

12/2/23

13

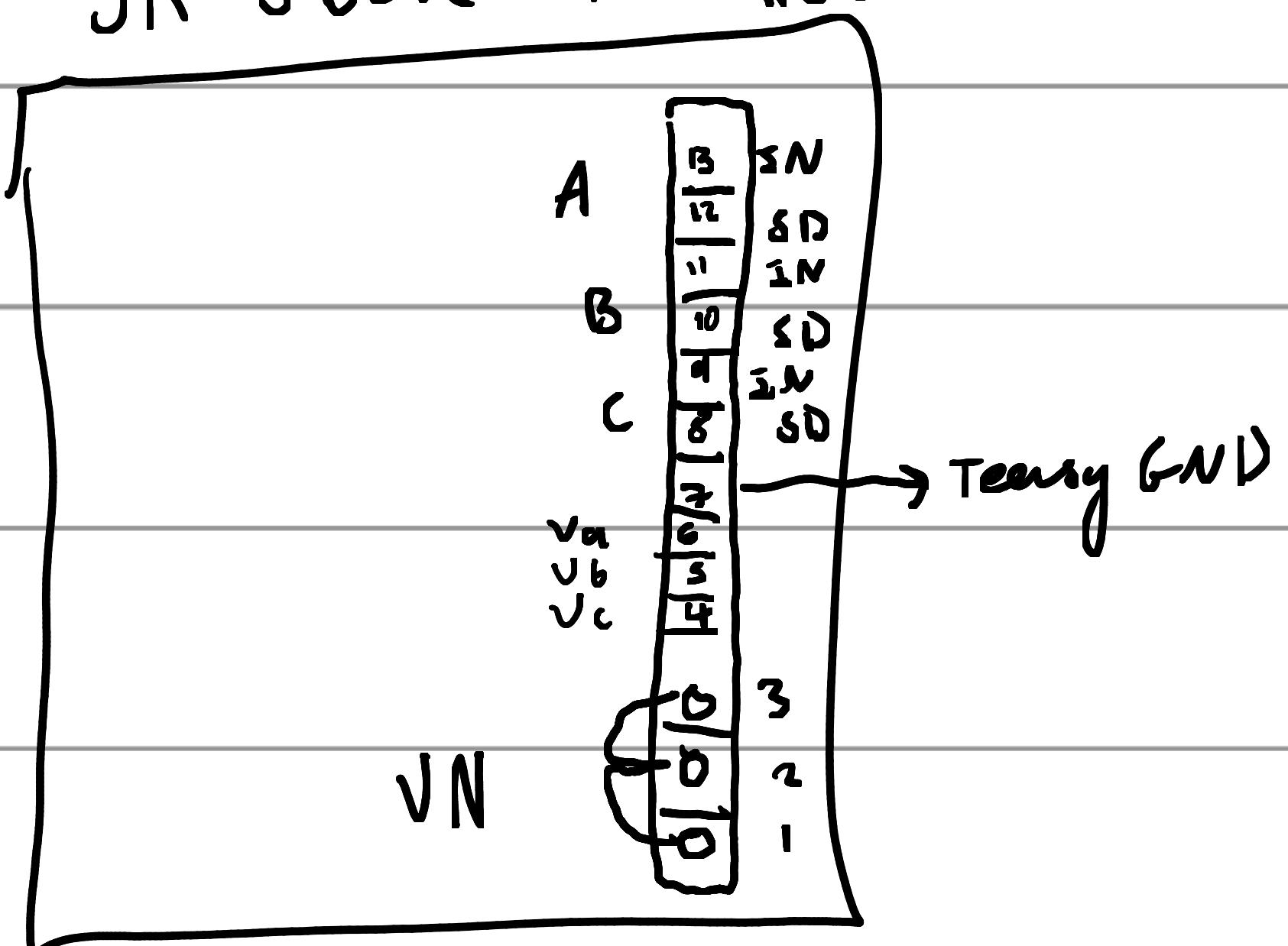
2.54 mm Pins

Circuit refresher

7

Goal to create an extension

JK-SBLDC-SMT-REV 2.4



which takes 2 teensy's

the SBLDC board, an encoder

I can perform both measurements

I drive a motor... with

enough pin outs to still

Header pins are C52928

cabrd enough pins for simulated

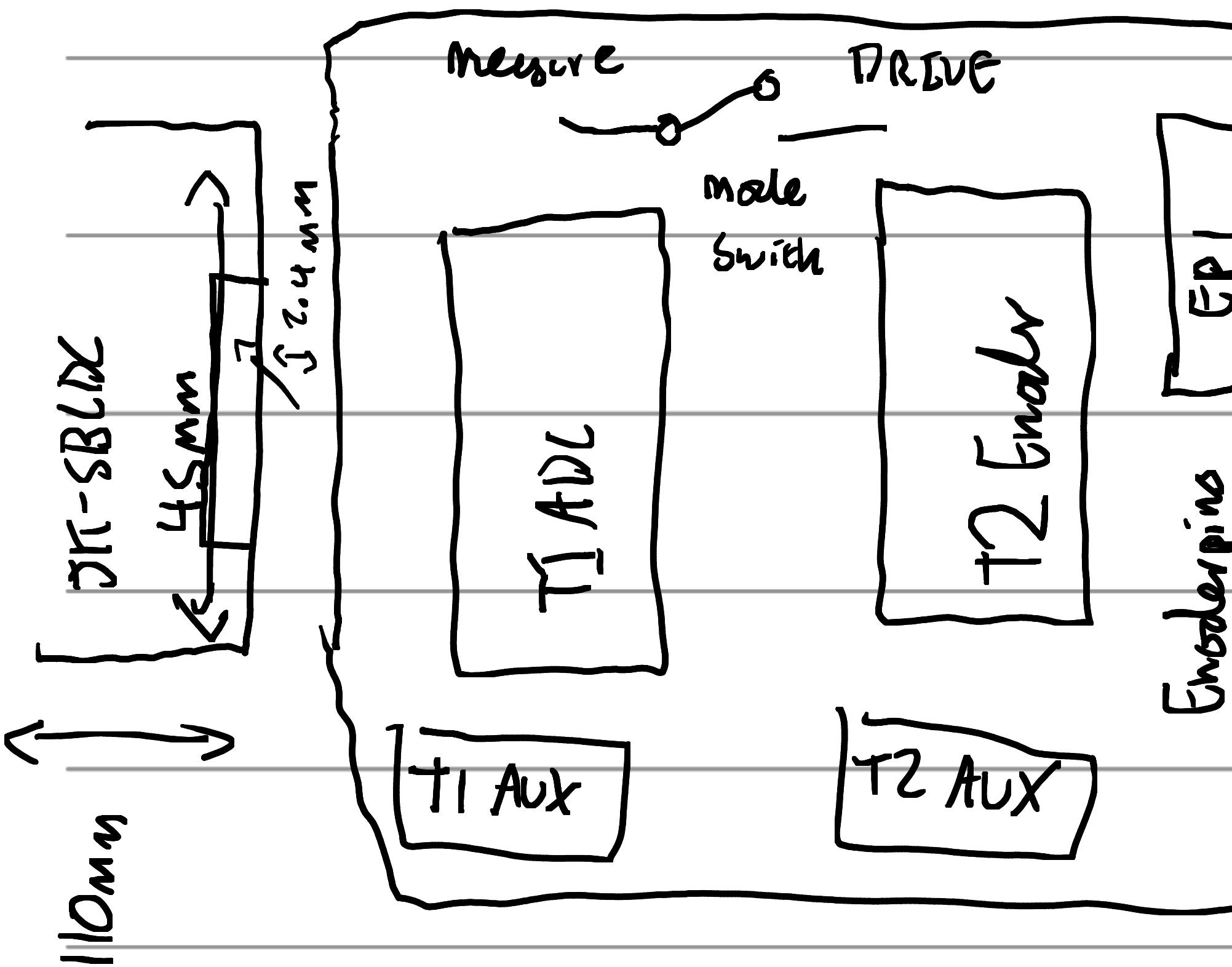
connection.

AS 5147

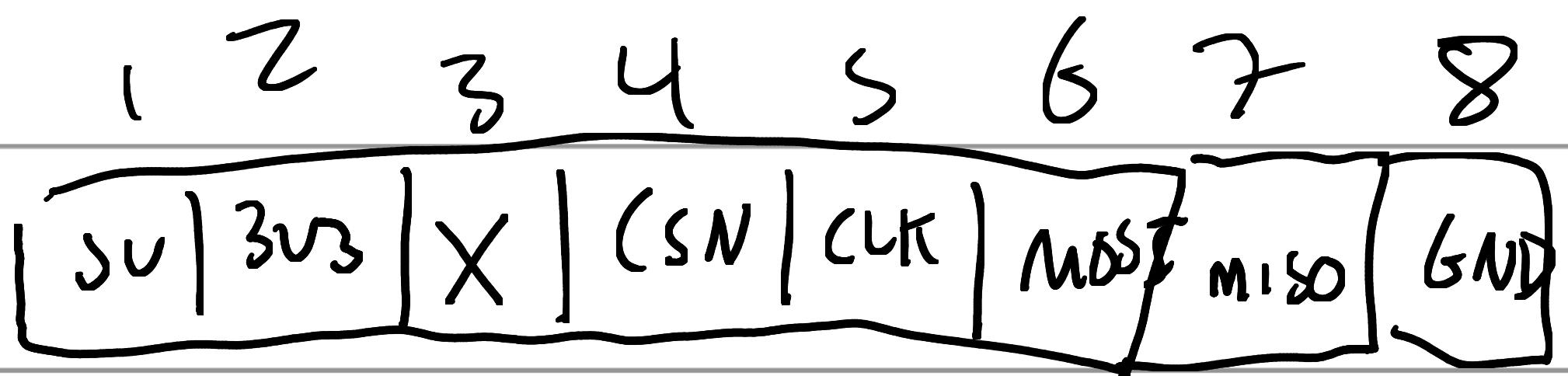
13 -2

⑦ ⑪

Shaded?



# ASS147P-TS\_Ek\_A13 Brechout

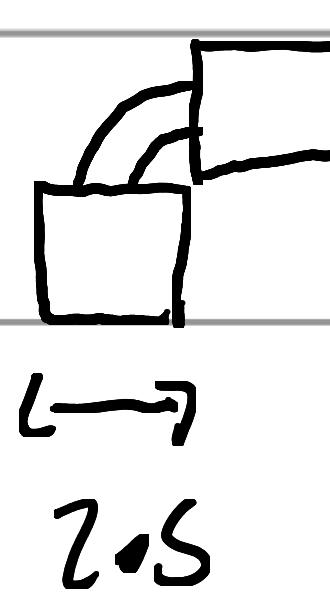
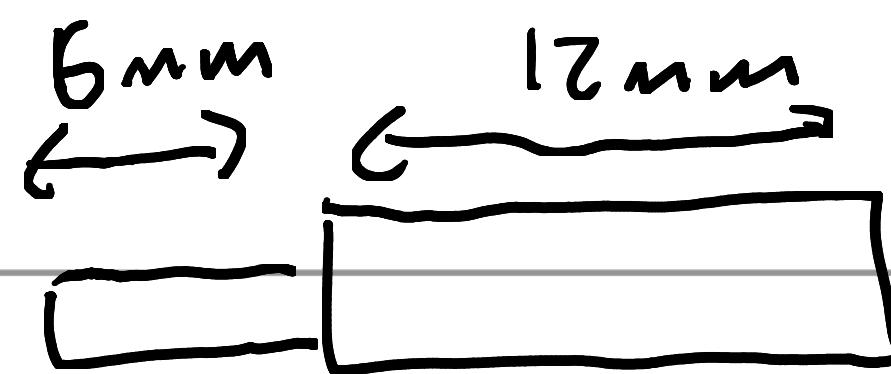


what AVK pins do we want from TI LTCZ,  
they have 0 → 23 (24) easily accessible / programmable

pins, & two GND, & 1 3.3V pin. & Vin.

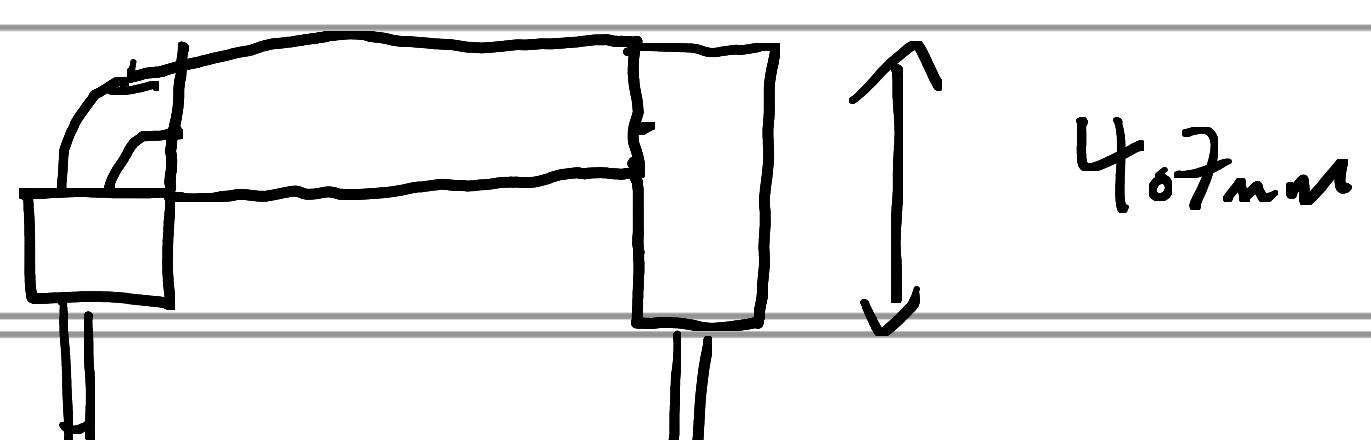
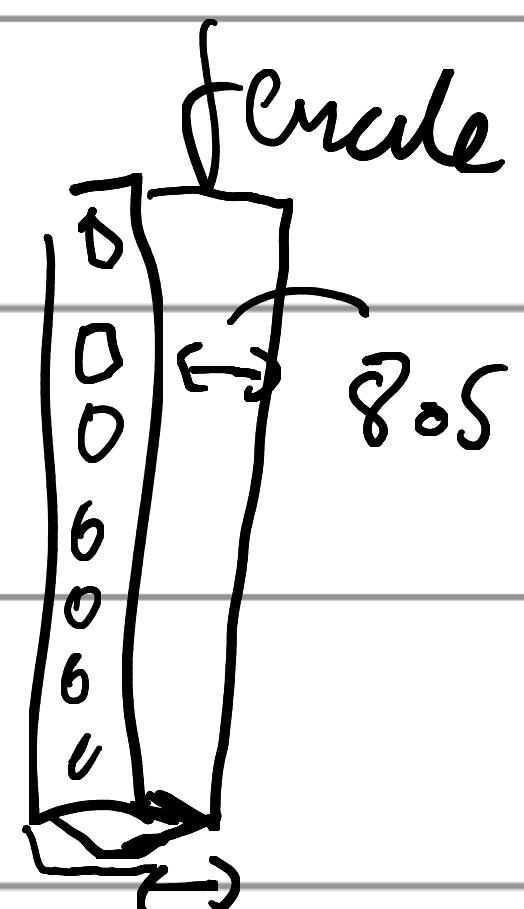
28 AVK pins total? yes.

HCIL PM2S4-1-13-W-8-S



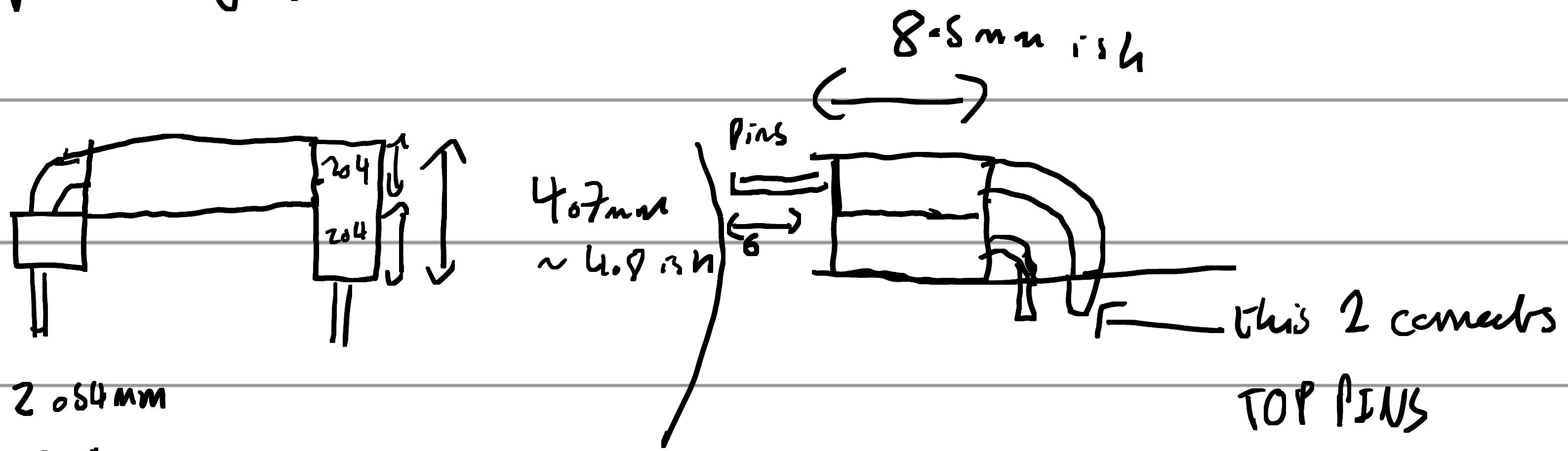
$$H = 2.05$$

want



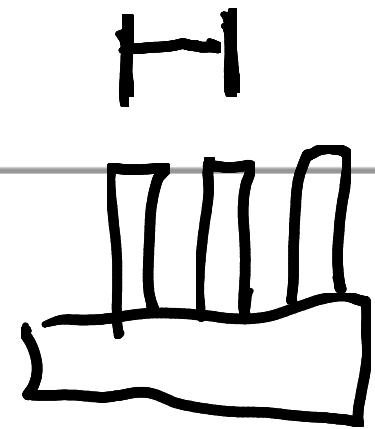
Best option as we have a  $\sim 2.4\text{mm}$  header

gap stand off from board is to have a  $2 \times 13$  pin header

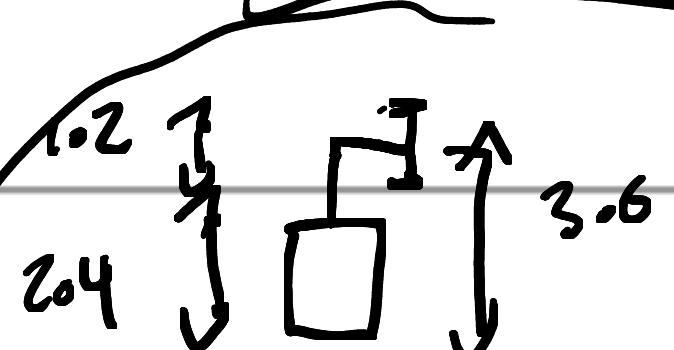


$2.054\text{mm}$

TOP PINS

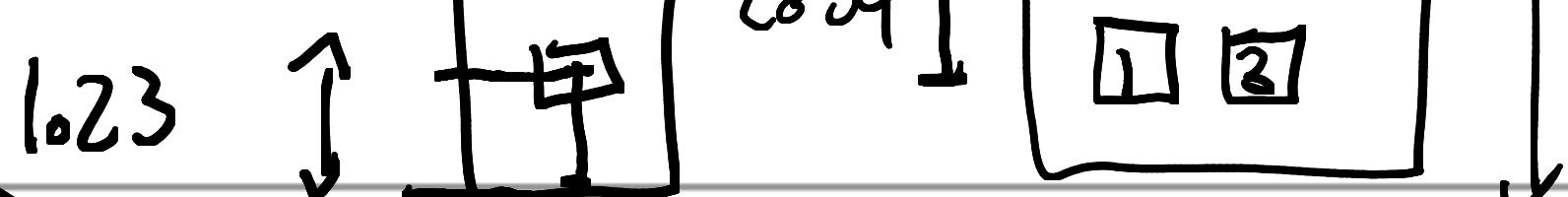


Candidates HCTL PM2S4-2-13-W-8.5  
1 WINNER



$$2.04 + 1.02 = 3.06$$

should be  $2.5\text{mm}$



$$1.023 + 2.054 = 3.077$$

5mm

$1.023$  | what we need

CL candidate  $1.078 + 2.054 = 4.32$  yes really

WCON

2189 - 213 R60 CUNT1

Next need 2 rail switch for the mode setting

SBLDC  
SMI  
204  
P1

PM 2S4 - Z-13-W-805

TOP

BOTTOM

2 3 4

3 5 6

4 7 8

5 9 10

6 11 12

7 13 14

8 15 16

9 17 18

10 19 20

11 21 22

12 23 24

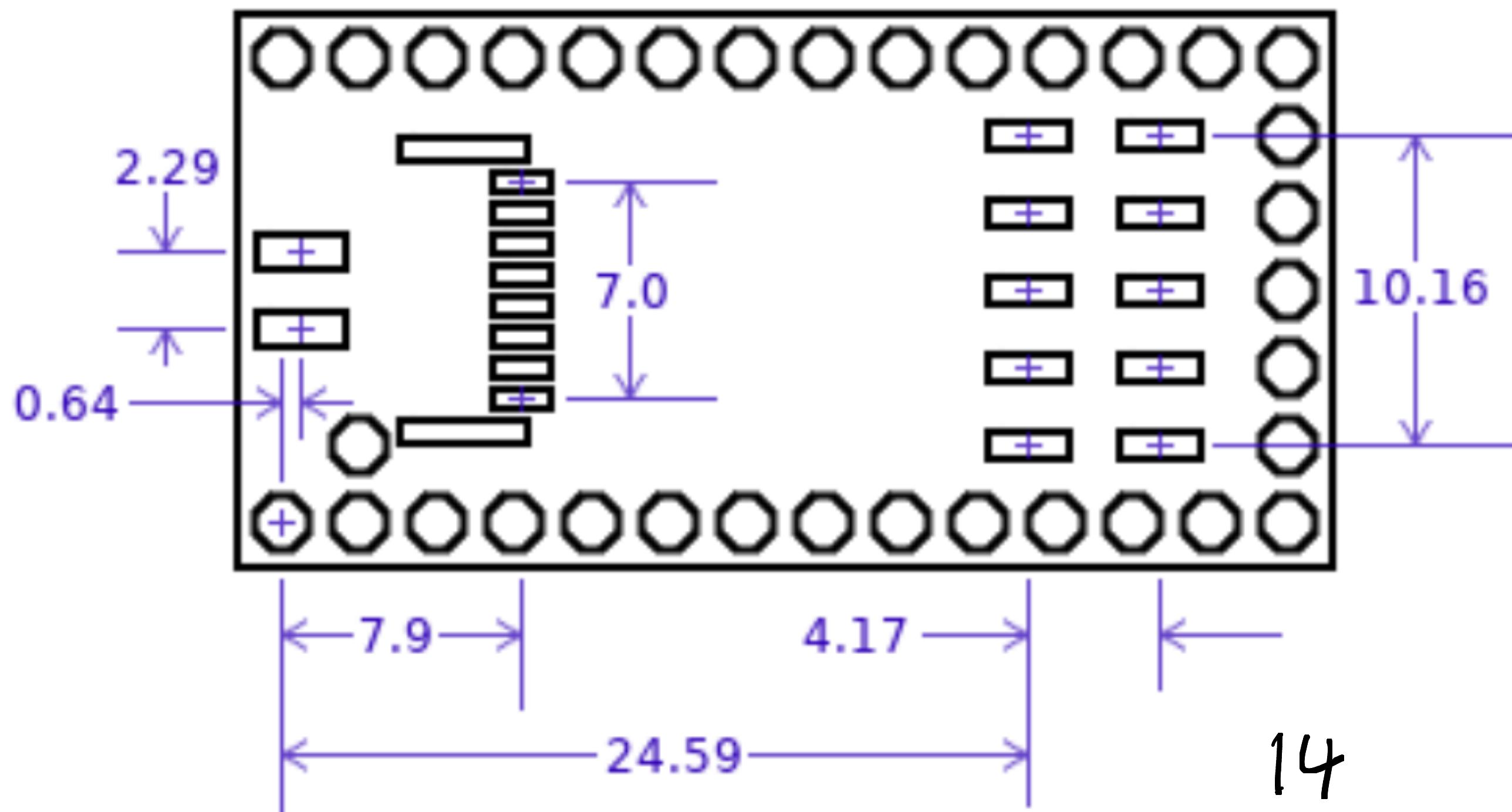
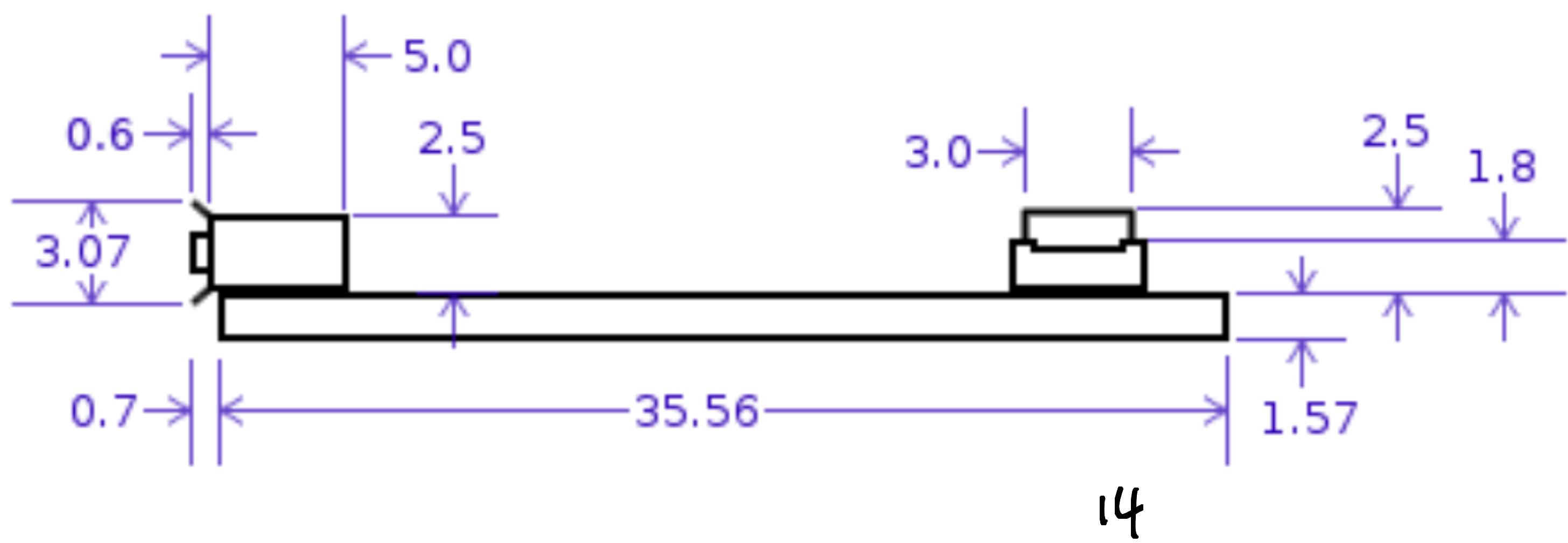
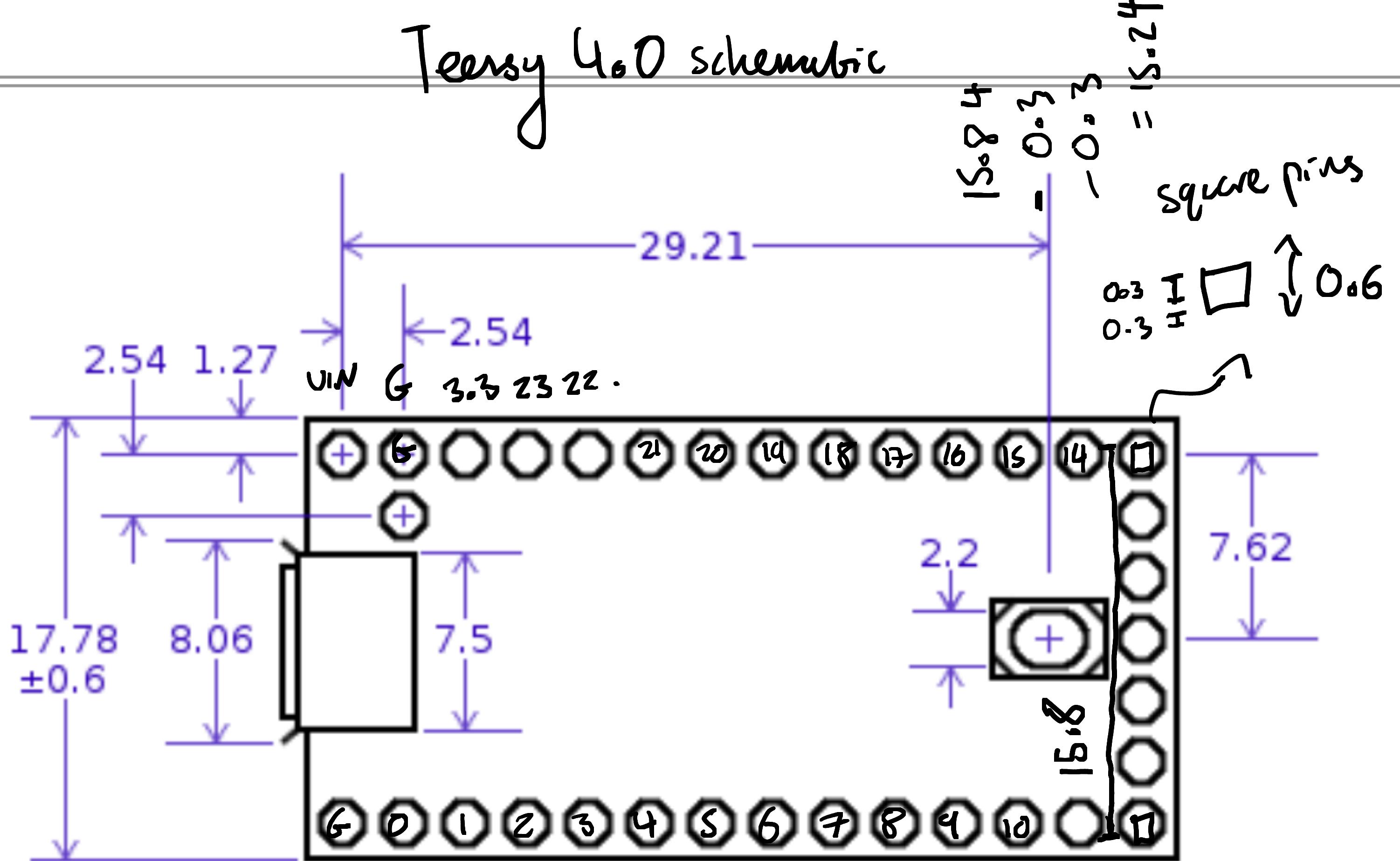
13 25 26



BOTTOM

TOP

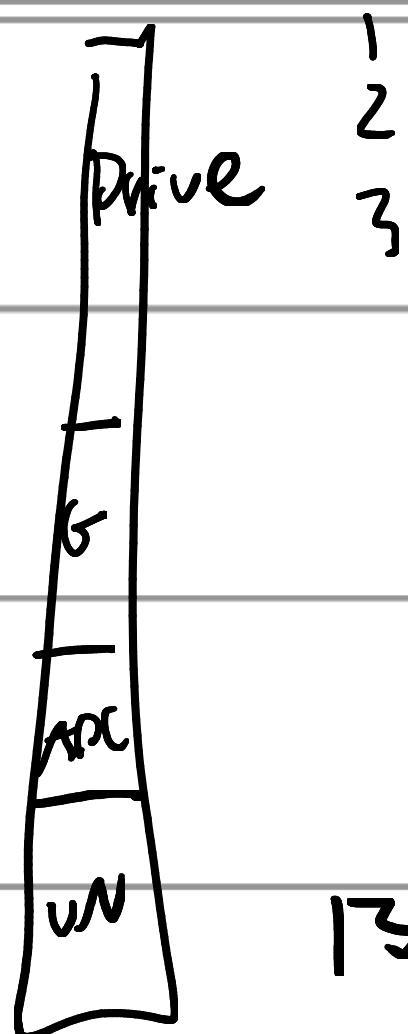
# Teensy 4.0 schematic



HCTL PM2S4-14 - Z-80S      need 4x of these

13

17/21/23



Fixing ASSI47 Netparts in every order.

13

S IS 60

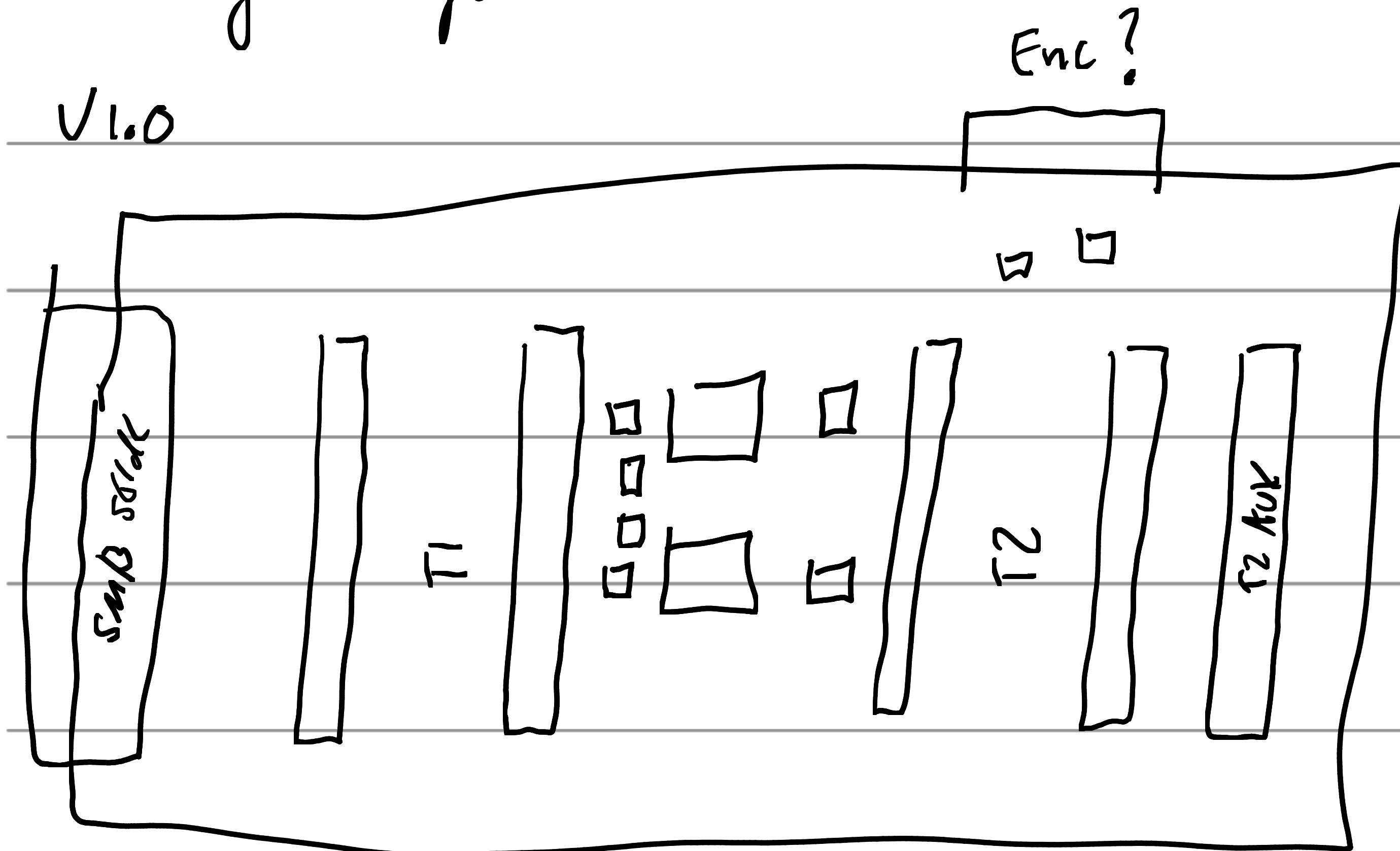
Kernel not insubtly again... Fixing that first. unne-r is S.1S.0-60-geni

old ones S.1S.0 - 58

Sudo apt remove 'linux\* 5.1S.0-43\*' done

circuit layout draft

V1.0

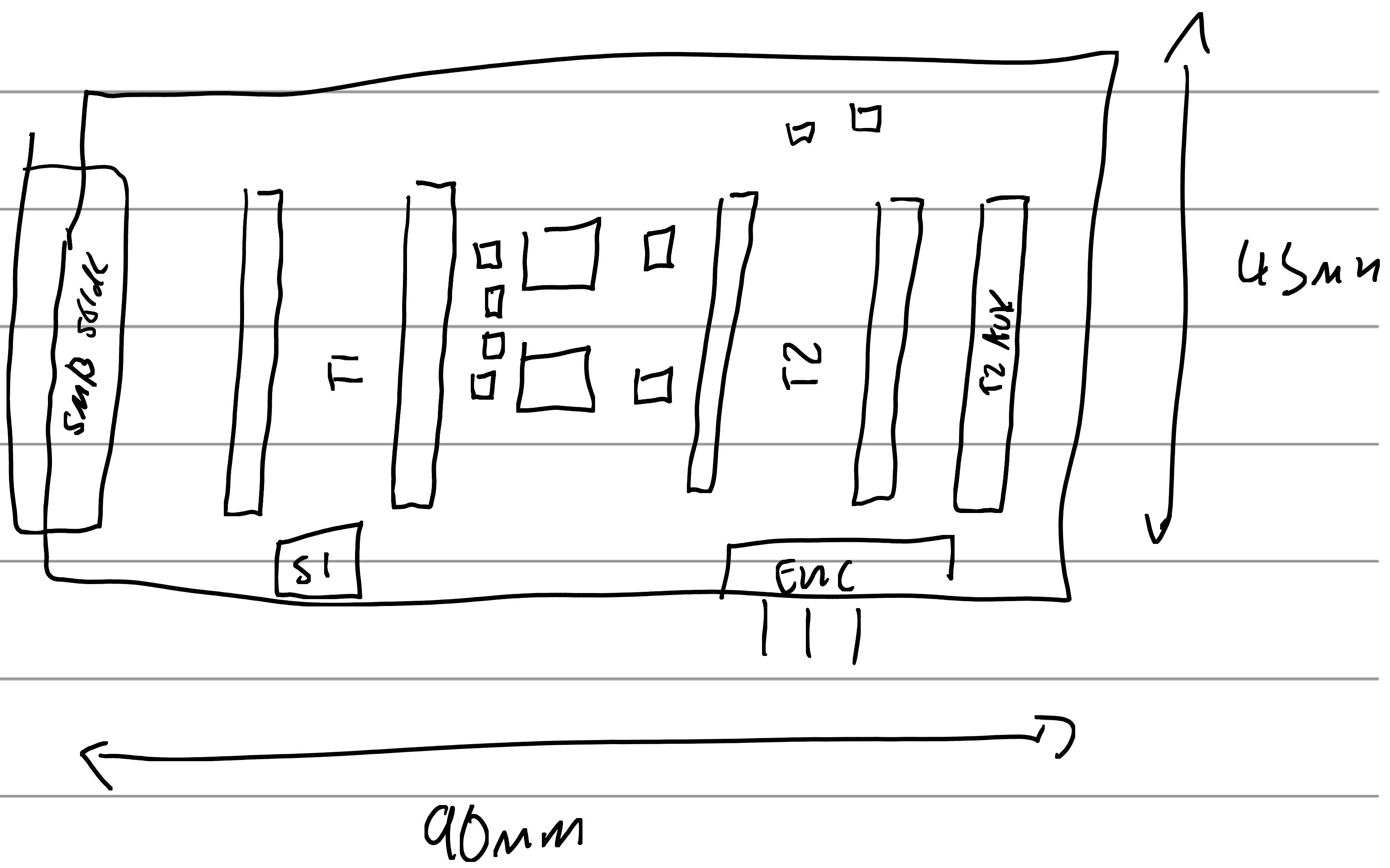


Design review, maybe enc pins l v66 T2 might interfere?

Same with switch! move to bottom.

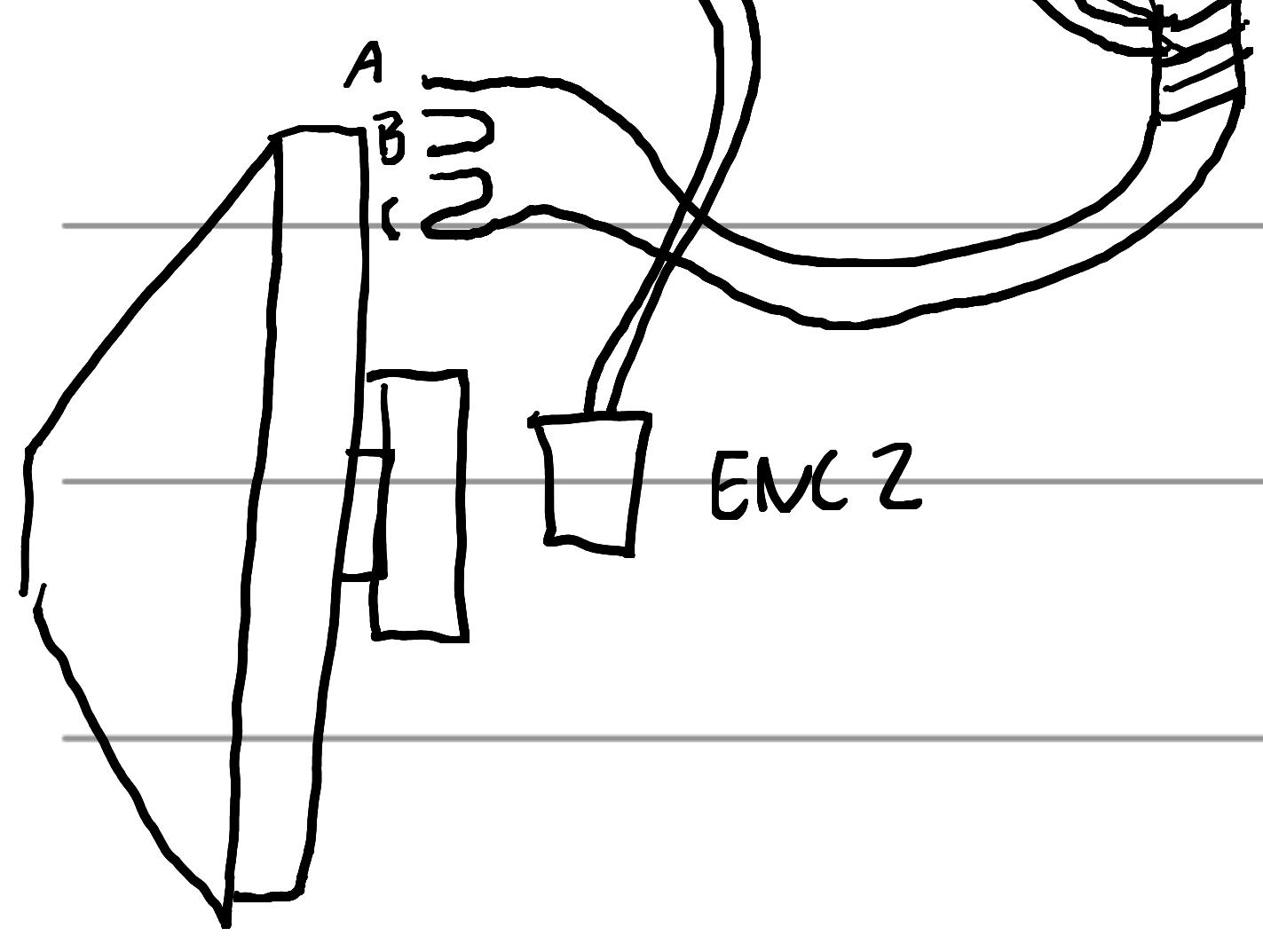
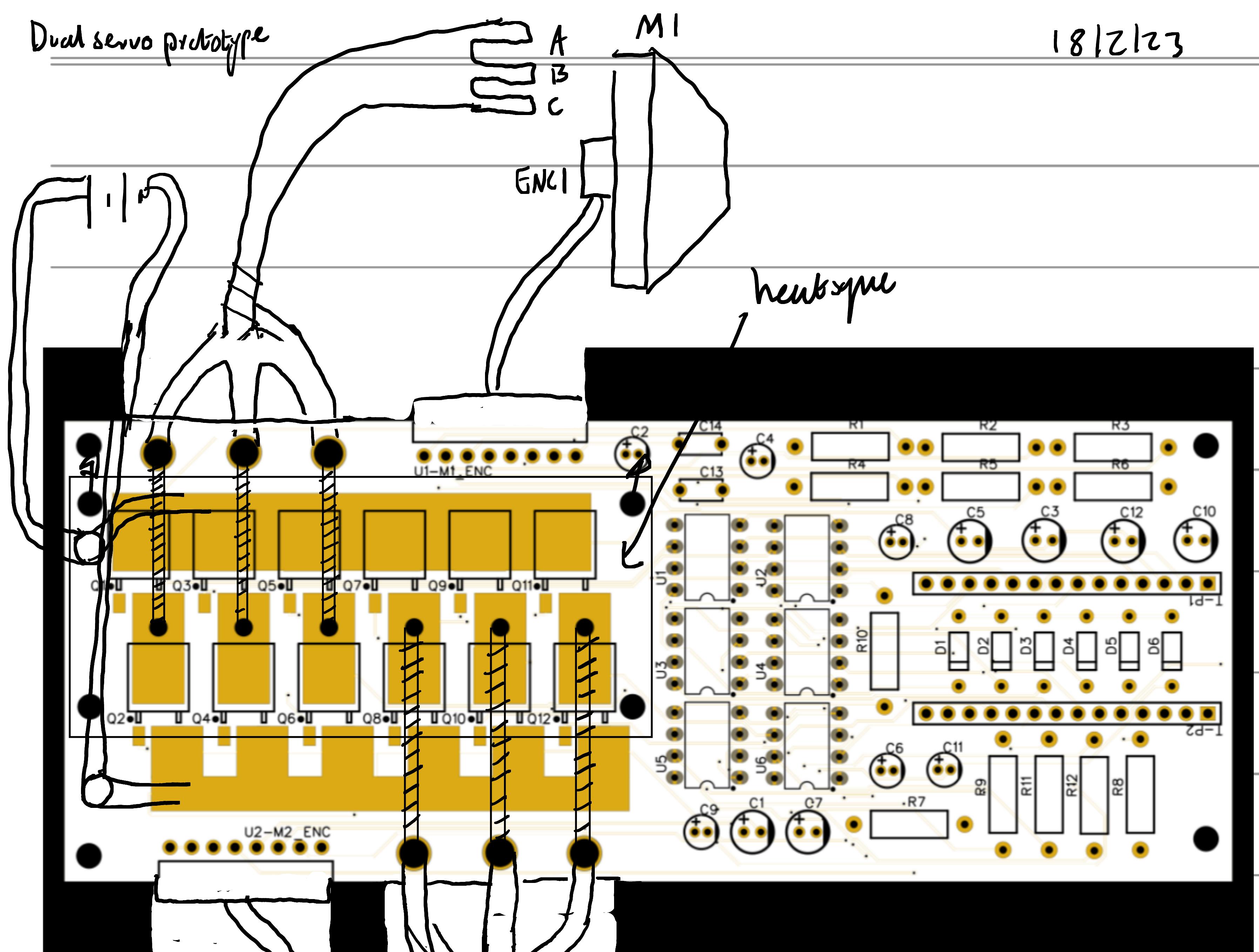
clear width  
0.4 0.7  
true

V1.0



Dual servo prototype

18/2/23



Fires, move drivers down,  
make VCC pad area larger (taller)  
allow space for power VCC/GND  
pads

## TZ PINS

2012123

S BIDC smt 13 AIN 2

12 ASD 1

11 BIN 9

10 BSD 0

9 CIN 8

8 CSID 7

7 GND SWAP to TZGND

6 VA

5 VB

4 VC

3 VN

2 VN

1 VN

Have not swapped GND, need to reintroduce

fault protection, watch out for AIN pin 13 as header

Connection is Good!

21/2/23

Tried several things today.

Could not get good encoder values would flip between 'valid' values & primarily 0 (sometimes 4, size), I tried to filter these values but was not getting around the circle with just 'skip steps', & iteration ctr would be tiny indicating a very small number of valid values.

Had a look at wiring & disconnected ADC & opto circuit, no difference... need to revert, reverse PWM logic.

Next: added ~100ns to the start of the 'end angle function'

this stopped the encoder returning junk & appears more stable.

NOTE SET COMPILE TO 600MHz + FAST NOT FASTEST.

noted ('way division') fault as it is useful feedback.

Remember PWM circuit + logic.

Tried spinning with hand drill, no skips! up to a respectable speed.

Sometimes randomly faults ('skipped steps') when resetting if PS4 trigger is held pressed down, also can fault if touching Teesy UD pins (good!) & if unplugging encoder ground (good!).

Next steps re-check pwm outputs + code & add battery. Double check throttle is off! Be ready with switch (Hn)

last run at iteration count was  $\sim 1,100,000$   $\frac{60 \times 1,100,000}{2^{12}}$

$\sim 16,113$  rpm. 1.113

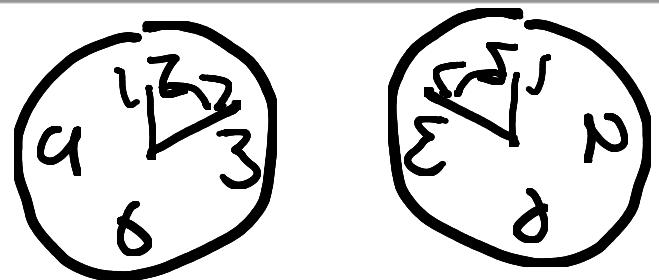
A2217 has 1000kv @ 4s  $4.2 \times 4$  16.8  $1000\text{kv} @ 16.8\text{v} = 16,800 \text{ rpm}$

95.9%

22/2/23

Attached Pwm pins again, needed to change pwm dependence to digital write Fast & added usm volatile. Motor spins with the battery, but spins in reverse direction, needs help to start, is silent when stalled... perhaps this thrust setting is too small, probably need startup program.

It turning the wrong direction might be as driving drill is 180° rotation



so cw measurement might be cww, need to

flip direction? The wrong direction prints are causing delays.

Startup program linear chirp with early escape when transitions  
in sequential states in the right direction

1) Fix direction bug before raising power.

2) Try power 70

3) Next implement startup program.

What would a startup program look like? Force state transition

even when out of position, inc switching frequency over time. Detect  
real reg. in each loop ...

Startup program... continued. still was skip protection with a hard off

need to make some fraction of an electrical cycle before assuming start

(condition is over.)

last-state      wrong-direction-ctr      right-direction-ctr  
last-expected-state  
<sub>next</sub>  
i    5000 → 0

delay (i)

current-state ++ / --    cw/cw  
commute-state()

read-cycle

↳ ideal state = last next expected state → escape?

↳ ideal next expected state

28/2/23

Blew up power circuit when messing with module sync! "n

Transistors failed on! (0 resistance)

Old camp has code bit with init-set (0) act (2) (swonker sync)

Attempted resolution don't call init-pwm when in fail statement

it's not an off switch

(are they there?)  
→ revert them?

Move init-pwm to before analogue writes / start of init motor

dedicated motor-off function.

renamed sync method, muted FORCE, added reflect from JT block  
projects but changed INIT-SEL from 0 → 2, this had 100K students  
before & after (thinks a big difference) & the non-use of force  
is an obvious one... code otherwise looks the same perhaps

start with INIT-SEL (0) & test ↴

why it could have failed → multiple init, init-pwm after  
motor

set frequency etc, no Idiot Check. Few arm (dsb), PWM duty  
of 70%, no stall protection.

Should not mess with module sync until we have a start program  
with a fixed very low duty.

Try to resolder mosfets (hard)

solder new PCB (need consider male header)

reinstate analogue write()

FUSE!

last state  
last expected next state  
start up again start up work == true

this state = startup[read cycle()]

expected next state = expected[loop[-cycle]]

if this state == last expected next state

progress++

If this state == last state

Nothing

Else

Error

skip or wrong way

113/23

Drafted Startup routine ... using escape microseconds to work at the stall speed, when the routine make 6 consecutive transitions successfully we exit the number of microseconds since the last transition & use this as a minimum transition time. If a transition takes longer we consider this a stall.  
We have startup stall, stall, step & way direction faults missing Startup way direction, startup way direction

Do have some code for submodule sync, be careful with init set (2) as this could be dangerous!

Startup routine forces commutation out of sync, with increasing frequency with the hope of catching the rotor with rotating magnetic fields.

When we get to SPWM will we need this startup program?

Start program will input the iteration ctr but by how much?

Can we reduce the nano second delay?

strand.

Order new connectors from China.

8 pin + 13 pin

check thrust = 0 logic motor off!

ADC ground experienced  $\rightarrow$  check cat circuit!

70 duty seems to be thresholding duty, although stated at  
60 bad for a Z 12

check vid for resolving laptop pin break up for debris cable, only

If it fails to load.

What if startup mode speed has to be less than stall speed?

Audit components

where is 812GB SSD now?

For each circuit needed

Number of transistors IRF7843TRPBF x 6

capacitors  $10\mu F \times 3$   $2.2\mu F \times 3$

resistors  $100\Omega \times 6$   $33k \times 6$   $7.8k \times 3$

Diodes IN4148 x 3

ICs IR2104PBF x 3

connectors 2.54-1x13P header x 1 (male)

S BLDC-High power PCB

\* Try one from each first!

IRLR7843 TR PBF (10x larger) (30x smaller)

10μF x6 (3 good ones)

202μF x4

100Ω ~100×

33K ~100×

70SK x6! 10k x4

IN 4148 x15

IRZ104 PBF x8

cameras x0

PCB slide x3

Thermal paste x1

XT-60 x6

## Calibration circuit

$\checkmark 1k \quad 10X + 8X$

$\checkmark 220\mu \quad 10X$

$\text{IN914} \sim 25X$  (~~yellow~~)

$\text{H111L1} \quad 10X$

3/2123

Ordered components from China for 3 power circuits  
encoder shielded pin connectors, need to order 3 encoders +  
breakouts.

Looking at calibration circuit, should I make a version  
with the driver pins connected?

what pins need connecting? (SBLDC)

Header ID Name Teeny 40 Pin

P1 - 13 A-IN 2

P1 - 12 A-SI 9

P1 - 11 B-IN 8

P1 - 10 B-SI 1

P1 - 9 C-IN 0

P1 - 8 C-SI 7

(calibration circuit errors.

only 2 GND pin not 3!

its the wrong way round P1-13 should be at the top!

Built 2 new versions Tester 3 which has the header

orientation fixed & fixed ground issue.

Tester4 which is the same as Tester 3 but with  
T2 3 PWM pins SD & 3 IN pins canceled... Check  
via breadboard that this works & does not  
interfere T2 encoder noise with T1 ADC! \*

(we assume Tx grounds are canceled internally; perhaps  
should explicitly connect them for safety? \*)

3/3/23

Weekly plan

Tuesday

ADC circuit resetup & GND check. ✓

Find project management tool. ✓

Wednesday

6x measurements (hope motor is ok)

→ check circuit v2 perf board

→ Tin circuit. (spwm)

Thursday

→ SPWM open loop test (which not or?)

→ Analyse results for 6x experiments

Friday

→ Check calibration circuit

→ Order new circuit (calibration)

→ ESC Thrust = 0

→ order x 3 encoders

7/3/23

looking at ADC calibration (calibration circuit)

Sync Pins	Master	Slave	Name	
3	3		RESET	✓
8	21		CLK	✓

Sync pins recommended

Now thinking about GND T1 T2 SBLDC,  
in order to test calibration circuit.

T1 ADC (GND for optos to T1 ✓)

T2 ENC (GND + 3.3V connect to breadboard)

SBLDC (GND for V  
SBLDC pin 7)

9/3/23

## Code review

- Fault class?
- Reset wrong direction (try when motor runs progress in startup.)
- Added if Thrust = 0 → Motor-off()
- re-arrange ifs & conditions so that normal method transition is the first if condition, this is to improve latency.

what do we want from a class version?

(configurable encoders

1 encoder || Motor or 2 Greater | 2 motors)

(configurable pins? (ENC Yes Motor No))



struct for arguments?

analogue write Freyung all phases.