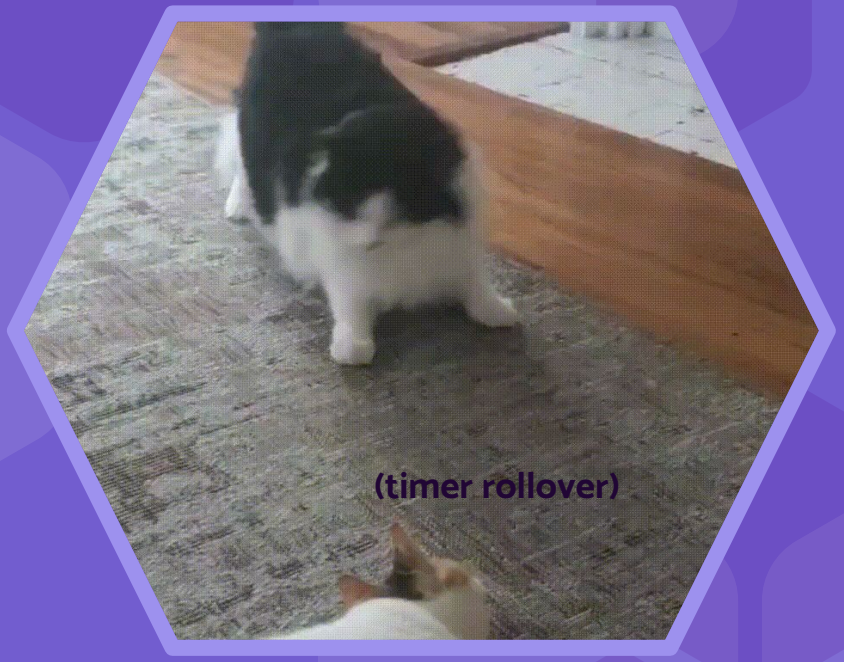
Scheduling, concurrency, timers, and clocks



Project

Reminder to post project ideas on Ed thread

You can also come to office hours to brainstorm ideas

Today

Where we've been:

Peripherals, embedded programming and CPU, memory models

Where we're going:

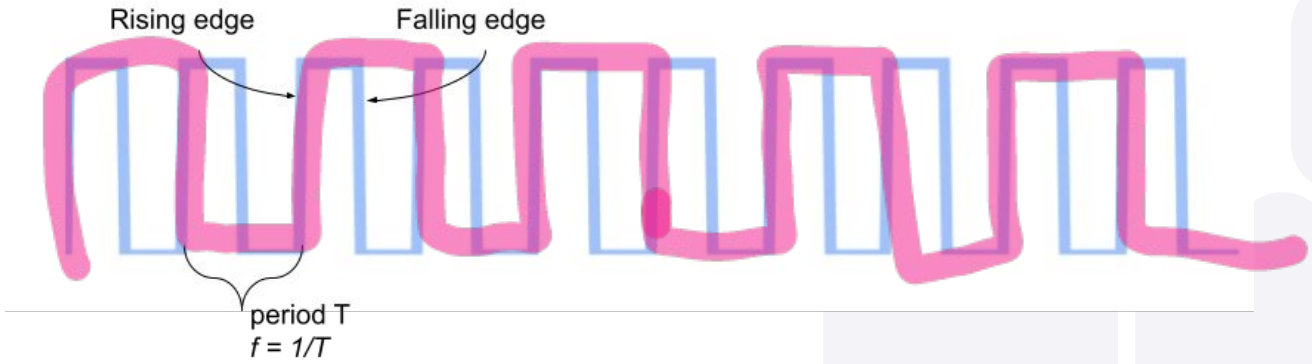
Time - clocks, timers, watchdogs

Brief introduction to scheduling (execution time, concurrency)

Keeping track of time: system clocks

Or “oscillators”

Basis of control of a CPU - instructions happen on “edges” of a clock (**why?**)



Counting time

Most basic way to keep track of time on a CPU: # of clock ticks

On an 8MHz CPU: 8 million clock ticks = 1 second

What is the largest unit of time we can keep track of in 32 bits on an 8MHz clock?

How do we keep track of longer time periods?

Timers

Keep track of time by incrementing every *n* clock ticks

- On MCUs: hardware support
- Often called something like TC (timer/counter) peripheral
- *Prescale* the clock (divide it by 2, 4, 8...) and increment on the clock ticks

Uses for timers

- Count to a specific number of clock ticks and generate an interrupt (you will do this in lab!)
  - How Arduino keeps track of time for millis()
- Check for rollover and use this as a low-overhead way to measure time
  - Polling or interrupt

Timer rollover math

48 MHz clock

Count every rising edge

32 bits: when will rollover happen?

$$\frac{2^{32}\ \text{"ticks"}}{48\times10^{6}\ \frac{\text{"ticks"}}{s}} = 89.48\ s$$

Keeping track of time without using floating point

Keep track of fractional seconds (say every 2^{-16} seconds)

- Precompute how many fractional seconds between each rollover (frequency math)
- Increment by that many fractional seconds in a counter

Quantization margins

With perfect timekeeping, # of fractional seconds expected in a day:

5662310400 $(24 \times 60 \times 60 \times 2^{16})$

48 MHz clock, pre-scaled by 16, 8 bit counter

- Effective frequency: 3 MHz (48/16)
- Rollover every 8.53×10^{-5} seconds
- = every 5.59 fractional seconds (~= 6)
- Rollovers in a day: 1012500000
- Fractional seconds counted: 6075000000
- Error: 7%

$\frac{2^8 \text{ "ticks"}}{3 \times 10^6 \frac{\text{"ticks"}}{\text{second}}}$

Clock drift

Imagine 32.768 kHz clock (common oscillator frequency - the SAM D21 has them too!)

0.001% drift rate (0.00001 seconds/second)

Drift during a day: ~.86 s

Drift during a year: ~ 315 s

Summary

What is the largest unit of time we can keep track of in 32 bits on an 8MHz clock? ~537 s

48 MHz clock, 32 bits: when will rollover happen? ~89 s

3 MHz clock, 8 bits: rollover every ~6 fractional second

When would you want to use a slower clock? A faster clock?

An 8-bit, 16-bit, or 32-bit counter?