

Analyse des Anlagenverhaltens

Inhalt

1	Dokumentorganisation.....	1
1.1	Autorenliste.....	1
1.2	Versionen.....	1
2	Vision	1
3	Requirementsanalyse	2
4	Werkstückbehandlung	3
5	Use-Cases	4
6	Embedded Player Spezifikation	12
7	Hardware Abstraction Layer - Klassendiagramm	14
8	Fehlersignalisierung und Quittierung.....	15

1 Dokumentorganisation

1.1 Autorenliste

Kürzel	Name
DS	Dennis Sentler
AH	Ahmed Hussein
DK	Daniel Kessener
MB	Martin Brak

1.2 Versionen

Version	Erstellt	Autor	Kommentar
0.1	16.04.18	DS	Vision geschrieben
0.2	16.04.18	DS, MB, DK, AH	Use-Cases und Werkstückbehandlung inkludiert
0.3	19.04.18	DS, DK	Mit den Diag. HAL, EMP und Fehlerquittierung ergänzt
0.4	22.04.18	DS	Mit kurzen Texten die Diagramme ergänzt

2 Vision

Eine eingebettete Softwarestruktur soll für eine Sortieranlage bestehend aus zwei Fließbändern entwickelt und implementiert werden. Dabei müssen Werkstücke, die auf das Fließband des Systems gelegt werden erkannt und am Ende des Prozesses in einer vorgegebenen Reihenfolge am zweiten Fließband herausgegeben werden. Die nicht der Reihenfolge entsprechenden Werkstücke, oder Werkstücke die nicht erkannt werden, müssen dabei aussortiert werden.

In einem realen Arbeitsprozess wie diesem können Situationen entstehen die nicht dem Musterablauf entsprechen. Deswegen muss das System Fehlerfälle erkennen und auch sinnvoll mit diesen umgehen.

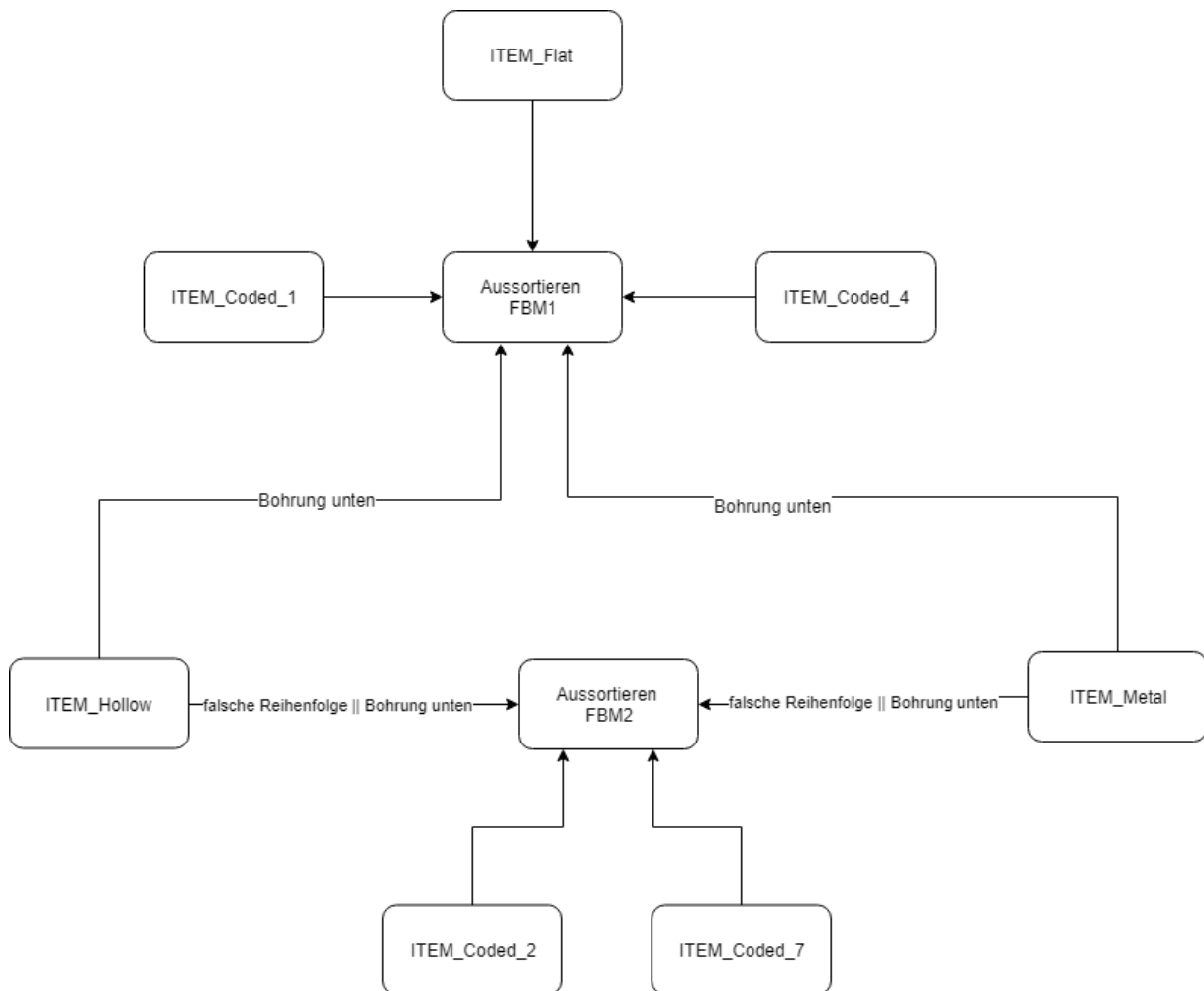
3 Requirementsanalyse

ID	Titel	Bezug	Requirement
R-01	Transportieren	-	Das System ist in der Lage ein Item, das auf FB1 aufgelegt wird, bis ans Ende von FB2 zu transportieren.
R-02	Portionierung	20	Ein Item darf nur auf FB2 überführt werden, wenn FB2 leer ist.
R-03	Sortieren	1, 13	Das System sortiert auf FB2 Items so aus, dass die Items, die am Ende von FB2 ankommen, einer prädefinierten Ordnung entsprechen.
R-03	ITEM_Flat	11	ITEM_Flat werden auf FB1 aussortiert.
R-04	ITEM_UpsideDown	12	ITEM_UpsideDown werden aussortiert.
R-05	ITEM_Coded_{1,4}	14	ITEM_Coded_{1,4} werden auf FB1 aussortiert.
R-06	ITEM_Coded_{2,7}	15	ITEM_Coded_{2,7} werden auf FB2 aussortiert.
R-07	Langsame Höhenmessung	23	Während der Höhenmessung laufen die FB langsam.
R-08	Item IDs	30	Das System vergibt eine ID an ein Item sobald dieses auf FB1 aufgelegt wird.
R-09	Anhalten von FB2	20, 24	Wenn ein Item das Ende von FB2 erreicht, hält dieses an, bis das Item entfernt wurde.
R-10	Informationsübersicht Item	25, 26, 27, 28, 29	Wenn ein Item das Ende von FB2 erreicht, werden ID, Typ, und die Höhenmesswerte beider FB auf der Konsole ausgegeben.
R-11	Info für ITEM_Coded_*	32, 33, 34, 35, 36	Wenn ein ITEM_Coded_* erkannt wird, werden Zeitstempel der Erkennung, ID, Binärcode und Höhenmesswert auf der Konsole ausgegeben.
R-12	Strom sparen	37	Wenn ein FB leer ist, hält es an.
R-13	Teilen der Rutschenkapazität	38, 39	Wenn die Rutsche eines FB voll ist, gelten FB-bezogene Sortierregeln nicht mehr; alle auszusortierenden Items werden von dem anderen FB aussortiert.
R-14	Schadensprävention der Weiche	44, 45	Die Weichen der FB dürfen nicht länger als ein paar Minuten am Stück geöffnet werden.
R-15	Replay	91	Das System kann aufgezeichnete Sensor-Daten einlesen und so einen Ablauf simulieren.
R-15	Schnellabschaltung	58, 59	Das Drücken des E-Stopp Schalters führt zum sofortigen Stillstand des Systems.
R-16	Betriebswiederaufnahme	60	Nach einer Schnellabschaltung bleibt das System stehen bis der E-Stopp Schalter wieder herausgezogen wird und eine der RESET Tasten gedrückt wurde.
R-17	Fehlendes Item	48	Das System erkennt, wenn ein Item unplanmäßig vom FB entfernt wurde.
R-18	Unerwartetes Item	49	Das System erkennt, wenn ein Item unplanmäßig auf ein FB gelegt wurde.
R-19	Rutschen voll	50	Das System erkennt, wenn beide Rutschen voll sind.

Mithilfe dieser Requirmentsanalyse kann immer überprüft werden ob das implementierte Systemverhalten der Planung entspricht und es kann gezielt gegen diese Requirements getestet werden.

Hinweis: Eine ausführlichere Variante mit zusätzlichen Infos finden Sie als Excel Tabelle im Verzeichnis work/design/Requirements.xlsx.

4 Werkstückbehandlung



Dieses Flow-Chart stellt dar, wie die einzelnen Bausteine der entsprechenden Kategorien (ITEM_Flat/Metal/Hollow/Coded_1/Coded_4/Coded_2/Coded_7) in dem finalen System behandelt werden sollen und falls sie aussortiert werden müssen, an welchem Modul dies geschehen muss.

5 Use-Cases

UC-000	
Name	Ein ITEM wird über die FB befördert
Autor	Kessener
Priorität	Critical
Auslöser	Ein ITEM wird auf FB1 gelegt
Akteure	Die gesamte Anlage, Nutzer
Vorbedingungen	<ul style="list-style-type: none"> - Das ITEM hat eine Bohrung auf der Oberseite - Die Rutsche von FB2 ist nicht voll
Ergebnis	Das ITEM erreicht das Ende von FB2 und kann abgeholt werden
Hauptszenario	<ol style="list-style-type: none"> 1. Ein ITEM wird vom Nutzer auf FB1 gelegt 2. FB1 befördert das ITEM durch die Weiche zum FB2 <ol style="list-style-type: none"> a. In der Höhenmessung wird das FB langsamer 3. FB2 befördert das ITEM zu seinem Ende <ol style="list-style-type: none"> a. In der Höhenmessung wird das FB langsamer 4. Auf der Konsole werden ID, Typ und Höhenmesswerte des ITEMS auf der Konsole ausgegeben 5. Das ITEM wird vom Nutzer entfernt
Alternativszenarien	<ol style="list-style-type: none"> 3. 1. Das ITEM entspricht nicht der gewünschten Reihung und __ wird durch die Rutsche von FB2 aussortiert 3. 2. Das ITEM hat sich überschlagen und wird durch die Rutsche __ von FB2 aussortiert
Ausnahmeszenarien	-

UC-001	
Name	ITEM_CODED wird sortiert
Autor	Kessener
Priorität	Standard
Auslöser	Ein ITEM_CODED wird auf FB1 gelegt
Akteure	Die gesamte Anlage, Nutzer
Vorbedingungen	- Beide Rutschen sind nicht voll
Ergebnis	Das ITEM_CODED wird identifiziert und korrekt platziert
Hauptszenario	<ol style="list-style-type: none"> 1. Das ITEM_CODED wird vom Nutzer auf FB1 gelegt 2. FB1 befördert das WS zur Höhenmessanlage <ol style="list-style-type: none"> a. In der Höhenmessung wird das FB langsamer 3. Das System identifiziert das ITEM als ITEM_CODED und gibt Zeitstempel, ID, Binärcode und Höhenmesswert des WS auf der Konsole aus 4. Der Binärcode des ITEMS ist nicht 1 oder 4 5. Das ITEM wird zu FB2 transportiert 6. FB2 befördert das ITEM zur Höhenmessanlage <ol style="list-style-type: none"> a. In der Höhenmessung wird das FB langsamer 7. Das System identifiziert das ITEM als ITEM_CODED und gibt Zeitstempel, ID, Binärcode und Höhenmesswert des ITEMS auf der Konsole aus 8. Der Binärcode des ITEM_CODED ist nicht 2 oder 7 9. Das ITEM_CODED wird zum Ende von FB2 transportiert und auf der Konsole werden ID, Typ und Höhenmesswerte ausgegeben 10. Das ITEM wird vom Nutzer entfernt
Alternativszenarien	<ol style="list-style-type: none"> 4. --- <ol style="list-style-type: none"> 1. Der Binärcode des ITEM_CODED ist 1 oder 4 2. Das ITEM_CODED wird durch die Rutsche von FB1 aussortiert 6. 1. Das ITEM hat sich überschlagen und wird durch die Rutsche __ von FB2 aussortiert 8. --- <ol style="list-style-type: none"> 1. Der Binärcode des ITEM_CODED ist 2 oder 7 2. Das ITEM_CODED wird durch die Rutsche von FB2 aussortiert

UC-002	
Name	FBM wechselt in den Betriebszustand
Autor	Brak
Priorität	Hoch
Auslöser	Start-Taste kurz gedrückt
Akteure	FBM, Nutzer
Vorbedingungen	- FBM befindet sich im Ruhezustand oder
Ergebnis	FBM befindet sich im Betriebszustand
Hauptszenario	<ol style="list-style-type: none"> 1. Nutzer drückt den Start Knopf einmal kurz 2. Das FBM wechselt in den Betriebszustand 3. Licht des FBM leuchtet grün 4. Sensoren senden Messwerte
Alternativszenarien	-
Ausnahmeszenarien	-

UC-003	
Name	FBM führt Kalibrierung und Selbsttests durch
Autor	Brak
Priorität	Standard
Auslöser	Start-Taste wird 3s lang gedrückt
Akteure	FBM, Nutzer
Vorbedingungen	- FBM befindet sich im Ruhezustand
Ergebnis	FBM hat Selbsttests durchgeführt und die Sensoren neu kalibriert
Hauptszenario	<ol style="list-style-type: none"> 1. Nutzer drückt die Start-Taste 3s lang 2. Das FBM wechselt in den Service-Mode 3. Licht des FBM blinkt grün 4. FBM führt Selbsttests und Kalibrierung durch 5. FBM wechselt in Ruhezustand 6. Grünes Licht des FBM wird ausgeschaltet
Alternativszenarien	-
Ausnahmeszenarien	-

UC-004	
Name	FBM wechselt in den Ruhezustand
Autor	Brak
Priorität	Hoch
Auslöser	Stop-Taste wird betätigt
Akteure	FBM, Nutzer
Vorbedingungen	<ul style="list-style-type: none"> - FB befindet sich im Betriebszustand - Es liegen keine Fehler oder Warnungen vor
Ergebnis	FBM befindet sich im Ruhezustand
Hauptszenario	<ol style="list-style-type: none"> 1. Nutzer drückt die Stop-Taste 2. Das FBM wechselt in den Ruhezustand 3. Licht des FBM wird ausgeschaltet 4. FB des FBM steht still 5. Sensoren senden keine Messwerte
Alternativszenarien	-
Ausnahmeszenarien	-

UC-005	
Name	E-Stopp des FBM wird betätigt
Autor	Brak
Priorität	Critical
Auslöser	E-Stop-Taste wird gedrückt
Akteure	Die gesamte Anlage, Nutzer
Vorbedingungen	- Mind. ein FBM der gesamten Anlage befindet sich im Betriebszustand
Ergebnis	Die gesamte Anlage ist abgeschaltet
Hauptszenario	<ol style="list-style-type: none"> 1. Nutzer drückt die E-Stop-Taste 2. Alle FBMe der Anlage werden abgeschaltet 3. Lichter aller FBMe in der Anlage schalten auf rot 4. Zustand aller FBMe wird gesichert 5. FB aller FBMe stehen still 6. Sensoren senden keine Messwerte
Alternativszenarien	<ol style="list-style-type: none"> 1. a. Nutzer zieht die E-Stopp-Taste wieder heraus
Ausnahmeszenarien	-

UC-006	
Name	Fehlerbehandlung
Autor	Brak
Priorität	Standard
Auslöser	FBM meldet einen Fehler
Akteure	FBM, Nutzer
Vorbedingungen	- FBM befindet sich im Betriebszustand
Ergebnis	Fehler ist behoben und FBM läuft weiter
Hauptszenario	<ol style="list-style-type: none"> 1. Ein Fehler im FBM tritt auf 2. FBM wechselt in Zustand "Anstehend unquittiert" 3. Licht des FBM blinkt rot (1 Hz) 4. Nutzer drückt Reset-Button 5. FBM wechselt in den Zustand "Anstehend quittiert" 6. Nutzer behebt den Fehler 7. FBM wechselt in Zustand "Ok"
Alternativszenarien	<ol style="list-style-type: none"> 6. 1. Fehler verschwindet von selbst 2. FBM wechselt in Zustand "Gegangen unquittiert"
Ausnahmeszenarien	<ol style="list-style-type: none"> 6. 1. Fehler lässt sich nicht beheben

UC-007	
Name	Ein ITEM wird von FB1 an FB2 übergeben
Autor	Brak
Priorität	Hoch
Auslöser	Item muss auf FB2 transportiert werden
Akteure	Die gesamte Anlage, Nutzer
Vorbedingungen	- ITEM hat die Weiche von FB1 passiert
Ergebnis	ITEM befindet sich auf FB2
Hauptszenario	<ol style="list-style-type: none"> 1. ITEM wird auf FB1 bis zu LB_End transportiert 2. Auf FB2 befindet sich kein ITEM 3. FB2 wird gestartet 4. ITEM wird FB2 übergeben
Alternativszenarien	<ol style="list-style-type: none"> 2. a. Auf FB2 befindet sich ein ITEM <ol style="list-style-type: none"> ___ 1. FB1 stoppt ___ 2. FB1 wartet bis sich kein ITEM mehr auf FB2 befindet ___ 3. FB1 wird gestartet
Ausnahmeszenarien	-

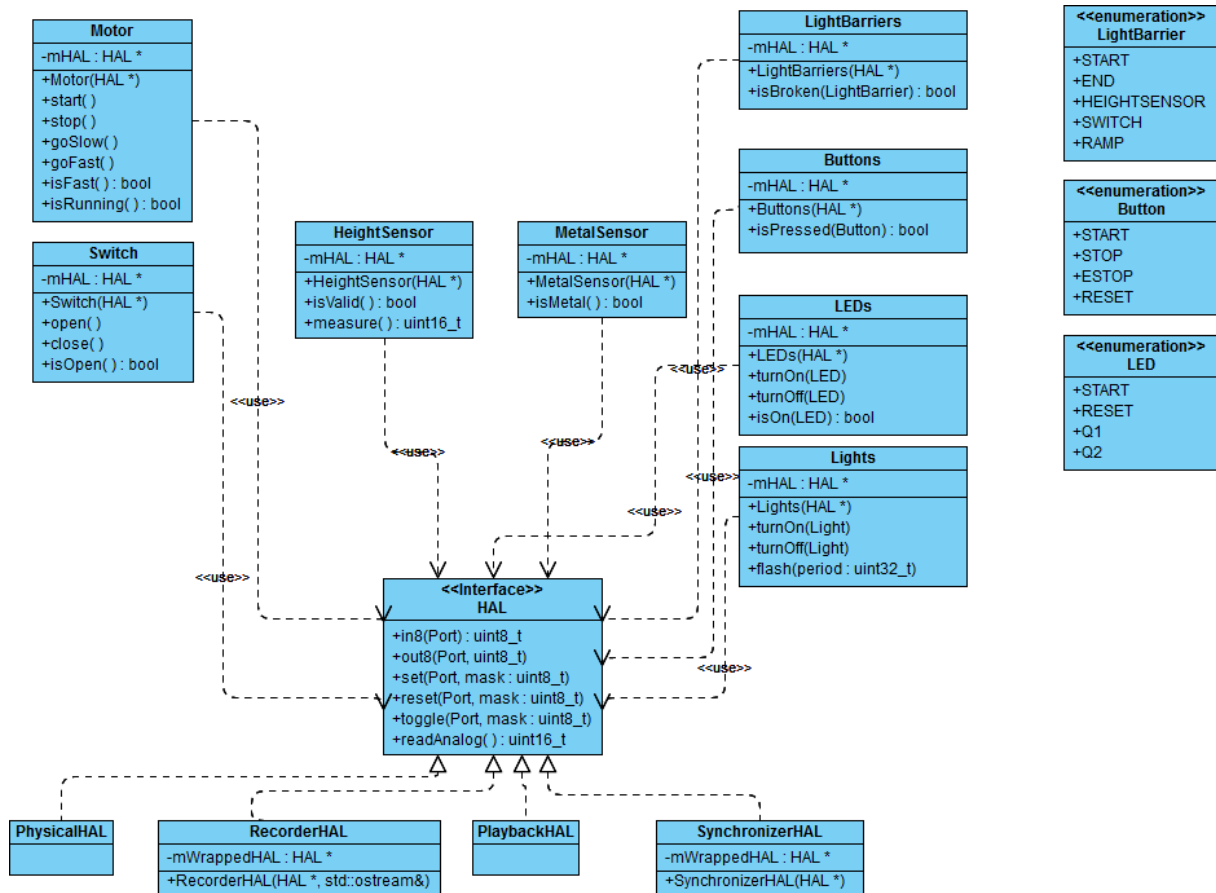
6 Embedded Player Spezifikation

Ein Embedded Player und Recorder wird hier spezifiziert. Dieser soll alle Vorgänge im System aufzeichnen und abspeichern

Spezifikation des EmbeddedPlayer (EMP) Format														
Version	1													
Beschreibung	<ol style="list-style-type: none"> 1. Das EMP Format ist ein menschenlesbares „plaintext“ Format 2. EMP ist Case-Insensitive 3. EMP Dateien bestehen aus einer Menge von Datensätzen 4. Ein Datensatz beschreibt eine Veränderung eines Sensors zu einem bestimmten Zeitpunkt 5. Jeder Datensatz ist in eine Newline-terminierten Zeile abgebildet 6. Ein Datensatz ist ein Tupel aus drei Whitespace separierten Werten (Timestamp, Sensor, Wert) (Siehe Tabelle A) 													
	<table> <tr> <th>Name</th><th>Beschreibung</th><th>Format</th></tr> <tr> <td>Timestamp</td><td>Menge von Millisekunden die seit Systemstart vergangen sind</td><td>Zahl im Dezimalsystem</td></tr> <tr> <td>Sensor</td><td>Der Sensor, dessen Wert sich geändert hat</td><td> LB_START LB_END LB_HEIGHTSENSOR LB_RAMP LB_SWITCH BUTTON_START BUTTON_STOP BUTTON_RESET BUTTON_ESTOP SWITCH_IN HEIGHTSENSOR_VALID HEIGHTSENSOR METALSENSOR </td></tr> <tr> <td>Wert</td><td>Der neue Wert des Sensors</td><td>Bei binärem Sensor:</td></tr> </table>	Name	Beschreibung	Format	Timestamp	Menge von Millisekunden die seit Systemstart vergangen sind	Zahl im Dezimalsystem	Sensor	Der Sensor, dessen Wert sich geändert hat	LB_START LB_END LB_HEIGHTSENSOR LB_RAMP LB_SWITCH BUTTON_START BUTTON_STOP BUTTON_RESET BUTTON_ESTOP SWITCH_IN HEIGHTSENSOR_VALID HEIGHTSENSOR METALSENSOR	Wert	Der neue Wert des Sensors	Bei binärem Sensor:	
Name	Beschreibung	Format												
Timestamp	Menge von Millisekunden die seit Systemstart vergangen sind	Zahl im Dezimalsystem												
Sensor	Der Sensor, dessen Wert sich geändert hat	LB_START LB_END LB_HEIGHTSENSOR LB_RAMP LB_SWITCH BUTTON_START BUTTON_STOP BUTTON_RESET BUTTON_ESTOP SWITCH_IN HEIGHTSENSOR_VALID HEIGHTSENSOR METALSENSOR												
Wert	Der neue Wert des Sensors	Bei binärem Sensor:												

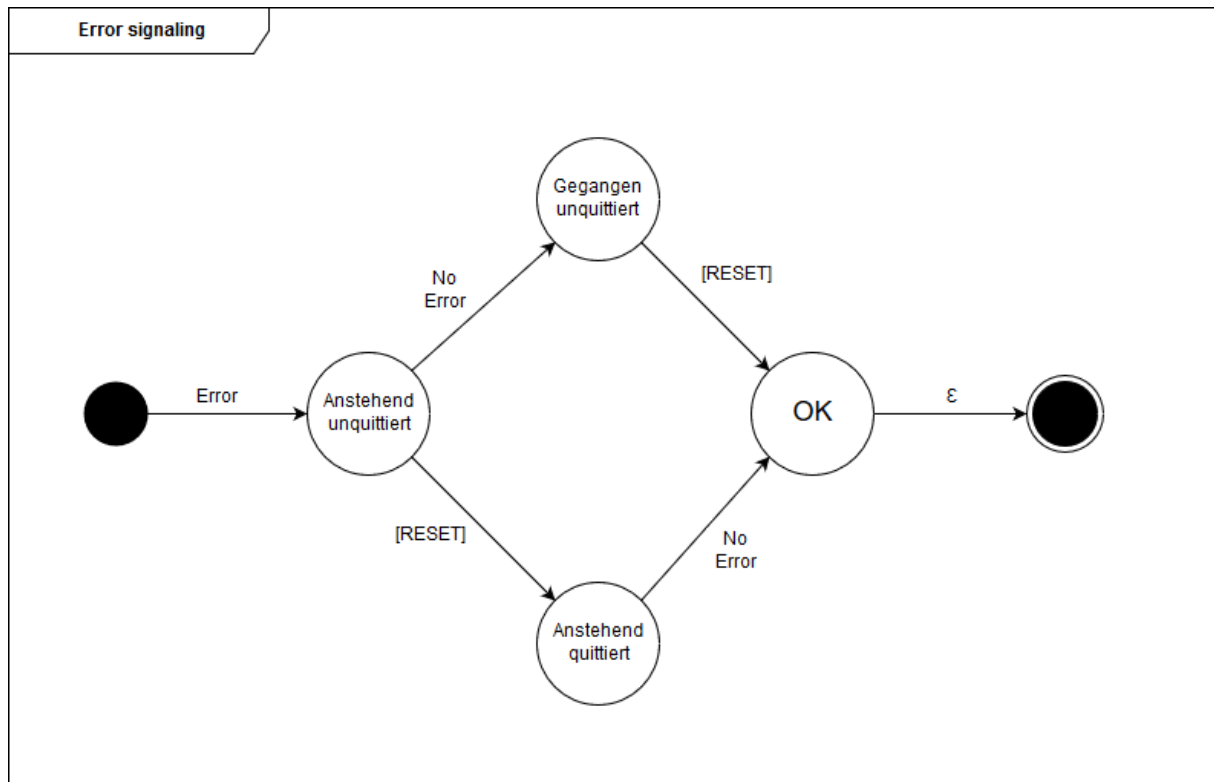
	<table><tr><td></td><td></td><td><div>0, false, f, low, lo</div><div>1, true, t, high, hi</div><div>Bei HeightSensor:<ul style="list-style-type: none">- 16bit binäre Zahl, muss exakt 16 Stellen lang sein- 16bit hexadezimale Zahl mit Präfix '0x' oder Präfix '\$'- 16bit dezimale Zahl; darf keine führende 0 haben</div></td></tr></table> <div>Tabelle A</div>			<div>0, false, f, low, lo</div> <div>1, true, t, high, hi</div> <div>Bei HeightSensor:<ul style="list-style-type: none">- 16bit binäre Zahl, muss exakt 16 Stellen lang sein- 16bit hexadezimale Zahl mit Präfix '0x' oder Präfix '\$'- 16bit dezimale Zahl; darf keine führende 0 haben</div>
		<div>0, false, f, low, lo</div> <div>1, true, t, high, hi</div> <div>Bei HeightSensor:<ul style="list-style-type: none">- 16bit binäre Zahl, muss exakt 16 Stellen lang sein- 16bit hexadezimale Zahl mit Präfix '0x' oder Präfix '\$'- 16bit dezimale Zahl; darf keine führende 0 haben</div>		
Formale Beschreibung	<div><div>EMP</div><div>:=</div><div>(Zeile '\n')+</div></div> <div><div>Zeile</div><div>:=</div><div>Satz? Kommentar?</div></div> <div><div>Satz</div><div>:=</div><div>Timestamp [\t]+</div><div>((BSensor [\t]+ BWert)</div><div> ('HEIGHTSENSOR' [\t]+ Number))</div></div> <div><div>Timestamp</div><div>:=</div><div>[0-9]+</div></div> <div><div>BSensor</div><div>:=</div><div>('LB_START' 'LB_END' 'LB_HEIGHTSENSOR' 'LB_RAMP' </div><div>'LB_SWITCH' 'BUTTON_START' 'BUTTON_STOP' </div><div>'BUTTON_RESET' 'BUTTON_ESTOP' 'SWITCH_IN' </div><div>'HEIGHTSENSOR_VALID' 'METALSENSOR')</div></div> <div><div>BWert</div><div>:=</div><div>'0' 'false' 'f' 'low' 'lo' '1' 'true' 't' 'high' 'hi'</div></div> <div><div>Number</div><div>:=</div><div>(([01]{16})</div><div> ('0x' Hex) ('\$' Hex)</div><div> ([1-9][0-9]{0-4}))</div></div> <div><div>Hex</div><div>:=</div><div>[0-9a-f]{1-4}</div></div> <div><div>Kommentar</div><div>:=</div><div>'#' [^\n]*</div></div>			
Beispiele	<div><div>0001000</div><div>LB_START</div><div>lo</div><div># Werkstück ist im Einlauf 1s nach Start</div></div> <div><div>0000000</div><div>BUTTON_ESTOP</div><div>1</div><div># E-Stopp Schalter ist NICHT gedrückt</div></div>			

7 Hardware Abstraction Layer - Klassendiagramm



Alle Hardware-Wrapper nutzen das HAL Interface um die Hardware Register auszulesen und um in diesen zu schreiben. Dabei wird die HAL entweder von der PhysicalHAL implementiert, welche die Schreib- und Leseoperationen durchführt, kann allerdings auch von der RecorderHAL oder der PlaybackHAL implementiert werden. In dem zweiten Fall können umhergehende Steuerbefehle aufgezeichnet werden um diese Sequenzen dann wieder zu Debugging-Zwecken abzuspielen.

8 Fehlersignalisierung und Quittierung



Dieses Diagramm stellt den Fehlerquittierungsverlauf dar. Dabei kann entweder die Fehlerursache vor oder nach dem Quittieren behoben werden. Ein Quittieren ist allerdings zwingend notwendig.