

Abstandsregelung über Ultraschallsensor mit dem Rover

Das State-Transition Programmierschema

1. Aufgabe (Abstand1.tns):

Stelle die Rover-Pappschachtel in einer Armlänge Abstand vor dem Rover auf. Schreibe ein Programm, das fortlaufend den Ultraschallsensor (Ranger) des Rovers ausliest und man jederzeit mit einer beliebigen Taste auf dem CX Rechner beenden kann. Je nachdem, welcher Abstand vom Ranger des Rovers gemessen wird, soll dein Programm folgende Funktionen ausführen:

- a. Das Hindernis befindet sich weiter als 30cm vom Rover entfernt: Der Rover bleibt stehen und die rote RGB LED des Rovers leuchtet.
- b. Das Hindernis befindet sich weniger als 30cm aber mehr als 13cm vom Rover entfernt: Der Rover fährt vorwärts, bis er weniger als 13 cm vom Hindernis entfernt ist.
- c. Das Hindernis befindet sich weniger als 13cm, aber mehr als 9cm vom Rover entfernt: Der Rover bleibt stehen und die grüne RGB LED leuchtet.
- d. Das Hindernis befindet sich weniger als 9cm vom Rover entfernt: Der Rover fährt rückwärts, bis der Abstand grösser als 9cm ist.

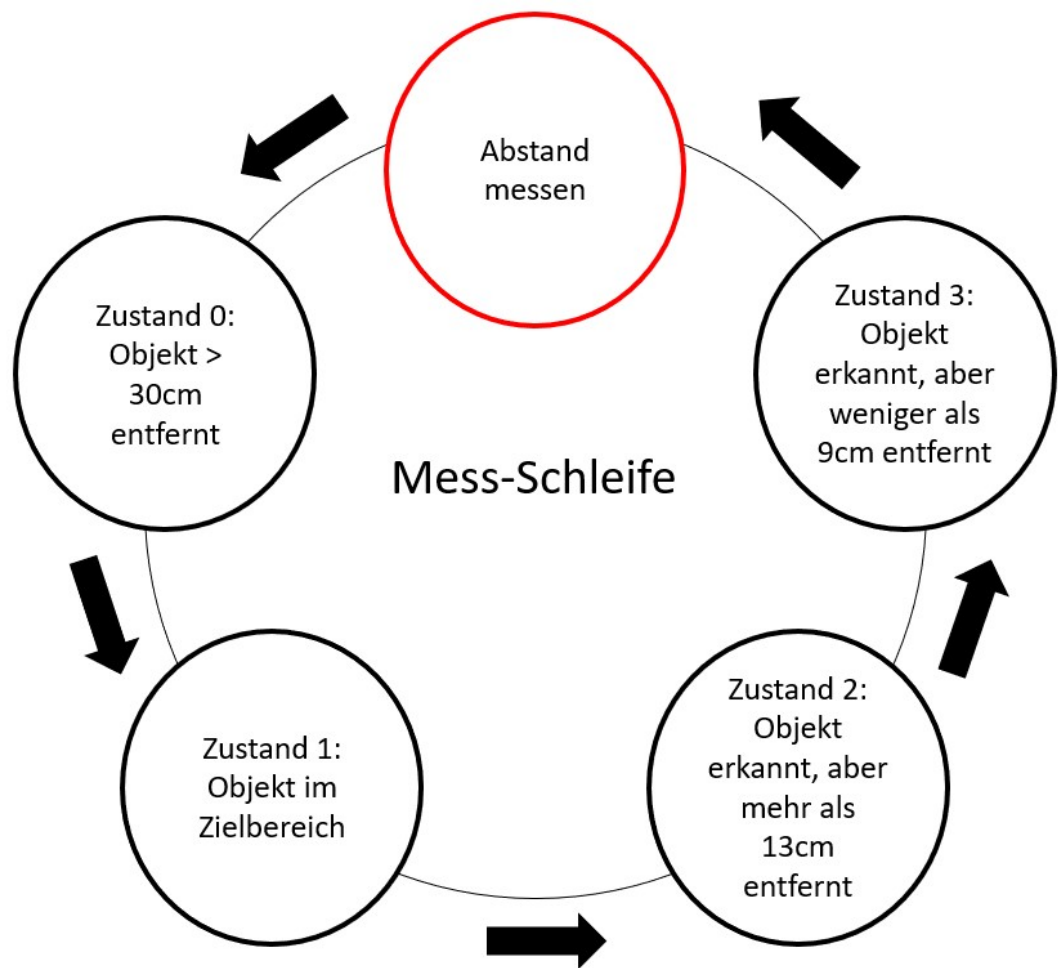
2. Welche Herausforderungen gibt es bei dieser Aufgabe?

- a. Alle Funktionen müssen innerhalb der innerhalb der <While> Schleife abgearbeitet werden, damit das Programm jederzeit per Tastendruck gestoppt werden kann.
- b. Es dürfen keine lokalen <While> Schleifen innerhalb der Funktionen programmiert werden, da sonst die Tasturabfrage nicht mehr durchgeführt wird.
- c. Es muss fortlaufend der Abstand mit dem Ultraschallsensor gemessen werden, damit die 4 unterschiedlichen Funktionen entsprechend ausgeführt werden. Je häufiger der Abstand gemessen wird, desto schneller reagiert der Rover auf Änderungen beim Abstand zum Objekt.
- d. Die Bewegungen des Rovers werden durch die <RV FORWARD> und <RV BACKWARD> Kommandos gesteuert. Beide Kommandos gehen in die sogenannte „Command Queue“ des Rovers. Fortlaufende RV FORWARD/BACKWARD Kommandos sind zu vermeiden, da sonst der Command Queue Speicher-Puffer des Rovers überläuft und einen Fehler während des Programmablaufs erzeugt.

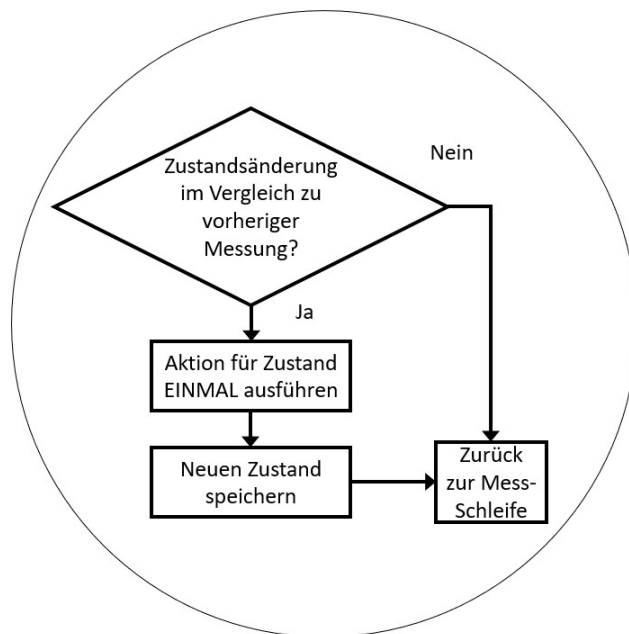
- e. Der Zielbereich, in dem der Rover stehen bleibt und seine RGB LED Grün leuchtet, muss hinreichend groß gewählt werden, da der Rover sonst permanent vorwärts und rückwärts fährt, d.h. zwischen der programmierten Minimal- und Maximalgrenze des Objektabstands hin- und her oszilliert. Daher der Minimalabstand von 9cm, bevor der Rover rückwärts fährt und der Maximalabstand von 13cm, bevor der Rover vorwärts fährt.

3. Lösung: Die State-Transition Programmierung

- a. Man erstellt Diagramme der möglichen Zustände in der Aufgabe und der Bedingungen, wann sich der Zustand ändert:



- b. In jedem der möglichen Zustände wird zunächst geprüft, ob der Zustand sich im Vergleich zum vorherigen Durchlauf der Mess-Schleife geändert hat oder nicht.



Wenn sich der Zustand nicht geändert hat, springt das Programm sofort zurück in die Mess-Schleife. Andernfalls wird die dem neuen Zustand zugeordnete Aktion ein einziges Mal ausgeführt, der neue Zustand gespeichert und zur Mess-Schleife zurückgekehrt.

Diese Programmiertechnik ist empfehlenswert für alle Aufgaben, bei denen es darauf ankommt, schnelle, regelmäßige Mess-Zyklen durchzuführen und vom Resultat abhängige, unterschiedliche Aktionen einzuleiten. Es ermöglicht eine schnelle, fortlaufende Messung in der Mess-Schleife, vermeidet lokale Schleifen innerhalb der Funktionen und ein Überlaufen des Speicher-Puffers des Rovers oder des Innovator Hubs.