1. #Übersicht Vorgängerprojekt
   1. Hardwareaufbau mit ARM, FPGA und NIOS
2. Breezyslam
   1. Parameter des Breezyslam
   2. #Übersicht Breezyslam
   3. #Eigene Erweiterungen von uns (z.b SetMap, SetPosition)
   4. #Was ist SLAM
   5. #Wie funktioniert Breezyslam
   6. #Warum Breezyslam und nix anderes
   7. Probleme mit Breezyslam
3. Howto’s, welche Tools(version), wie einstellen, wie ist Projekt aufgebaut?
4. #Erweiterte GUI
   1. #Erweiterung Kommunikationskanal
5. #Einleitung Autonomes Fahren
   1. #Voraussetzung SLAM
6. #Lichtansteuerung/Bugfixes
7. Auswertung/Ergebnisse
8. Verbesserungen/Ausblick
   1. Evtl ROS?
9. #Sensoren des Fahrzeugs
10. #Projektanforderungen
11. Einleitung: Autonomes Fahren
    1. Voraussetzung SLAM
12. Projektanforderungen
    1. Übersicht Vorgängerprojekt
       1. Was ist vorhanden/worauf bauen wir auf? kurz
    2. Ziel unseres Projekts (Implementation SLAM)
13. Vorgängerprojekt
    1. Hardwareaufbau mit ARM, FPGA und NIOS, Sensoren (Nebensatz, Rotsensor funktioniert nicht, wird aber auch nicht gebraucht)
    2. Lidarsensor? Wie misst er, Verbreitung, Kenndaten (Geschwindigkeit)
    3. Bestehende Software: welche Komponenten gibt es, wie arbeiten sie zusammen
    4. Einarbeitung
       1. -> Bugfix Lichtansteuerung
       2. Probleme bei Einarbeitung
          1. Projektstruktur/keine Makefiles o. Projektdateien
          2. Unbekannter Compiler (ARM Projekt)
          3. Unbekannte Qt-Version
          4. NIOS2 common/ARM\_NIOS/using\_shared\_memory\_example.cpp
          5. Inbetriebnahme: falscher Port
       3. Verweis auf HowTos im Anhang
14. SLAM
    1. Was ist SLAM?
    2. Für welchen/Warum SLAM haben wir uns entschieden?
    3. Wie funktioniert Breezyslam?
    4. Unsere Anpassungen (restliche Gruppe)
15. GUI
    1. Vorhandene GUI -> Erweiterungen
    2. Aufbau und Funktion der GUI (warum wurde sie erweitert?)
    3. Erweiterter Kommunikationskanal (evtl. restliche Gruppe)
16. Wie/was wurde getestet (Gruppe)
    1. Positionsfindung durch Lidarsensor
       1. Anpassung logmovie, damit er Sensor einliest
    2. Erstellen der Karte mit ferngesteuertem Auto
       1. Wie wurde bestehender Code angepasst und wie funktionierte er? (z.B speichern der Karte nach 1 Minute)
    3. Erstellen der Karte mit getragenem Auto
       1. Programm: Karte wird nach Tastendruck erstellt
    4. ->Problem mit Drehung
    5. Geschwindigkeitsmessung für Odometriedaten
       1. Anpassung GUI: 3 Sekunden fahren
       2. Wegen Beschleunigung schwierig
       3. Evtl. MPU Daten/Gyroskope
17. Ergebnisse der Tests
    1. Breezyslam mit versch. Parametern (z.B mit und ohne Odometrie)
    2. Erstellte Karten und Bewertung/Abweichungen vom Optimum
18. Ausblick und Verbesserungen
    1. ?
19. Anhang: Howto‘s