

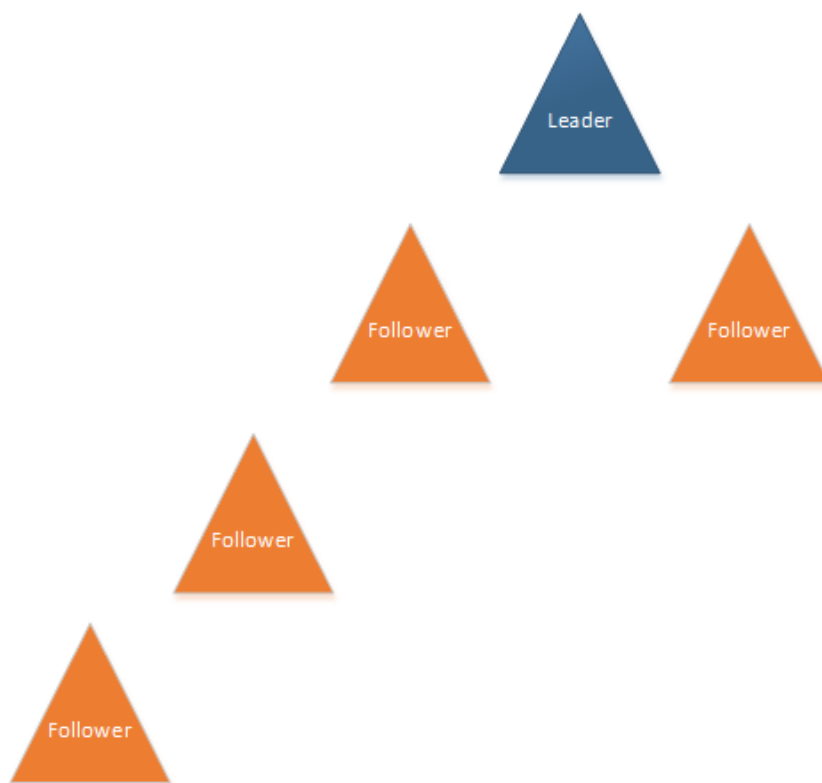
Beschreibung: Mindstorm V-Formation

Hinweis: Aufgabe wurde aufgrund mangelnder Kommunikation zwischen den Mindstorms im Zuge der Vorlesung angepasst!

Wie besprochen wurde das bei der Besprechung erstellte Dokument als "Vorarbeit.pdf" mit angehängt.

Darin enthalten sind die Ideen für die ursprüngliche Aufgabenstellung.

Darstellung der Formation



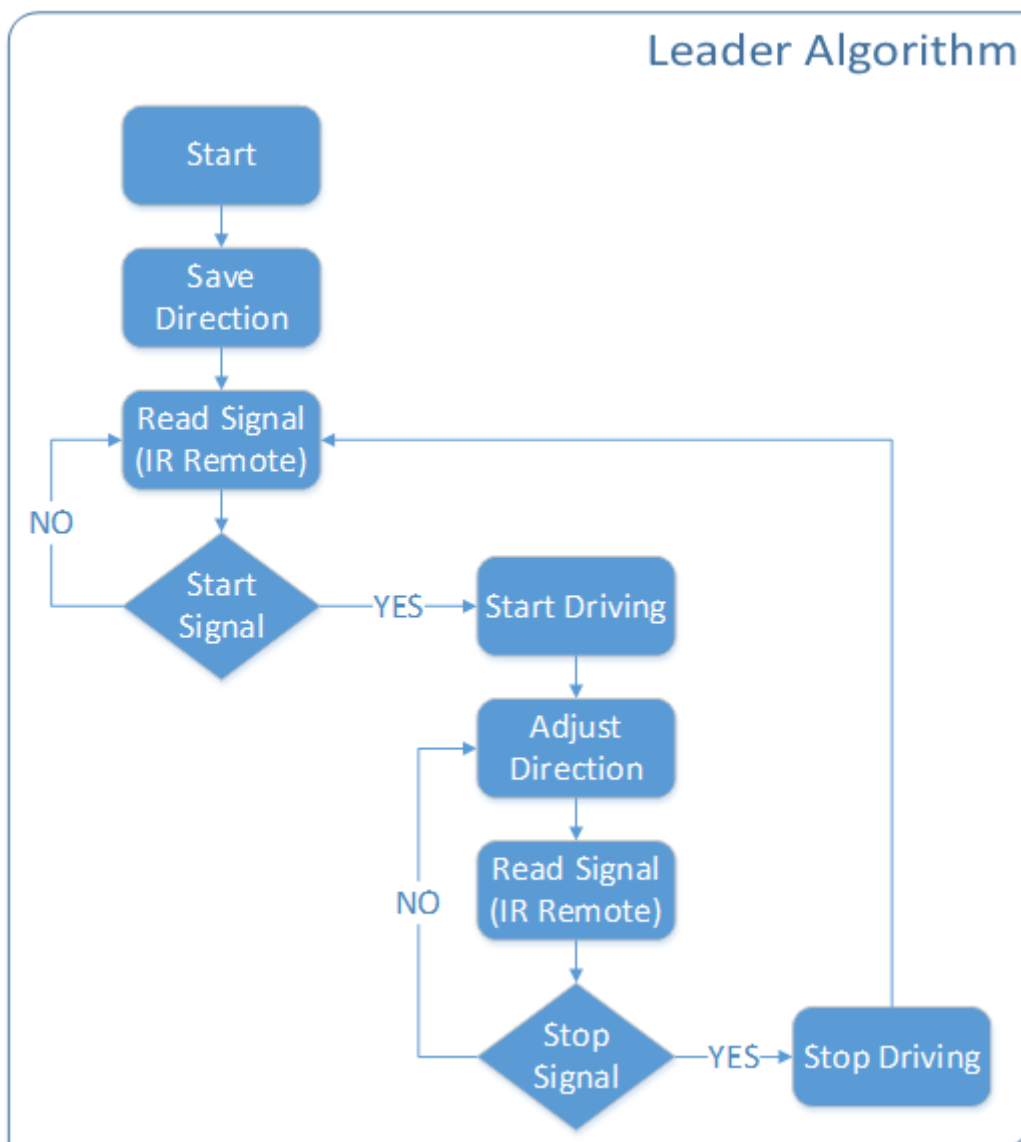
In der gegebenen V-Formation sollen die Follower dem Leader folgen. Wenn sich dieser nach vorne bewegt, sollen sich auch die Follower nach vorne bewegen. Der Abstand zum Vordermann wird dabei dynamisch beim Programmstart festgelegt. Dadurch wird der Abstand durch den Aufbau der Formation festgelegt.

Da die Mindstorms durch Ungenauigkeiten nicht geradeaus fahren, wird der Gyroskop-Sensor verwendet, um die Ausrichtung der Roboter während der Fahrt anzupassen. Die initiale Ausrichtung wird beim Programmstart festgehalten. Anhand dieser werden Lenkbewegungen zur Korrektur ausgeführt.

Die Videos für die V-Formation befinden sich im Abgabeordner (V-Formation_1.mov und V-Formation_2.mov).

Verhalten des Leader

- Die initiale Ausrichtung des Roboters wird bei Programmstart durch den Gyroskop-Sensor ausgelesen und gespeichert
- Wartet auf das Signal der Infrarot Fernbedienung zum Start
- Bei entsprechendem Tastendruck fährt der Leader los
- Der Leader nutzt seinen Gyroskop-Sensor um die ursprüngliche Richtung beizubehalten
- Der Leader fährt so lange vorwärts, bis das Stoppsignal durch die Fernbedienung erfolgt

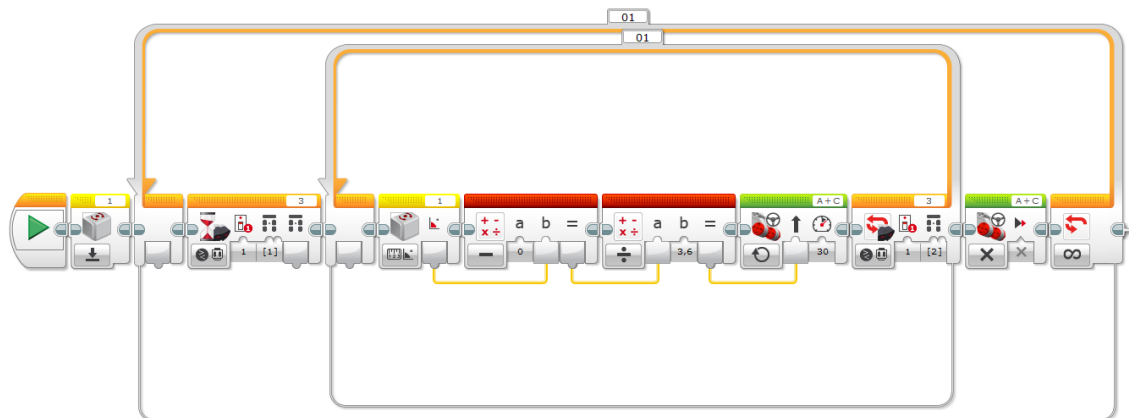


Der Algorithmus ist sowohl in Python als auch mit der Lego Mindstorm Anwendung implementiert.

Der Pythoncode ist abgelegt unter: `leader/main.py`

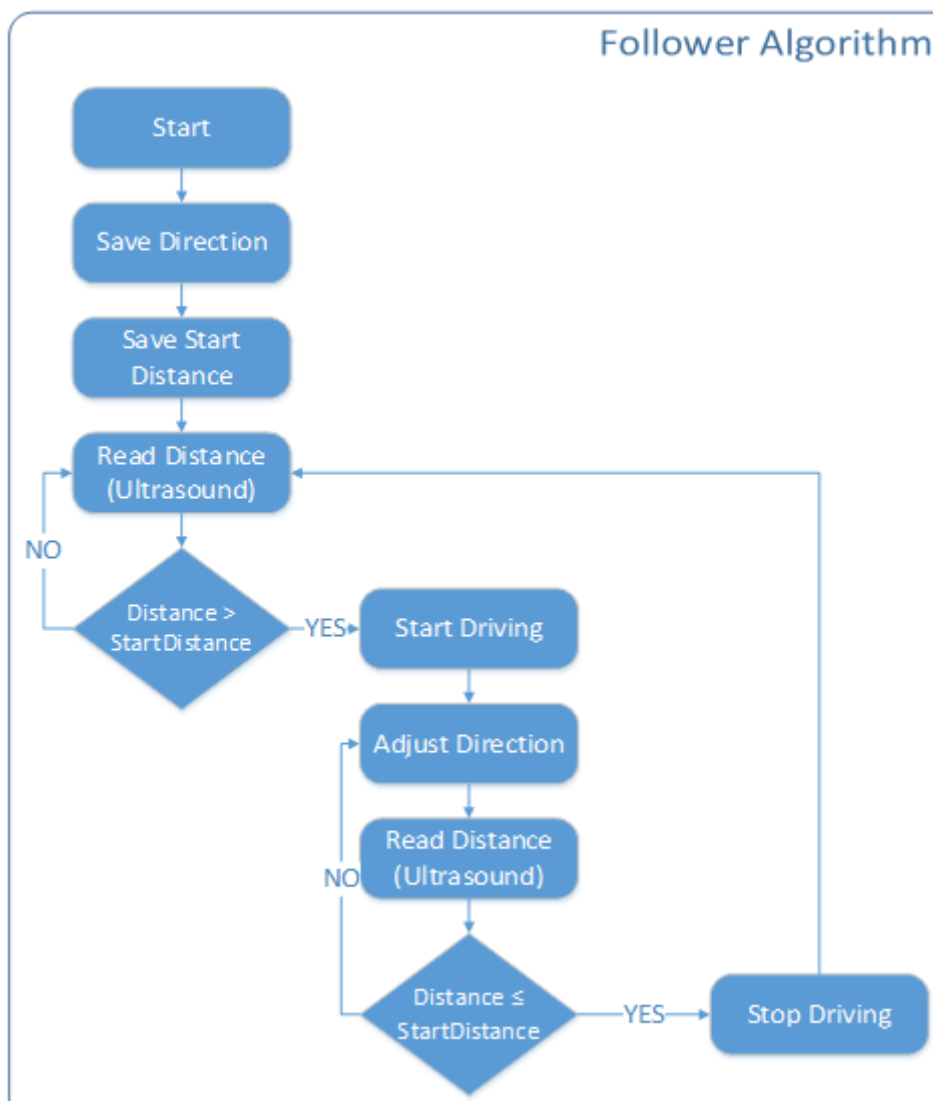
Der Lego Mindstormcode unter: `V-Formation-Gesamt.ev3`

Das folgende Bild zeigt den Algorithmus des Leaders in der Lego Mindstorm Anwendung:



Verhalten der Follower

- Die initiale Ausrichtung des Roboters wird bei Programmstart durch den Gyroskop-Sensor ausgelesen und gespeichert
- Die Distanz zum Vordermann wird gespeichert
- In einer Schleife:
 - Überprüfung der Distanz zum Vordermann
 - Falls sich die Distanz kleiner der ursprünglichen Distanz ist, fährt der Mindstorm los
 - Falls die Distanz kleiner gleich dem Startwert ist, hält der Mindstorm an
 - Falls sich die Fahrtrichtung vom Anfangswert geändert hat, korrigiert der Mindstorm die Richtung



Der Algorithmus ist sowohl in Python als auch mit der Lego Mindstorm Anwendung implementiert.

Der Pythoncode ist abgelegt unter: `follower/main.py`

Der Lego Mindstormcode unter: `V-Formation-Gesamt.ev3`

Das folgende Bild zeigt den Algorithmus des Followers in der Lego Mindstorm Anwendung:

