Motor Control Logic
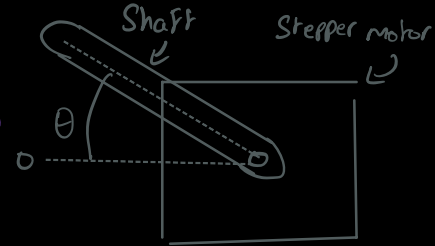
↳ Goal: get every motor to θ simultaneously
↳ target angle

↳ Given: AccelStepper library

Shaft   Stepper motor

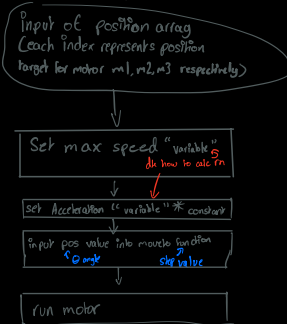**AccelStepper library functions:** ★ that i think are useful
↳ run: poll motor and step/return true if step due
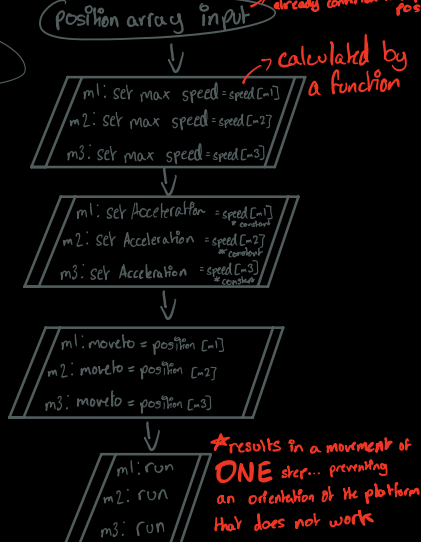↳ moveTo: set target pos
↳ setSpeed: set max speed (run function will attempt to accelerate to this speed)
↳ setAcceleration: set acceleration rate (expensive call and avoid it possible)

General logic V1:

Input of position array
(each index represents position target for motor m1, m2, m3 respectively)

↓

Set max speed "variable"
↳ idk how to calc rn

↓

set Acceleration "variable" ★ constant

↓

input pos value into moveTo function
↳ θ angle → step value

↓

run motor

General logic V2:

position array input

↓ ← calculated by a function

m1: set max speed = speed [m1]
m2: set max speed = speed [m2]
m3: set max speed = speed [m3]

↓

m1: set Acceleration = speed[m1] ★ constant
m2: set Acceleration = speed[m2] ★ constant
m3: set Acceleration = speed[m3] ★ constant

↓

m1: moveTo = position [m1]
m2: moveTo = position [m2]
m3: moveTo = position [m3]

↓

m1: run
m2: run
m3: run

assuming angle value is already converted to step pos

★ results in a movement of **ONE** step... preventing an orientation of the platform that does not work

---

Speed Calc function V1:

Input: i value representing motor #

↓

get current pos of motor

↓

(absolute (current pos - target) ★ constant) = speed

↓

constrain speed from 0 to 1000
↳ Done to prevent skipped steps