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 mtime = cnt + mdtmdtimeout = mdt0motor ctr = 400mdtimeout-motor ctr--Read/write to hub stat ctr = 5Set pulse width Read encoders stat ctr--Ш epochs waitcnt(mtime) mdtimeout = 0motor timer-initialize distances, directions, stat ctr = 0speeds, all timers, all counters, 1 pulse Write stats to hub some flags Shift registers epoch = 400 pulses = 20 msmotor timer \leftarrow caller right_enc, left_enc bits shifted in II motor timer--= 20 µs = each encoder reading, transition indicated by difference in two LSBs (twice as many transitions as spokes, II ndt = 4000 clocksso 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 g below done on encoder transition inmeout period 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 motor timer = 0**vel ctr** set to 0 and incremented by Stop motors 1 based on dominant wheel, Clear stats accumulated for 20ms, latest full Wait for next command mtime += mdtvalue stored in cur vel and used to motor ctr = 0

motor_ctr = 0 stat_ctr--Adjust PWM with P(ID) on speed

0 value stored in **cur** set **nominal_pwm**. 1