NC STATE UNIVERSITY

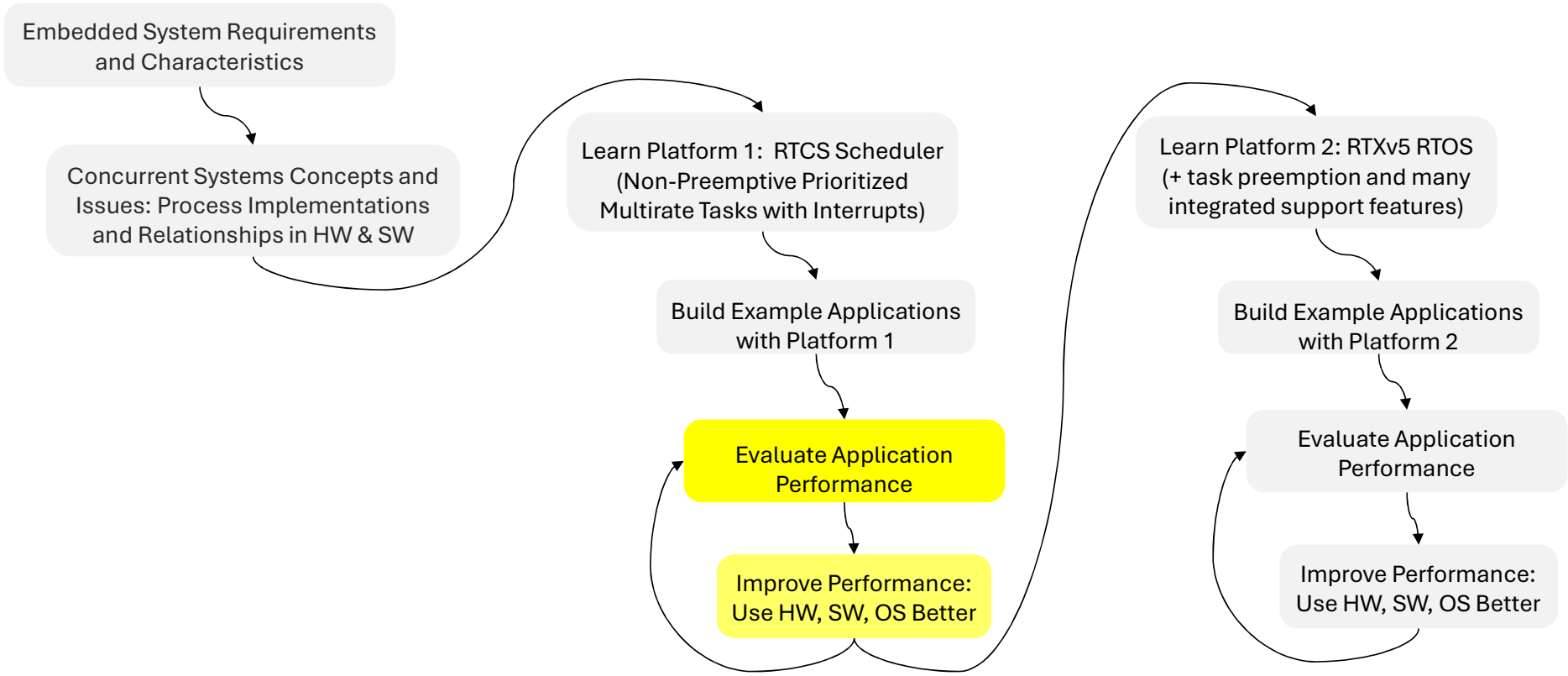
PERFORMANCE ANALYSIS OF EXAMPLE APPLICATIONS ON PLATFORM 1 (RTC SCHEDULER WITH INTERRUPTS, BASIC PERIPHERAL USE)

V2

9/30/2025

NC STATE UNIVERSITY

Where are we in the class?



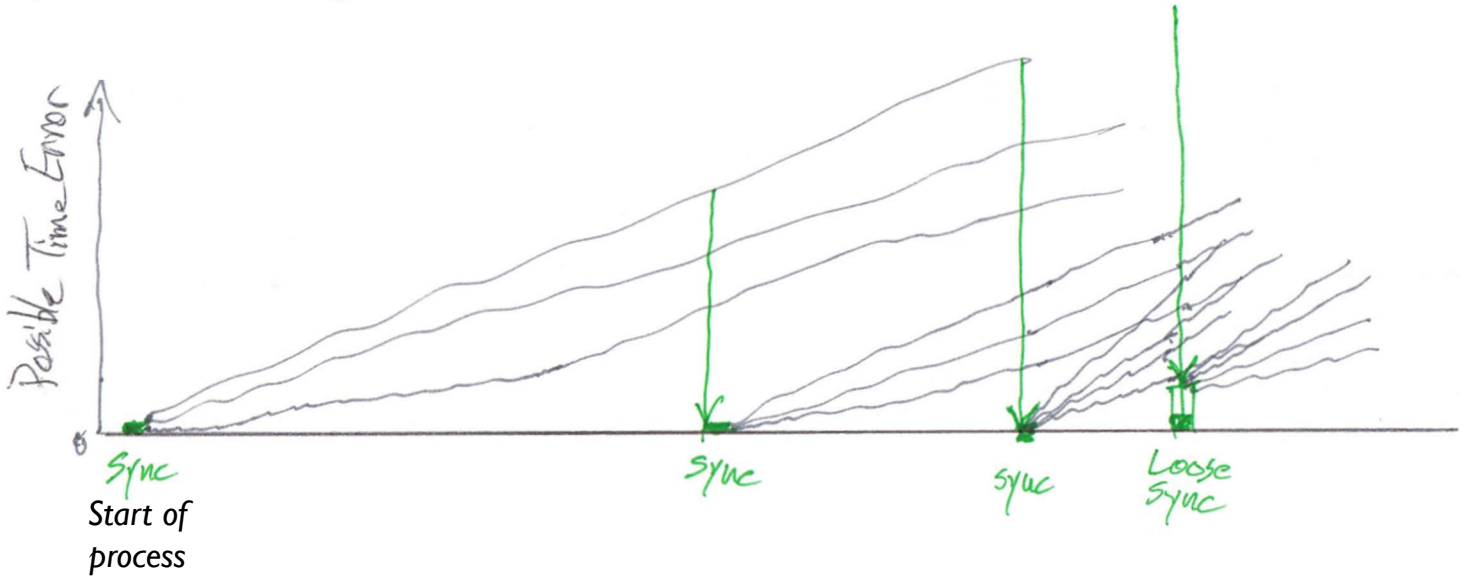
NC STATE UNIVERSITY

LN12 – Performance Analysis

- Synchronization: why, where and how
- Summarize key characteristics of first implementations
  - Processes, triggers (external sync), detectors, schedulers, workers, internal sync in workers
- Characterize basic performance limits of Platform 1 (48 MHz CM0+)
  - Interrupt System
    - Latency: 15 clock cycles
  - RTCS
    - Time-triggered scheduling resolution
    - Scheduler latencies: time to examine table, find next ready task
- Evaluate first implementations vs. performance limits
  - Identify performance gaps and risks, mark in table
- Evaluate response times for first implementations vs. requirements
  - Identify performance gaps and risks, mark in table

PROCESS SYNCHRONIZATION: WHY, WHERE AND HOW?

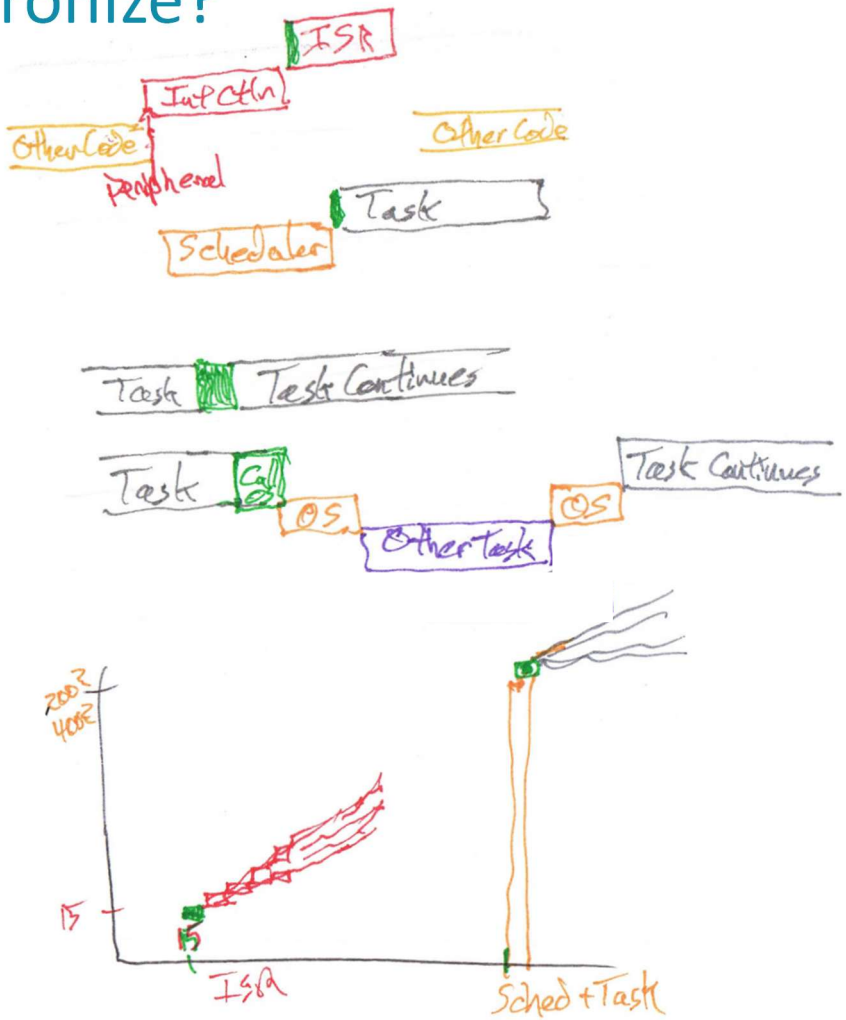
Why Do We Synchronize Processes?



- One reason: sync to external events/timing requirements
  - Timing variability (-> error) accumulates as process executes
  - Synchronization removes accumulated timing error in process since last sync.
- Note
  - Synchronizing code C to event E means code C will not run **before** E, but **after** E (if at all).
  - How long after? It depends….

Where and How Can a Process Synchronize?

- At start of process
  - Start of interrupt handler
    - **Interrupt system** synchronizes start of handler (ISR) to **after** trigger event (at least 15 cycles after for Cortex-M0+ CPU)
  - Start of task function
    - **Task scheduler** synchronizes start of task function to **after** trigger event
- Anywhere within process
  - **Explicit synchronization code** within task code
    - Test for condition. If true, then execute code
  - **Synchronization call to OS** within task code
    - Available? Depends on task scheduling
      - Probably not supported if non-preemptive (e.g. RTCS)
      - Probably supported if preemptive (e.g. RTXv5, FreeRTOS, etc.)
    - How?
      - Task calls an OS sync. function. Function doesn't return (blocks) until after sync. event occurs.
      - **Scheduler** synchronizes **resumption** of task to **after** trigger event
- Tightness of synchronization depends on method used



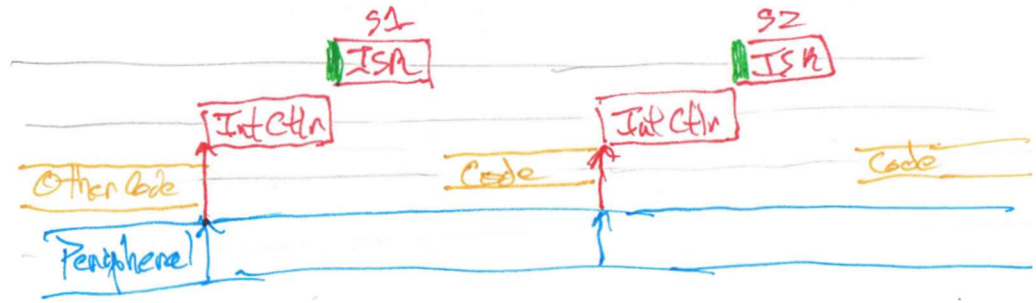
NC STATE UNIVERSITY

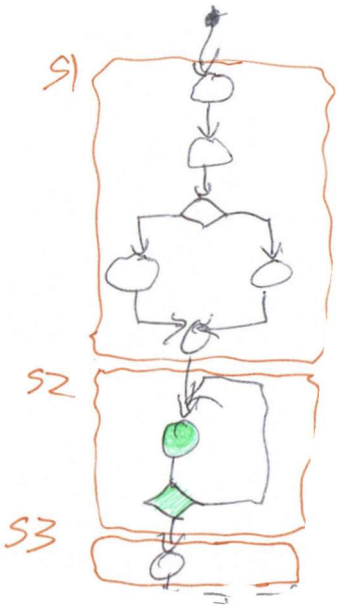
Why Does/Doesn't OS Support It?

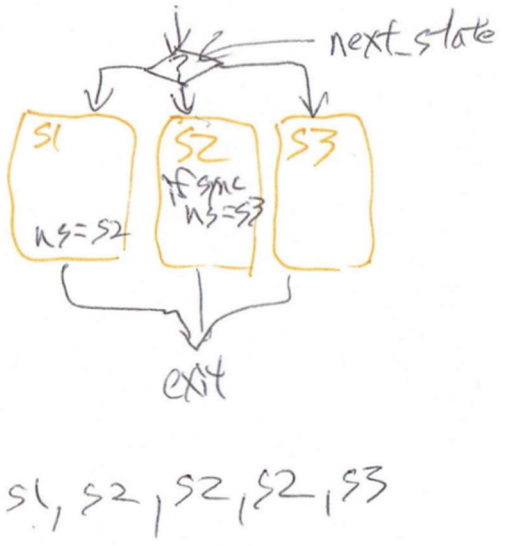
- With task preemption, tasks may be preempted partway through, so scheduler must be able to resume task partway through.
- Not much more work needed to offer synchronization (which may require blocking) partway through task

How to Sync Within a Run-to-Completion Process?

- Background
  - Applies to all ISRs (even with task preemption), all tasks with RTC scheduler
  - Interrupt system, RTC scheduler can only run the process from its first instruction
- Convert process to finite state machine
  - Each call to process executes code for one state, allowing this process to block while sharing CPU with other processes
  - Example: I^2C ISR in KL25Z Reference Manual







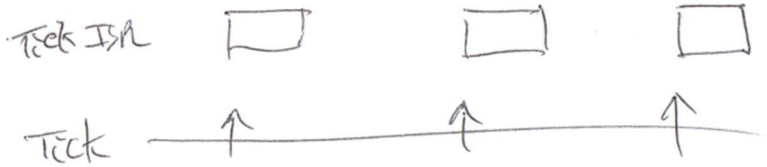
NC STATE UNIVERSITY

PLATFORM LIMITS

NC STATE UNIVERSITY

Platform 1 Performance Limits

- Characterize basic performance limits of Platform 1 (48 MHz Cortex-M0+)
  - Interrupt System
    - Latency: 15 clock cycles
  - RTCS
    - Time-triggered scheduling resolution: 1 tick



f_{tick}	CPU cycles (48 MHz)
1 kHz	48,000
10 kHz	4,800
100 kHz	480
1 MHz	48

    - Scheduler latencies: time to examine table, find next ready task, call task function

NC STATE UNIVERSITY

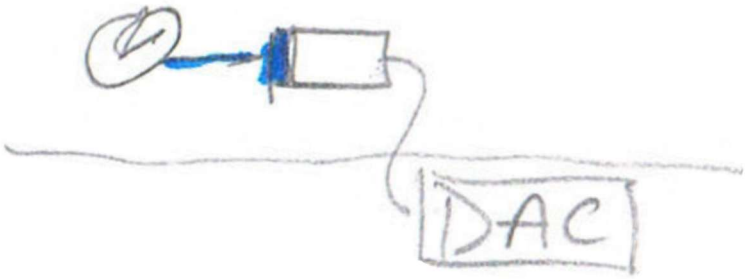
SUMMARY AND EVALUATION OF INITIAL APPLICATION DESIGN APPROACHES

Examples

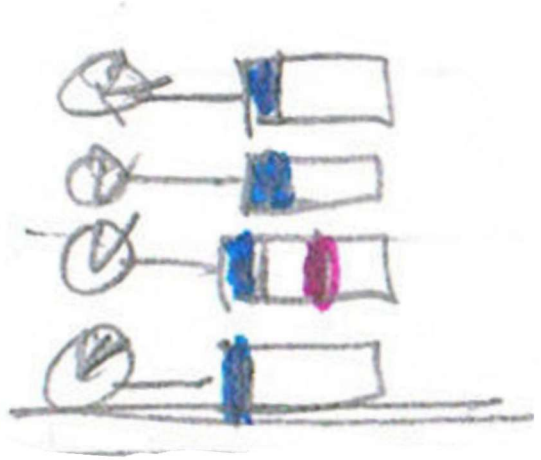
Event Trig.
Time Trig.

- Quadrature Decoder w/Limit Switch

- Waveform Generator

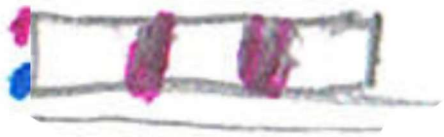


- Blinky Control Panel



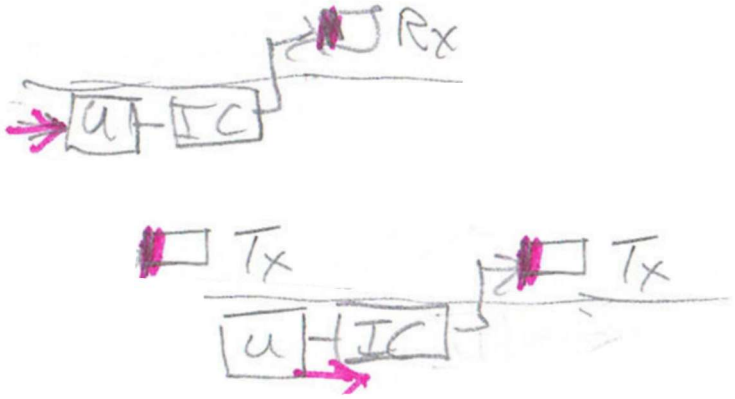
Examples

- Touchscreen

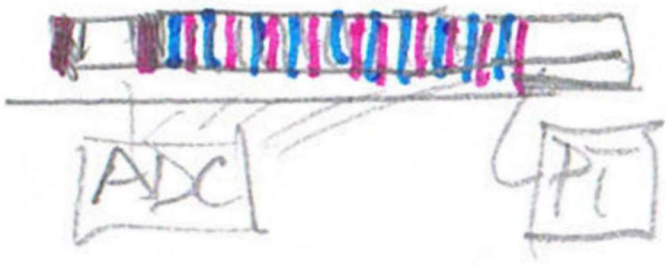


- LCD Controller

- Serial UART Communications

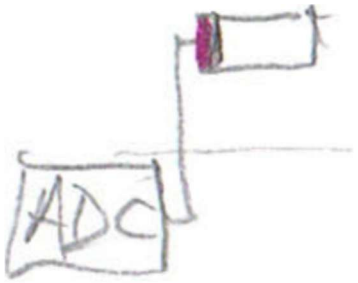


- Scope

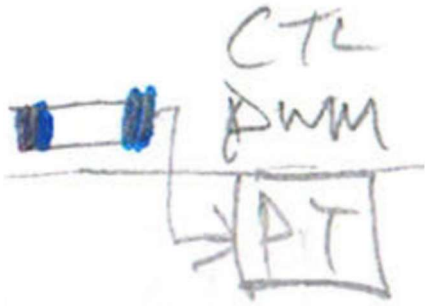


Examples

- SMPS Controller



- PWM



NC STATE UNIVERSITY

Evaluation of Performance Gaps and Risks

- Quadrature Decoder w/Limit Switch
- Waveform Generator
- Blinky Control Panel
- Touchscreen
- Serial UART Communications
- LCD Controller
- Scope
- PWM
- SMPS Controller