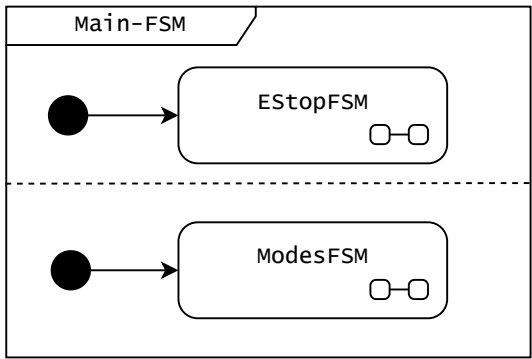
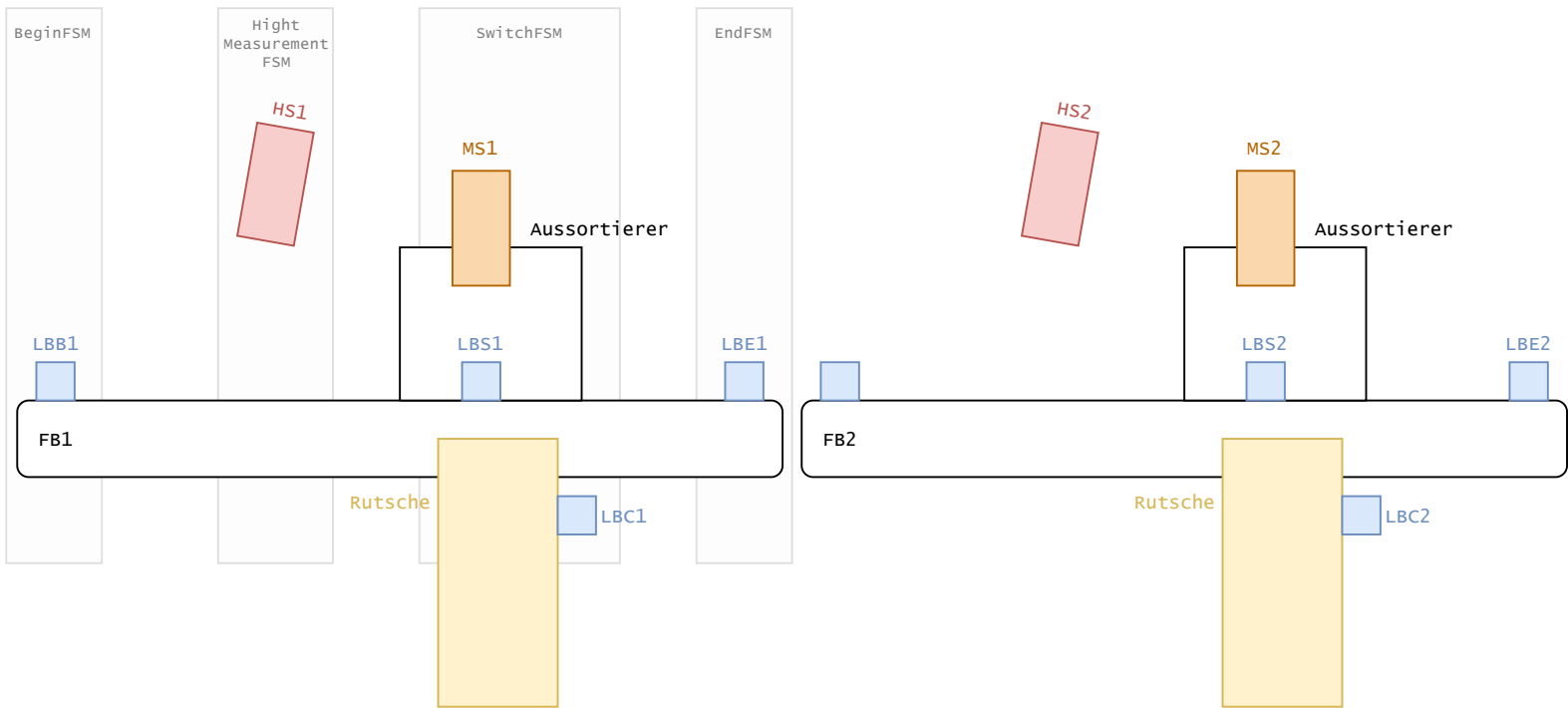
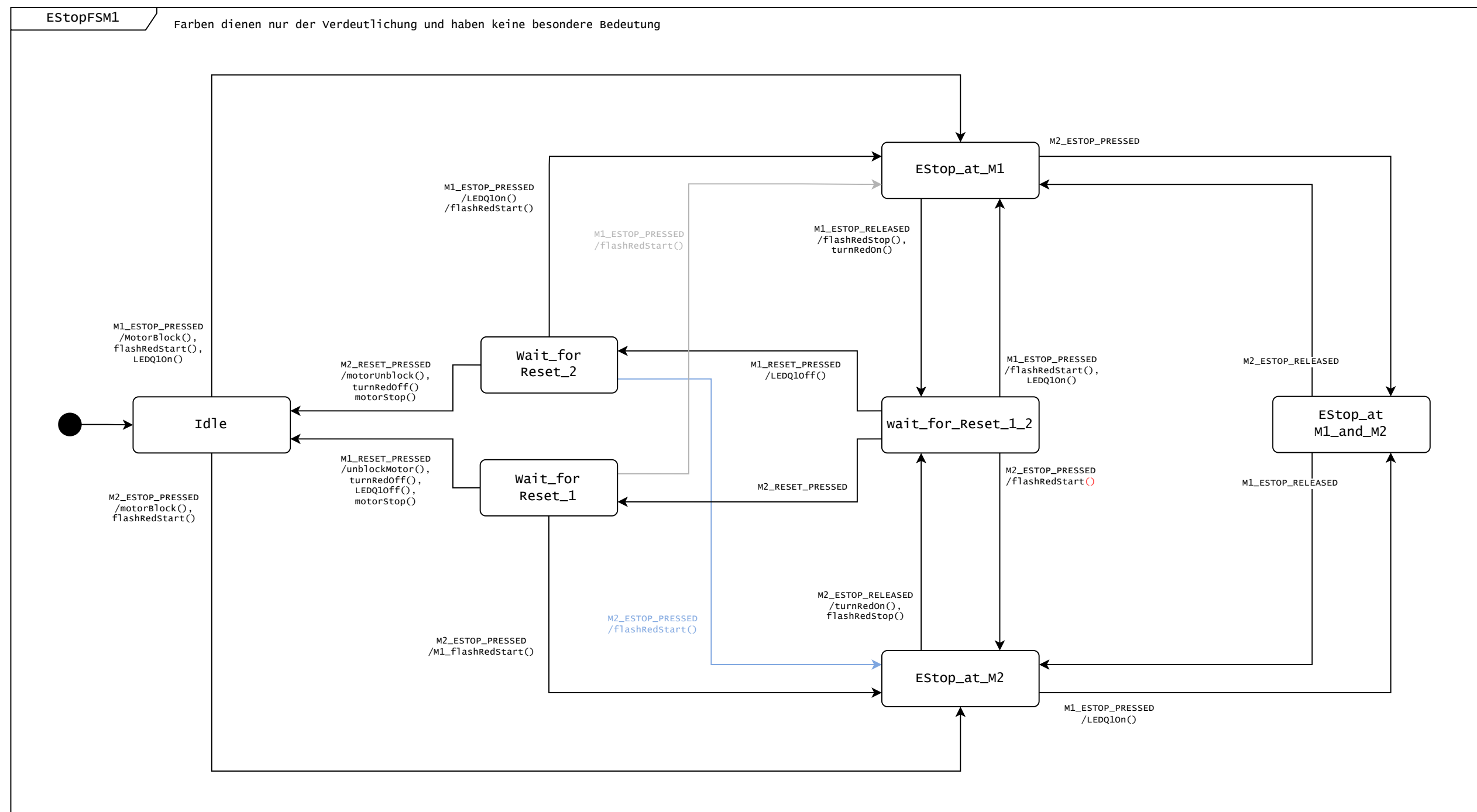
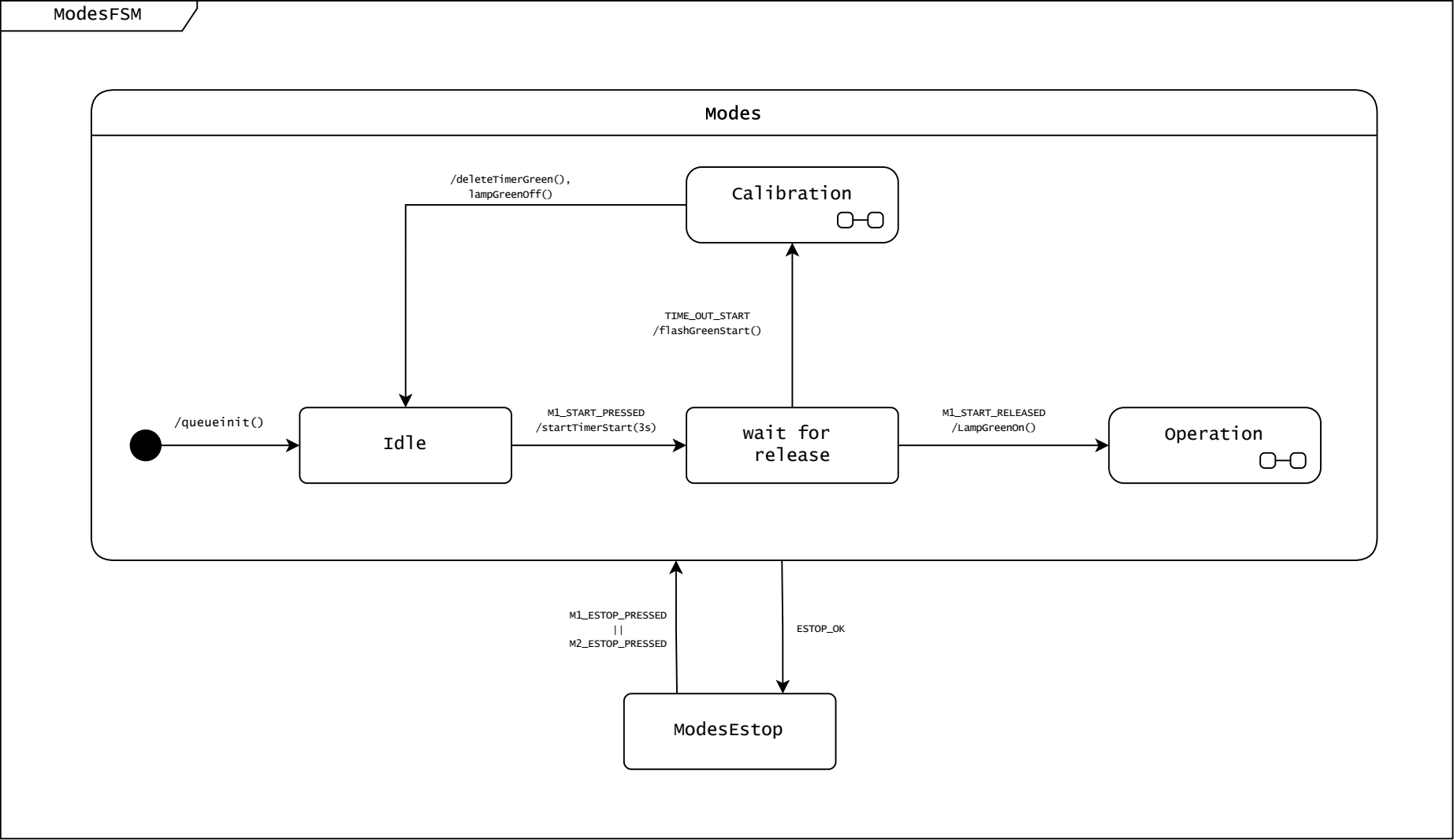


# FSMs der Anlage1:

Alle FSMs wurden trotz hierachischer Modellierung flach implementiert.



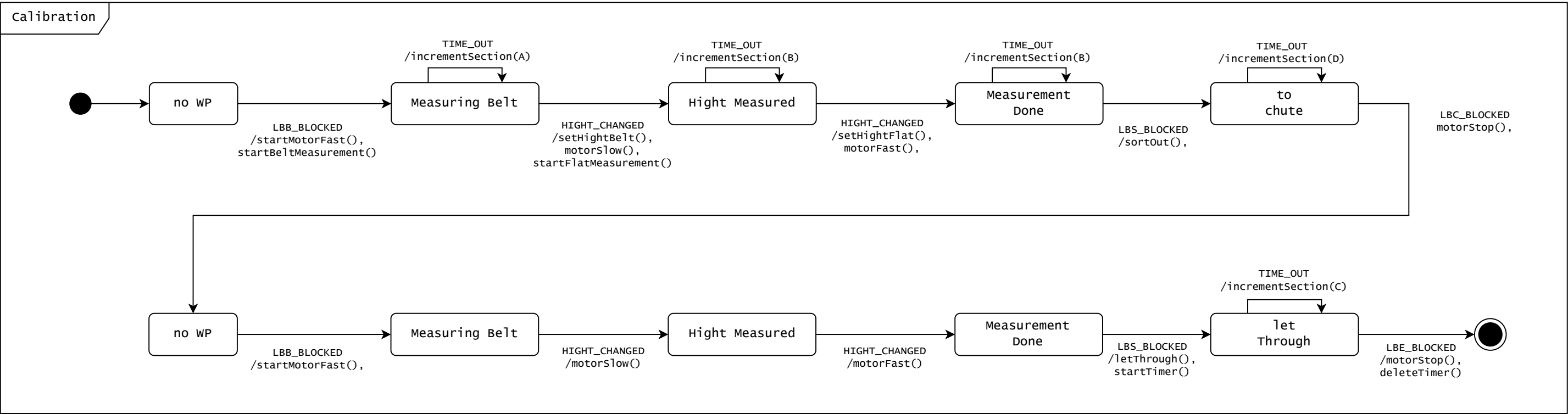




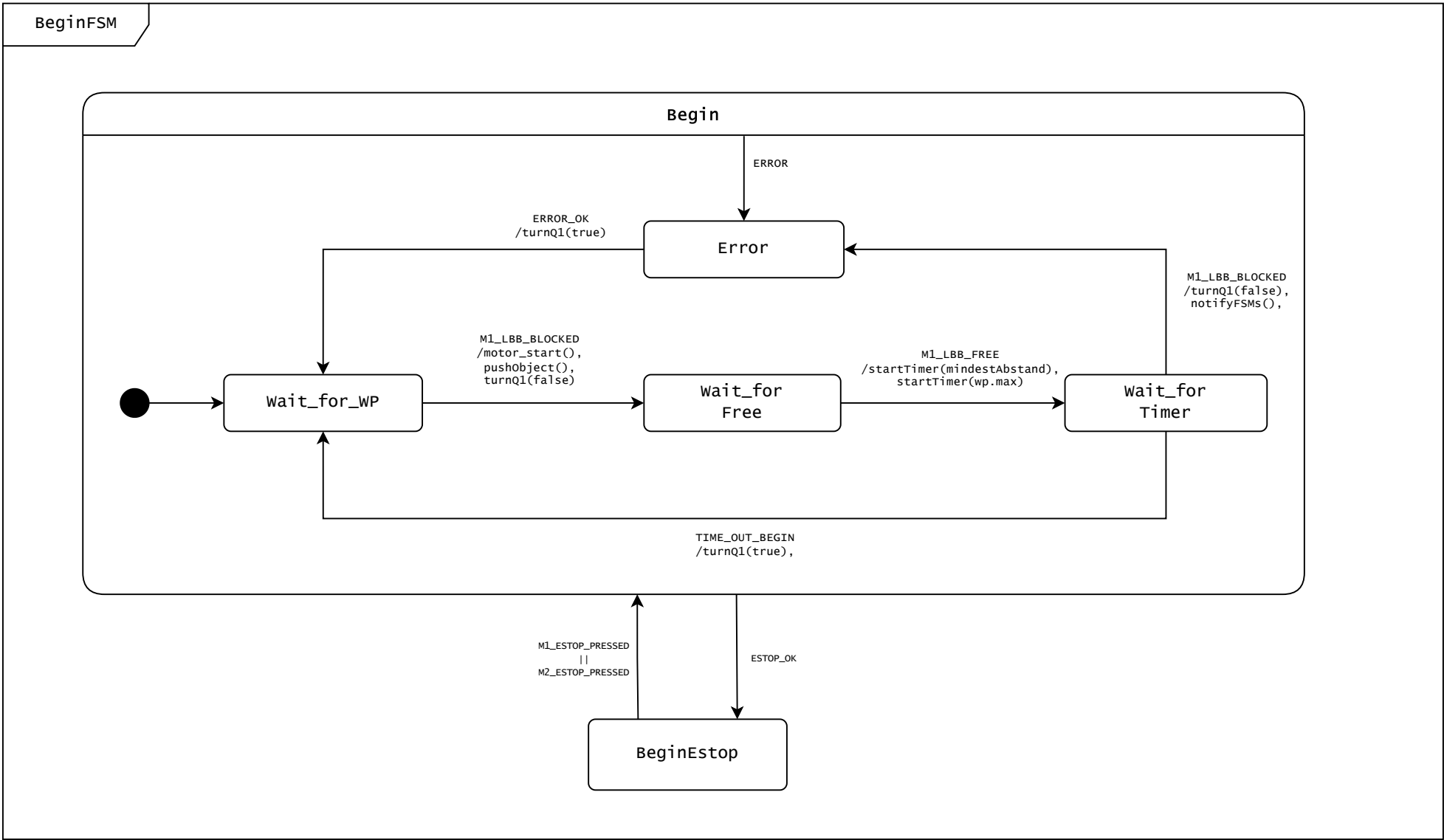
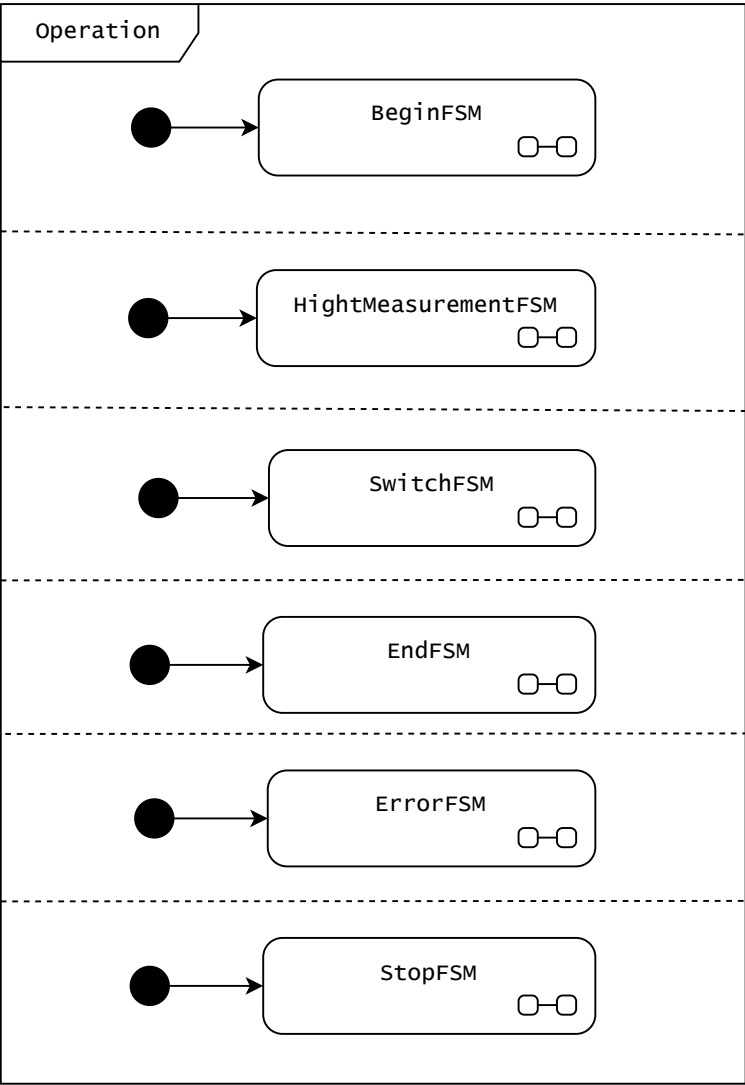
Die Queue enthält WP Objects

Calibration wurde trotz hierachischer Modellierung flach implementiert.

Bei der Calibration wird nur der Encoder (und keine FSMs!) kalibriert.

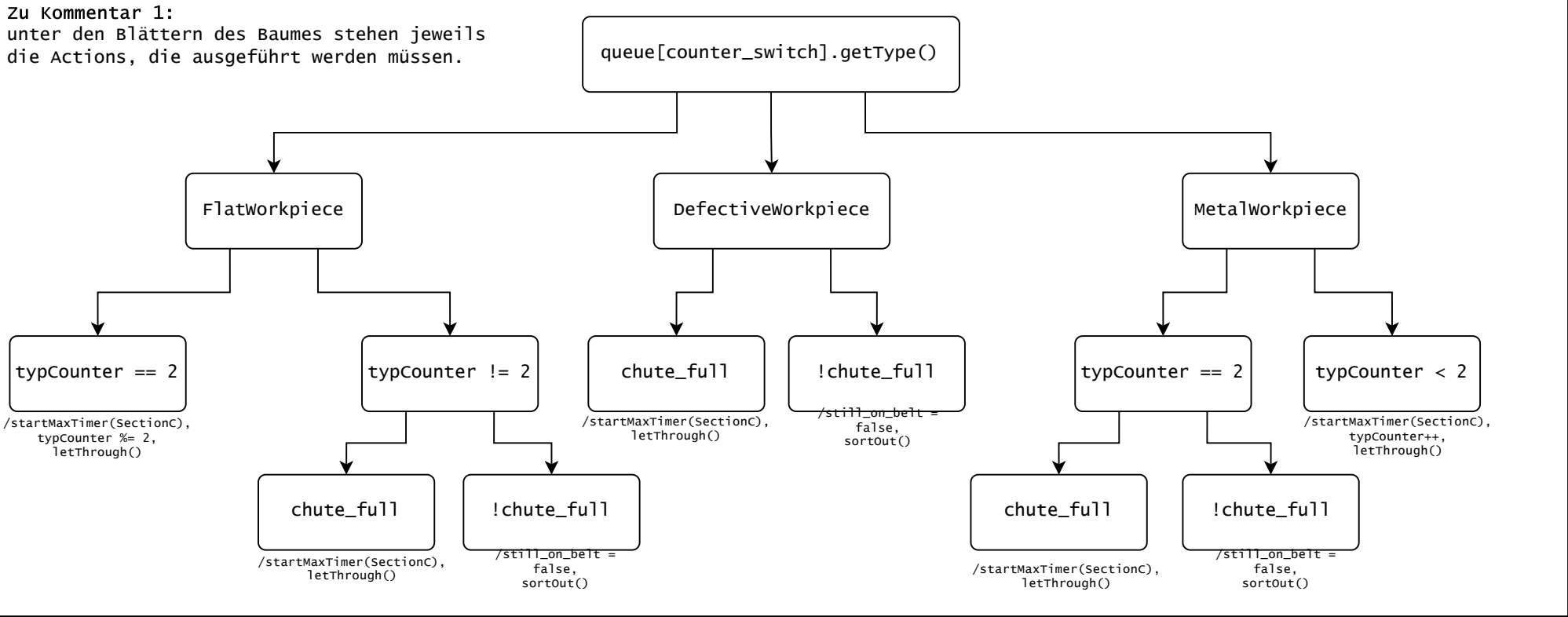
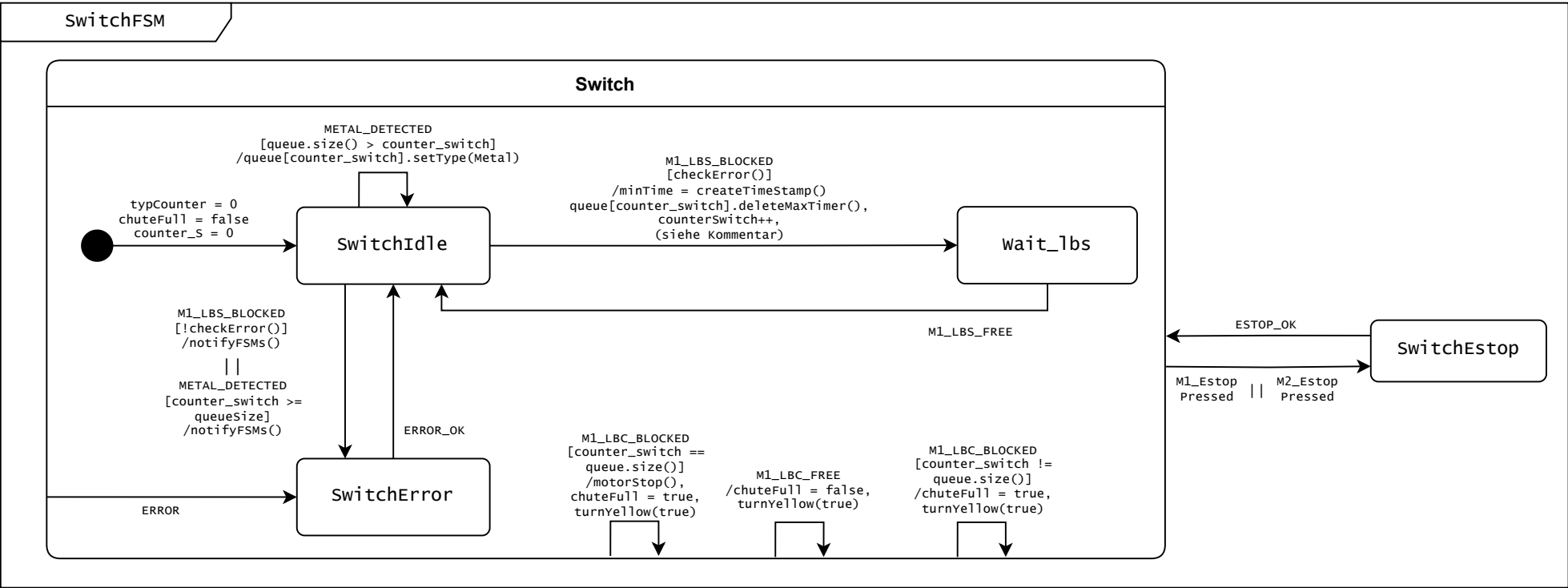


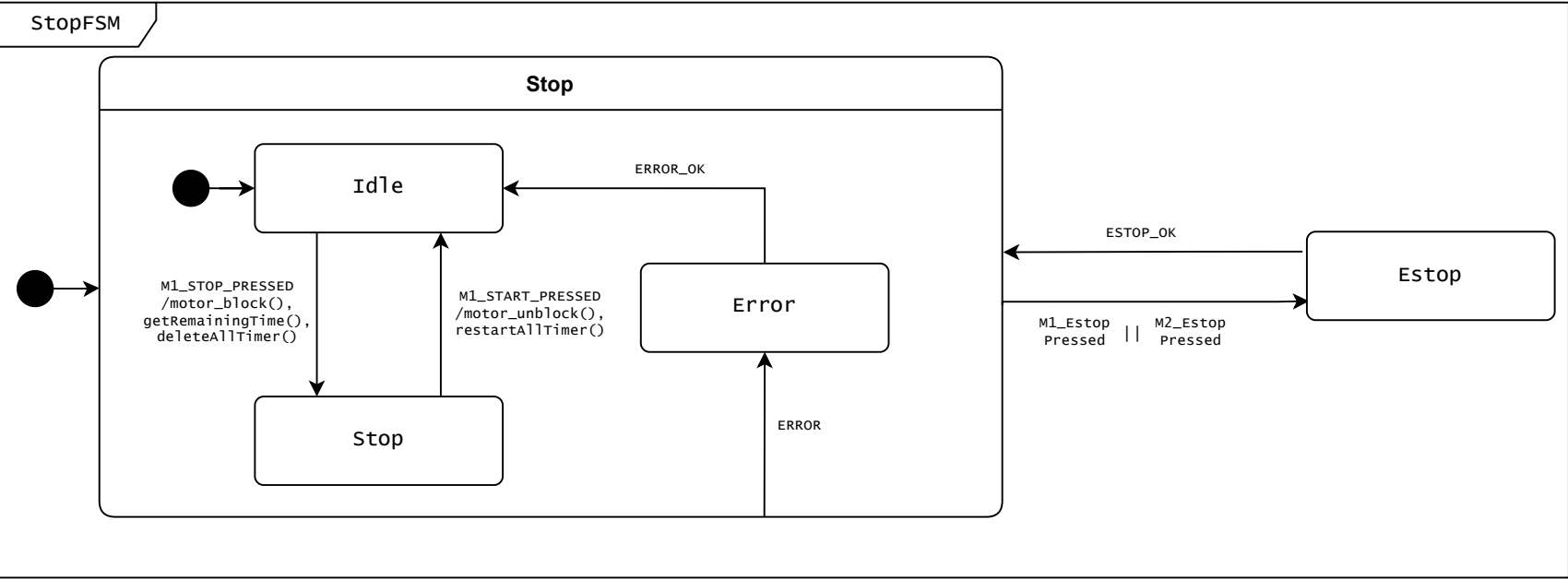
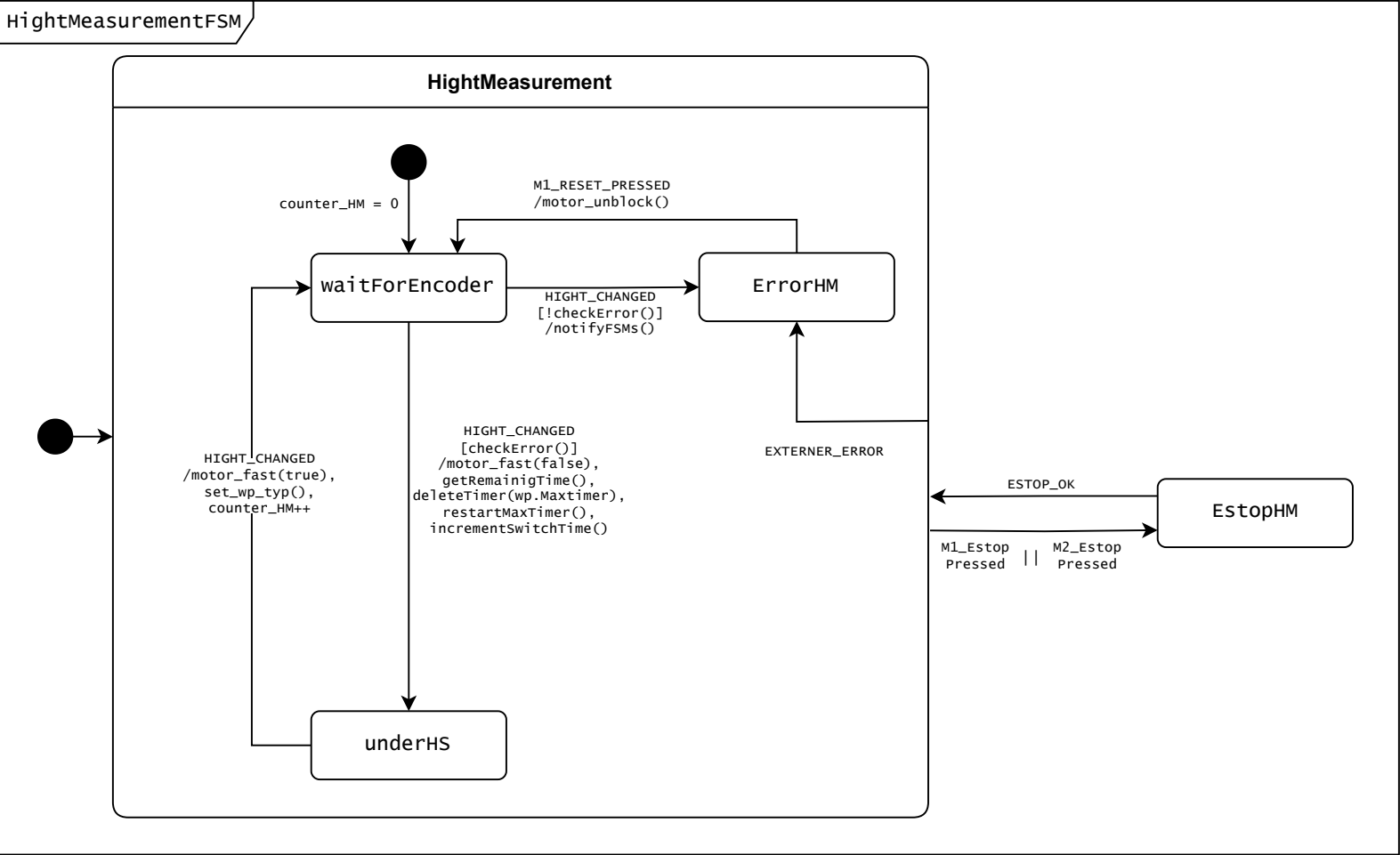
Jede FSM hat einen ESTOP zustand, der von jedem zustand aus erreicht werden kann.  
In diesem Zustand ignorieren die FSMs alle events bis auf "ESTOP\_OK", was nur von ESTOP\_FSM geschickt wird

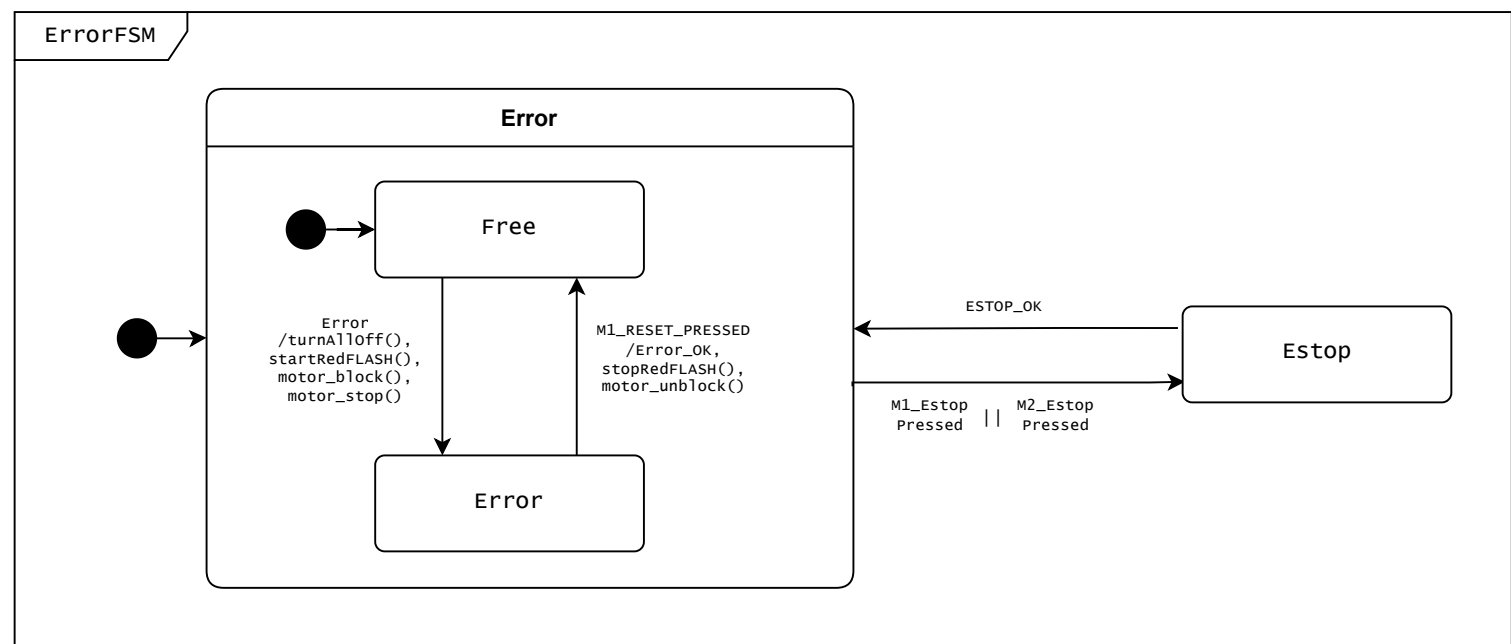
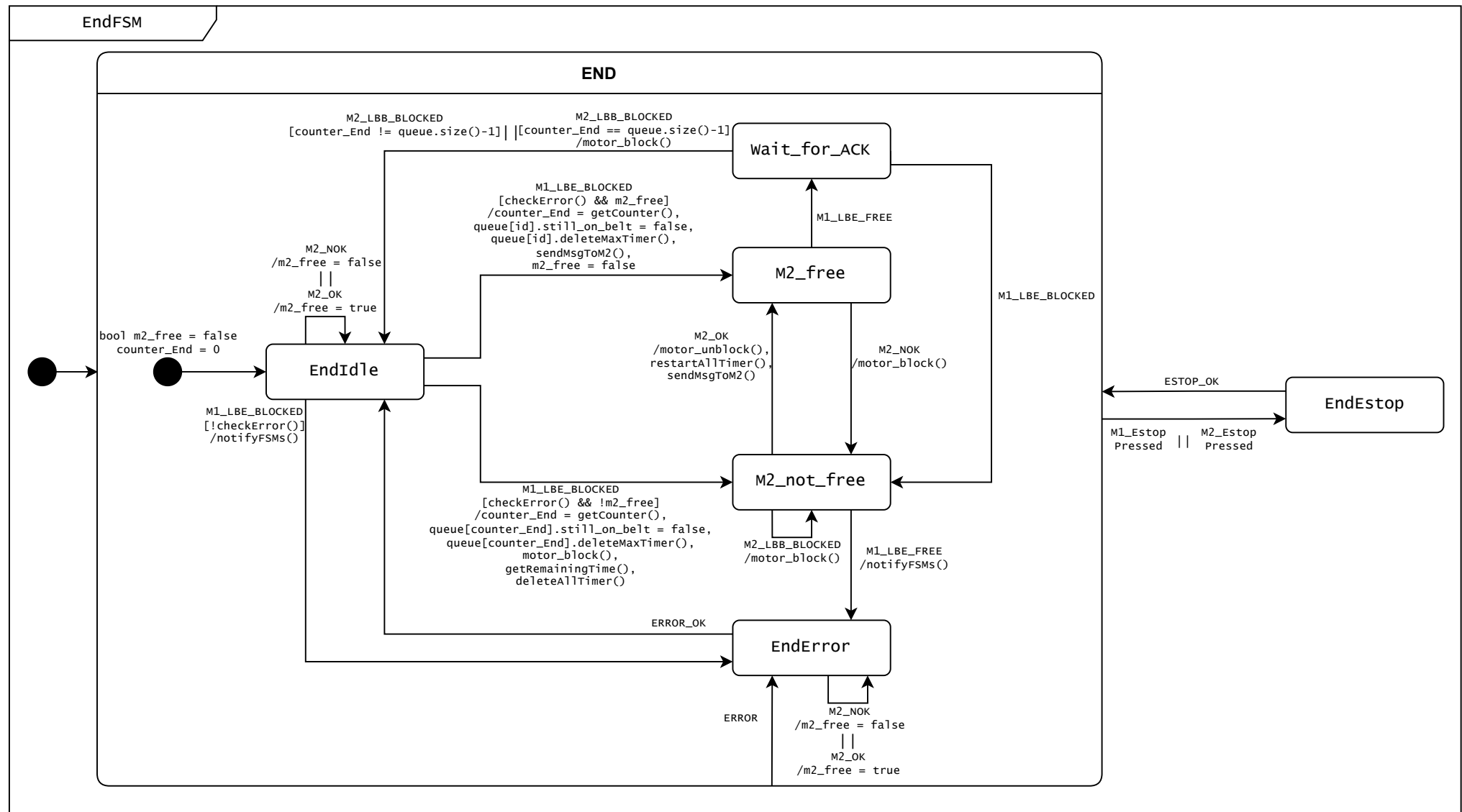


BeginFSM startet 2 timer (nicht periodisch):  
1) der Timer, der für den Mindestabstand zuständig ist  
2) der Max timer vom wp

Kommentar 1:  
Beim Übergang vom Zustand "SwitchIdle" zum Zustand "wait\_lbs" sind weitere Bedingungen zu überprüfen.  
Je nach dem welche Bedingungen erfüllt sind, müssen ggf. verschiedene Actions ausgeführt werden.  
Damit die Modellierung nicht riesig wird, haben wir uns entschieden, die Bedingungen Seperat zu modellieren. (s. unten)

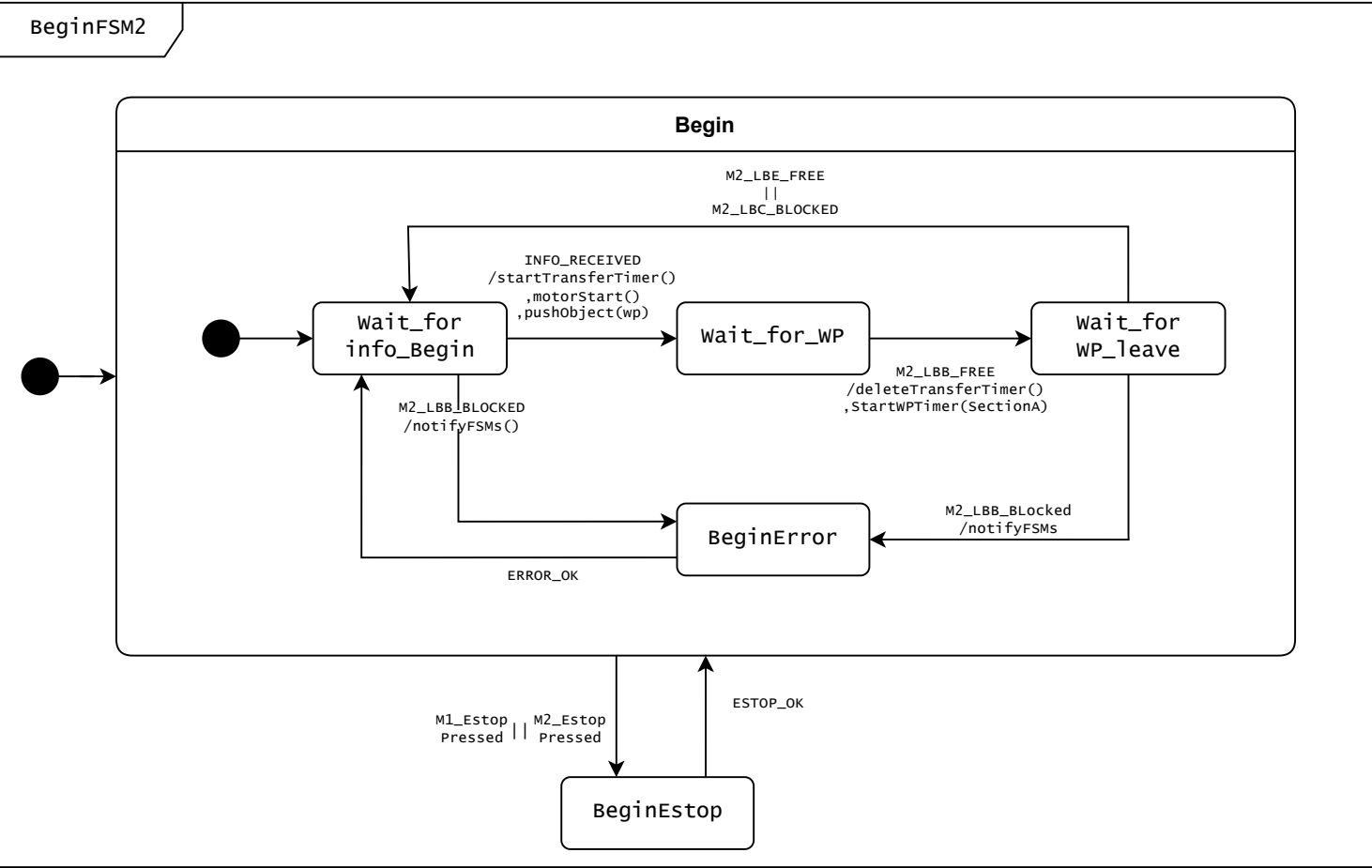






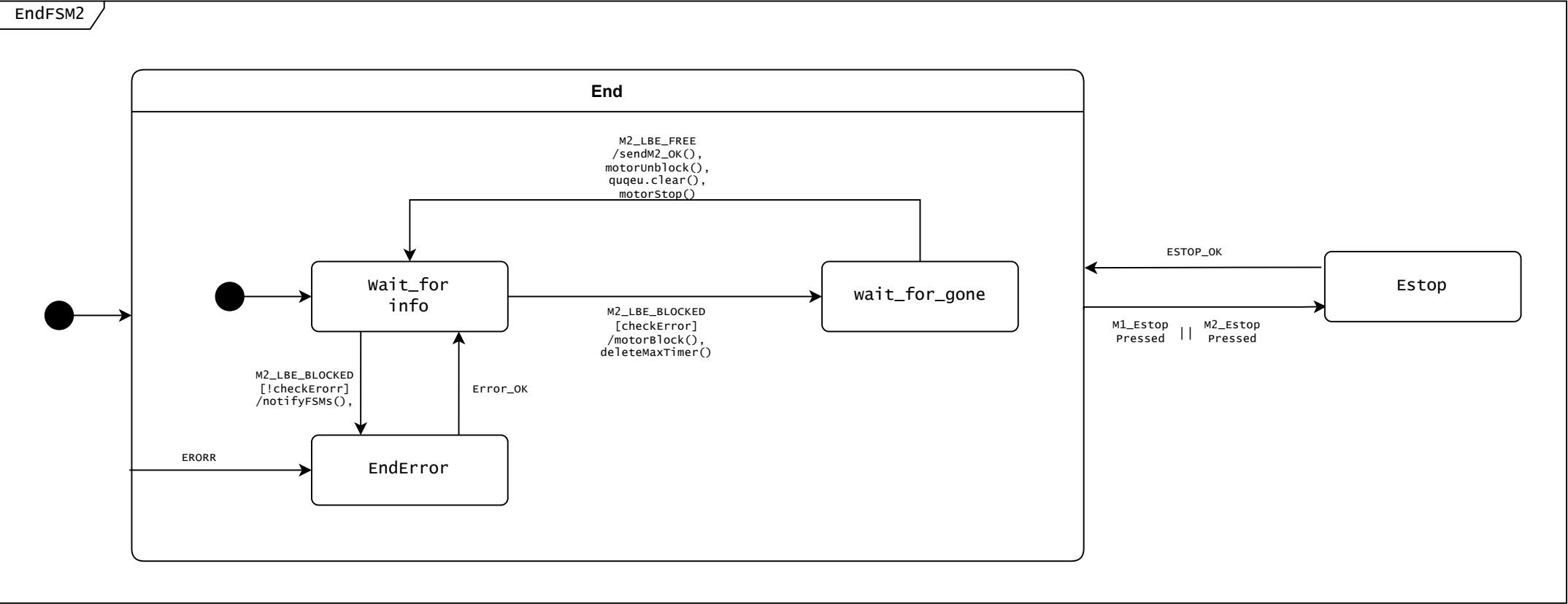
# FSMs der Anlage2:

Anlage2 hat dieselbe Struktur wie Anlage1



Es gibt sehr kleine Unterschiede zwischen den FSMs in A1 und A2:

- Estop1 vs Estop2 : Der einzige Unterschied sind die Actions
- Switch1 vs Switch2 : Wenn aussortiert werden muss aber die Rutsche ist voll -> Error-zustand
- HM1 vs HM2 : die FSM prüft, ob der gesendete Typ von A1 mit dem WP übereinstimmt.





Diese FSM hat die Aufgabe, Anlage2 mitzuteilen, wann sie bereit ist für ein neues WP und wann nicht.

