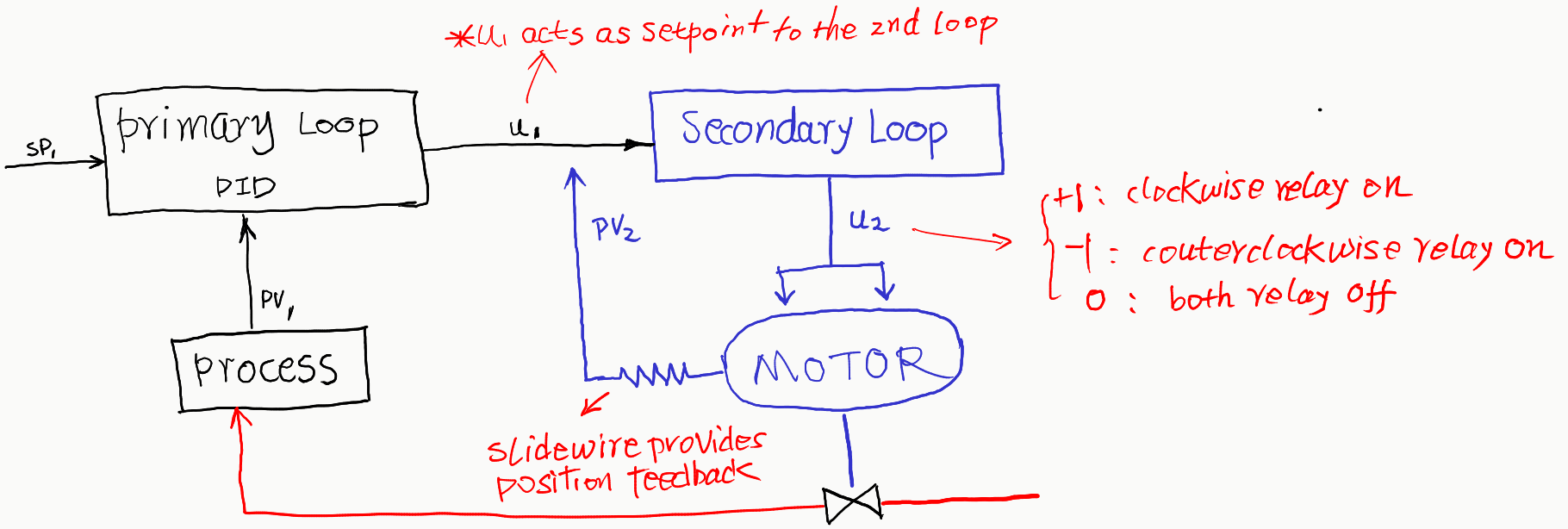
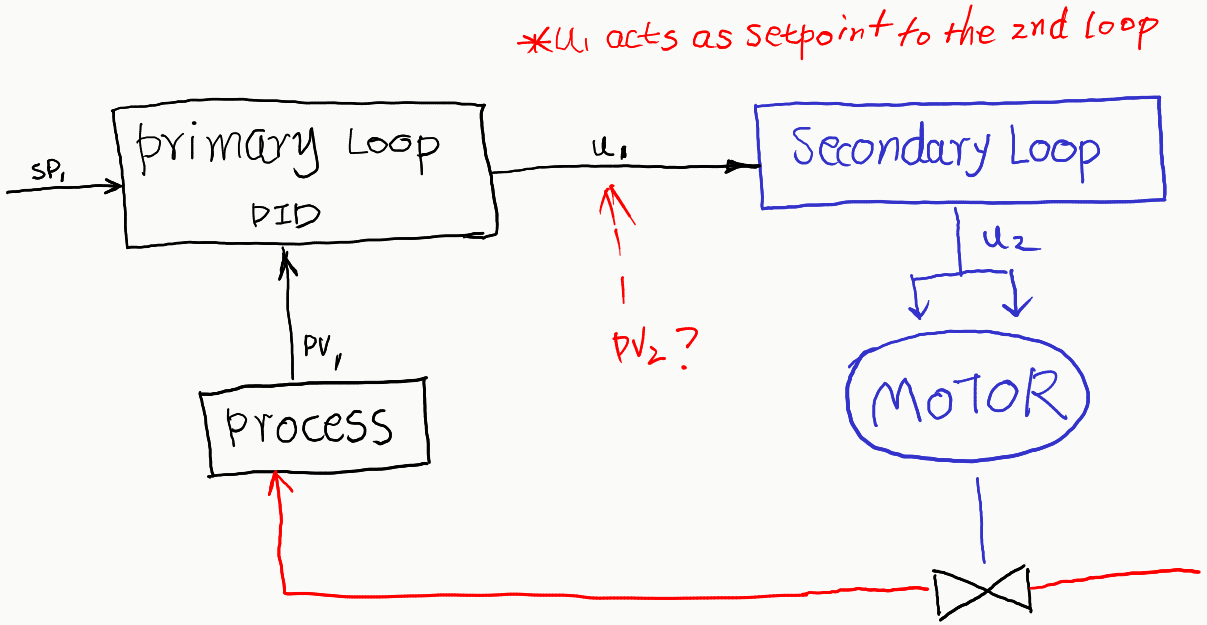
POSITION PROPORTIONAL



Sec. Loop : *Any small difference in input ($u_1 - PV_2$) will cause full band change in the output (u_2), hence the sec. loop is a high gain proportional controller. (⌐)

* A positive numbre θ (dead band) will applied to the ($u_1 - PV_2$), when $|u_1 - PV_2| < \frac{\theta}{2}$, the output u_2 will be limited to zero.

THREE POSISION STEP CONTROL



No feedback from motor, hence the PV_2 has to be estimated:

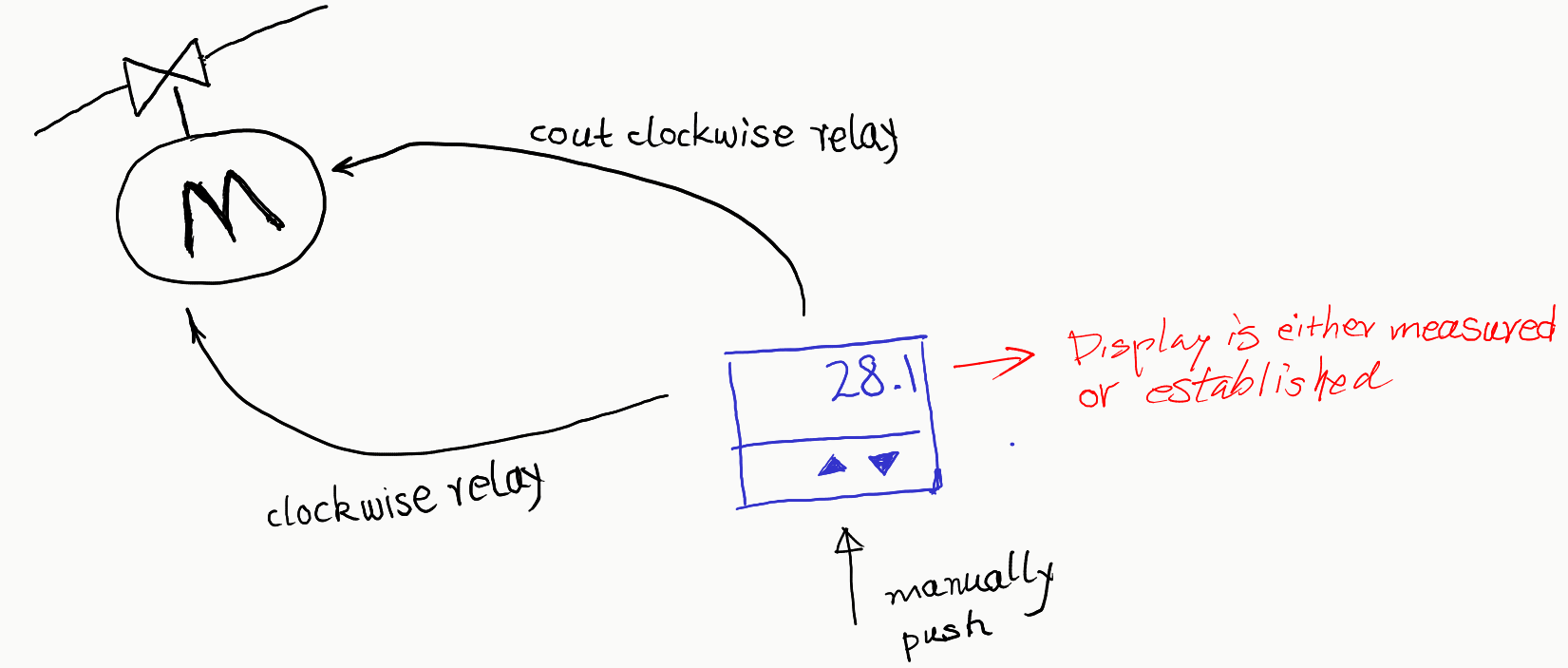
* How long time each relay was "on"
* Motor speed (stroke time) is known

Sec. Loop: In additional to the P.P. algorithm, the 3PSC will estimate it's PV as:

$$\frac{d}{dt} PV_2 = \frac{u_2}{T_s} \qquad (1)$$

T_s: Motor Stroke Time

Manual Operation of Valve



STORY 1

The device should turn on or turn off the two relays to make the slidewire reading match the primary loop's control output.

Note ① If the slidewire reading and the primary output are within the threshold error range $\frac{\theta}{2}$, the relays're not allowed to operate.

② The primary output can be continously changing

③ The output and the reading are all in percentage values: [0, 100]

④ If the primary output is not varying, the stablizing time that the slidwire reading mathes the primary output should only be limited by the motor speed (motor stroke time).

⑤ Both relays "on" is a forbidden condition, that will damage the motor.

STORY 2

When there is no slidewire feedback, the device must estimate the valve position using method described in the 3PSC algo. with the assumed "PV_2" value, it should still archive the purpose of moving the valve position as depicted by the primary output "ui".

Note
① After device power cycle, it should continue from its previous estimation state of valve position
② There is no dead-band (θ) threshold as that in the slidewire-feedback case.

STORY 3

I should be able to manually push the ▲ and the ▼ button on the front panel to move the motor clockwisely or counterclockwisely in order to control the valve position.

NOTES

① The display should show me the valve position while I push the buttons (0% – 100%)

② The display values could be measured ones or assumed ones if no slidewire feedback.

③ I should still able to move the motor when the display values have already hit it's limitations

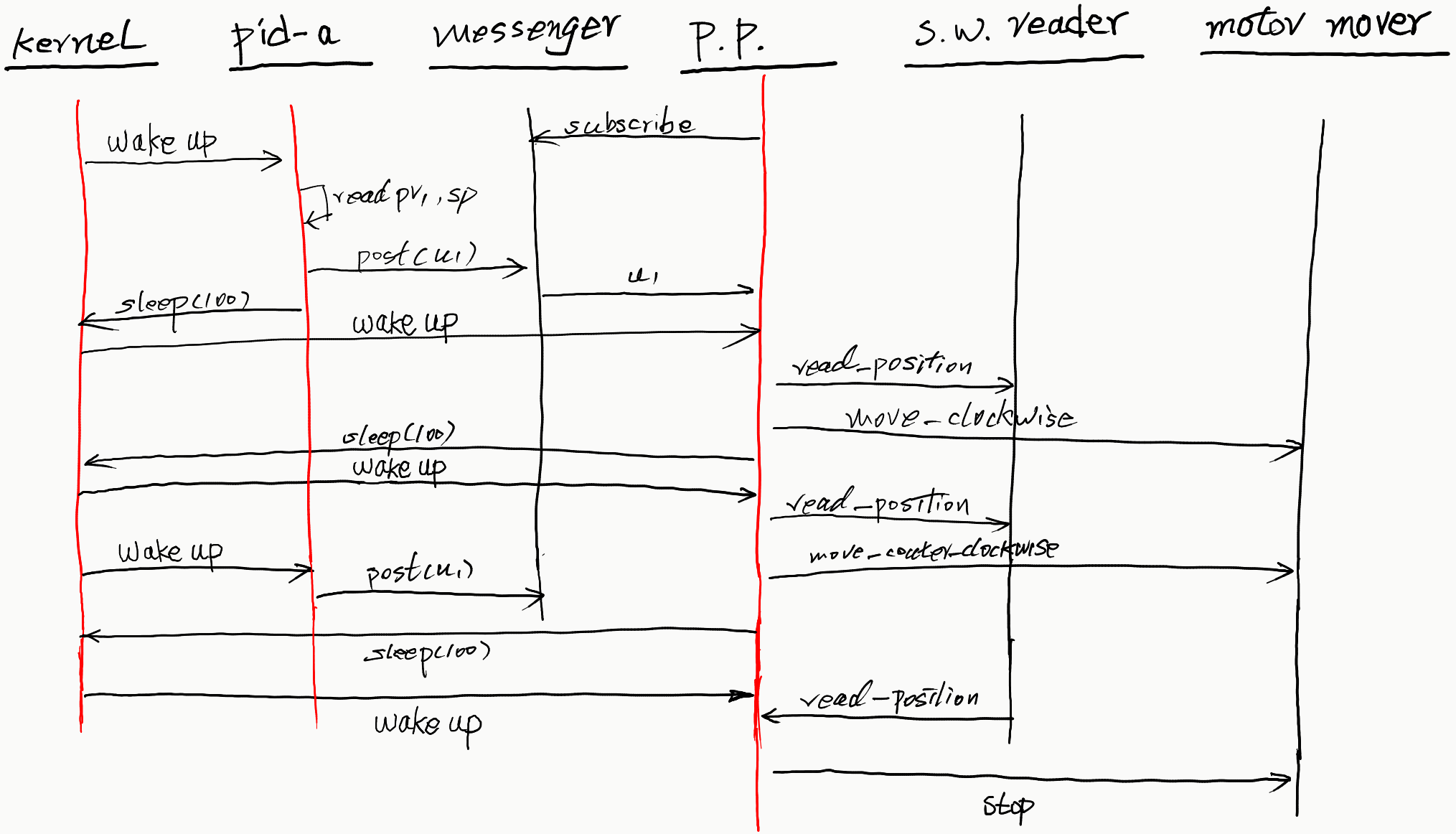
Questions

① what types of motor the P.P. and the 3PSC are expected to work with? (solenoid?)

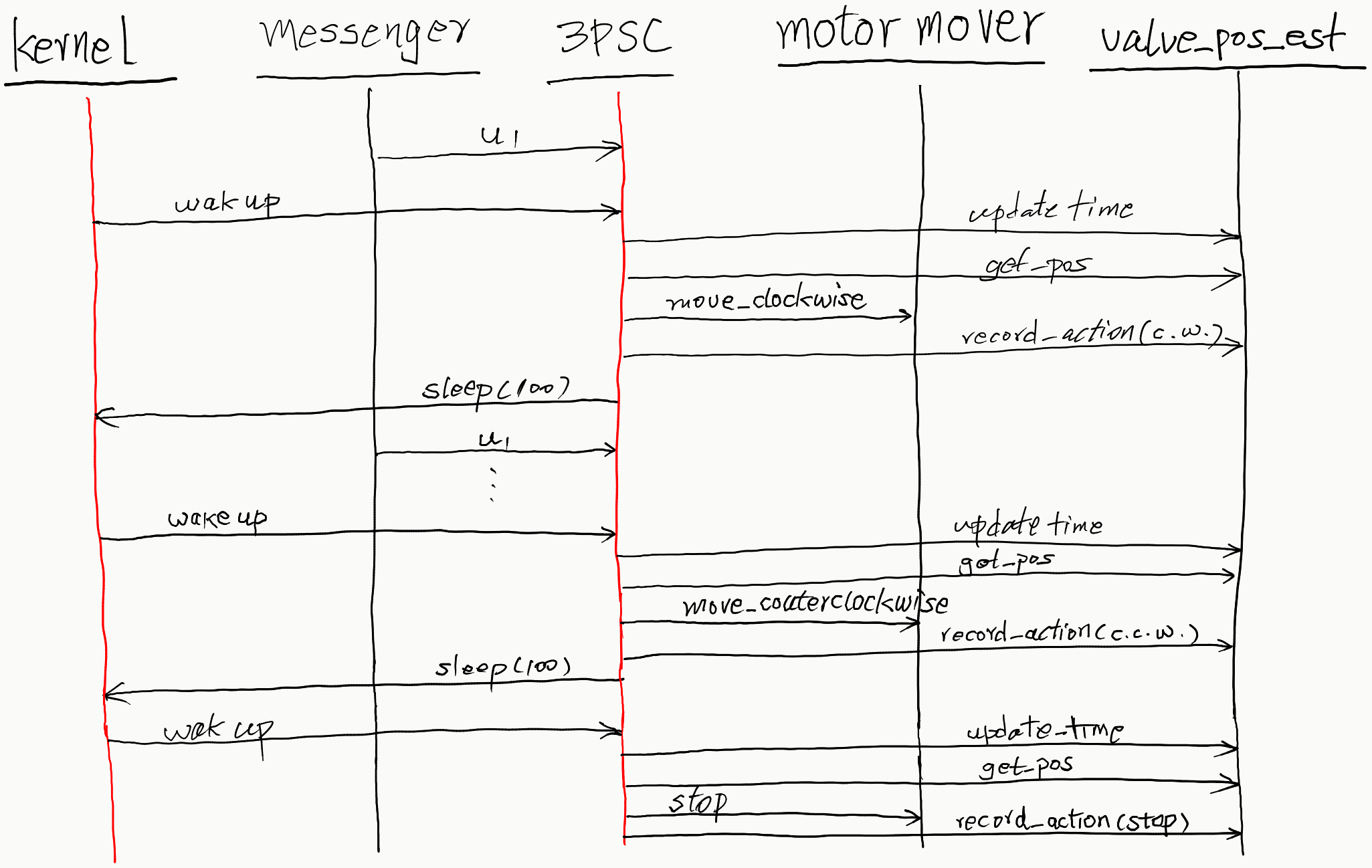
② In the case of 3PSC, how the device get the initial position of the valve?

③ In the case of 3PSC, when the valve position and the assumed value out of sync, how an operator to fix it in the field?

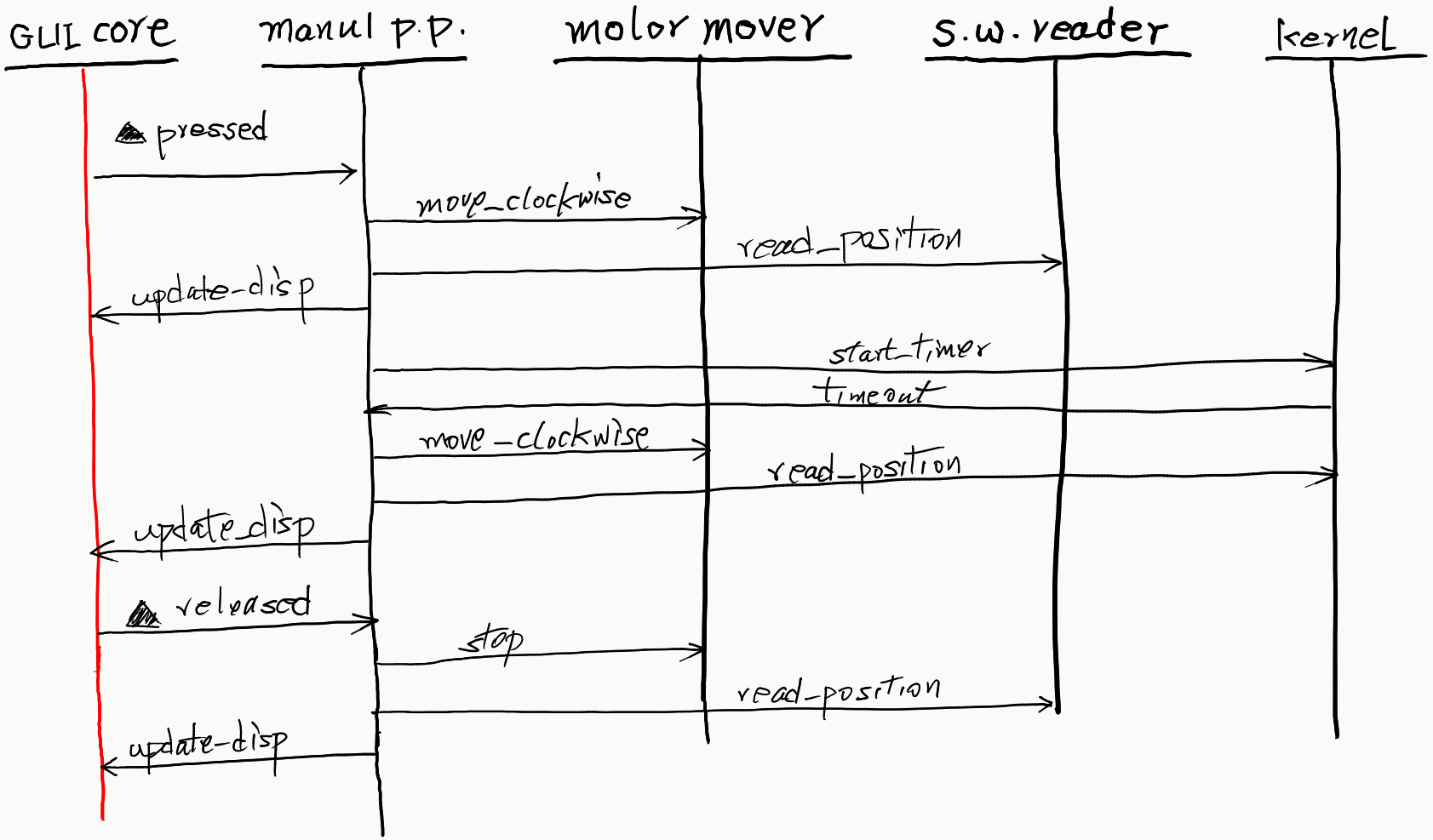
STORY 1 SD-1.1


kernel
pid-a
messenger
P.P.
s.w. reader
motor mover
wake up
subscribe
read pv, sp
post(u1)
u1
sleep(100)
wake up
read_position
move_clockwise
sleep(100)
wake up
read_position
move_counter_clockwise
wake up
post(u1)
sleep(100)
read_position
wake up
stop

STORY 2 SD-2.1 (Interactions with PID are omitted)



STORY 3 SD-3.1


GUI core
manul p.p.
molor mover
s.w. reader
kernel
pressed
move_clockwise
read_position
update-disp
start_timer
timeout
move_clockwise
read_position
update_disp
released
stop
read_position
update-disp

STORY 3 SD-3.2 manual 3psc

T.B.D.

WORK ASSIGNMENT

adaptations

	P.P. task	S.W. reader	motor mover	Valve_pos_est	3psc task	GUI
Lacey				✓		✓
Vincent		✓	✓			
Zhongwei	✓				✓	