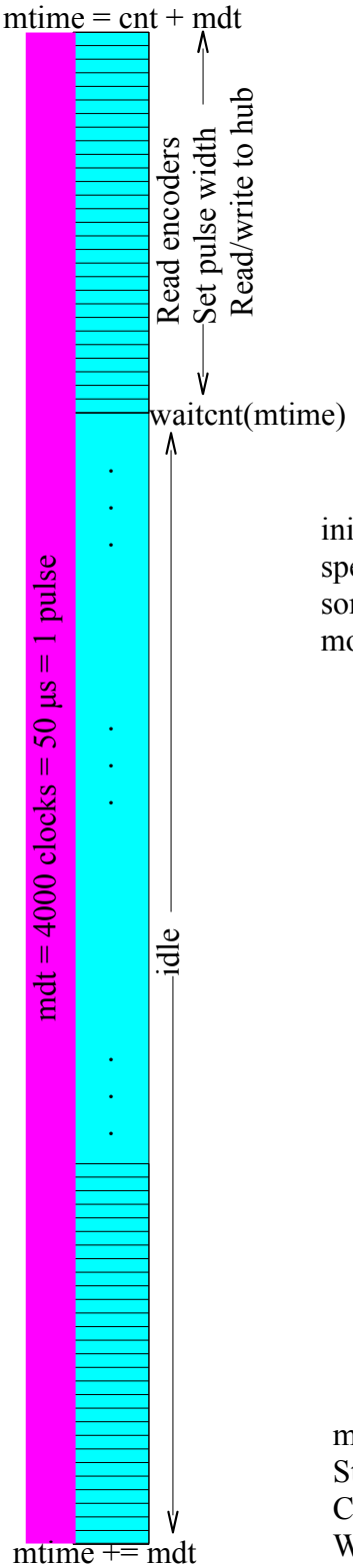
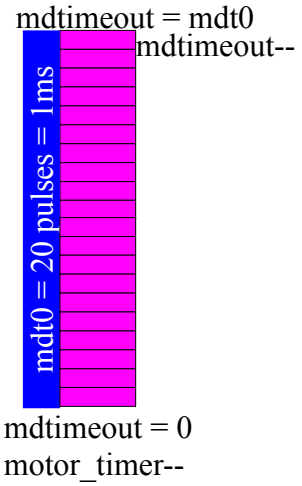
Timer Timeline for the S2 motor driver. Matt Greenwolfe 6-20-12

clock frequency = 80MHz, so 1 clock = 12.5ns

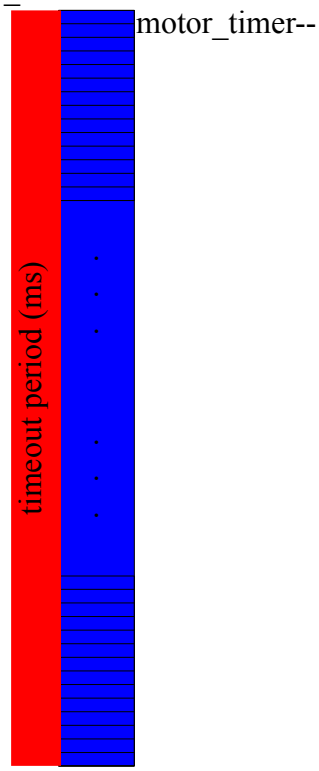
retrieve addresses for hub variables, initialize values (including shift registers) *once* when cog is started

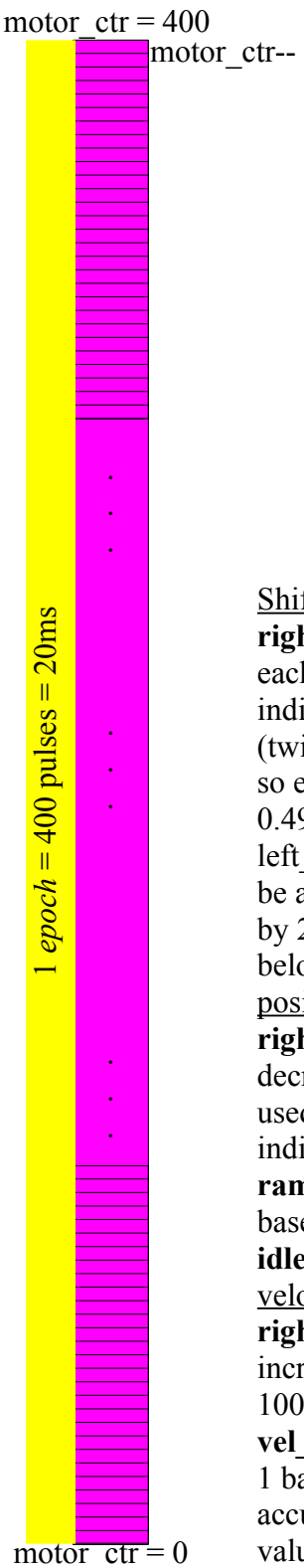


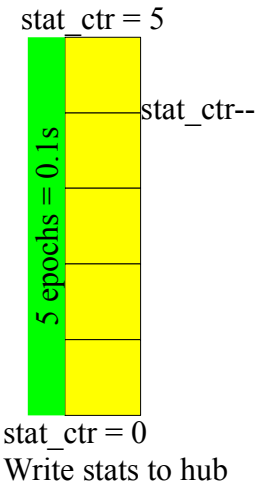


initialize distances, directions, speeds, all timers, all counters, some flags

motor_timer ← caller







Shift registers

right_enc, left_enc bits shifted in each encoder reading, transition indicated by difference in two LSBs (twice as many transitions as spokes, so each transition represents 0.491mm/2) right_distance (rd) and left_distance (ld) below assumed to be already corrected by multiplying by 2. Allincrementing/decrementing below done on encoder transition

position counters

right_count, left_count set to rd*ld, decremented by ld, rd respectively, used to synchronize wheels and indicate end of command.

ramp_count incremented by 1 based on dominant wheel

idler_cnt

velocity counters

right_vel, left_vel set to 0 and incremented by 1, accumulated for 100ms and used for stats only

vel_ctr set to 0 and incremented by 1 based on dominant wheel, accumulated for 20ms, latest full value stored in **cur_vel** and used to set **nominal_pwm**.