

Integracja usług RTOS i raport dowodów

barrier, IRQ, timer, pipeline, pressure, heap i stack HWM

Karta	K16 / 16	Czas	30 minut
Platforma	Hazard3 / RV32I	Trace	24 events
Dynamic	1 queue / 128 B	Digest	c474ac09
Wersja	v00.01	UUID	05b4f486-...
		karty	

Kompozycja, bez nowego prymitywu

```
StaticTask producer, processor, reporter;  
StaticQueue<Sample, 2> input;  
Queue<Report> output{2};           // one measured dynamic service  
EventGroup<ServiceBit> startup;  
PeriodicTimer<TimerContext> sampler;
```

Kontrakt K16. ISR i timer tylko budzą task. Pipeline, transformacja, backpressure i raport są task-context work. Każdy storage, priority, queue policy, failure i pomiar ma jawne uzasadnienie.

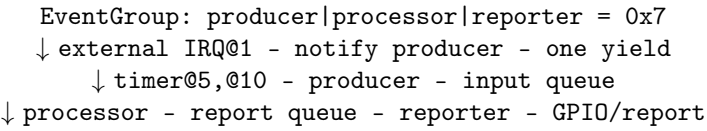
Plan 30 minut

Czas	Tryb	Dowód
0–5	inventory	static delta 0; dynamic queue 128 B
5–10	startup	barrier mask 0x7
10–15	input	IRQ@1, timer@5/10
15–20	pipeline	3 reports, sum 120
20–25	pressure	peaks 2/2, drop seq 4, recovery
25–30	report	latency 6, HWM, digest

Predykcja

Który element może być dynamiczny i dlaczego? _____. Co musi się stać po zapełnieniu queue? _____.

1 Service graph i startup barrier



Pomiar	Predykcja	Odczyt
startup barrier mask	0x7	_____
ISR wake / yield calls	1 / 1	_____
ISR kick / producer tick	1 / 1	_____
reporter resume widzi kick	true	_____
timer callback ticks	5, 10	_____
timer active po stop	false	_____

Granice wykonania

- external ISR: clear source, notification FromISR, one yield;
- timer daemon: bounded `xTaskNotifyGive(producer)`;
- producer: tworzy sample i realizuje pressure policy;
- processor: transformuje value $\times 2$;
- reporter: agreguje pod `CriticalSection`, ustawia GPIO.
Dlaczego transformacji nie wykonuje timer callback?

Timestamped trace

Pierwszy sample: tick _____. Ostatni report: tick _____. Latencja: _____ ticków. Liczba eventów: _____.

2 Pressure injection i recovery

Zdarzenie	Predykcja	Odczyt
sample seq 1 / value 10	accepted	_____
burst seq 2 i 3	accepted	_____
seq 4 przy full input	drop-newest	_____
input queue peak	2	_____
report queue peak	2	_____
end marker po zwolnieniu miejsca	accepted	_____

Wynik pipeline

$$10 \cdot 2 + 20 \cdot 2 + 30 \cdot 2 = \underline{\hspace{2cm}}$$

Produced / processed / reported: _____ / _____ / _____. Dropped sequence: _____.

3 Memory inventory

Pomiar	Predykcja	Odczyt
static services heap delta	0 B	_____
dynamic report queue cost	128 B	_____
oversized allocation result	null	_____
free heap przed/po failure	equal	_____
minimum-ever free	$\leq$ current	_____

Stack report

Task	HWM words	Interpretacja
producer	169	_____
processor	183	_____
reporter	189	_____

Czy dodatni HWM sam dowodzi wystarczającego zapasu dla wszystkich wejść? _____. Dla-
czego? _____.

4 Hazard3/GDB i raport końcowy

```
make check
riscv64-unknown-elf-gdb build/task01_integration/prog.elf
b integration_debug_checkpoint
```

```
p/x g_barrier_mask
p g_isr_wake_requested
p g_timer_ticks
p g_samples_produced
p g_samples_processed
p g_reports_received
p g_input_queue_peak
p g_report_queue_peak
p g_pressure_dropped_sequence
p g_dynamic_queue_heap_cost
p g_allocation_failure_recovered
p g_producer_hwm
p g_processor_hwm
p g_reporter_hwm
p g_end_to_end_latency
p/x g_trace_digest
p g_integration_pass
```

Zaliczenie

- ☐ uzasadniam graph, priorities, ownership i storage;
- ☐ pokazuję barrier 0x7, IRQ@1 oraz timer@5/10;
- ☐ trzy sample dają reports i sumę 120;
- ☐ peaks 2/2, seq 4 drop i end-marker recovery;
- ☐ static delta 0, dynamic queue 128 B, failure bez wycieku;
- ☐ raportuję latency 6, HWM 169/183/189 i digest c474ac09.

Wyjście

Który invariant uzasadnia każdy użyty wrapper?

Jak odtworzysz ten raport bez screenshotu?

Koniec bloku K02–K16: dalszy projekt rozszerza aplikację, ale nie zmienia zaliczonych kontraktów storage, lifetime, ISR i evidence.