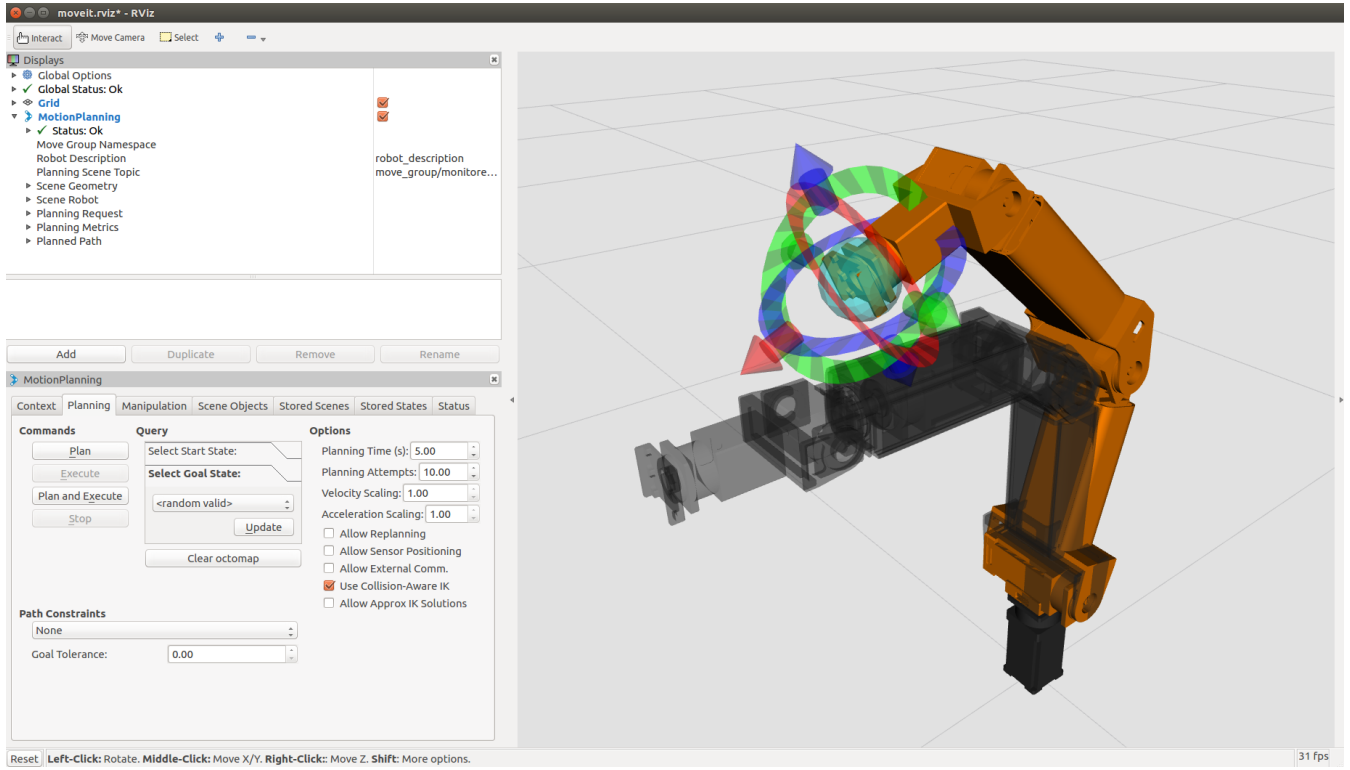
Improving the DIY SCARA robot arm

Bioscara v2

Aim

- Enable DIY Arm for Lab Integration
- ROSify the SCARA Arm
- Easy Manipulation through MoveIt



Automation Pipeline

Closed Loop Control
Ustepper S32 Firmware

I2C driver
lgpio

Communication

I2C

Communication Protocol

Communication API

Homing/Stall Reset

ROS2 Control Hardware Interface

ROS2 Control

(Homing/Stall Reset)

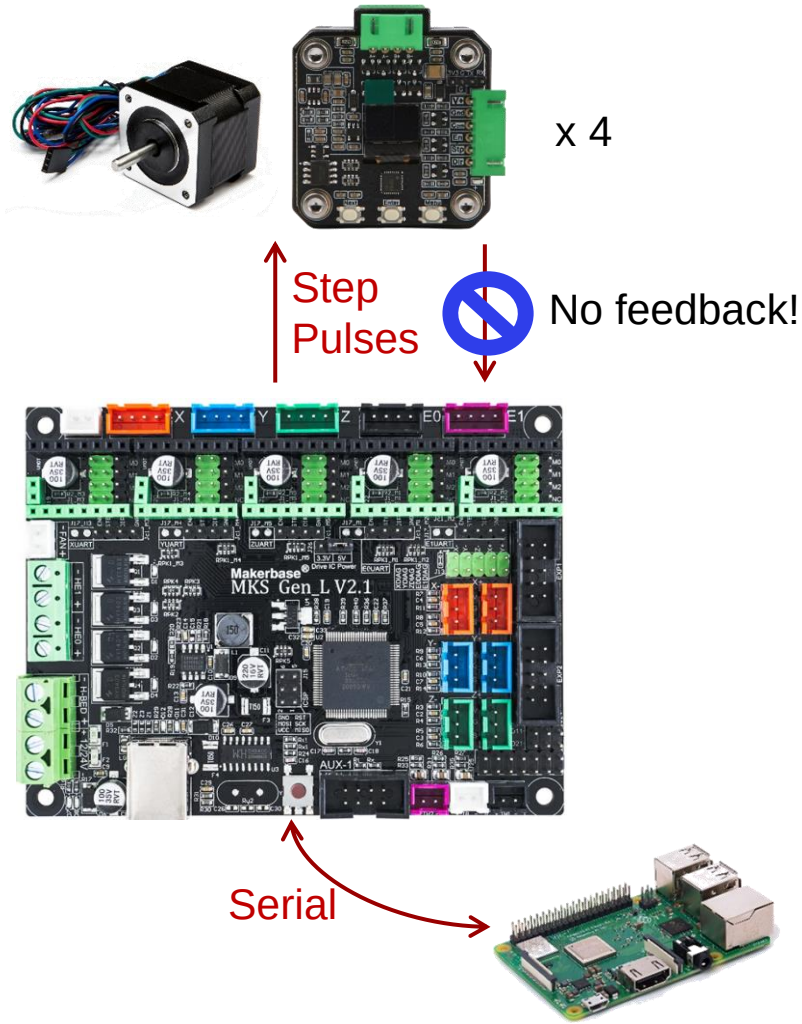
MoveIt Motion Planning

Rviz GUI

Motion Sequencer

Protocol	Controller Level [Location]	Methods
SiLa SiLa	Automation Scheduler [Lab] Action Controller [Arm]	**Task** – Transfer labware **Action** – pick – place
ROS-SiLa Bridge: SiLa Server & ROS Client		
ROS2	Motion Sequencer [Arm]	**Motion Sequence** – follow trajectory sequence – grip labware
ROS2 (*MoveIt*)	Motion Controller [Arm]	**Motion Primitives** – follow trajectory (joint/cartesian) – get/set gripper width
Hardware Interface (Custom/Manufacturer Provided)		
I2C, UART, CAN, ...	Embedded (Sensors/Actuators) [Joint]	**Joint Primitives** – get/set joint position/velocity – get/set gripper joint position

Architecture (old)



MKS Servo42C

- NO support/ Poor Doc.
- Buggy Firmware
- No sensorless homing

- Stall detection but no feedback to controller!
- Serial control advertised but not realisable since position request **cancles** movement
- Potentially Step Signals (3 wires)
  + Parallel UART request to read state? (2 wires)

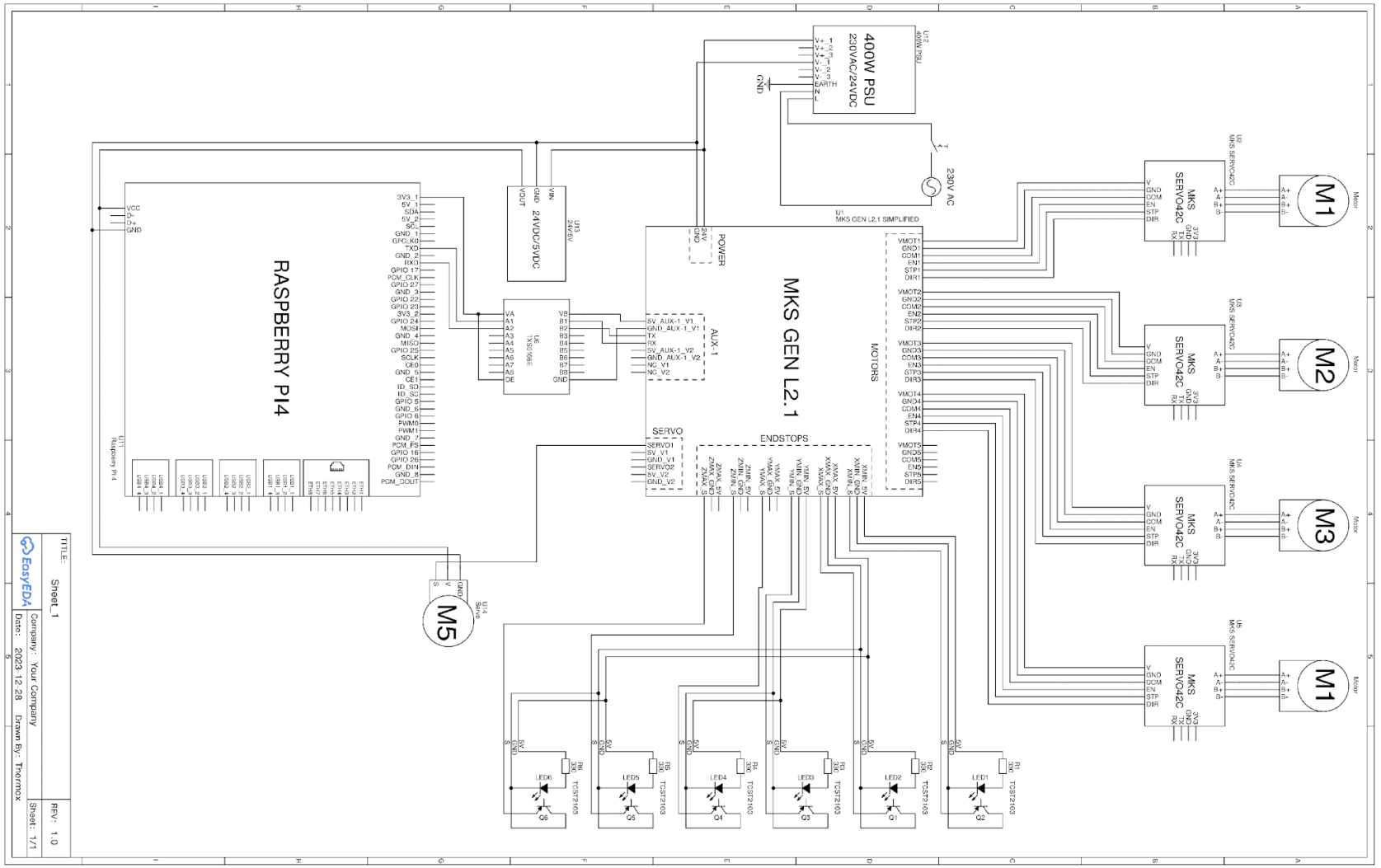
MKS Printer Board

- Marlin 3D printer firmware
- Slightly modyfied for 4th axis but still ”thinks” its a 3D printer
- G-Code via Serial from Raspberry Pi
- Emulates postion feedback

Raspberry Pi

- Handles Inverse Kinematics
- Communication
- Python Scripts

Old Schematic



4x Joint each:

- 2x Power
- 4x Signal

1x Servo

- 2x Power
- 1x Signal

6x Endstop

- 2x Power
- 1x Signal

==========

45 cables

Architecture (new)

x 4

I2C

feedback!

Ustepper S32

- Freely programmable Stepper driver

+ Lots of freedom

- More work

- Many different ways for stall detection but tricky to use.
- Sensorless Homing Possible!

Raspberry Pi

- Custom Communication Protocol
- Robot Kinematics are handled by ROS2 - MoveIt

Change of Focus	Old	New
Custom	• Robot Kinematics • Trajectory Generation	• Joint – Controller Interfacing
Pre-made	• Joint – Controller Interfacing	• Robot Control and everything above

+2088 dkk Ustepper S32

-470 dkk MKS Servo 42C

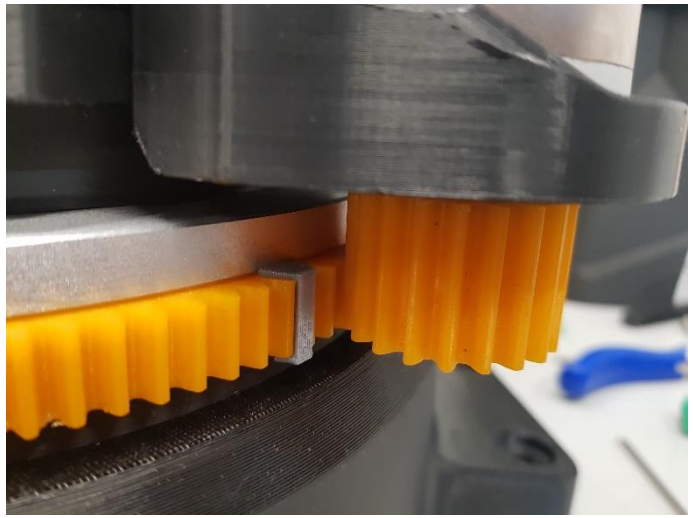
-207dkk Printer Board

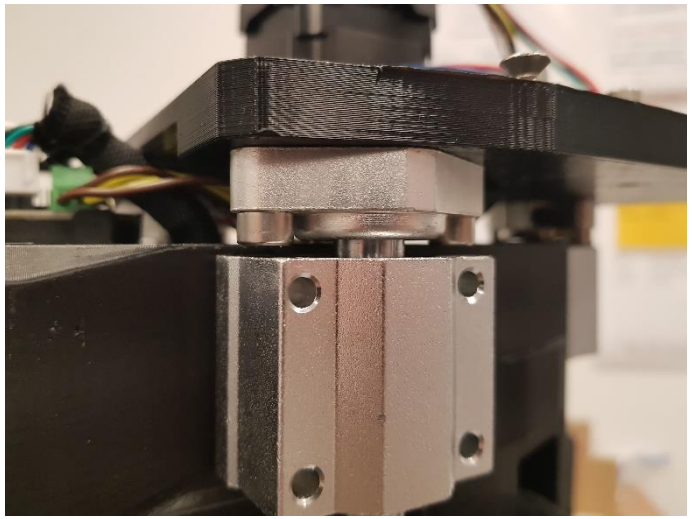
-108 dkk logic converter

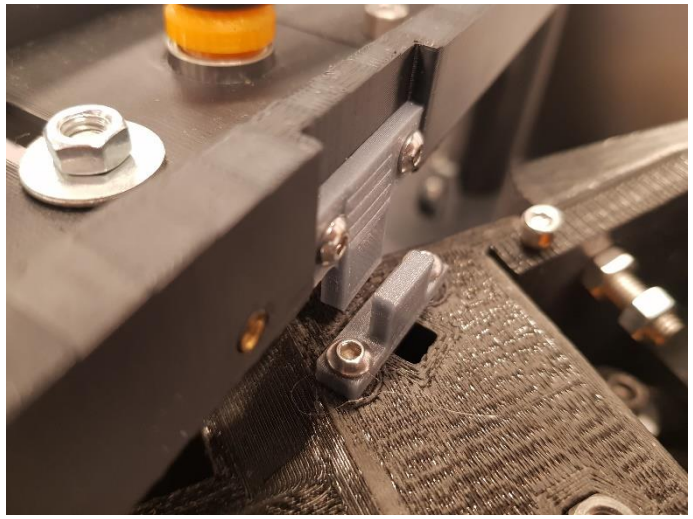
========

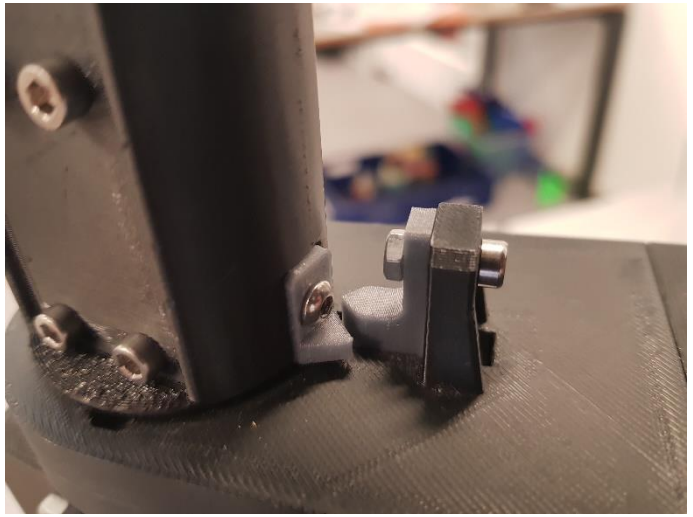
+1303 dkk (4500 dkk -> 5800 dkk)

Sensorless Homing





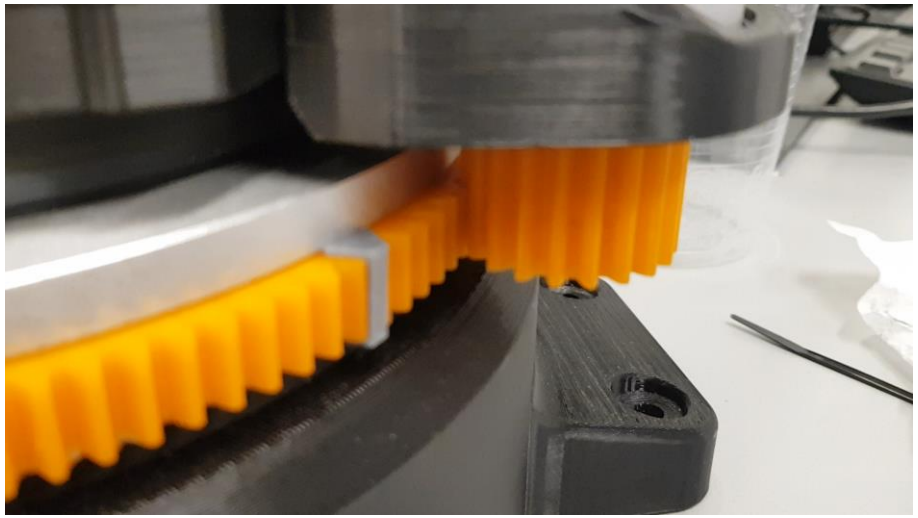


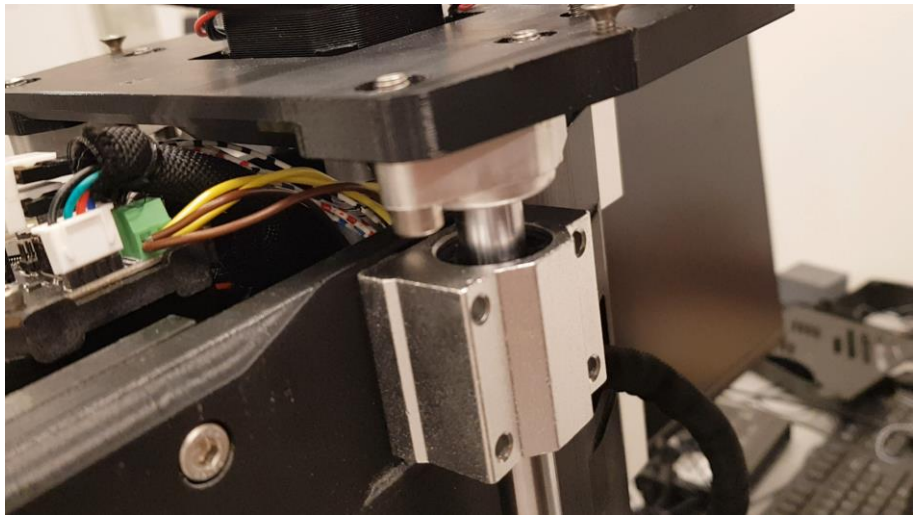


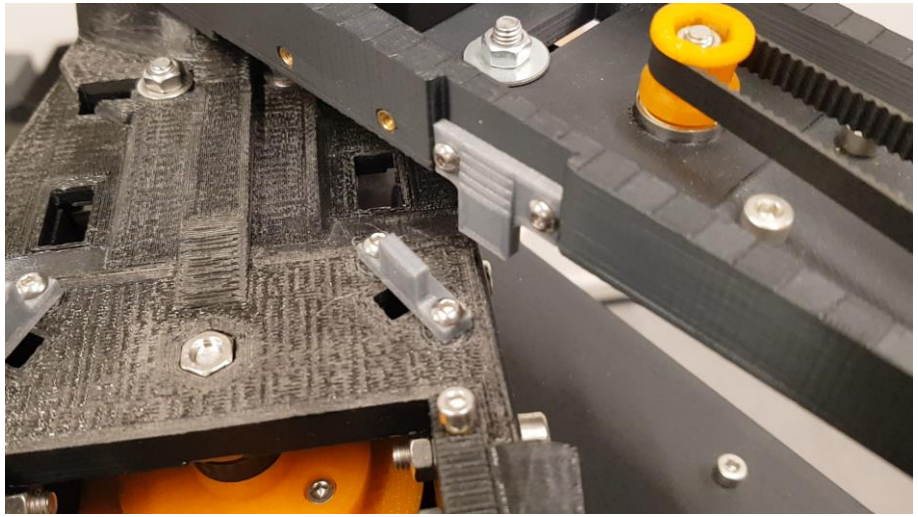
Stall detection via encoder error

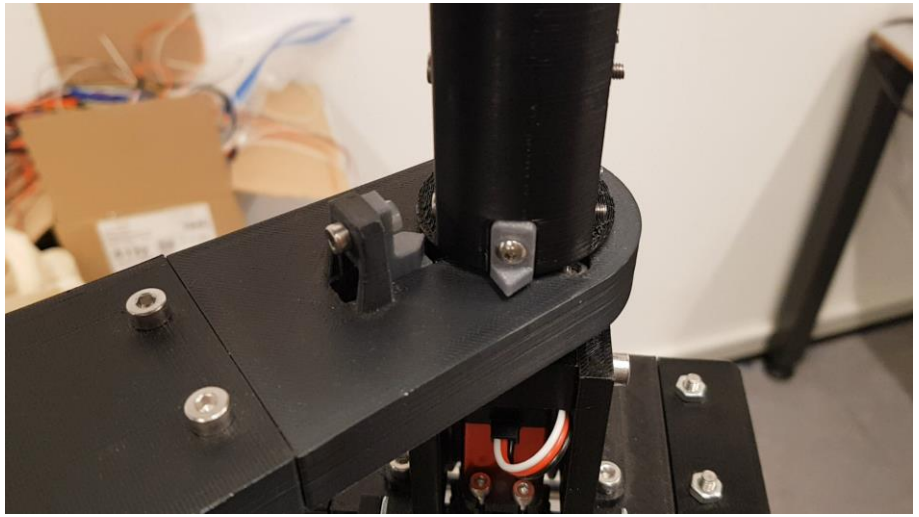
- Adjustable Speed, direction and sensitivity
- Reduced Range on J1 and J4
- Repeatability? Accuracy?

Videos

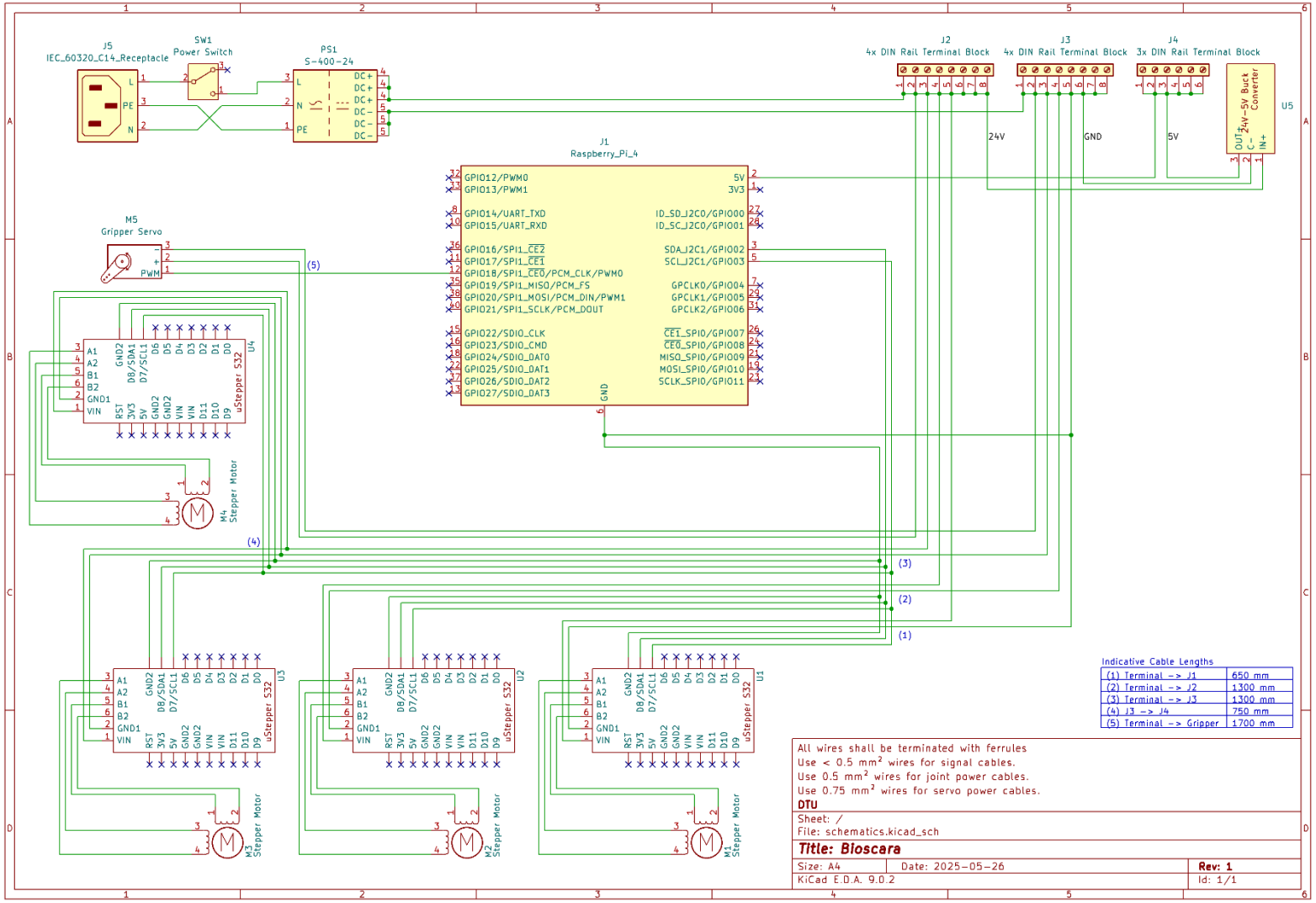








New Schematic



Sensorless Homing removes need for endstops

BUT limits freedom: no continous joints.

4x Joint each:

- 2x Power
- 2x Signal

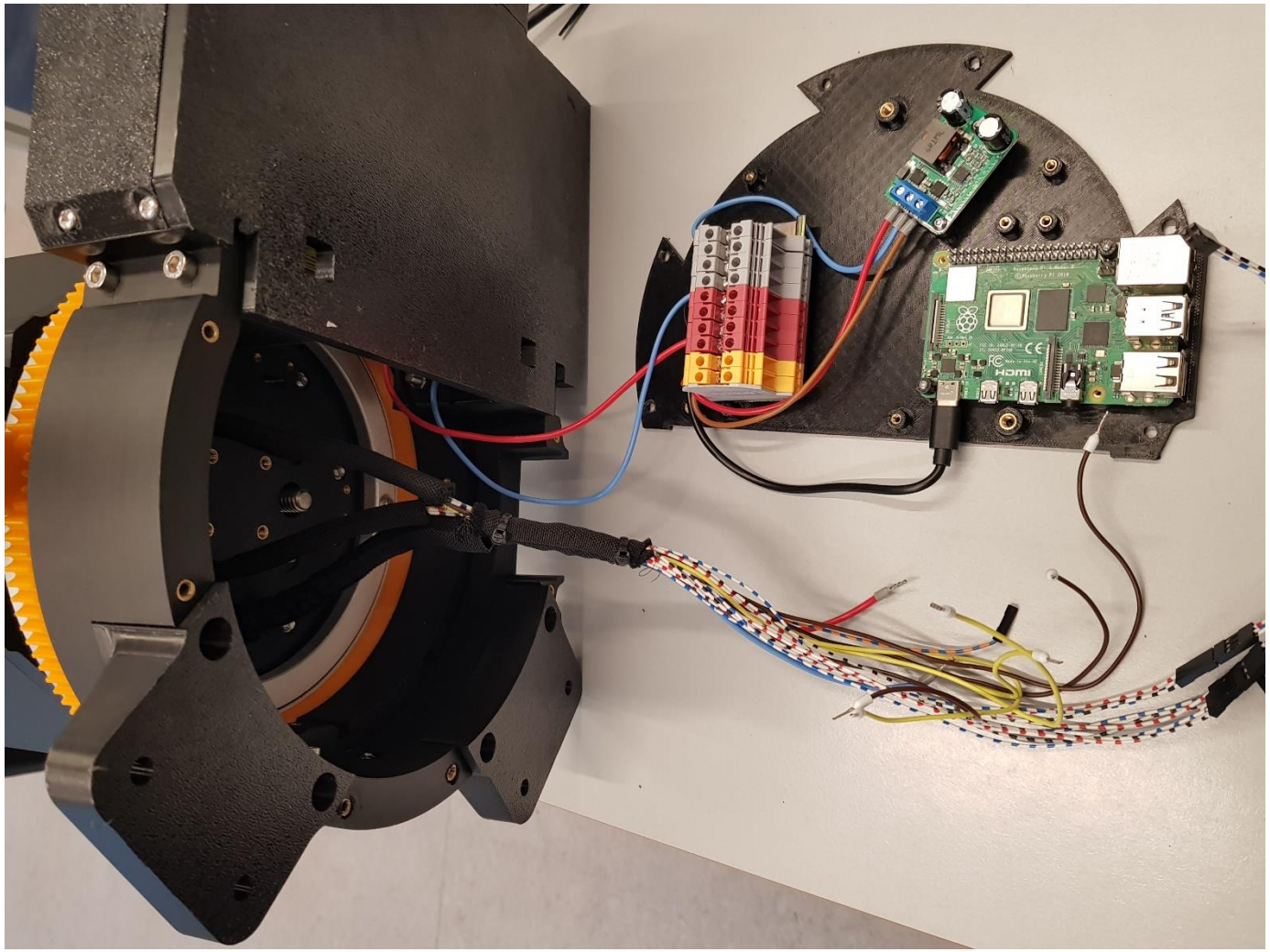
1x Servo

- 2x Power
- 1x Signal

==========

19 cables

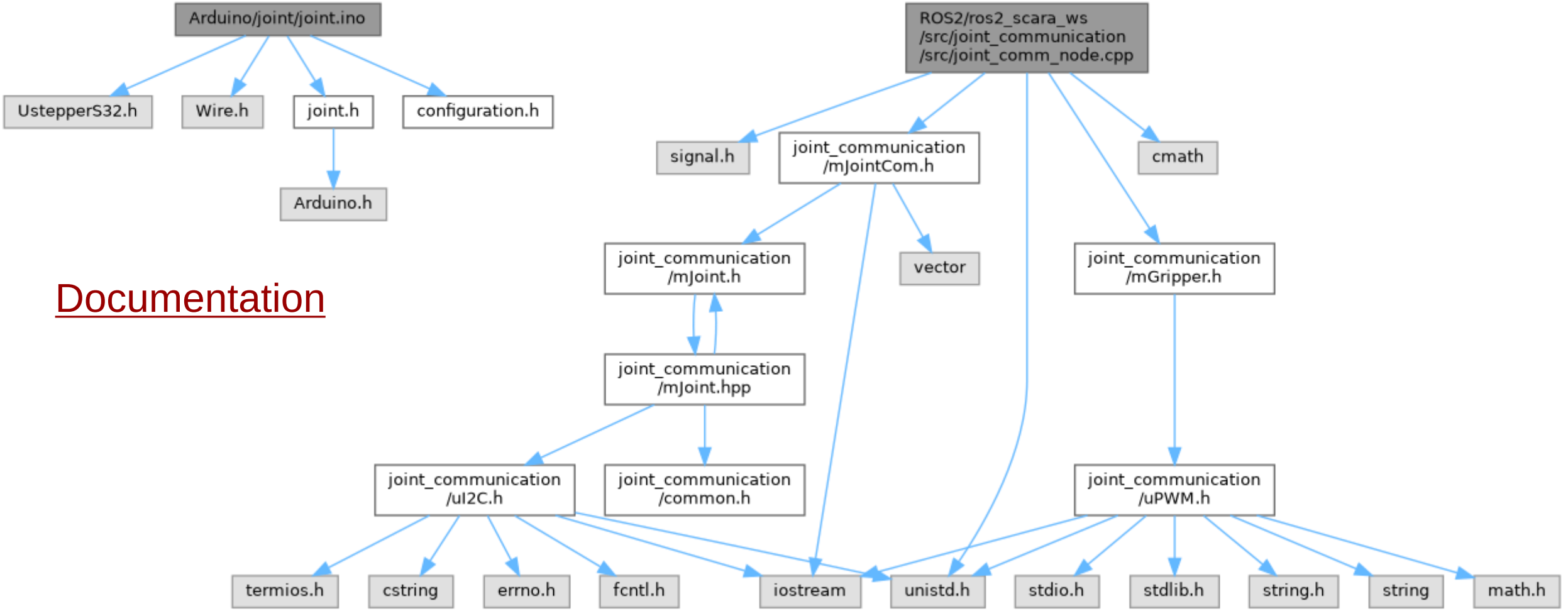
58% decrease



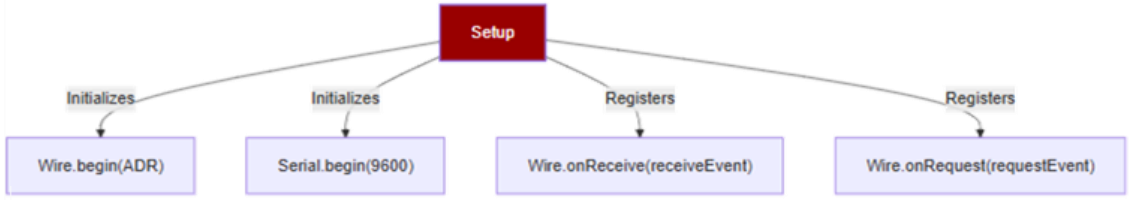
- Power Distribution via Terminal Blocks
- Signal cables from Raspberry Pi with Y-splitter

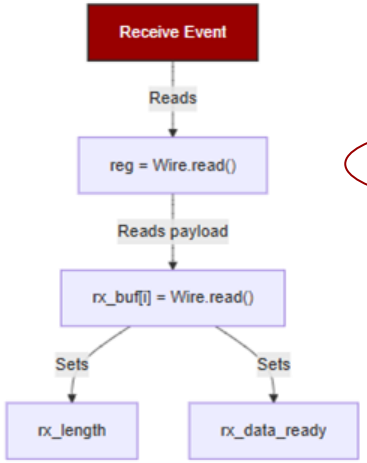
Code Dependencies and Documentation

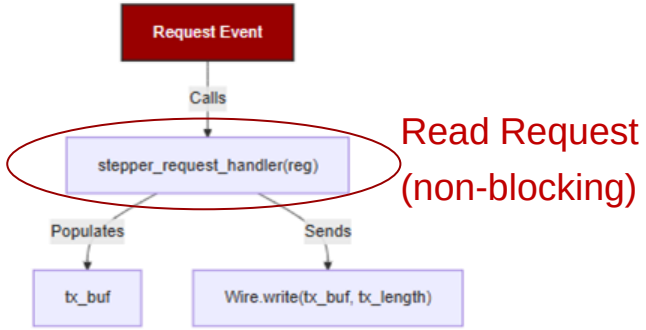
Documentation



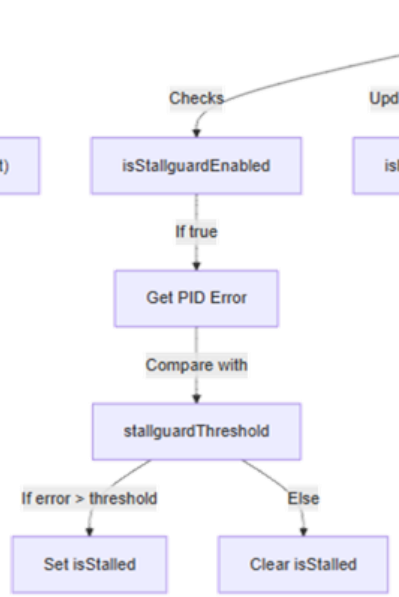
Joint Communication Protocol







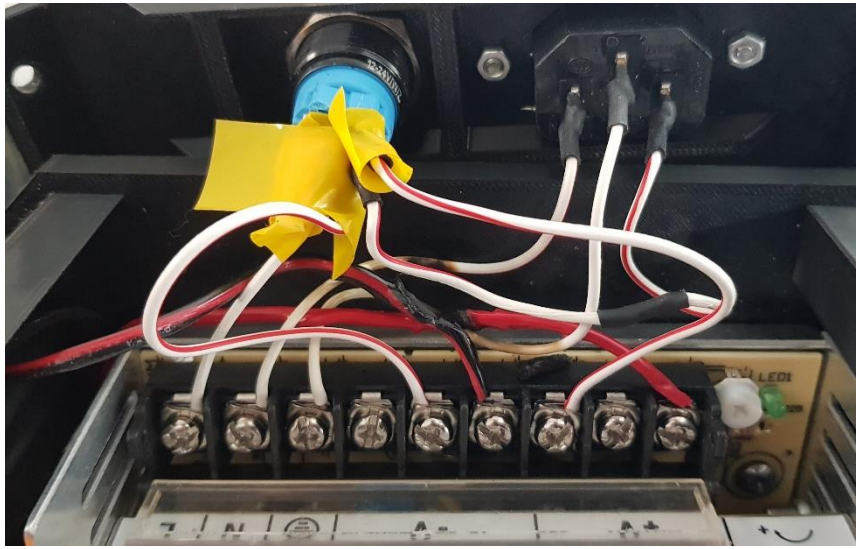
Read Request
(non-blocking)



Commands
(blocking)

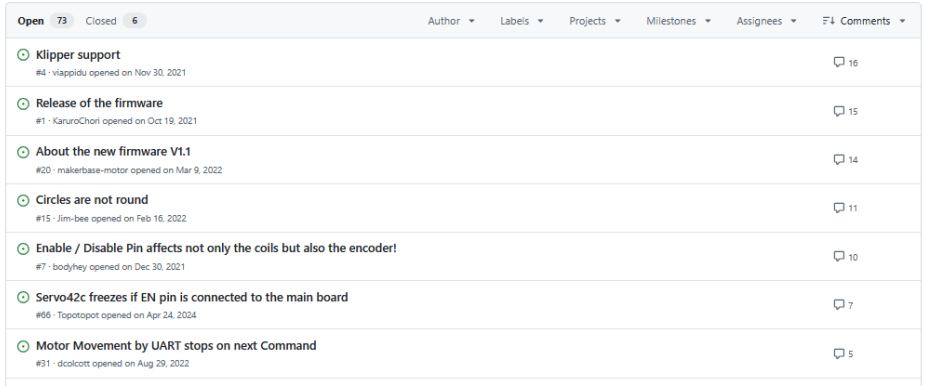
Buy Cheap, buy twice

- YES it is possible to source the cheapest components from AliExpress BUT:



Another Example:

- MKS Servo42C firmware issues
- Power button welded close



TODO

- Mechanical:
  - J3 lower arm to newest version
  - J3 tension pulley slip:
    - Fix with set screw or similar
    - Increase friction by roughing surface
  - Gripper redesign with sufficient clearance

- Test Accuracy and Precision of sensorless homing
- Test Robustness of Communication Protocol
- Homing Script (through ROS or as plain script)

- For motion with MoveIt:
  - Create ROS2 Control Hardware Interface
  - Create URDF description file
  - Adapt simple example