

```

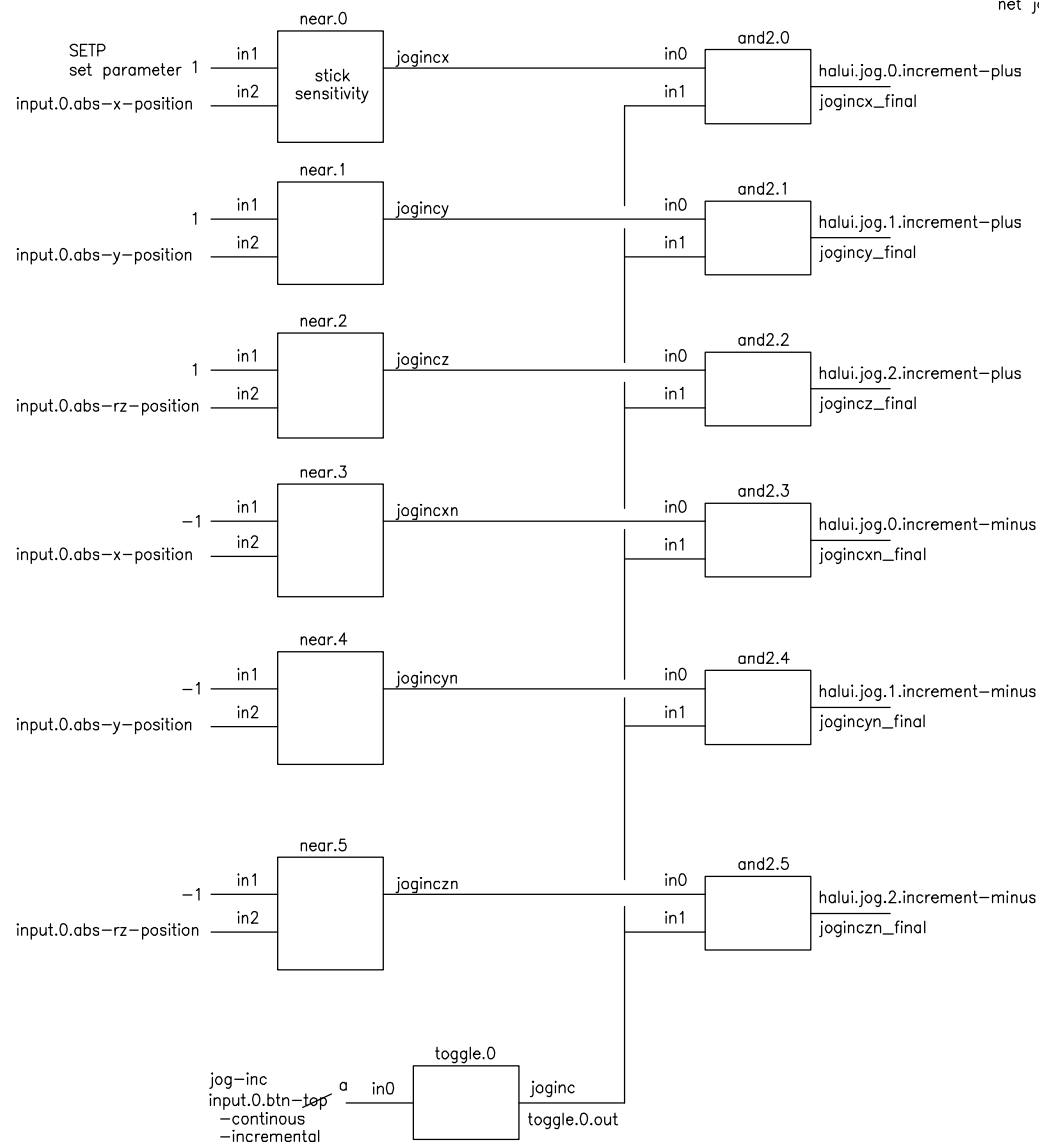
## if toggle is set for incremental and joystick is 1 or -1 make an incremental move
## hook up near outputs to and2. inputs
net jogincx and2.0.in0 <= near.0.out
net jogincy and2.1.in0 <= near.1.out
net jogincz and2.2.in0 <= near.2.out
net jogincxn and2.3.in0 <= near.3.out
net jogincyn and2.4.in0 <= near.4.out
net joginczn and2.5.in0 <= near.5.out

```

```

## and2 outputs to halui incremental jog
net jogincx_final halui.jog.0.increment-plus <= and2.0.out
net jogincy_final halui.jog.1.increment-plus <= and2.1.out
net jogincz_final halui.jog.2.increment-plus <= and2.2.out
net jogincxn_final halui.jog.0.increment-minus <= and2.3.out
net jogincyn_final halui.jog.1.increment-minus <= and2.4.out
net joginczn_final halui.jog.2.increment-minus <= and2.5.out

```



NEAR  
Upoređuje analogni izlaz  
joystika da li je 1 ili -1 to  
jest da li je joystick u krajem  
polozaju

AND  
Ako je joystick u krajem  
polozaju i ako je TOP dugme  
pritisnuto onda se zadaje  
koracni signal + ili -

```

## toggle from continuous to incremental jogs
net joginc and2.0.in1 and2.1.in1 and2.2.in1 and2.3.in1 and2.4.in1 and2.5.in1
mux2.0.sel mux2.1.sel mux2.2.sel <= toggle.0.out

```

push-on  
push-off  
from momentary  
push-button

```
net jog-inc toggle.0.in <= input.0.btn-topa
```