 bis 4095) Diskretisierungsschritten. Die beiden niederwertigsten Bits (1 und 0) werden jedoch aufgrund des internen Signalverarbeitungsalgorithmus nicht für die Berechnung verwendet, sodass sich die realisierte Auflösung daher auf 10 Bit (d.h. 0 bis 1023) reduziert (siehe AR102X-Datenblatt [22]). Tatsächlich sind damit Y1:Y0 und X1:X0 Nullen.

Basierend auf durchgeführten Referenzmessungen für die Toleranzbetrachtung hat sich ergeben, dass aufgrund von unvermeidlichen Spannungsschwankungen im System (d.h. beim Touch-Panel-Controller und beim HIL-Testsystem) Bit 3 und Bit 2 als Schwankungsbreite für das Akzeptanzkriterium zugelassen werden müssen, sodass der Referenzwert  $\pm 4$  Bit erlaubt ist.

Nachdem das Testdesign und das Akzeptanzkriterium in RQM definiert worden ist, können die Anforderungen aus DOORS mit dem Testfall verbunden werden. In Abbildung 3.12 wird gezeigt, dass dieser Test mit den bereits in Kapitel 3.1.1.1 und Kapitel 3.1.1.2 identifizierten Anforderungen verknüpft ist.

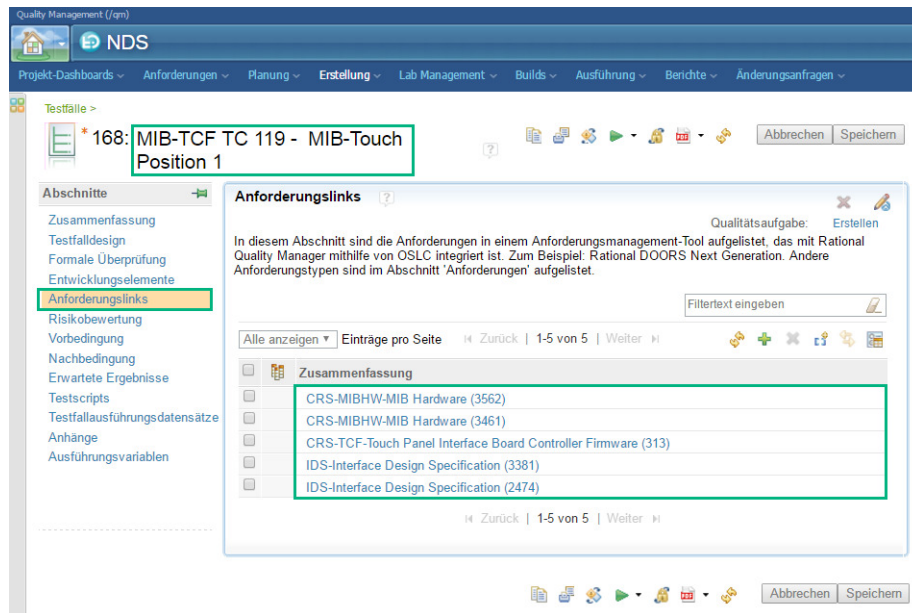


Abbildung 3.12: Screenshot aus RQM: Anforderungen aus DOORS für Testfall 119 in RQM integriert

### 3.1.3 Anforderungen an das Messsystem

Nach der Extraktion der Anforderungen aus DOORS und der Erstellung des Testplans in RQM können die Anforderungen an die Adapterleiterplatte formuliert werden.

Anforderungen an die Adapterleiterplatte:

- 17 analoge Eingänge
- 3 analoge Ausgänge
- 12 digitale Eingänge
- 10 digitale Ausgänge
- 4 Spannungsausgänge
- 1 USB-Schnittstelle
- 2 RS232-Schnittstellen

Diese Leiterplatte wurde innerhalb von B. Braun anhand der spezifizierten Anforderungen realisiert. Die Leiterplatte ist nicht Teil dieser Masterarbeit, wird aber in dieser Masterarbeit verwendet. Für ein besseres Verständnis über die Funktion der Adapterleiterplatte wurde ein Blockschaltbild erstellt, welches in Abbildung 3.13 zu sehen ist.

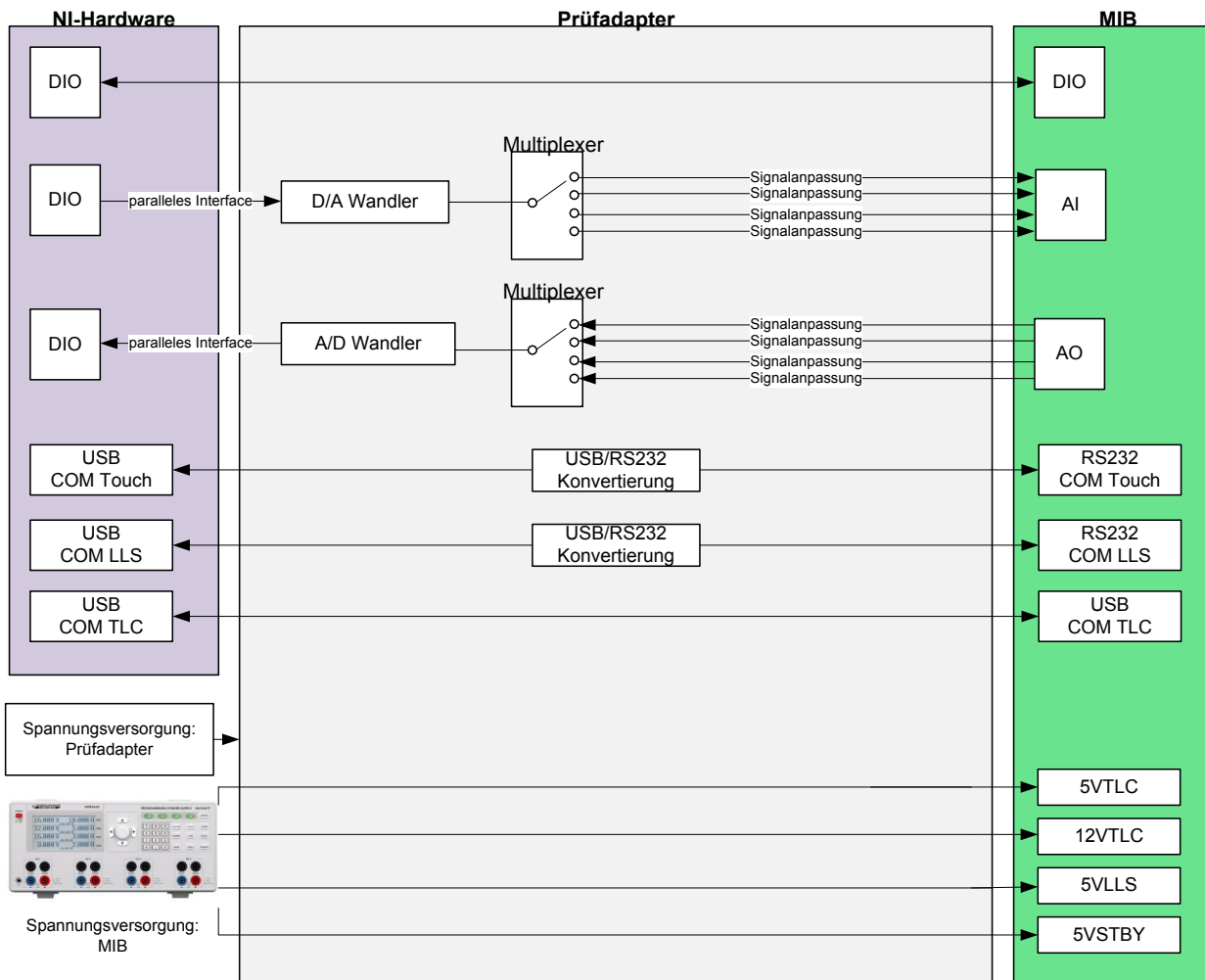


Abbildung 3.13: Blockschaltbild: Funktion der Adapterleiterplatte

### 3.1.3.1 Schnittstellenkonvertierung

Das MIB verwendet als Kommunikationsschnittstellen für MSF-LLS und TCF-TLC jeweils eine serielle RS232. Eine Konvertierung der RS232 Schnittstellen erscheint sinnvoll, da immer mehr Computer keinen RS232-Port mehr aufweisen. Um das HiL-Testsystem möglichst computerunabhängig zu gestalten, wurde auf der Adapterleiterplatte eine Konvertierung der RS232 Schnittstellen auf USB Schnittstellen vorgesehen.

### 3.1.3.2 Signalanpassung: Pegelwandler

Von der NI Hardware werden lediglich low voltage TLL Signale mit 3.3 V Pegel ausgegeben, daher werden Komparatoren als Pegelwandler verwendet, um das MIB mit den notwendigen 5V TTL Logiksignalen zu versorgen. Hierbei wird die Signallogik invertiert. Zusätzlich realisieren die Komparatoren in dieser Konfiguration die auf dem UIB verwendeten Pull-Up-Widerstände und simulieren den Tastendruck.

### 3.1.3.3 Analog-Digital-Wandler/Digital-Analog-Wandler

Die im Meßsystem verwendete Messkarte verwendet FPGAs, welche aufgrund ihrer hohen Geschwindigkeit die Steuerung der Analog-Digital-Wandler/Digital-Analog-Wandler (A/D-Wandler, D/A-Wandler) mit Hilfe einer Parallelschnittstelle ermöglicht. So kann eine mikrosekundengenaue Synchronisation zum Wandlungszeitpunkt sichergestellt werden.

Da die Tests nicht parallel stattfinden, ist es möglich, die Signale über Multiplexer an den A/D-Wandler und D/A-Wandler anzubinden. Der D/A-Wandler verfügt über eine Auflösung von 12 Bit und funktioniert nach dem R2R<sup>6</sup> Prinzip, um hohe Wandlungsraten bei gleichzeitig simpler Funktionsweise zu ermöglichen. Der ADC verfügt ebenfalls über eine Auflösung von 12 Bit und funktioniert nach dem Prinzip der sukzessiven Approximation (auch Wägeverfahren genannt).

---

<sup>6</sup> Widerstandsnetzwerk mit den Werten R und 2R

## 3.2 Implementation der Verifikationsplattform

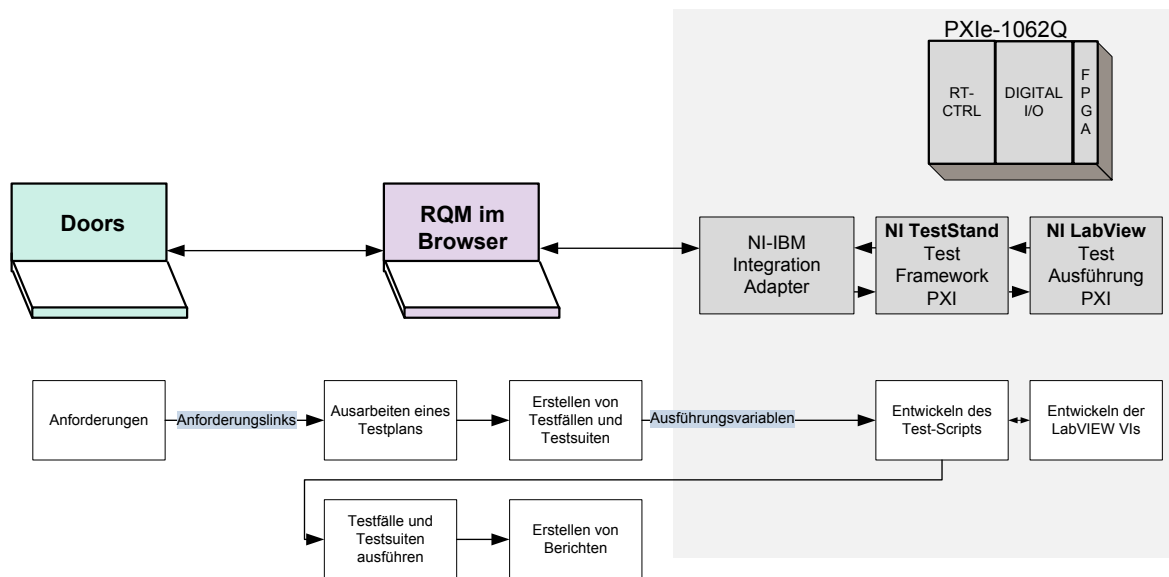


Abbildung 3.14: Aufbau des SW-Frameworks bestehend aus IBM DOORS, IBM RQM, NI TestStand und NI LabVIEW. DOORS dient als Anforderungsmanagement Tool, welches die Hardware- und Software Design Spezifikationen enthält. Anhand dieser Anforderung wird die Hard- und Software entwickelt. RQM ist ein Test-Management-Tool, welches anhand der in DOORS enthaltenen Anforderungen dazu benutzt werden kann, automatische Test ablaufen zu lassen. Nach Abschluss der Tests können die Ergebnisse über einen automatisch erstellten Report eingesehen werden. Über den NI-IBM Integrationsadapter wird die Verbindung zwischen IBM und NI hergestellt. In TestStand kommt es zum Design und zur Ausführung der Testsequenz und der einzelnen Testschritte (LabVIEW). Nach Ablauf der Testschritte wird ein Testreport generiert, der RQM zur Verfügung gestellt wird.

### 3.2.1 Aufbau des HiL-Testsystems

Um medizinische Elektronikkomponenten wie das MIB in einem HiL-Testsystem zu prüfen, muss eine möglichst realitätsnahe Simulationsumgebung geschaffen werden. Man setzt den Prüfling der Simulationsumgebung aus, erfasst die Antwort des Prüflings und wertet diese entsprechend der Anforderungsspezifikation aus. Um die Signale des Prüflings zu simulieren, wird ein Prüfadapter verwendet, der die Schnittstelle zwischen den NI Komponenten und dem MIB bildet. Dieser Adapter wurde anhand der Anforderungen, die aus der Signalanalyse hervorgehen, von B. Braun entwickelt.

Von National Instruments wird das PXIe-1062Q Chassis mit acht Steckplätzen verwendet. In dem Chassis befindet sich ein Controller (NI PXIe-8115 - PXI-Controller mit 2,5-GHz-Dualcore Prozessor), welcher einen seriellen Anschluss, Ethernetanschlüsse, 6 Hi-Speed USB Schnittstellen und weitere Anschlüsse für Peripheriegeräte besitzt. Da der



PXI-Controller ein vollständiges System (Festplatte, RAM, CPU) enthält, wird kein externer PC benötigt.

Außerdem wird das Digital-RIO-Modul NI PXI-7813R verwendet, welches über einen rekonfigurierbaren FPGA-Chip verfügt. Dieses Modul besitzt 160 Digitalkanäle, die je nach Bedarf als Ein- oder Ausgänge oder als Counter bei Frequenzen von bis zu 40 MHz konfiguriert werden können. So können Funktionen mit einer Auflösung von 25 ns realisiert werden. Die als Ausgänge konfigurierten Kanäle können Spannungen von 0 bis 3.3 Volt ausgeben. Die als Eingänge konfigurierten Kanäle können 0 bis 5 Volt einlesen.

### **3.2.2 Schnittstellen der Hardwarekomponenten des HiL-Testsystems**

Das MIB wird über eine 78-polige D-Sub-Steckerverbindung mit der Adapterleiterplatte verbunden. Von der Adapterleiterplatte sorgen drei USB- und zwei 68-polige VHDCI-Steckerverbindungen für eine Verbindung mit der NI Hardware. Der Aufbau des HiL Testsystems ist Abbildung 3.15 zu entnehmen. Die genaue Pinbelegung des HiL-Testsystems kann den Tabellen im Anhang in Kapitel 8.3 entnommen werden.

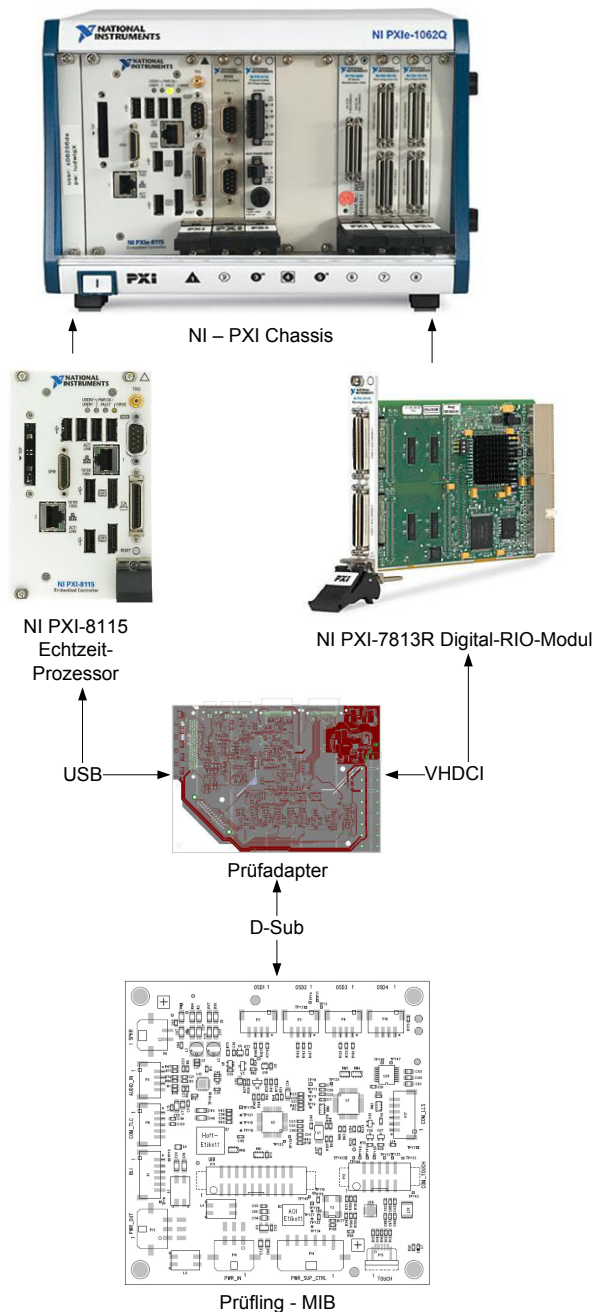


Abbildung 3.15: Schnittstellen der Hardwarekomponenten des HiL-Testsystems

### 3.2.3 Entwicklung der Testumgebung in TestStand

Der Testplan, der in RQM erstellt wurde, wird im nächsten Schritt in die Testmanagementsoftware TestStand eingepflegt. Die Architektur der zugehörigen LabVIEW VIs wird in Kapitel 3.2.4 erläutert.

Mit Hilfe des TestStand *Sequence Editors* kann eine Testsequenz erstellt werden. Der Grundbaustein einer jeden Sequenz sind die Testschritte. Es gibt fünf verschiedene Arten

von Testschritten (Pass/Fail Test, Numeric Limit Test, Multiple Numeric Limit Test, String Value Test und Action). Diese werden in Abbildung 3.16 dargestellt.

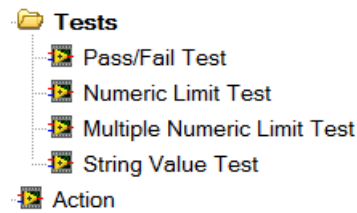


Abbildung 3.16: TestStand: Arten von Testschritten

Die verschiedenen Arten von Testschritten werden dazu genutzt, um Code Module sequenziell aufzurufen und Messungen über definierte Grenzen zu evaluieren. Es ist in TestStand möglich diverse Code Module zu verwenden, um Eingangssignale zu simulieren oder Messungen durchzuführen. Es werden Code Module von LabVIEW, C, C++ und VB.NET von TestStand unterstützt. Im Zuge dieser Arbeit wird rein mit LabVIEW Code gearbeitet. Das Ziel besteht darin klar definierte, unabhängige und modular aufrufbare Bausteine zu erstellen, da in dieser Weise eine einfach wartbare und vielseitig einsetzbare Sequenz entsteht. Einfache Operationen können auch in TestStand ausgeführt werden, sobald jedoch mit größeren Datenmengen gearbeitet wird, ist es ratsam ein Code Modul in die TestStand Sequenz einzufügen. Anhand des Testdesigns und der Akzeptanzkriterien, welche in RQM definiert worden sind, kann in TestStand die Sequenz erzeugt werden. Ein Überblick über den Aufbau einer Testsequenz kann Abbildung 3.17 entnommen werden.

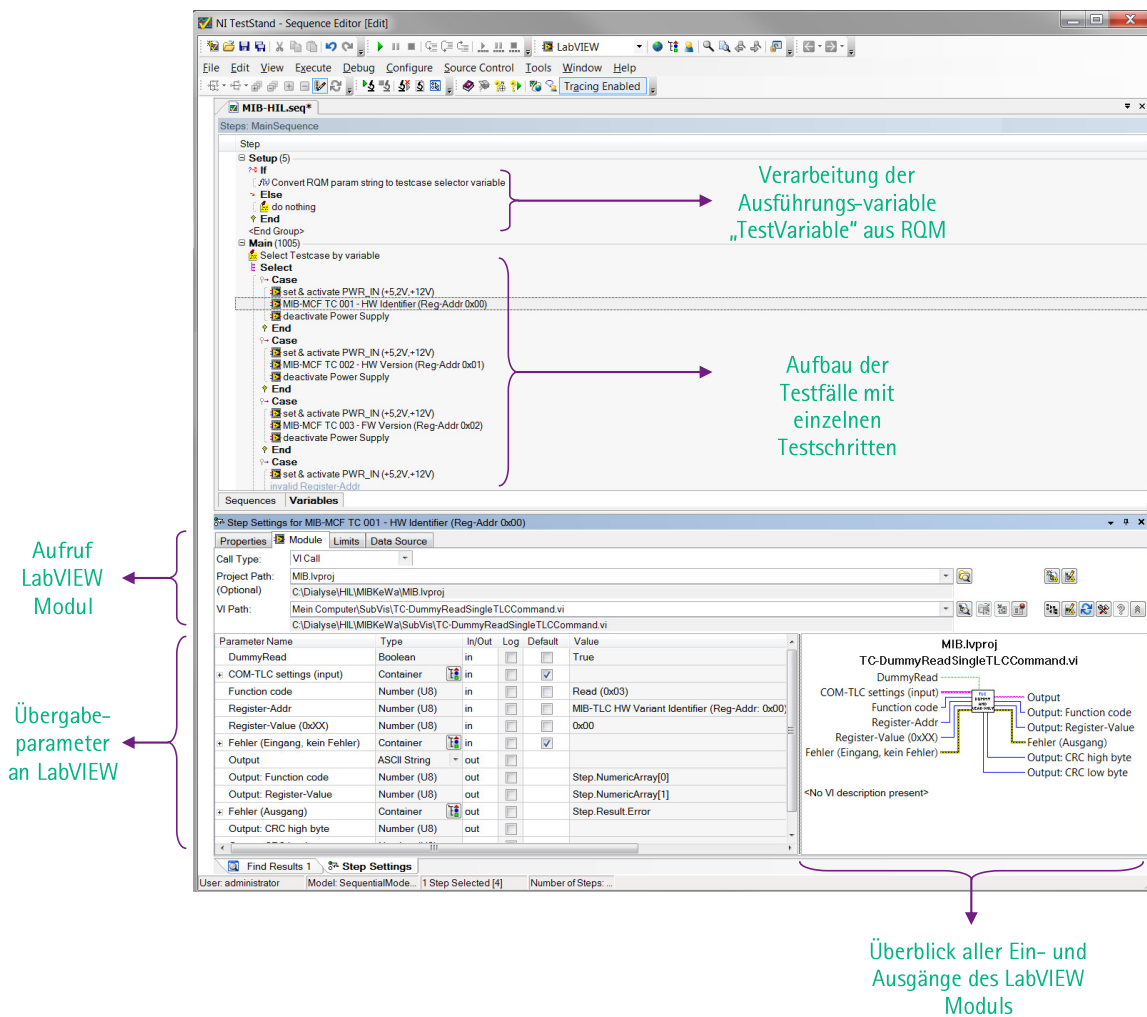


Abbildung 3.17: TestStand: Aufbau der Sequenzdatei

Über die Ausführungsvariable `TestVariable` wird die Traceability von IBM-RQM zu NI-TestStand umgesetzt. Damit IBM-RQM mit NI-TestStand kommunizieren kann ist ein Adapter notwendig. Dieser Adapter heißt NI Test Integrations Adapter. IBM-RQM übernimmt die Ablaufsteuerung der Testfälle, indem RQM die entsprechende Ausführungsvariable an TestStand sendet, wobei die Ausführungsvariable der Testfallnummer entspricht. Da Ausführungsvariablen als String von RQM an TestStand übergeben werden, muss diese Variable in TestStand in eine Zahl umgewandelt werden. In der Main-Sequenz von TestStand wird dann über eine Case Struktur der entsprechende Testfall durchgeführt. Nachdem der Testfall ausgeführt worden ist, liefert TestStand einen Report an RQM. Außerdem werden die Ergebnisse der Testdurchführung auch in DOORS mit den entsprechenden Anforderungen verlinkt.

- *Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 5:*  
Abbildung 3.18 zeigt die einzelnen Testschritte dieses Testfalls. Zuerst wird die Spannungsversorgung mit dem `HMP4040initandsetvoltage.vi` aktiviert. Im nächsten

Schritt wird die Firmware Version des MSF über das LLS\_COM.vi ausgelesen und über einen Numeric-Limit-Test bewertet, wobei das Ergebnis 0x40 sein muss. Abschließend wird die Spannungsversorgung über das HM04040deactivateandclose.vi deaktiviert.

The screenshot displays the TestStand interface for Case 99. The top section shows the test steps: 'set & activate PWR\_IN (+5,2V,+12V) and set & activate PWR\_OUT (+5,2V,+12V)', 'MIB-MSF TC 099 - FW Version (Transmit Reg 2)', and 'deactivate Power Supply'. Below this is the 'Properties' window for the 'Limits' tab, showing a 'Numeric Limit Test' with a comparison type of 'EQ (==)', a low limit of '0x40', and a high limit of '0xb'. The 'Parameter Name' table lists various inputs and outputs for the 'LLS\_COM.vi' module, including 'CRC automatic', 'COM-LLS settings (input)', 'Input', 'CRC8 (manual)', 'Fehler (Eingang, kein Fehler)', 'Output', 'Output: Data', 'Output: PDC', 'Fehler (Ausgang)', 'no output after Transmit register', and 'CRC8 (automatic)'. The 'Limits' table shows the comparison type 'EQ (==)', low limit '0x40', high limit '0xb', and numeric format 'Hexadecimal'. To the right, a diagram shows the 'LLS\_COM.vi' module with its inputs and outputs connected to the test steps.

Parameter Name	Type	In/Out	Log	Default	Value
CRC automatic	Boolean	in	<input type="checkbox"/>	<input checked="" type="checkbox"/>	True
COM-LLS settings (input)	Container	in	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Input	Number (U8)	in	<input type="checkbox"/>	<input type="checkbox"/>	0x80
CRC8 (manual)	Number (U8)	in	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
Fehler (Eingang, kein Fehler)	Container	in	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Output	ASCII String	out	<input type="checkbox"/>	<input type="checkbox"/>	
Output: Data	Number (U8)	out	<input type="checkbox"/>	<input type="checkbox"/>	Step.Result.Numeric
Output: PDC	Number (U8)	out	<input type="checkbox"/>	<input type="checkbox"/>	
Fehler (Ausgang)	Container	out	<input type="checkbox"/>	<input type="checkbox"/>	
no output after Transmit register	Number (I16)	out	<input type="checkbox"/>	<input type="checkbox"/>	
CRC8 (automatic)	Number (U16)	out	<input type="checkbox"/>	<input type="checkbox"/>	

Comparison Type	EQ (==)
Low	0x40
High	0xb
Units	
Numeric Format	Hexadecimal

Abbildung 3.18: TestStand: Testschritte von Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)). Oben im Bild erkennt man die Architektur des Testfalls. Darunter werden die Parameter, welche an das LabVIEW VI übergeben werden angezeigt. Ganz unten im Bild sieht man das Akzeptanzkriterium für den Numeric-Limit-Test.

- Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 5:

Abbildung 3.19 zeigt die Realisierung in TestStand der einzelnen Testschritte dieses Testfalls. Zu Beginn wird die Spannungsversorgung mit dem HMP4040initandsetvoltage.vi aktiviert. Danach wird die Touch-Position mit dem TC-TouchSetPosition.vi gesetzt. Als nächstes wird überprüft, ob der TCF die richtige Touch-Position an den TLC übermittelt. Diese Überprüfung erfolgt über einen Multiple-Numeric-Limit-Test und das TC-HostTouchPosition.vi. Bei einem Multiple-Numeric-Limit-Test wird ein Array mit den Ergebnissen, die dem Test unterzogen werden, erstellt und über das Akzeptanzkriterium überprüft. Dies sind X-Position, Y-Position sowie eine Boolesche Variable die Aufschluss darüber gibt, ob die Standardabweichung der detektierten X- und Y-Positionswerte innerhalb von 4 Bit liegt. Abschließend wird die Spannungsversorgung über das HM04040deactivateandclose.vi deaktiviert.

The screenshot displays the LabVIEW TestStand interface for a test case. At the top, a list of steps for 'Case 119' is shown, including actions like 'set & activate PWR\_IN (+5.2V)', 'set Touch voltage', 'cyclic measurement of X & Y position data', 'MIB-TCF TC 119 - MIB-Touch Position 1', 'clear Touch signal', and 'deactivate Power Supply'. Below this, the 'Properties' tab is active, showing 'Call Type: VI Call', 'Project Path: MIB.lvproj', and 'VI Path: Mein Computer\HostVi\TC-HostTouchPosition.vi'. A table lists parameters for the VI, such as 'standard deviation within X...', 'mean - X - Digit (Touch)', and 'data message repeat rate'. To the right, a diagram shows the VI 'MIB.lvproj TC-HostTouchPosition.vi' with its inputs and outputs. At the bottom, a 'Limits' table defines acceptance criteria for the test steps.

Parameter Name	Type	In/Out	Log	Default	Value
standard deviation within X a...	Number (I16)	out	<input type="checkbox"/>		Step.NumericArray[2]
mean - X - Digit (Touch)	Number (DBL)	out	<input type="checkbox"/>		Step.NumericArray[0]
standard deviation - X - Digit ...	Boolean	out	<input type="checkbox"/>		
mean - Y - Digit (Touch)	Number (DBL)	out	<input type="checkbox"/>		Step.NumericArray[1]
standard deviation - Y - Digit ...	Boolean	out	<input type="checkbox"/>		
data message repeat rate	Number (I32)	out	<input type="checkbox"/>		

Index	Measurement Name	Settings
0	mean - X - Digit (Touch)	>= 819 And <= 849
1	mean - Y - Digit (Touch)	>= 1492 And <= 1522
2	standard deviation within X and within Y <= 4 Bit?	== 1

Abbildung 3.19: TestStand: Testschritte von Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1). Oben im Bild erkennt man die Architektur des Testfalls. Darunter werden die Parameter, welche an das LabVIEW VI übergeben werden angezeigt. Ganz unten im Bild sieht man das Akzeptanzkriterium für den Numeric-Limit-Test.

### 3.2.4 Entwicklung der LabVIEW Softwareumgebung

Es gibt zwei mögliche LabVIEW Bereiche, zum einen Host-VIs mit klassisch PC basierten Programmen, die das Interface zu Teststand haben, zum anderen können durch Verwendung der FPGA Messkarte FPGA-VIs erstellt werden. Diese sind rekonfigurierbar, können Echtzeit-Probleme bis in Mikrosekunden-Bereich lösen und sämtliche Ein- und Ausgänge der Messkarte ansprechen. Aus Tabelle 3.14 kann entnommen werden, für welche Aufgaben es notwendig ist ein FPGA-VI zu erzeugen und bei welchen VIs ein Host-VI ausreicht.

Tabelle 3.13: Überblick der LabVIEW Implementierungstypen: Host-VIs

Beschreibung	LabVIEW Implementierungstyp	Kommentar
Kommunikation MIB-MCF mit TLC	Host-VI	VISA-Gerätekommunikation
Kommunikation MIB-MSF mit LLS	Host-VI	VISA-Gerätekommunikation
Kommunikation MIB-TCF mit TLC	Host-VI	VISA-Gerätekommunikation
Kommunikation Netzteil	Host-VI	VISA-Gerätekommunikation

Tabelle 3.14: Überblick der LabVIEW Implementierungstypen: Host-VIs & FPGA-VIs

Beschreibung	LabVIEW Implementierungstyp	Kommentar
Setzen der Hardkey-Pegel: (BP-)-, BP START/STOP-, (BP+)-, AQ-, EQ-, POWER ON/OFF Taste	Host-VI	Festlegen von Bedienelementen im FPGA-VI auf dem FPGA
	FPGA-VI	FPGA I/O Node: DO <sup>7</sup> über FPGA setzen
Hardkey-LED-Pegelbestimmung: BP START/STOP-, AQ-, EQ-, POWER ON/OFF LED	Host-VI	Abfragen von Anzeigeelementen im FPGA-VI auf dem FPGA
	FPGA-VI	FPGA I/O Node: DI <sup>8</sup> über FPGA abfragen
Audiosignal simulieren und messen	Host-VI	Erstellen des analogen Signalverlaufs (Sinus, Rechteck), Auslesen des DMA-FIFOs, Konvertierung des digitalen Werts in eine Spannung
	FPGA-VI	Ansteuerung des D/A-Wandlers für die Audio-Signalausgabe, Ansteuerung des A/D-Wandlers für die Audio Signal Messung, Verwendung von DMA-FIFOs für die Übertragung großer Datenmengen zum Host-VI
Bestimmung des Pegels der Hintergrundbeleuchtung des Monitors der NDS (BLI ENA)	Host-VI	Abfragen eines Anzeigeelements im FPGA-VI auf dem FPGA
	FPGA-VI	FPGA I/O Node: DI über FPGA abfragen
Touch-Signal simulieren	Host-VI	Konvertierung der analogen Spannungswerte in digitale Werte, Festlegen von Bedienelementen (Touch-Position) im FPGA-VI auf dem FPGA
	FPGA-VI	Ansteuerung DAC für die Simulation der Touch-Position (FPGA I/O Node: DI über FPGA abfragen, DO über FPGA setzen)
Messen der bisher nicht genannten AI <sup>9</sup> Signale (Frequenzmessung der OSD, analoge Spannungsmessung OSD LED, analoge Messung der Spannungsversorgungen (12V TLC, 5VTLC, 5VCMB, 5VLLS, 5V STBY))	Host-VI	Auslesen des DMA-FIFOs, Auswerten des Empfangssignals (Spannungs-, Frequenzmessungen)
	FPGA-VI	Messungen über den ADC (zeitkritische Ansteuerung) Verwendung von FIFOs für die Übertragung großer Datenmengen
Setzen des externen Reset-Pegels	Host-VI	Festlegen von Bedienelementen im FPGA-VI auf dem FPGA
	FPGA-VI	FPGA I/O Node: DO über FPGA setzen
Bestimmung der Einschaltzeit des pulsbreitenmodulierten Signals der Hintergrundbeleuchtung des Monitors der NDS (BLI PWM)	Host-VI	Auslesen des DMA-FIFOs, Abfragen eines Anzeigeelements im FPGA-VI auf dem FPGA
	FPGA-VI	FPGA I/O Node: DI über FPGA abfragen und die Einschaltzeit mit Hilfe von FGPA-I/O Methodenknöten

<sup>7</sup> digitaler Ausgang

<sup>8</sup> digitaler Eingang

<sup>9</sup> analoger Eingang

Das Host-VI dient der High-Level-Verarbeitung. Damit werden im HiL-Testsystem alle Pins angesprochen, deren Aufrufzeitpunkt nicht zeitkritisch ist, unter anderem, die Ansteuerung der Schnittstellen zum MIB (TLC, LLS) und die Ansteuerung des Netzteils zur Spannungsversorgung. Die Umsetzung der USB-basierten Gerätekommunikation geschieht mit NI- Virtual Instrument Software Architecture (VISA). Dieser Softwarestandard kann dazu genutzt werden, um zum Beispiel USB-Schnittstellen zu konfigurieren oder zu programmieren. Dadurch bietet NI-VISA den Vorteil einer vereinfachten Systemintegration. Anhand von Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 6 wird verdeutlicht, wie der Grundaufbau zur USB-Kommunikation mit NI-VISA aussieht.

- *Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 6:*

Für den Testfall 99 benötigt man kein FPGA-VI, da dieser Testfall keine zeitkritischen Aktionen beinhaltet. Auf die wesentlichen Schritte reduziert, wird in Abbildung 3.20 das LLS\_COM.vi dargestellt, welches für die Kommunikation zwischen dem MIB und dem LLS herangezogen wird.

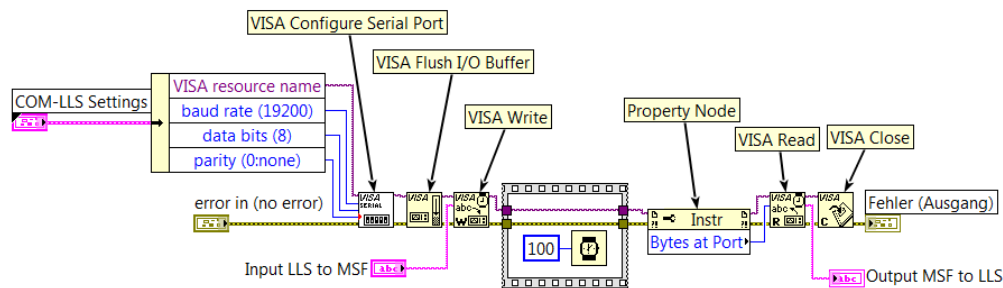


Abbildung 3.20: LabVIEW VI zur Kommunikation mit dem LLS

Die Gerätekommunikation über NI-VISA besteht aus fünf Grundbausteinen. Zuerst ist es notwendig den COM-Port über ein `VISA Configure Serial Port.vi` zu konfigurieren. Der `VISA resource name` entspricht dem COM-Port des Computers, er kann dem Geräte-Manager entnommen werden. Die Parameter der Datenübertragung beinhaltet die IDS-3286. Im Anschluss an die Konfiguration, wird der Buffer über das `VISA Flush I/O Buffer.vi` geleert, um ungültige Daten aus dem Buffer zu löschen. Danach wird mit dem `VISA Write.vi` ein Befehl an den MSF gesendet 100 Millisekunden gewartet und anschließend die Antwort mit dem `VISA Read.vi` ausgelesen. Je nachdem wie viele Bytes am Port anliegen, wird über den `Property Node` ein Integer mit genau dieser Zahl an Bytes an das `VISA Read.vi` weitergegeben. Abschließend muss die serielle Kommunikation über ein `VISA Close.vi` geschlossen werden, wodurch die Ressource freigegeben wird.



Analog dazu erfolgt die Kommunikation mit dem TCF und dem MCF. Auch die Ansteuerung des Netzteils erfolgt über die NI-VISA Gerätekommunikation.

Durch den Einsatz von Field-Programmable Gate Arrays (FPGAs) kann in LabVIEW ein modularer und vielseitig einsetzbarer Code erzeugt werden. Gewünschte Hardwarefunktionen, können ohne Änderung der physikalischen Messhardware, umgesetzt werden. Um die gewünschte Konfiguration zu erreichen, sind Logikblöcke und programmierbarer Routing-Ressourcen im FPGA vorhanden. Je nachdem wie die im FPGA enthaltenen Komponenten verbunden werden, ergibt sich, nachdem eine Schaltungskonfiguration kompiliert worden ist, die gewünschte Funktionalität. Diese ist vollständig rekonfigurierbar und kann speziell an die Messaufgabe angepasst werden. Man verwendet ein Host-VI, um Informationen zwischen dem Host-Computer und dem FPGA-Zielsystem auszutauschen (siehe Abbildung 3.21). Der Host kann ein Windows basierter Computer oder ein Realtime-Target sein. In dem realisierten HiL-Testsystem wird ein Digital-RIO-Modul NI PXI-7813R mit rekonfigurierbaren FPGA verwendet.

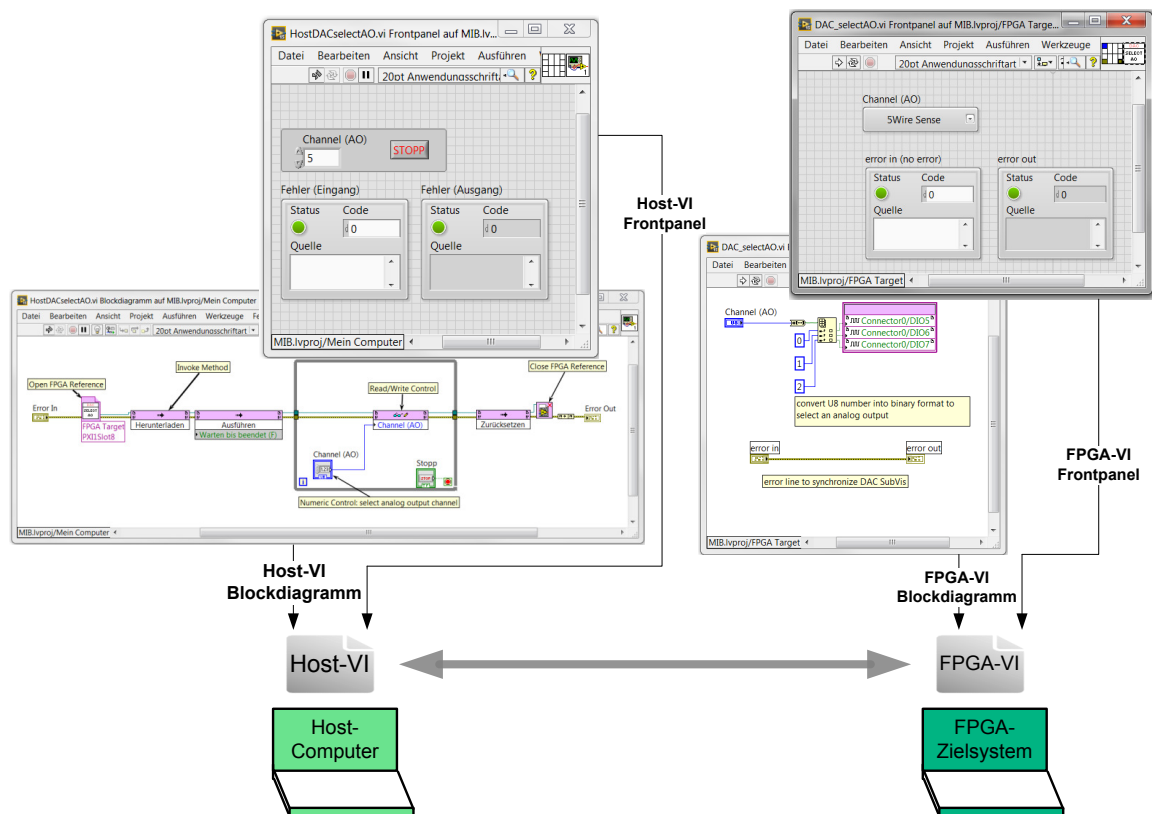


Abbildung 3.21: Architektur FPGA-Programmierung

Durch die Verwendung von FPGAs ergibt sich in LabVIEW eine zweiteilige Architektur. Es gibt das FPGA-VI und das Host-VI. Zu beachten ist, dass das Host-VI und das zugehörige FPGA-VI nicht dieselben LabVIEW-Standard-Paletten zur Verfügung stellen.

Viele Tools sind nur im Host-VI zugänglich. Darunter befinden sich Tools zur Datenverarbeitung und Tools zur Datenspeicherung.

Alle zeitkritischen Pins des HiL-Testsystems werden über den FPGA angesteuert. Dazu zählen Pins an denen Frequenzen gemessen oder ausgegeben werden müssen, sowie Pins die zur Ansteuerung der Multiplexer oder zur Ansteuerung der AD-/DA-Wandler dienen. Die Grundstruktur (siehe Abbildung 3.22) von Host-VIs lässt sich durch folgende Schritte definieren:

- Zuerst wird die Referenz auf das FPGA-VI geöffnet (`Open FPGA Reference.vi`).
- Danach wird die zum Testfall passende FPGA Konfiguration heruntergeladen und ausgeführt.
- `Controls`<sup>10</sup> und `Indicators`<sup>11</sup> sind im FPGA-VI die einfachste Methode, um Daten mit einem Host-VI auszutauschen. Dieser Datenaustausch erfolgt programmatisch über das `Read/Write Control.vi`.
- Zusätzlich gibt es sogenannte `Invoke Methods`, die dazu genutzt werden, große Datenmengen zwischen dem Host und dem FPGA auszutauschen. Dies geschieht über das "First In – First Out" Verfahren oder über Buffer mit direkten Speicherzugriff (`DMA`<sup>12</sup>).
- Zuletzt muss die FPGA Referenz geschlossen werden.

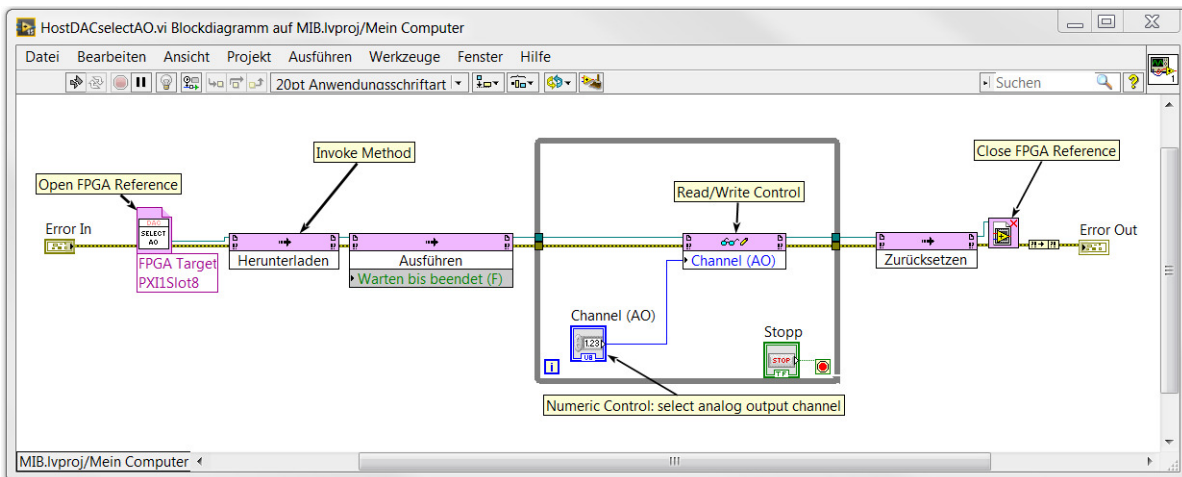


Abbildung 3.22: Darstellung der Grundstruktur eines Host-VIs

<sup>10</sup> Bedienelemente

<sup>11</sup> Anzeigeelemente

<sup>12</sup> Direct Memory Access

Die Programmierung des FPGAs erfolgt über ein FPGA-VI. Bei der Kompilierung wird das FPGA-VI in VHDL Code übersetzt, wobei die Übersetzungsdauer zwischen zwei und 20 Minuten, je nach Komplexität des VIs, liegt. Anhand von Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 6 wird in weiterer Folge erklärt, wie der prinzipielle Umgang mit FPGA-VIs aussieht und welche NI-Elemente genutzt werden müssen.

- *Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 6:*

Für die Simulation des Touch-Signals ist die Programmierung des FPGAs notwendig, da sehr viele zeitkritische Ereignisse im Mikrosekundenbereich zu berücksichtigen sind. Wie bereits in Schritt 4 dieses Beispiels erklärt worden ist, ist für die Simulation ein festgelegter 5-Wire-Sense Signalverlauf einzuhalten, sodass das MIB eine Touch-Position erkennt und auch die Koordinaten der Touchposition an den TLC sendet. Abbildung 3.23 zeigt diesen Ablauf im LabVIEW Blockdiagramm.

Zuerst wird die 5-Wire-Sense Leitung über den Multiplexer ausgewählt, dazu dient das `DAC_selectA0.vi`. Wenn eine Touch-Position simuliert werden soll, muss über das Host-VI eine Variable `Touch` als "true" übergeben werden, sodass die (X-)Leitung auf Masse gezogen wird und der Touch-Controller "erwacht". Damit sichergestellt werden kann, dass der Touch-Controller die Aktivierung erkennt, wird der "Pull-down" für 200  $\mu s$  durchgeführt.

Im Anschluss wird über die 5-Wire-Sense-Leitung die vom Host-VI erhaltene Spannung in Y-Richtung ausgegeben. Dies geschieht mit Hilfe des `DAC_writeloadRegisterA0.vi` solange, bis an der (Y-)Leitung eine steigende Flanke erkannt wird oder mindestens 300  $\mu s$  vergangen sind. In dem `DAC_writeloadRegisterA0.vi` werden die Steuerleitungen des D/A-Wandlers so angesteuert, dass die entsprechende analoge Spannung an 5-Wire-Sense angelegt wird. Danach wird für die Dauer von 300  $\mu s$  oder bis an der (X-)Leitung eine fallende Flanke erkannt wird, die Spannung in X-Richtung über die 5-Wire-Sense-Leitung eingespeist.

Dieser Ablauf geschieht in einer Sequenz, wobei die Simulation der Touch-Position solange durchgeführt wird, bis die Boolesche-Variable `Touch` "false" ist. Wenn die Variable `Touch` "false" ist, wird der Pin der (X-)Leitung nicht auf Masse gezogen, wodurch der Touch-Controller keine Touch-Position erkennen kann und dadurch auch keine Daten an den TLC sendet.

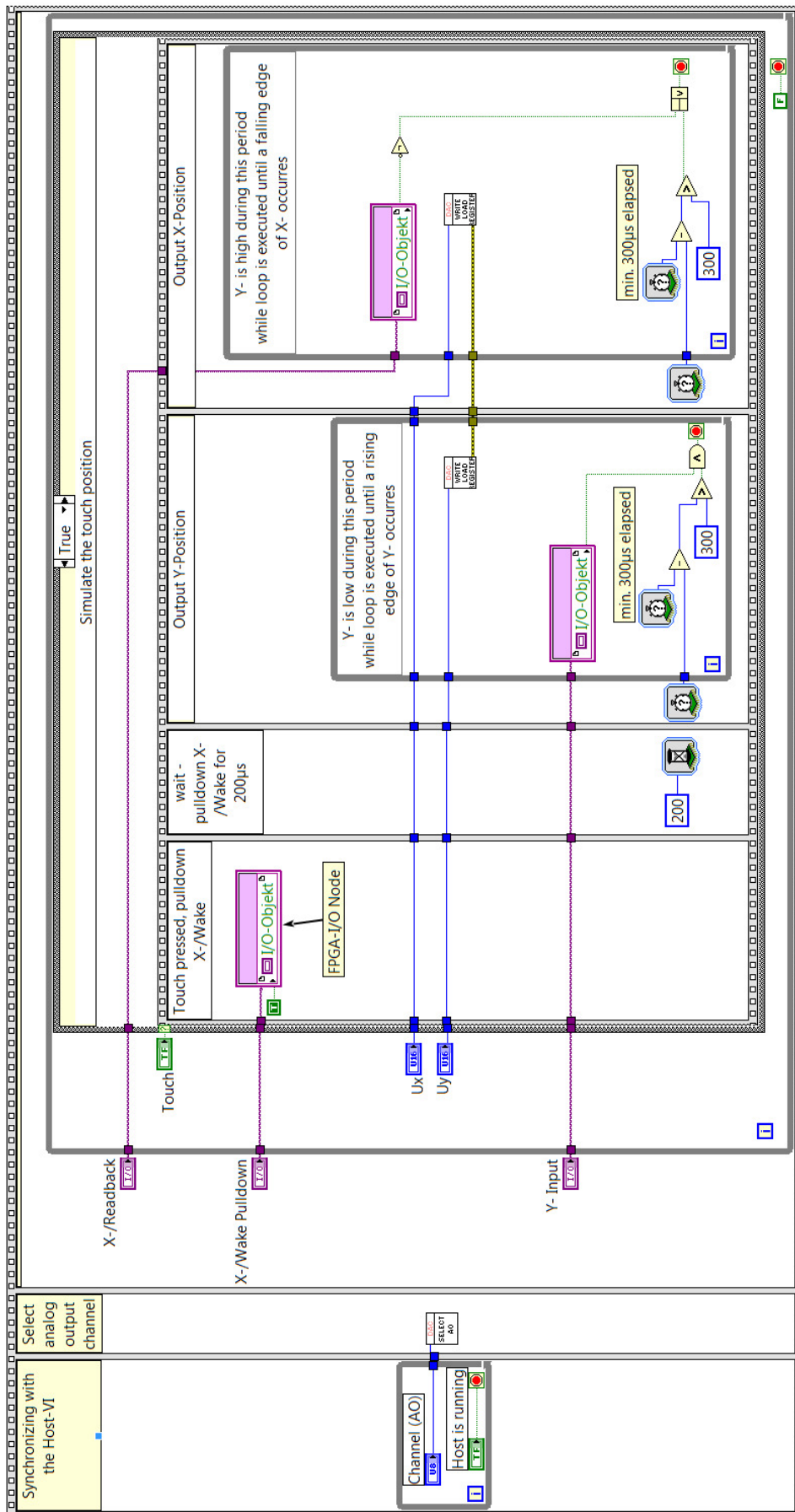


Abbildung 3.23: Simulation einer Touch-Position über die 5-Wire-Sense- und (X)-Leitung



---

## 4 Nachweis der Wirksamkeit des HiL-Testsystems

Dieser Abschnitt behandelt die Verifikation des HiL-Testsystems, welches gemäß Kapitel 3.2.2 nach Abbildung 3.15 aufgebaut wurde. Die HiL-Schnittstellenbeschreibungen sowie die genaue Pinbelegung kann den Tabellen im Anhang, in Kapitel 8.3, entnommen werden. Das vollständige Verifikationsdokument über den Prüfaufbau wurde dem Betreuer zur Verfügung gestellt [29].

Anhand der Verifikation wird der Nachweis erbracht, dass das erstellte HiL-Testsystem bestehend aus Software und Hardware fehlerfrei funktioniert und die vorgegebenen Anforderungen, welche im Testdesign verankert sind, erfüllt. Dazu ist es notwendig, Fehlfunktionen zu simulieren. Die jeweilige Fehlfunktion kann durch folgende drei Ereignisse beeinflusst werden:

- Veränderung des Prüflings (MIB Firmware)
- Unterbrechung der Schnittstellenverbindungen
- Veränderung der Eingangssignale in TestStand

Um den Verifikationsaufwand ressourcenschonend durchzuführen, wurden Äquivalenzklassen gebildet. Diese umfassen Testfälle, welche in vergleichbarer Weise funktionieren. Falls ein Fehler in einem Testfall der Äquivalenzklasse auftritt, so ist es aufgrund der Äquivalenzklassenbildung sichergestellt, dass dieser dann auch in den anderen Testfällen der Klasse fehlschlägt. Falls kein Fehler bei dem jeweiligen Testfall der Äquivalenzklasse auftritt, wird erwartet, dass auch alle anderen Testfälle derselben Äquivalenzklasse keine Fehler entdecken. Wenn die erzeugten Fehler von dem HiL-Testsystem erkannt werden, kann die Aussage getroffen werden, dass das Testsystem verifiziert ist und als Prüfumgebung verwendet werden kann.

Für die Verifikation der Prüfumgebung werden in einem ersten Schritt Verifikationshilsmittel (hier MIB-Firmware) verändert. Dazu wurden die Anforderungen an drei spezielle Prüflinge gemäß Tabelle 4.1 definiert.

Tabelle 4.1: Auflistung der Prüfmuster für den Nachweis der Wirksamkeit des HiL-Testsystems

Prüfling	Beschreibung	Kommentar
<b>MIB-PCB1</b>	M0340030802	MIB V5 mit FW-Set 1.00 "Gutmuster": keine Veränderung der FW
<b>MIB-PCB2</b>	M0340030802 MCF Modifizierungen <ul style="list-style-type: none"> <li>• FW-Name: MCF-HIL-Test-PCB2</li> <li>• HW Identifier geändert auf 0x5D</li> <li>• HW Identifier geändert auf 0x5D</li> <li>• HW Version geändert auf 0x00</li> <li>• FW Version geändert auf 0.3.0</li> <li>• IDS-3192: Controller Status: MIB-TLC-Controller Status (Register Addr 0x04: Bit 6 konstant auf 1 setzen)</li> <li>• Detailed FW Version "enabled"</li> <li>• Default: Lautstärke auf 0% geändert</li> <li>• Default: OSD (Register Addr 0x23) geändert auf blue</li> <li>• Default: OSD Helligkeit auf 0% geändert</li> <li>• Default: Test Sequence Control (Reg Addr. 0x25) auf 0x01 geändert</li> <li>• Default: OSD Alarm Environment (Reg Addr. 0x26) auf 0x07 geändert</li> <li>• Default: Backlight Brightness (Reg Addr. 0x27) auf 0% geändert</li> <li>• Default: Loudspeaker Amplifier Mute (Reg Addr. 0x29) geändert auf "unmuted"</li> </ul> MSF Modifizierungen <ul style="list-style-type: none"> <li>• FW-Name: MSF-HIL-Test-PCB2</li> <li>• Detaillierte FW-Version-Informationen werden gesendet (Receive Register 1)</li> <li>• Geänderte FW-Version auf 0.3.0 (Transmit Register 2)</li> </ul> TCF Modifizierungen <ul style="list-style-type: none"> <li>• FW-Name: TCF-HIL-Test-PCB2</li> <li>• Geänderte FW-Version auf 0.3.0</li> <li>• Detaillierte FW-Version-Informationen werden gesendet</li> </ul>	MIB V5 mit FW-Set 1.00 "Schlechtmuster": Veränderung der FW, um "Fail"-Verhalten zu überprüfen
<b>MIB-PCB3</b>	M0340030802 MCF Modifizierungen <ul style="list-style-type: none"> <li>• Default: Reset Control (Reg Addr. 0x20) geändert auf 0x08</li> <li>• FW-Name: MCF-HIL-Test-PCB3</li> <li>• Besitzt sämtliche Änderungen von MIB-PCB2</li> </ul>	Enthält alle Änderungen von MIB-PCB2 und zusätzlich die Änderungen der Spalte "Beschreibung" von MIB-PCB3 "Schlechtmuster": Veränderung der FW, um "Fail"-Verhalten zu überprüfen

---

Hierzu wurde auf drei identisch bestückte MIB-Leiterplatten (MIB-PCB) je drei unterschiedliche MIB-HiL Firmware Testversionen programmiert. Das erste Testmuster (MIB-PCB1) wird als "Gutmuster" bezeichnet, da es sich dabei um eine aktuell freigegebene MIB-Version, gemäß dem Schaltplan M0340030802 handelt. Dieser Prüfling besitzt die Firmware (FW) Version 1.00.

Um das HiL-Testsystem auch auf die ordnungsgemäße Erkennung von Fehlern zu überprüfen, wurde der Nachweis auch anhand von zwei "Schlechtmustern" (MIB-PCB2, MIB-PCB3) durchgeführt. Diese Schlechtmuster sind ebenfalls gemäß Schaltplan M0340030802 bestückt, aber sie besitzen eine veränderte FW-Version, deren absichtlich eingebauten Fehler aber dazu führen, dass bestimmte Testfälle fehlschlagen. Das HiL-Testsystem muss die Änderungen durch Fehlschlagen des zugehörigen Testfalls anzeigen.

Da nicht alle Testfälle durch Variation der Verifikationshilfsmittel überprüft werden können, gibt es Testfälle für die Verifikation, die einen Fehler mit einer Unterbrechung der HiL-Schnittstellenverbindung simulieren. Auf diese Weise kann zum Beispiel die Detektion einer nicht funktionierenden OSD LED simuliert werden. Ein anderes Beispiel ist die Abfrage der Datenwiederholungsrate der Touch-Position bei einer unterbrochenen Verbindung der COM-TOUCH-Schnittstelle (P12 - siehe Anhang Tabelle 8.21) zum TLC. In diesen Fällen muss der Testfall einen Fehler erkennen und dies durch ein Nicht-Bestehen des Testfalls anzeigen.

Im dritten und letzten Schritt gibt es Testfälle für die Verifikation, in welchen die Eingangssignale von TestStand verändert werden und überprüft wird, ob der Test ordnungsgemäß fehlschlägt. Dazu wurden verschiedene Testfälle entwickelt, um die mit dem HiL-Testsystem verbunden Funktionen zu überprüfen. Mit Hilfe von veränderten Eingangssignalen und Eingangsparametern gemäß der nachfolgenden Auflistung kann eine Reihe der Tests abgedeckt werden:

- MCF: Alternatives Setzen der Hardkey-Hinterleuchtung
- MCF: Alternative Aktivierung eines simulierten Hardkeys
- MCF: Modifikation des eingespeisten Audiosignals (Frequenz)
- MCF: Alternatives Aktivieren der Signalfarbe der OSDs
- MCF: Alternative Aktivierung der Blinkfrequenz der OSDs
- MCF: Alternative Helligkeitsansteuerung der OSDs



- MCF: Verifizierung der Fehlererkennung des Softwareprotokolls bei Übertragungsfehlern durch Übertragung geeigneter Prüfnachrichten
- MCF: Alternative Aktivierung/Deaktivierung eines externen MIB-Reset
- MCF: Alternative Aktivierung/Deaktivierung von MIB-TLC Schnittstellensignalen durch entsprechendes Setzen der Spannungsversorgung
- MSF: Alternatives Setzen der Mute-Funktion des Lautsprechers
- MSF: Alternative Übertragung der Resetsequenz
- MSF: Alternatives Aktivieren der roten OSDs
- MSF: Alternative Aktivierung der Blinkfrequenz der roten OSDs
- MSF: Modifikation des eingespeisten Audiosignals (Frequenz)
- MSF: Alternatives Setzen des AQ-Hardkey-Hinterleuchtung
- TCF: Alternatives Setzen der Touchposition

Zusätzlich zu den zuvor genannten drei Schritten kann man durch Setzen von geeigneten Breakpoints<sup>13</sup> in TestStand aktuell am Prüfling anliegende Schnittstellen-Signale manuell mit dem Multimeter oder Oszilloskop messen. Somit kann verifiziert werden, ob das HiL-Testsystem mit TestStand und die manuelle Messung vergleichbare Messwerte liefert. Vergleichbar in diesem Zusammenhang bedeutet, dass der TestStand Messwert innerhalb eines Toleranzbereichs um den manuell gemessenen Wert liegt.

Anhand der beiden Testfälle 99 und 119 soll im Folgenden veranschaulicht werden, wie das HiL-Testsystem verifiziert wird.

- *Testfall 99 (MIB-MSF TC 99 - FW Version (Transmit Reg 2)) - Schritt 7:*

Um zu zeigen, dass Testfall 99 ordnungsgemäß funktioniert, wird der Test zudem mit dem MIB-"Gutmuster" durchgeführt und an diesem die Firmwareversion 0x40 ausgelesen. Wird der Test jedoch mit dem MIB-"Schlechtmuster" (MIB-PCB2) durchgeführt, sollte der Testfall fehlschlagen, weil dieser Prüfling eine definiert falsche Firmware Version hat. Als Ergebnis der beiden Tests folgt, dass beim "Gutmuster" die korrekte FW-Version, beim "Schlechtmuster" aber die fehlerhafte FW-Version korrekt als falsch erkannt wird. Dieser Test ist somit korrekt verifiziert.

---

<sup>13</sup> Haltepunkte in der TestStand-Sequenz

- 
- *Testfall 119 (MIB-TCF TC 119 - MIB-Touch Position 1) - Schritt 7:*

In der Testsequenz in TestStand gibt es drei Testfälle, die eine Touch-Position simulieren (119, 120, 121) und die Richtigkeit der übertragenen Position überprüfen. Aufgrund der Tatsache, dass die Testschritte bis auf das unterschiedliche Setzen der Touch-Position identisch sind, können diese drei Testfälle zu einer Äquivalenzklasse zusammengefasst werden.

Die Verifikation erfolgte für die "richtig positiv"- und "richtig negativ"-Beurteilung des Testfalls mit dem MIB-"Gutmuster". Es kann hier mit einem Verifikationshilfsmittel (MIB-PCB1) gearbeitet werden, da das Eingangssignal, also die analoge Position des Touch-Berührungspunkts, verändert wird. Für die "richtig positiv"-Beurteilung wird der Test laut Testdesign durchgeführt. Für die "richtig negativ"-Beurteilung wird das analoge Signal, welches für die Simulation der Touch-Position verantwortlich ist (5-Wire-Sense) verändert und überprüft, ob ein Fehler detektiert wird. Dazu ist es notwendig das analoge Signal der Touch-Position soweit zu verändern, dass die digital empfangene Touch-Position sich nicht im Toleranzbereich des Akzeptanzkriteriums befindet.

Alle Schritte konnten erfolgreich durchgeführt werden. Somit konnte der Nachweis eines korrekt funktionierenden Testdesigns nachgewiesen werden, welches die an das Design gestellten Anforderungen vollständig erfüllt.



---

## 5 Diskussion und Ausblick

Zielsetzung der vorliegenden Arbeit war es, eine Testplattform bestehend aus IBM RQM, NI TestStand und NI LabVIEW nach dem HiL-Prinzip aufzubauen. Dieses Ziel wurde im Zuge dieser Arbeit erreicht. Nach der Signalanalyse, der Kozeptionierung eines Lösungskandidaten, der Erstellung eines Verifikationskatalogs, des sogenannten Testdesigns, der Implementierung und Verifikation im Zuge des Nachweises der Wirksamkeit, ist ein Software-Framework entstanden, welches eine End-to-End Traceability von den Anforderungen bis hin zu den einzelnen Testfällen ermöglicht. Es wird anhand der medizinischen Elektronikkomponente MIB gezeigt, dass die Kommunikation der verschiedenen Softwarekomponenten nahtlos miteinander möglich ist. Der folgende Abschnitt diskutiert neben der Umsetzung der Verifikationsplattform auch während des Verlaufs dieser Arbeit entstandene neue Ideen und Verbesserungsvorschläge.

Eine wesentliche Kernkomponente der Verifikationsplattform ist der von NI verwendete NI Test Integrations Adapter. Über diesen Adapter ist es möglich, die IBM Komponente RQM mit den NI Komponenten TestStand und LabVIEW zu vereinen. Erst dadurch kann eine lückenlose Traceability von den Anforderungen bis hin zu den einzelnen Testfällen von LabVIEW, welche über TestStand zu einer Testsequenz formiert werden, erreicht werden.

Mit Hilfe dieser Verifikationsplattform werden nur die funktionellen hardwareseitigen Anforderungen abgedeckt. Anforderungen, welche zum Beispiel die chemische Beschaffenheit, die elektromagnetische Verträglichkeit oder die Mechanik betreffen, werden nicht von der realisierten Plattform abgedeckt. Durch Hinzufügen von weiteren Attributen in DOORS können diese jedoch in zukünftigen Projekten als Grundlage für eine Erweiterung der Testfälle herangezogen werden. Bei einer Anpassung der Anforderungen ist es notwendig, dass Testfälle in RQM gegebenenfalls neu definiert werden. In diesem Fall müssen dann zusätzlich parallel auch Testfalldesign, Akzeptanzkriterien und Anforderungslinks aktualisiert werden. Diese Tests können auch unabhängig von LabVIEW zum Beispiel durch manuelle Tests implementiert werden.

Durch die Verwendung von TestStand ist es möglich, die Testimplementierung sehr modular zu gestalten. Erstellte LabVIEW VIs können mehrfach verwendet werden. Durch die Verwendung von klar-definierten und unabhängigen LabVIEW Programmen ist ein einfacher Einsatz möglich, wobei kleinere Subprogramme eine einfachere Wartung ermöglichen.

Derzeit dauert die vollständige Überprüfung des Prüflings (MIB) 20 Minuten. Die Verifikationsdauer wird dabei im wesentlichen von den nachfolgenden Faktoren beeinflusst:

- Wartezeiten, die bei der Kommunikation zwischen RQM und TestStand entstehen;
- Wartezeiten, die dazu dienen, einen definierten Ausgangspunkt zu Beginn des Tests zu erreichen;
- Durchlaufzeiten des Tests, die aufgrund der Durchführung von kontinuierlichen Messungen über einen definierten Zeitraum entstehen.

Da die Datenübertragungszeit von TestStand zu RQM nicht weiter verbessert werden kann und jeder Test eine gewisse Durchlaufzeit benötigt lässt sich die Performanz nur durch Clustering von Testfällen sowie durch einen optimierten Einsatz des zu Beginn eines Tests durchgeführten Resets, der am Prüfling befindlichen Mikrocontroller, durch die Spannungsversorgung erreichen. Derzeit ergibt sich alleine durch die in den Testfällen implementierten Reset, hervorgerufen durch die Spannungsversorgung eine unabhängig von den restlichen Testinhalten benötigte Testdauer von acht Minuten.

Derzeit bietet RQM keine Möglichkeit das Testdesign in einem leicht weiter zu verarbeitenden Dokumenttyp (zum Beispiel Word oder PDF) auszuleiten. Daher muss das Testdesign in einem externen Dokument gepflegt werden, sodass dadurch mindestens zwei parallel zu führenden Dokumente entstehen. Um diesen Mehraufwand zu vermeiden, muss ein Testdesign Modul zum Publishing erworben und angewendet werden.

Für eine ordnungsgemäße Verwendung der Verifikationsplattform zum Test des Prüflings (MIB) ist die Verwendung eines Adapters zwischen dem Prüfling (MIB) und der NI-Hardware erforderlich, da das verwendete NI-Modul nicht über analoge Ein- und Ausgänge verfügt. Optional kann anstelle dieses Adapters auch eine andere NI-Messkarte gewählt werden, wobei bei der Entscheidung für den Adapter oder die NI-Messkarte die vorhandenen Ressourcen unter anderem Zeit, Funktionalität, Wartungsaufwand, Wiederverwendbarkeit und Kosten, berücksichtigt werden müssen.

---

Der erstellte Aufbau dient als Grundlage für kommende Teststrategien und Testumsetzungen im Unternehmen B. Braun. Zukünftig soll diese Verifikationsplattform als Grundlage für weitere HiL-Projekte dienen und dessen Funktionalität weiter ausgebaut werden. Unter anderem sollen Coverage-Analysen selektiv für jedes Anforderungsmodul (zum Beispiel: IDS, CRS) und die Testfälle einer medizinischen Elektronikkomponente durchgeführt werden können. Außerdem können die Auswirkungen auf die Testfälle aufgrund von Anforderungsänderungen analysiert und die Sicherstellung der Traceability zwischen Anforderungen und Tests gewährleistet werden.

HiL-Systeme werden in der Industrie bereits sehr erfolgreich eingesetzt. Ein Alleinstellungsmerkmal dieser Verifikationsplattform ist allerdings die durchgängige Traceability von den Anforderungen bis hin zu den Testfällen mit einer in sich geschlossenen "Toolchain". Dabei werden die Ergebnisse des in TestStand durchgeführten Testfälle automatisch an RQM in Form eines Reports weitergeleitet, wobei dessen Inhalte jederzeit an die Bedürfnisse des Anwenders angepasst werden können, um so diesen von administrativen Tätigkeiten zu entlasten.

Zusammenfassend lässt sich sagen, dass diese Verifikationsplattform manuell durchgeführte zeitaufwendige und damit fehleranfällige Tests durch automatisierte HiL-Tests ersetzt, wobei neben der automatischen Generierung von auditsicheren Testreports auch die Traceability der Anforderungen mit den Testfällen sichergestellt ist.



---

## 6 Zusammenfassung

Zusammenfassend lässt sich sagen, dass die in der Aufgabenstellung formulierten Punkte vollständig abgearbeitet wurden und das Ziel, eine Testplattform, bestehend aus IBM RQM, NI TestStand und NI LabVIEW, nach dem HiL-Prinzip aufzubauen, erreicht worden ist.

Um eine Entwicklung und Evaluation einer automatischen Verifikationsplattform für medizinische Elektronikkomponenten durchzuführen, ist es zunächst notwendig, sich ein umfangreiches Wissen über die zu verifizierende Elektronikkomponente anzueignen. Im Anschluss wird der Entwurf für die Verifikationsplattform erstellt. Dazu müssen die Anforderungen aus der Anforderungsmanagement Datenbank DOORS auf die funktionell testbaren Anforderungen gefiltert werden. Für die Filterung ist nicht nur entscheidend, dass es sich um funktionelle Anforderungen handelt, sondern auch, dass sie auf elektronischer Ebene testbar sind. Diese Anforderungen werden in der vorliegenden Arbeit mit dem Attribut *Testable* in DOORS versehen, sodass im Anschluss eine spezifische Sicht (*MIB-Verification*) erstellt werden kann. Nach der Analyse der Anforderungen an die Prüfumgebung kann nach dem V-Modell eine Prüfumgebung erstellt werden.

Zunächst ist dafür die Erstellung des Testplans in RQM erforderlich. Die Basis dieses Testplans bilden die aus DOORS gefilterten Anforderungen. Jeder Testfall des Testplans wird einer Testsuite zugeordnet, um funktionell zusammengehörige Tests zu clustern. Nachdem das Testfalldesign und die Akzeptanzkriterien für jeden Test definiert worden sind, können die Anforderungen aus DOORS über Anforderungslinks mit RQM verknüpft werden.

Damit bestimmte Funktionen des Prüflings mit der NI-Hardware getestet werden können, war es im Zuge dieser Arbeit notwendig, eine Adapterleiterplatte zu verwenden. Dazu ist es im nächsten Schritt notwendig die Anforderungen an das Messsystem festzuhalten. Informationen aus DOORS und dem Testdesign liefern die entsprechenden Anforderungen an diesen Adapter. Dieser wurde entsprechend den Anforderungen innerhalb von B. Braun realisiert. Da die verwendete NI-Hardware (NI PXI-7813R) als solches nur digitale Ein- und Ausgänge zur Verfügung stellt, besitzt dieser Adapter A/D-Wandler für das Messen von analogen Signalen und D/A-Wandler für die Ausgabe von analogen Signalen.



Außerdem muss eine Signalanpassung anhand von Pegelwandlern durchgeführt werden, da die NI-Hardware lediglich Low-Level-TTL-Signale mit 3.3V Pegel ausgeben kann. Weiterhin ist es erforderlich, die seriellen RS232-Schnittstellensignale zur Kommunikation zwischen dem TCF und dem TLC sowie zwischen dem MSF und dem LLS mit Hilfe von RS232-to-USB-Konvertern umzuwandeln, da die NI-Hardware, ebenso wie handelsübliche Computer, nicht eine ausreichende Anzahl von physikalischen RS232-Schnittstellen aufweisen.

Nachdem die Anforderungen gefiltert, das Konzept aufgestellt und ein Adapter für die Verifikationsplattform erstellt worden ist, erfolgt die Design-Implementierung der Plattform, wobei das in RQM ausformulierte Testdesign in Testschritten einer TestStand-Sequenz realisiert wird. Diese Testschritte greifen auf LabVIEW-Code-Module zu, wobei es sich dabei um modular und vielseitig-einsetzbare LabVIEW-Code-Module handelt. Über die Ausführungsvariable `TestVariable` wird eine Traceability von RQM bis hin zu TestStand und in weiterer Folge LabVIEW erreicht.

Im Anschluss an die Implementierung wird die Prüfumgebung einer Verifikation unterzogen, um den Nachweis der Wirksamkeit des HiL-Testsystems zu erbringen. Dabei reicht es nicht aus die positive Funktion des Testsystems nachzuweisen, viel mehr muss nachgewiesen werden, dass die Prüfumgebung auch Fehler korrekt erkennt. Dafür ist es notwendig, zusätzliche Testfälle zu erstellen, die überprüfen, ob das HiL-Testsystem "richtig positiv" und "richtig negativ"- Beurteilungen des Testfalls durchführen kann. Für die Verifikation des HiL-Testsystems muss es wie spezifiziert, bestehend aus Hardware und Softwarekomponenten, aufgebaut werden. Die Spannungsversorgung, der Adapter, der Prüfling und die NI-Hardware bilden die Hardwarekomponenten. Softwarekomponenten, bei welchen es entscheidend ist einen definierten Stand aus dem Versionsmanagement zu benutzen, umfassen TestStand und LabVIEW. In dieser Arbeit erfolgt die Verifikation anhand des B. Braun MIBs. Bei dem MIB handelt es sich um eine Leiterplatte, welche zum Informationsaustausch zwischen Anwender und der Dialysemaschine NDS dient. Die Verifikation ist sowohl bei den "richtig positiv"- als auch bei den "richtig negativ"- Beurteilungen erfolgreich durchgeführt worden.

Das erstellte HiL-Software-Framework bindet die reale Hardwarekomponenten MIB in eine simulierte Umgebung ein und erreicht eine nahtlose Traceability, welche von den Anforderungen bis hin zum implementierten LabVIEW Testfall führt und somit zeigt, dass das Ziel dieser Arbeit vollständig erreicht wurde.

---

## 7 Literaturverzeichnis

- [1] Abourida S, Dufour C, Bélanger J: **Real-Time and Hardware-In-The-Loop Simulation of Electric Drives and Power Electronics: Process, problems and solutions.** *Proceedings of the International Power Electronics Conference (IPEC-Niigata)*, (2005).
- [2] Alexander H, Breuch G: **Fachpflege Nephrologie und Dialyse.** *Elsevier, Urban&FischerVerlag* (2008).
- [3] Automation Consultants: **Rational Quality Manager.** <http://www.automation-consultants.com/index.php/products/ibm-products/rational-quality-manager> (Version: 2016) [Zugriff: 05.04.2016].
- [4] B. Braun Avitum AG: **Broschüre Patienteninformation.** *Internes Dokument, Melsungen* (2015).
- [5] B. Braun Avitum AG: **CRSMCF - Component Requirement Specification Monitor Interface Board Controller Firmware, ID: NDS-0800 Version 6.0.** *Internes Dokument, Melsungen* (2016).
- [6] B. Braun Avitum AG: **CRSMIBHW - Component Requirement Specification Monitor Interface Board Hardware, ID: NDS-0803 Version 5.0.** *Internes Dokument, Melsungen* (2016).
- [7] B. Braun Avitum AG: **CRSMSF - Component Requirement Specification Monitor Interface Board Supervisor Firmware, ID: NDS-0798 Version 6.0.** *Internes Dokument, Melsungen* (2016).
- [8] B. Braun Avitum AG: **CRSTCF - Component Requirement Specification Touch Panel Interface Board Controller Firmware, ID: NDS-0792 Version 6.0.** *Internes Dokument, Melsungen* (2016).
- [9] B. Braun Avitum AG: **Dialog iQ Challenge the Thinking.** *Internes Dokument, Melsungen* (2016).
- [10] B. Braun Avitum AG: **Dialog iQ Dialysis Machine Service Manual SW 1.00.00 EN.** *Internes Dokument, Melsungen* (2015).
- [11] B. Braun Avitum AG: **IDS - Interface Design Specification, ID: NDS-0492 Version 6.0.** *Internes Dokument, Melsungen* (2016).

- [12] B. Braun Melsungen AG: **Welcome to B. Braun - A Company Portrait 2015**. <https://www.bb Braun.de/de/unternehmen/organisation-zahlen-und-fakten.html> (Version: 2015) [Zugriff: 29.02.2016].
- [13] Broy M, Rausch A: **Das neue v-modell® xt**. *Informatik-Spektrum, Springer*, (2005).
- [14] Hörl WH, Koch KM, Lindsay RM, Ronco C, Winchester JF: **Replacement of renal function by dialysis**. *Kluwer Academic Publishers* (2004).
- [15] IBM: **Anforderungsmanagement für Systeme und innovative IT-Anwendungen - Rational DOORS**. <http://www-03.ibm.com/software/products/de/ratidoor> (Version: 2016) [Zugriff: 05.04.2016].
- [16] IBM: **Rational Quality Manager**. [http://www.ibm.com/developerworks/rational/library/08/1216\\_kelly/index.html](http://www.ibm.com/developerworks/rational/library/08/1216_kelly/index.html) (Version: 2008) [Zugriff: 05.04.2016].
- [17] International Electrotechnical Commission: **IEC 60601-1-8: Medical electrical equipment. Part 1-8: General requirements for basic safety and essential performance. Collateral standard: General requirements, tests and guidance for alarm systems in medical electrical equipment and medical electrical systems**. Edition 2.0 (2006-10).
- [18] Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung (KBSt): **Entwicklungsstandard für IT-Systeme des Bundes: Vorgehensmodell (V-Modell). Regelungsteil**. *Bundesanzeiger* (1997).
- [19] Koordinierungs- und Beratungsstelle für die Informationstechnik in der Bundesverwaltung: **Planung und Durchführung von IT-Vorhaben: Vorgehensmodell**. *Bundesanzeiger* (1992).
- [20] Langner F, Alsmann U, Meyer J: **Fahrversuch versus HiL-Test, High-Tech Unterstützung für die Qualitätssicherung durch HiL-Testsysteme im Vergleich zu Entwicklungsfahrzeugen**. *berner & mattner optimizing your development - white paper*, (2008).
- [21] Lu, Bin and Wu, Xin and Figueroa, Hernan and Monti, Antonello: **A low-cost real-time hardware-in-the-loop testing approach of power electronics controls**. *IEEE Transactions on Industrial Electronics*, (2007).
- [22] Microchip Technology Inc.: **AR1000 Series Resistive Touch Screen Controller**. Datenblatt. <http://ww1.microchip.com/downloads/en/DeviceDoc/40001393C.pdf> (Version: 2016) [Zugriff: 17.03.2016].

- 
- [23] Moeller D: **Production Specification - MIB V5: Production Data (Schematic and Layout and Parts List) - 2016-03-16 Status 49 - 2013-03-16**. *Internes Dokument B. Braun Avitum AG 04.03-2017 [V03], Melsungen* (2016).
- [24] National Instruments Corporation: **LabView**. <http://www.ni.com/labview/d/> (Version: 2016) [Zugriff: 05.04.2016].
- [25] National Instruments Corporation: **NI Test Integration Adapter for IBM Rational Quality Manager**. <http://www.ni.com/pdf/manuals/375182b.html> (Version: 2016) [Zugriff: 05.04.2016].
- [26] National Instruments Corporation: **TestStand**. <http://www.ni.com/teststand/d/> (Version: 2016) [Zugriff: 05.04.2016].
- [27] Schönweiß G: **Dialysefibel 3**. *Abakiss Verlag* (2006).
- [28] Schwarz RF: **DAS V-MODELL® XT**. <ftp://ftp.heise.de/pub/ix/projektmanagement/vmodell/V-Modell-XT-Gesamt-Deutsch-V1.3.pdf> (Version: 2006) [Zugriff: 15.02.2016].
- [29] Wallner K: **Test Specification (TS) - Equipment Qualification MIB HiL - 2.00**. *Internes Dokument B. Braun Avitum AG 06.05-0647 [V02], Melsungen* (2016).
- [30] Wallner K: **Test Specification (TS) - HiL MIB v4a FWs 1.00 - 1.00**. *Internes Dokument B. Braun Avitum AG 06.03-3327 [V01], Melsungen* (2016).



---

## 8 Anhang

Der Anhang wurde den Betreuern und Begutachtern ausgehändigt, da er vertrauliche Informationen und Daten des Unternehmens B. Braun Avitum AG beinhaltet.