Control Structure
Encoder
Calibration

ADD to ARM MOTOR CONTROL STRUCT:
target & current degrees,
delta degrees, too?
↳ (would be over-dimensioning) (task3)
☆ → MAXIMUM SPEED for motor.

ARM CONTROL

STEP 2 In case you're wondering why these aren't just code comments, it's b/c I'm not strong enough in C yet to just plunk down code. I'll get there, eventually.

FINDING DELTAS

DELTA MA = TARGET MA - CURRENT MA ⟹ MA.DELTA = MA.TARGET DEG - MA.CURRENT DEG

MA. DELTA DEG = MA. TARGET DEG - MA. CURRENT DEG — BASE LIFT
C. DELTA DEG = C.' ' - C.' ' — ELBOW LIFT
MB DELTA DEG = B.' ' - B.' ' — WRIST/HAND LIFT

(FUNCTION)float get Delts (STRUCT ARM CONTROL STRUCTURE. CURRENT) {
return CURRENT. TARGET DEG - CURRENT. CURRENTDEG;
}

STEP 3 : MOVEMENT

probably the hardest part - putting conceptual into reality.

inputs ARM CONTROL STRUCTURES, DO I NEED SOMETHING LIKE A TIMEWINDOW?

3.1 - function to move to specified degrees, given control structure & abstracted degree data.
3.2 - TASK ↳ SLOW SINE CURVES MIGHT BE NICE, TOO. - speed up towards the middle, slowdown towards the end.
3.2 - TASK-BASED CONTROL OF SAID FUNCTION.

3.4 - CURRENT QUESTION - HOW DO I HANDLE MULTIPLE MOTORS at once? SCALING seems like a bit much.
IF I'M using a task, it's nbit harder to sit & wait, too.

MOVING ARMS IN THE MAKING THE MOVEMENTS HAPPEN AT ONCE:

~~W = F · D~~ Time = Distance / ~~DEGREES~~ STEPS PER DEGREE

~~PSEUDO TIME~~ De

PSEUDO TIME = $\frac{\Delta \text{ Degrees}}{\text{Steps per degree}}$ × MAXIMUM MOTOR POWER.

AVERAGE "TIME" = $\frac{\text{PSEUDO TIME 1} + \text{PSEUDOTIME2} + \text{PSEUDOTIME3}}{3}$

TIME = $\frac{\Delta \text{ DEG}}{\text{Steps}}$ · POWER TIME = $\frac{\text{POWER} \cdot \Delta \text{DEG}}{\text{STEPS}}$ STEPS · TIME = POWER · Δ DEG $\frac{\text{STEPS} \cdot \text{TIME}}{\Delta \text{DEG}}$ = POWER