

# Fehlerbehandlung

## Inhalt

Fehlererkennung .....	1
Error Manager .....	1
Fehlerfälle.....	2
Fehlersignalisierung und Quittierung.....	3

## Fehlererkennung

Ob es sich bei einem Systemzustand um einen fehlerhaften Zustand handelt oder nicht entscheidet immer der Master – die oberste Entscheidungslogik. Wo und wie ein Fehler erkannt wird hängt allerdings davon ab, welcher Manager gerade aktiv ist und das Modul steuert. So können zum Beispiel Fehlerfälle sowohl im Run Manager, als auch im Config Manager erkannt werden. In jedem Fall wird die Fehlernachricht zum Master weitergereicht ohne sich um die Behandlung des Fehlers zu kümmern. Eine Ausnahme bildet da das Unterbrechen der seriellen Kommunikation, denn in diesem Fall hat nur ein Modul die Möglichkeit den Master zu benachrichtigen. In diesem Ausnahmefall wird also unverzüglich agiert und das System wird heruntergefahren.

Löst ein behandelbarer Fehler eine Fehlernachricht aus wird der Master den aktuellen Manager pausieren und stattdessen den Error Manager einschalten. Dieser bekommt die Fehlernachricht mit und kann Fehlerspezifisch arbeiten.

## Error Manager

So wie die anderen Manager implementiert der Error Manager ebenfalls das IManager Interface (*siehe Diagramm 1, Seite 2*), wodurch alle ankommenden HAL-Events direkt an diesen weitergeleitet werden. Ist der Error Manager aktiv, hat er also die volle Kontrolle über die HAL. Außerdem implementiert er ebenfalls die Schnittstelle um Nachrichten von allen anderen Steuereinheiten zu erhalten (*IRecipient*), so bleibt eine Kommunikation mit dem Master und anderen Managern erhalten.

Der Error Manager bleibt zur jedem Zeitpunkt erhalten und wird bei Bedarf eingeschaltet und nach Behebung des Fehlers wieder inaktiv gestellt.

Wird der Error Manager eingeschaltet bekommt er stets eine Nachricht mit der Fehlerursache. Zu diesem Zeitpunkt kann vom Error Manager eine Fehlerbehandlung instanziiert werden, deren Lebenszeit der Existenz des tatsächlichen Fehlers entspricht. Diese Fehlerbehandlung wird vom Error Manager mit einem Pointer gehalten.

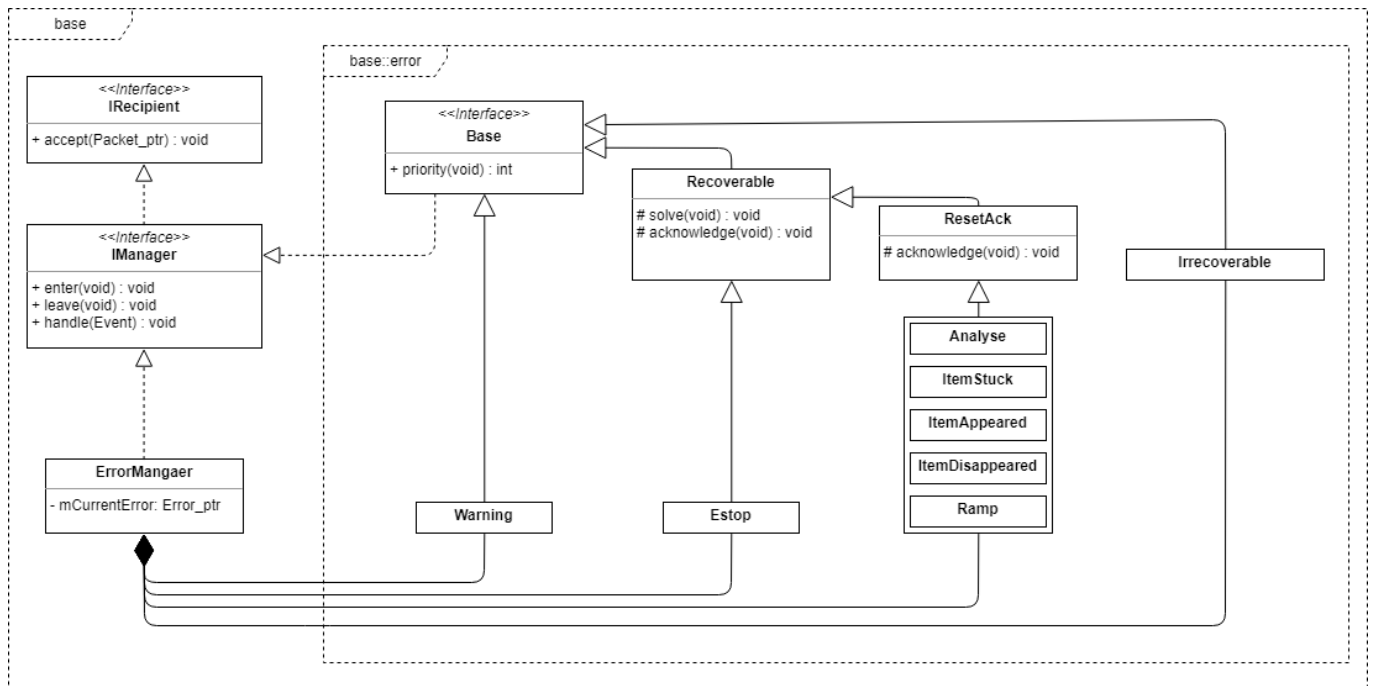


Diagramm 1 – Klassendiagramm des ErrorManagers und der Fehlerfälle

Jeder Fehlerfall implementiert für das error::Base Interface eine Priorität, damit eine sinnvolle Bearbeitungsreihenfolge der Fehlerfälle möglich ist.

Es gibt behandelbare Fehler (*Recoverable*), aber auch den nicht behandelbaren Fehler bei Fehlschlag der seriellen Kommunikation. Die behandelbaren Fehlerfälle implementieren eine Behandlungsroutine.

## Fehlerfälle

Fehlerfall	Beschreibung	Behandlung durch Benutzer
Warning	Ein anderes Modul befindet sich in aktiver Fehlerbehandlung. Ein pausierter Zustand.	-
Estop	Der E-STOP wurde gedrückt.	E-STOP wieder herausgezogen.
Analyse	Ein Werkstück konnte nicht eindeutig erkannt werden.	Ausgabe dieser Fehlernachricht.
Ramp	Beide Rutschen sind voll.	Beide Rutschen werden leergeräumt.
ItemAppeared	Ein Werkstück ist unerwartet aufgetaucht.	Dieses Werkstück wird vom Modul entfernt.
ItemDisappeared	Ein Werkstück ist verschwunden.	-
ItemStuck	Ein Werkstück ist innerhalb einer Lichtschranke blockierend hängen geblieben.	Dieses Werkstück wird vom Modul entfernt.
Irrecoverable	Die serielle Verbindung wurde getrennt oder die Konfiguration ist fehlgeschlagen.	-

## Fehlersignalisierung und Quittierung

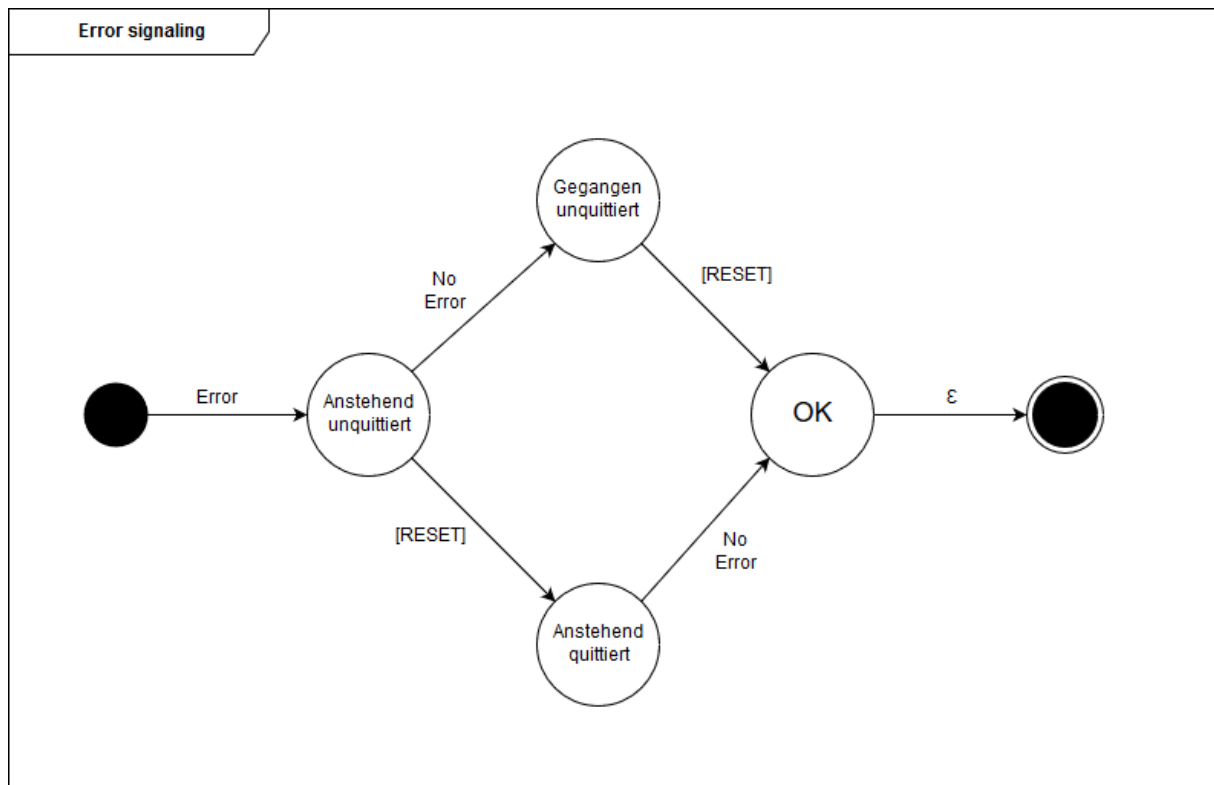


Diagramm 2 – Fehlerquittierung

Tritt eine Fehlerbehandlung ein, so muss dieser Fehler quittiert werden. Dies kann sowohl nach der Behandlung, als auch vorher erfolgen. Der Fehler muss also beseitigt und quittiert werden. Dies geschieht mit dem RESET Knopf.