|  |  |  |
| --- | --- | --- |
|  |  |  |

Titel

Autor: Julien Stengel

Datum: 11.01.2018

Inhaltsverzeichnis

[1 Historie 1](#_Toc503453694)

[2 Voraussetzungen 2](#_Toc503453695)

[3 Anmerkungen 2](#_Toc503453696)

[4 Bedienung 2](#_Toc503453697)

[5 Parameter 3](#_Toc503453698)

# Historie

| Datum, Version | Beschreibung | Autor |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# Voraussetzungen

* Der Benutzer des Knotens weiß wie man mit Ros und Linux umgeht
* Der Befehl catkin\_make wurde bereits im Ros-Workspace ohne Fehlermeldung ausgeführt
* Alle nötigen Bestandteile für Mavlink-Communication sind vorhanden
* Der Backup CamNode ist vorhanden und wird verwendet (Das ist der alte CamNode, der noch über Aves eingestellt werden kann)
* Man ist über LAN mit dem Odroid verbunden und greift entweder über remote Desktop oder über Putty (via ssh) auf das Odroid zu
* Folgende Lane Detection Version wird genutzt:

„https://svn.berner-mattner.com/svn/FT0118/functions/lane\_detect/software/branches/vers\_0\_8\_2“

# Anmerkungen

* Lenksregelung wird über die Fernbedienung gesteuert
  + Wie man die Fernbedienung bedient werden muss kommt auf das Gerät an.
    - Die Bedienung in Berlin kann man sich von Malte, Felix oder Nicolai genauer erklären lassen
* Sollte das STM-Board rot blinken so muss ein USB-Kabel angeschlossen werden
* In der Anleitung wird davon ausgegangen, dass man sich für die remote mit dem Desktop verbunden hat

# **Bedienung**

1. Konsole/Terminal starten
2. Roscore starten
3. Neues Terminal-Fenster öffnen (entweder komplett neu oder mit „strg+shift+T“ einen neuen Tab in der bereits bestehenden Konsole starten)
4. CamNode starten
   1. In den Ros-workspace navigieren
   2. „source devel/setup.bash“ ausführen
   3. „rosrun camera\_stream CamNode“
5. Lane Detection starten
   1. Neues Terminal-Fenster öffnen (entweder komplett neu oder mit „strg+shift+T“ einen neuen Tab in der bereits bestehenden Konsole starten
   2. In den Ros-workspace navigieren
   3. „source devel/setup.bash“ ausführen
   4. “rosrun lane\_detect lane\_detect 20 0 90 60 0 20 900 200 200 35 744 480 0 0.5 2 10 1” ausführen
      1. Was diese Parameter bedeuten ist weiter unten aufgeführt
      2. Wenn alles geklappt hat, dann sollten sich Momentanaufnahmen der Kamera und ein Koordinatenkreuz öffnen
6. Mavlink auf Ros-Seite starten
   1. Neues Terminal-Fenster öffnen (entweder komplett neu oder mit „strg+shift+T“ einen neuen Tab in der bereits bestehenden Konsole starten
   2. In den Ros-workspace navigieren
   3. „source devel/setup.bash“ ausführen
   4. „rosrun communication StmCommunication“
7. Die Kamera mit Aves konfigurieren
   1. Links im Reiter auf „Connections“ drücken
      1. Dann auf „SSH“ klicken
      2. Eine Verbindung auswählen und verbinden
   2. Das Menü über den oberen Linken Pfeil öffnen
   3. Weiter im Reiter „Connections“
      1. Wähle „ProcessingEngine“ aus
      2. Wieder die gewünschte Verbindung auswählen
      3. Den Port auf 5405 stellen
   4. In das „Addons“ Untermenü
      1. PeDebugger auswählen
      2. Ganz rechts den Menüpunkt wählen
      3. Gain und Expo. Soweit anpassen bis Linien in der Ros-Ausgabe von Lane-Detection zu sehen sind
8. Das Auto auf die Fahrbahn setzen, und mit der Fernbedienung die Geschwindigkeit steuern, während das Auto selber lenkt (Zum Umschalten auf Lane Keeping hier nochmal der Verweis auf den ersten Punkt in der Anmerkung)

# Parameter

1. Kamera Höhe
2. Kamera Winkel
3. Öffnungswinkel x
4. Öffnungswinkel y
5. Kemeratranslation x
6. Kemeratranslation y
7. Gradient threshold
8. Linien min Abstand
9. Linienverwaschung
10. Min Linien Länge
11. Video Breite
12. Video Höhe
13. Roi – x - Verhältnis
14. Roi – y - Verhältnis
15. Polynomgrad
16. Boolscher Wert für das Ausgeben der Bilder