Autonomous Indoor Drone

Team IMP
Rafael Mäuer et al.

Agenda

1. Goal
2. Current State
3. Hardware
4. Software
5. Problems and Limitations
6. Next Steps
7. Live-Demo

1. Goal

- Construction of an autonomous, indoor flying drone
- The drone should fly to goals and avoid obstacles
- A map of the room will be created by the drone and visualized by an application

2. Current State

- Hardware
  - Drone is built
  - Drone flies with RC
  - Problem with Trim-Calibration
  - Power-Supply is stable
  - Sensors deliver values
- Software
  - Simulation
  - Lift off
  - Creating a map in real-time
  - Flight towards point on map
  - Stabilization

2. Current State

3 Hardware

- Drone
  - Necessary Parts
  - Schematic (wiring)
  - Build process
- Sensors
  - 2D Laser Scanner
  - US Distance sensors

3.1 Hardware - Drone

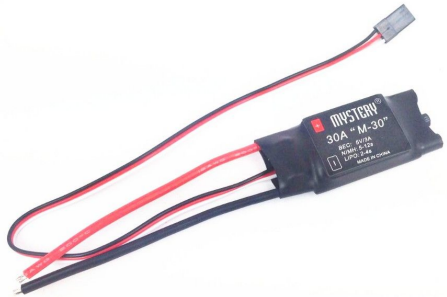
- Drone Parts
  - Basic Components
  - Flight Controller
  - Companion Computer
  - Power-Supply
  - Remote Control
  - Sensors
  - Status-LED
- Schematics
- Build Setup

3.1 Hardware - Drone

- Main Frame

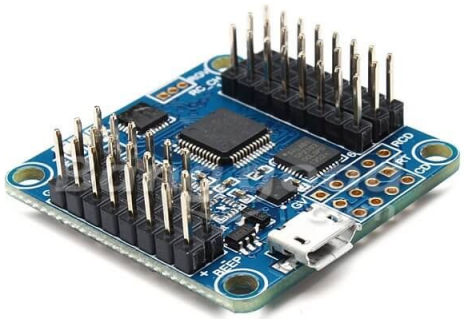
3.1 Hardware - Drone Parts

- 4x Motors, Props, Prop-Guards, ESCs

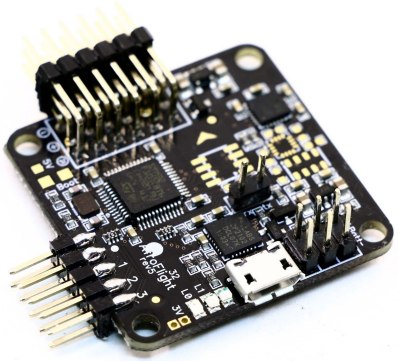


3.1 Hardware - Flight Controller

- F1 Flight Controller
- ROSflight firmware



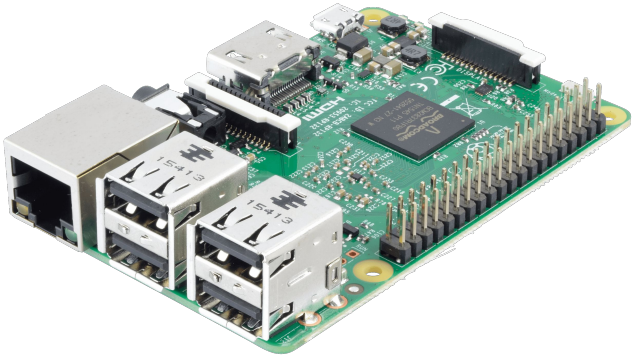
Flip32



Naze32

3.1 Hardware - Companion Computer

- Companion Computer
  - Ubuntu, ROS

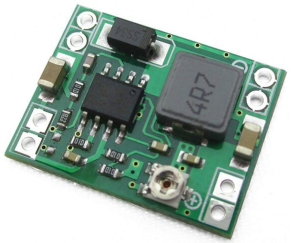


Raspberry Pi 3

Nano Pi Neo Air

3.1 Hardware - Power Supply

- LiPo-Battery
- Power-Converter
- LiPo-Alarm



3.1 Hardware - Control

- Flysky Receiver
- Flysky Transmitter

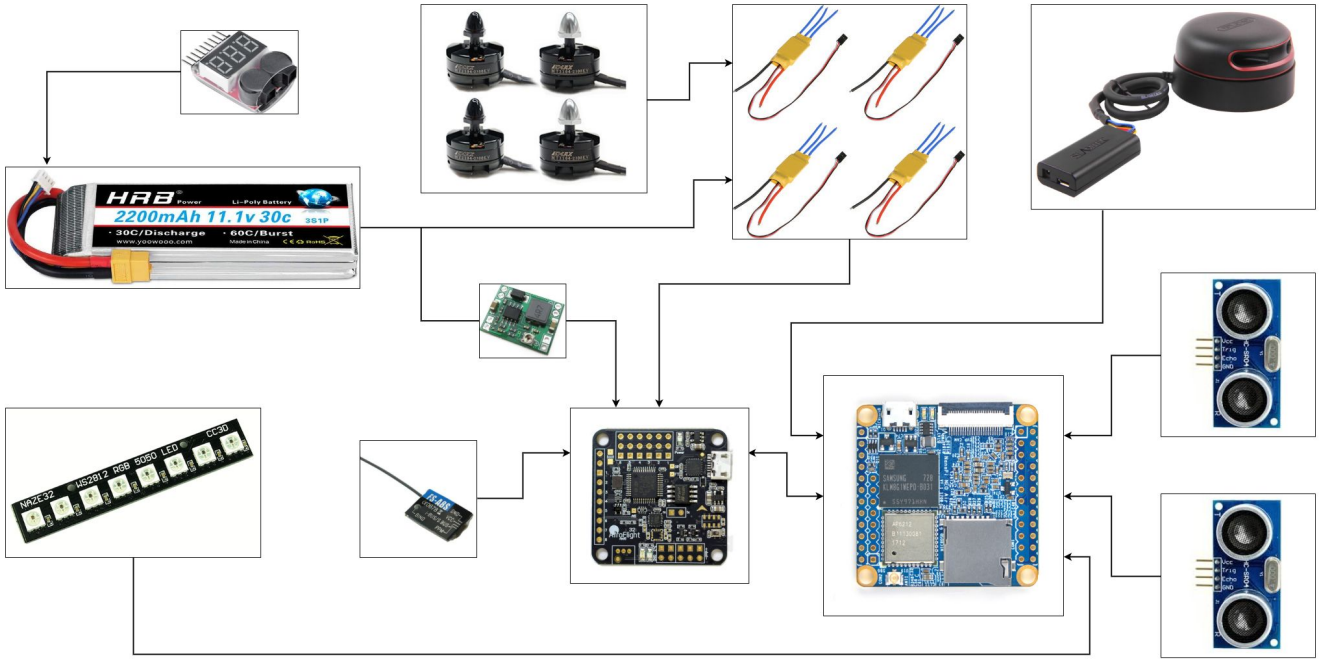
3.1 Hardware - Sensors

- Ultrasonic (HC-SR04)
- Lidar (RPLIDAR A2)

3.1 Hardware - Status LED

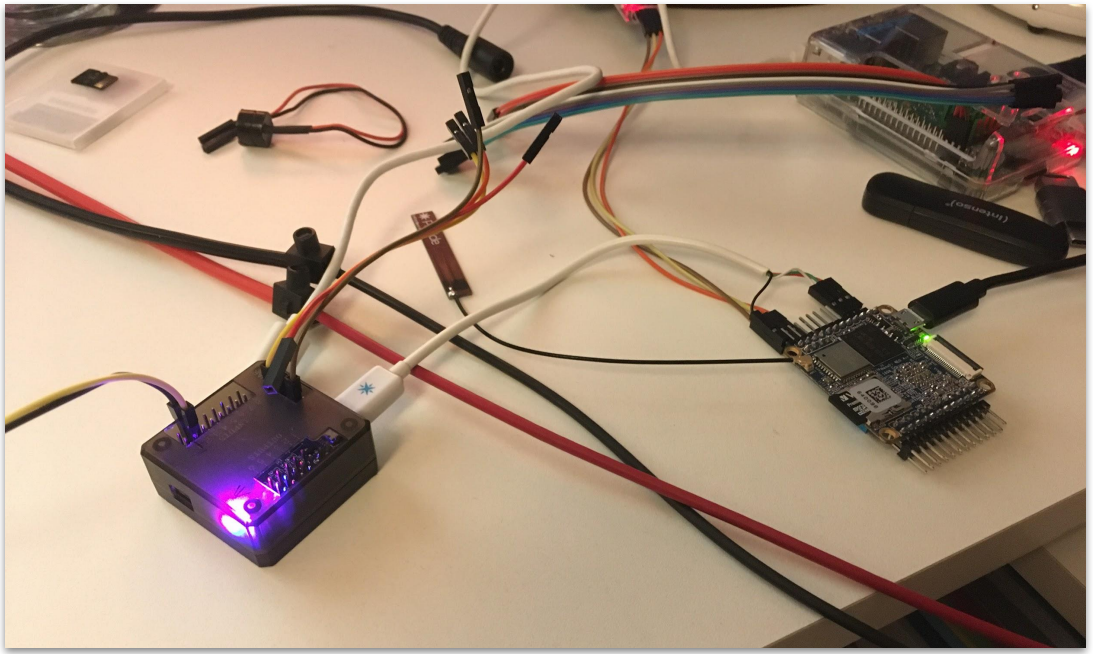
- Naze32 WS2812 RGB-Strip
  - 1: Operating System
  - 2: ROS
  - 3: Flight Controller
  - 4: IMU Calibration
  - 5: Ultrasonic Sensors
  - 6: LIDAR Sensors
  - 7: Remote Control
  - 8: ARM Status

3.2 Hardware - Schematic



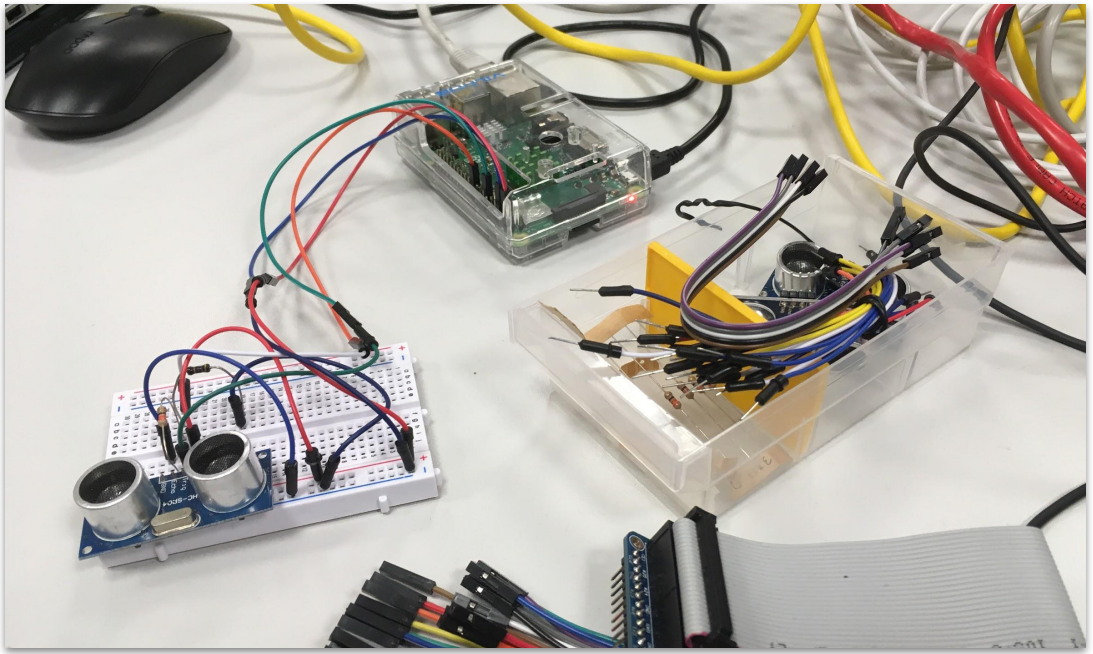
3.3 Hardware - Tests

- Establish communication between Flight-Controller and Companion-Computer



3.3 Hardware - Tests

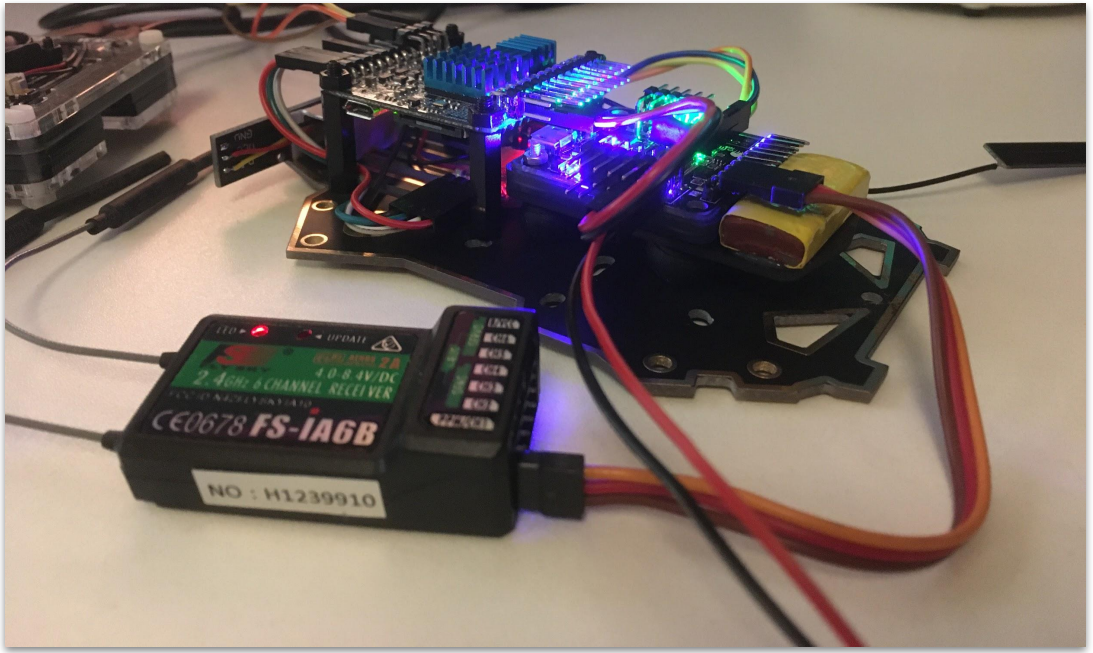
- Getting data from Ultrasonic Sensor



3.4 Hardware - Build 1

- Setup Rev. 1

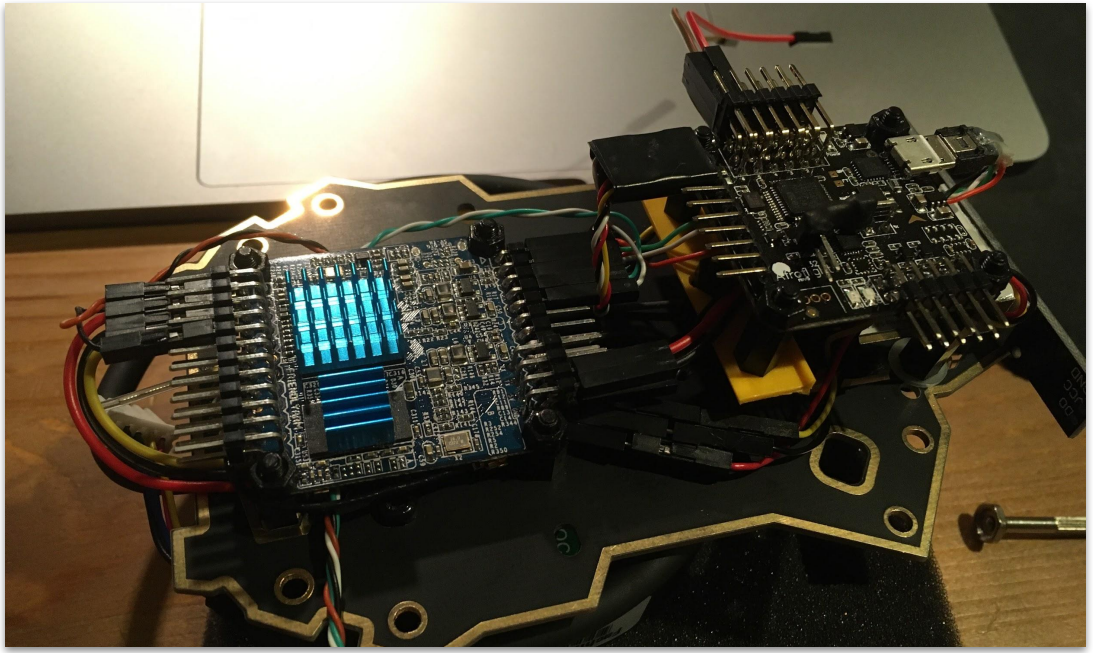
- FC mounted with rubber to avoid vibrations
- Doesn’t fit due to cable pinout positioning



3.4 Hardware - Build 2

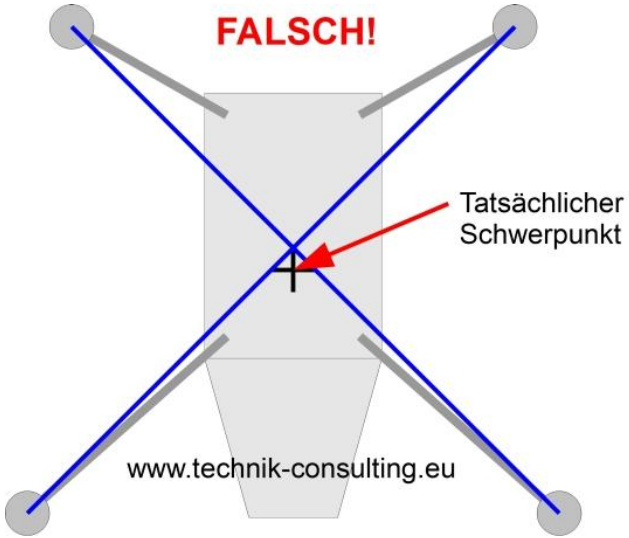
- Setup Rev. 2

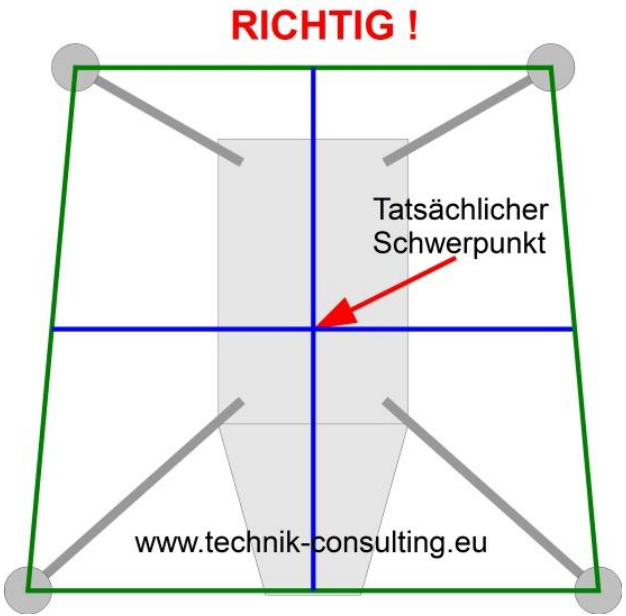
- Change Positions of Flight Controller and Companion Computer
- Better position for correct Center of Gravity



3.5 Hardware - Center of Gravity

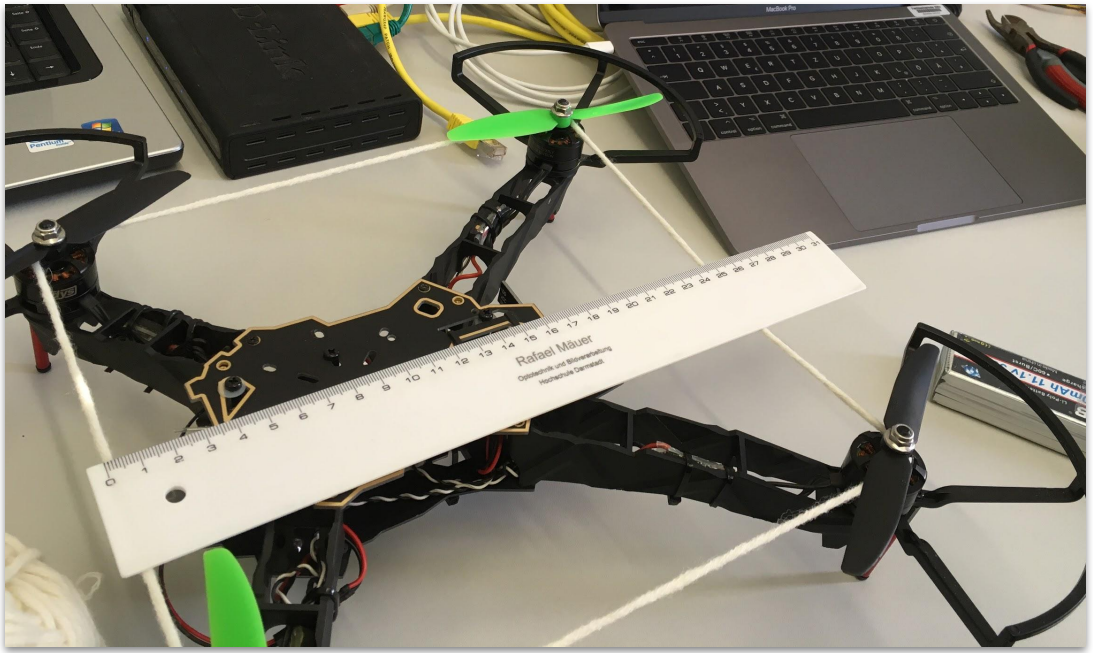
- Center of Gravity





3.5 Hardware - Center of Gravity

- Center of Gravity



3.6 Hardware - Sensor Mounting

- Lidar Sensor
- Ultrasonic Sensors

3.7 Hardware - Sensors - Horizontal

- Horizontal 2D Laser Scanner using RPLidar A2 by Slamtec
  - Range: 0.2m - 12m
  - Angular Range: 0° - 360°
  - Scan Rate: 10Hz
  - Scan Range Flatness: -1.5° - 1.5°
  - Distance Resolution: <0.5mm [< 1.5m] ; <1mm
  - Angular Resolution: 0.9° [@10Hz]
  - Laser Class: FDA I

3.7 Hardware - Sensors - Horizontal

- Published Data: *sensor_msgs/LaserScan*
  - Timestamp
  - Angle informations
  - Range informations
  - Range- & intensity data

- ROS node from existing package “rplidar_ros” by Slamtec

3.8 Hardware - Sensors - Vertical

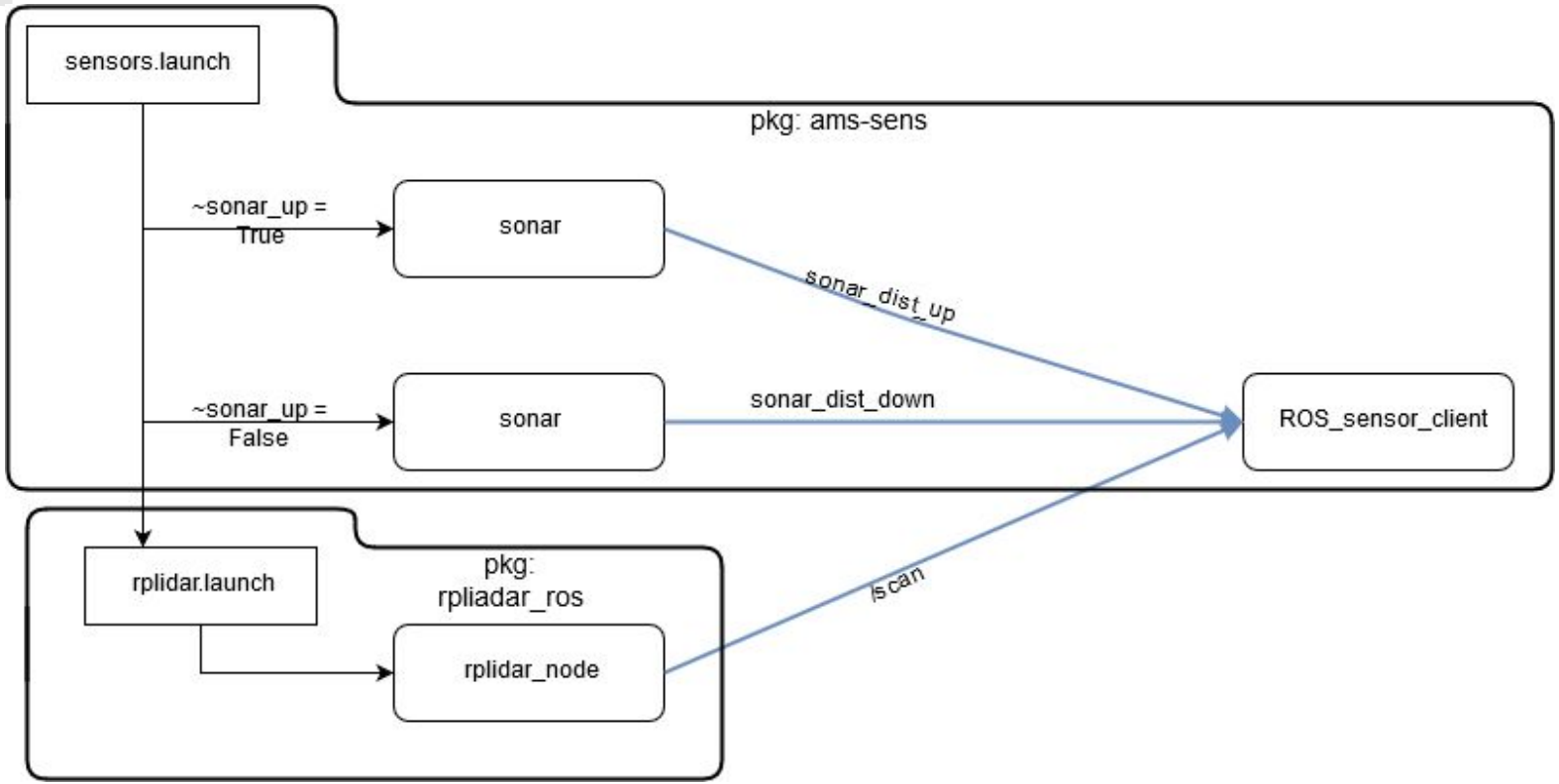
- Vertical distance scan using HC-SR04 ultrasonic sensor
  - Distance Range: 2 cm - 400 cm
  - Measurement Angle: 30°
  - Resolution: 0.3 cm
- Scan Data: *sensor_msgs/Range*
  - Timestamp
  - Radiation type [US / IR]
  - Distance Range
  - Distance Value

3.9 Hardware - Sensors - Scan Data

- ROS node for measuring US Sensor and calculating distance
  - Started twice with different setup variables - Up-/Down-Sensor
  - Publishing data in three different ROS topics
    - Laser scan: “/scan” , Type: *sensor_msgs/LaserScan*
    - US upwards: “sonar_dist_up” , Type: *sensor_msgs/Range*
    - US downwards: “sonar_dist_down” , Type: *sensor_msgs/Range*

- Importing Rplidar launch file in custom sensor launch file and starting all nodes together

3.9 Hardware - Sensors - Scan Data



4 Software

- ROSflight
- Simulation
- Mapping
- Navigation
- Drone Control
- Overview
- Client Application

4.1 Software - ROSflight

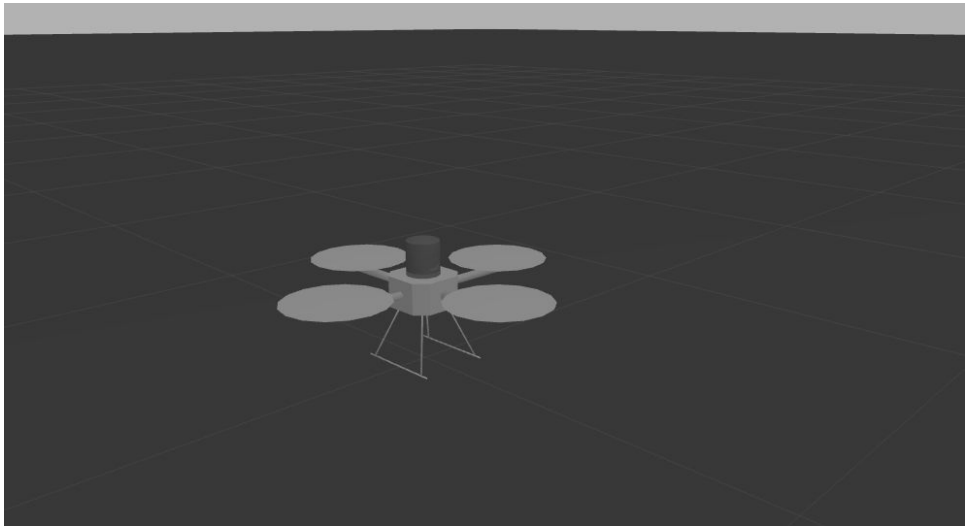
- Software for flight vehicles based on ROS
- Consists of two parts
  - Firmware on Flight-Controller
  - ROS node on Companion Computer
  - Communication via Serial (MAV-Link)

4.2 Software - Simulation

- Drone: ROSflight Sim (Gazebo Simulation) including IMU and Sonar sensors
- Remote Control: ROSflight Joy (Keyboard/Joystick RC)
- Lidar Sensor: Velodyne Laser
- Room: Gazebo Model (created with Building Editor)

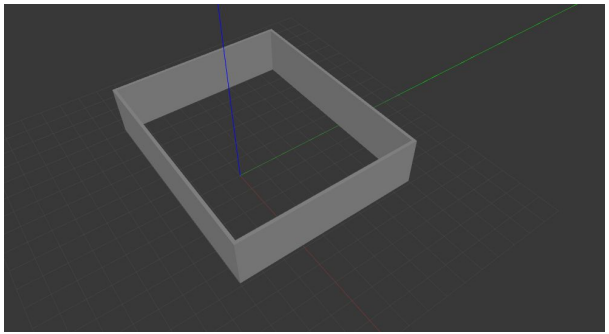
4.2 Software - Simulation

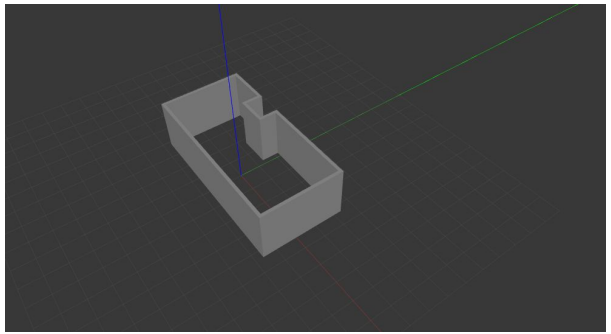
- Drone + Lidar:

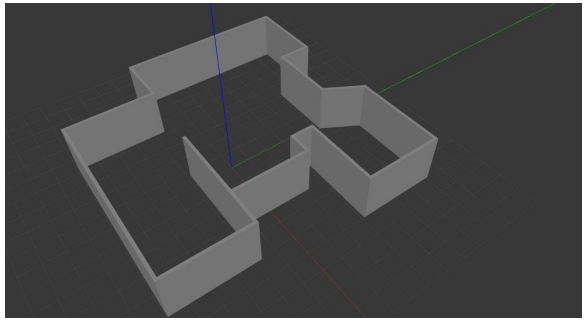


4.2 Software - Simulation

- Rooms:







4.3 Software - Mapping

- Common ROS package: gmapping
  - Problem: no odometry data given

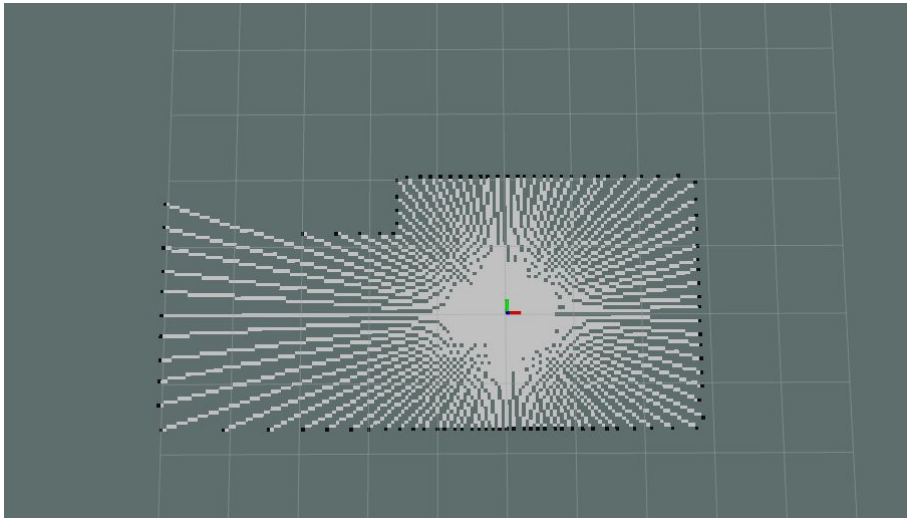
- Solution: Hector Mapping
  - Requirements: Laser and IMU data
  - Features:
    - Map
    - Robot position

4.3 Software - Mapping

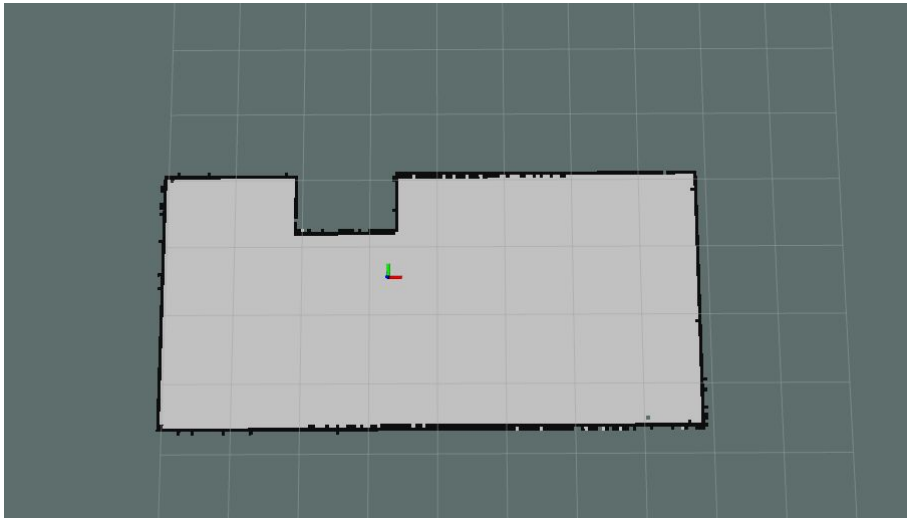
- Trade-offs:
  - Room should be irregular and not too big
  - Map can not be loaded from storage
    - needs to be created again every time
  - Creation of map unstable in the beginning

- High noise of the robot position, especially in the early map creation phase
  - Kalman Filter for reducing the noise

4.3 Software - Mapping



Map creation

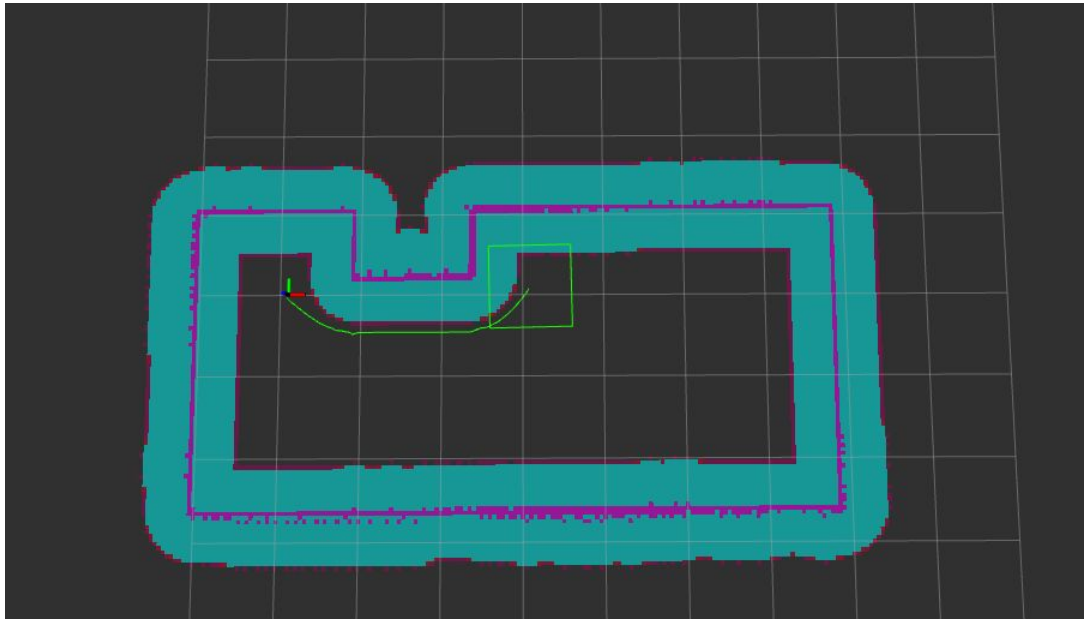


Final map

4.4 Software - Navigation

- ROS navigation stack
  - Requirements:
    - ROS move_base package
    - Planner and Costmap parameter (.yaml) files
    - Map
    - Robot position
  - Features:
    - Path
    - Velocity commands

4.4 Software - Navigation



Cost map & path

4.5 Software - Drone Control

Take-off:

- Sonar sensor for estimating the height
- Setting drone throttle to keep height

Landing:

- Reducing drone throttle

4.5 Software - Drone Control

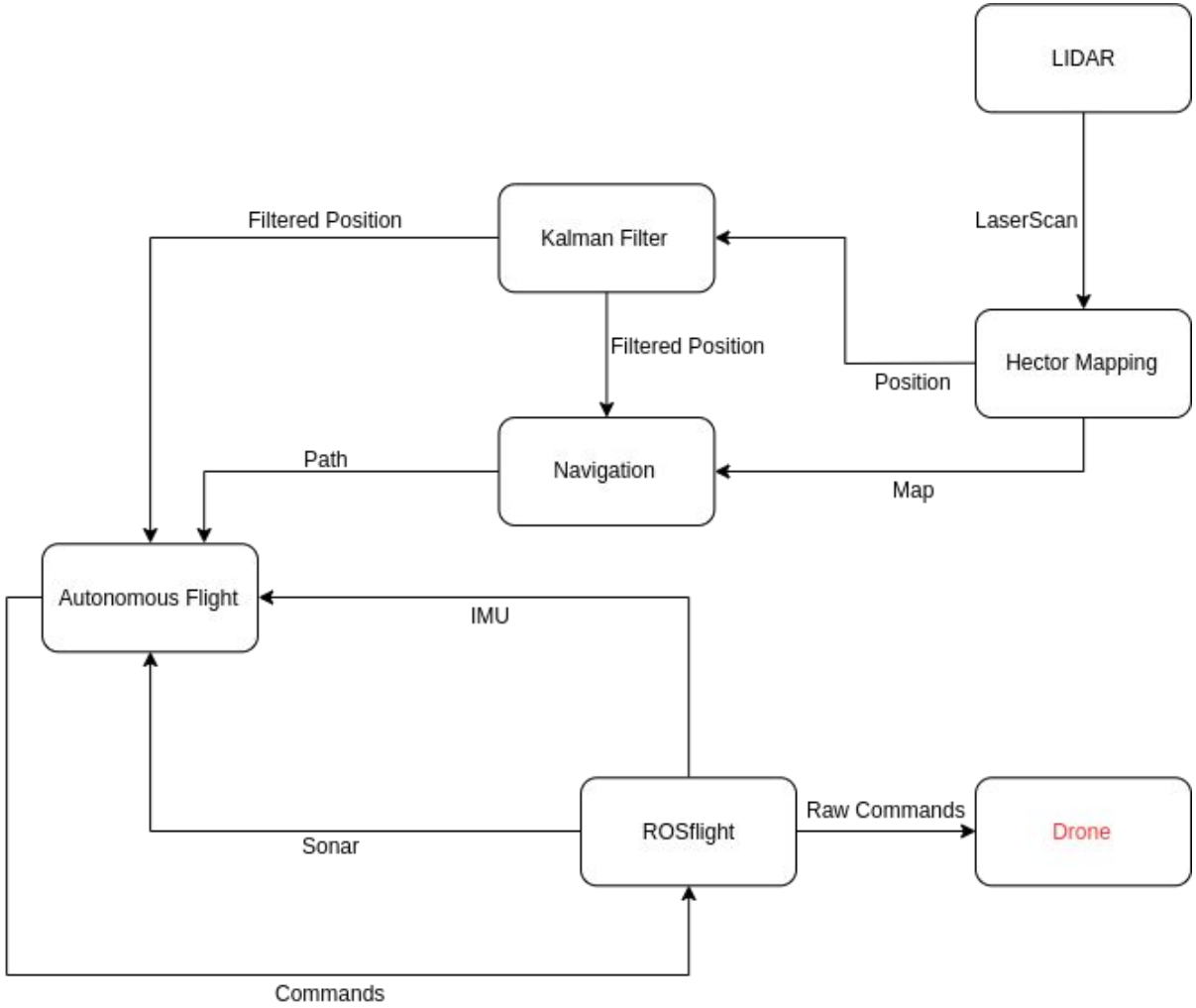
Path following:

- Navigation path needs to be translated into drone commands
- Velocity commands seemed to be more difficult to handle
- An algorithm was implemented to follow single points
  - Filtered position from hector mapping as reference
  - Roll and pitch were set according to reach a particular x-y-coordinate within a given tolerance
  - IMU acceleration was used to control velocity

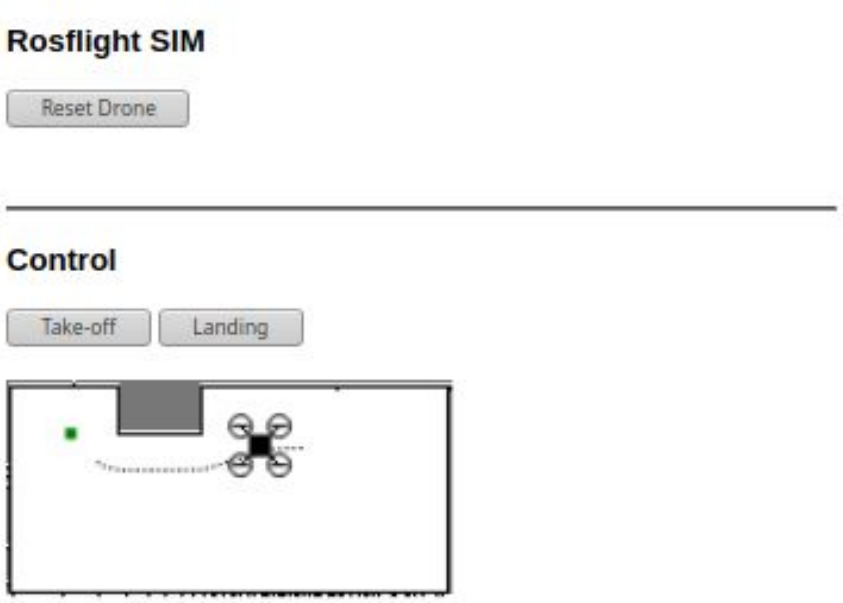
4.6 Software - Overview

ROS packages:

- velodyne
- hector_mapping
- hector_imu_attitude_to_tf
- navigation


LIDAR
LaserScan
Kalman Filter
Filtered Position
Filtered Position
Hector Mapping
Position
Navigation
Path
Map
Autonomous Flight
IMU
ROSflight
Raw Commands
Drone
Sonar
Commands

4.7 Software - Client Application



5. Problems and Limitations

- Drone:
  - Very limited space to fit all parts into/onto frame
  - Two build setup revisions were necessary to find working setup
  - Serial communication with Naze32 only working with USB
  - PID adjustments problematic due to automatic TRIM correction
  - Calibration not possible while LIDAR is running (movement)
- Companion Computer
  - Different software versions on RasPi and NanoPi (Ubuntu, ROS)
  - Software library for status LED Strip (missing support for NanoPi)

5. Problems and Limitations

- Sensor measurement:
  - Ultrasonic: Custom launch file prevents exit of services
  - Lidar: Motor not stopping when stopping linux service
- Software:
  - Currently only 2D map
  - Simulation:
    - Not the same LIDAR and Sonar as on the real device
    - Clean room without obstacles not a real scenario
  - Fixed height

6. Next Steps

- Wiki/Documentation
- Open Source it
- Fix problem with automatic Trim-Calibration
- Adaption of software to work with real vehicle

7. Live Demo

Resources (Images)

- http://d1f3xz7hjchz45.cloudfront.net/images2/q330/storm-q330-big017.jpg
- https://fpv.tv/wp-content/uploads/2014/12/Flip32-Flight-Controller-With-32-bit-STM32-and-Naze32-for-Multirotors-no2.jpg
- https://3bn82z33ols32bjthc3grk4d-wpengine.netdna-ssl.com/wp-content/uploads/2014/11/IMG_1218.jpg
- https://cdn-reichelt.de/bilder/web/xxl_ws/A300/RASP_03_01.png
- http://www.friendlyarm.com/image/catalog/details/air_02.jpg
- https://www.arrishobby.com/images/MT-EMAX-MT2204-KV2300-2-Pairs.jpg
- https://ae01.alicdn.com/kf/HTB1p5NQMpXXXXb5XVXXq6xXFXXXW/200013145/HTB1p5NQMpXXXXb5XVXXq6xXFXXXW.jpg
- https://images-na.ssl-images-amazon.com/images/I/4149DaXvfIL._AC_.jpg
- https://images-na.ssl-images-amazon.com/images/I/51T0HfnKY2L._SL1024_.jpg
- https://images-na.ssl-images-amazon.com/images/I/51phenXYoaL._AC_SL1001_.jpg
- https://images-na.ssl-images-amazon.com/images/I/61ZEGIXch5L._AC_SL1200_.jpg
- https://sep.yimg.com/ay/yhst-90268561309754/arris-ws2812b-rgb-5050-full-color-led-lighting-board-for-naze-32-cc3d-flight-controller-19.gif
- https://images-na.ssl-images-amazon.com/images/I/71%2BBOf-%2BNvL._AC_SL1290_.jpg
- https://cdn-global-hk.hobbyking.com/media/catalog/product/cache/1/image/660x415/17f82f742ffe127f42dca9de82fb58b1/legacy/catalog/77931_1_high_2__3.jpg
- https://images-na.ssl-images-amazon.com/images/I/61bAVEuVjdL._AC_SL1002_.jpg
- https://www.technik-consulting.eu/kategorien/Optimierung/dateien/Diagonalmethode.jpg
- https://www.technik-consulting.eu/kategorien/Optimierung/dateien/Schwerpunkt-richtig.jpg
- https://i.ebayimg.com/images/g/4GcAAOSw~gZcj6DK/s-l300.png
- https://images-na.ssl-images-amazon.com/images/I/61-2fYKuyKL._SY355_.jpg