

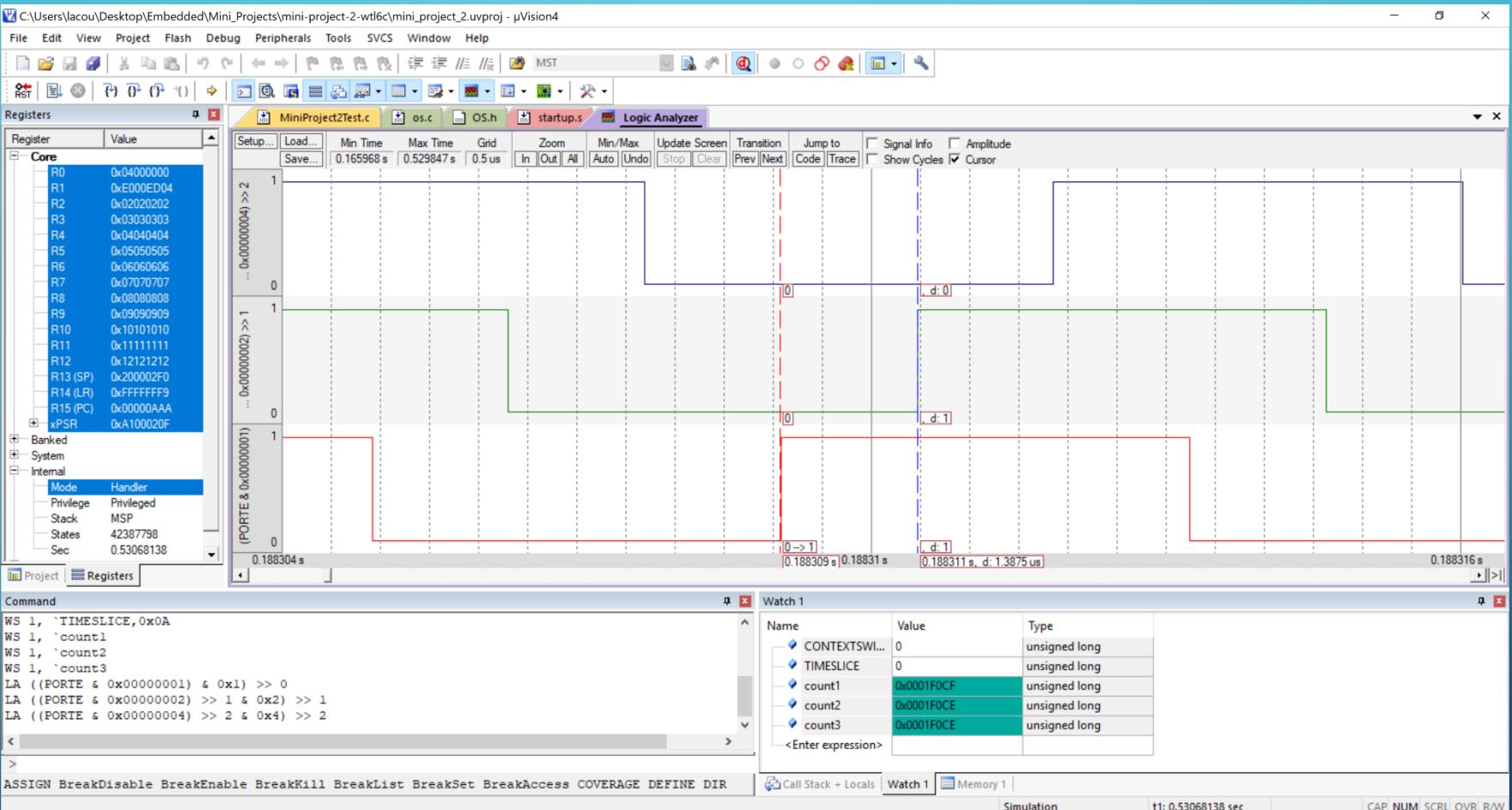


MINI PROJECT 2

TCB MANAGEMENT, BASIC SCHEDULING, TIMING FEATURES

WILLIAM TRACE LACOUR (WTL6C)

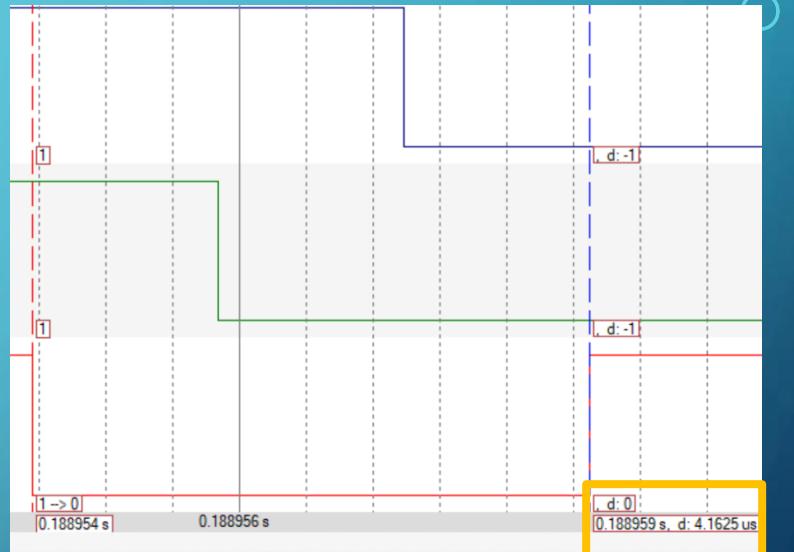
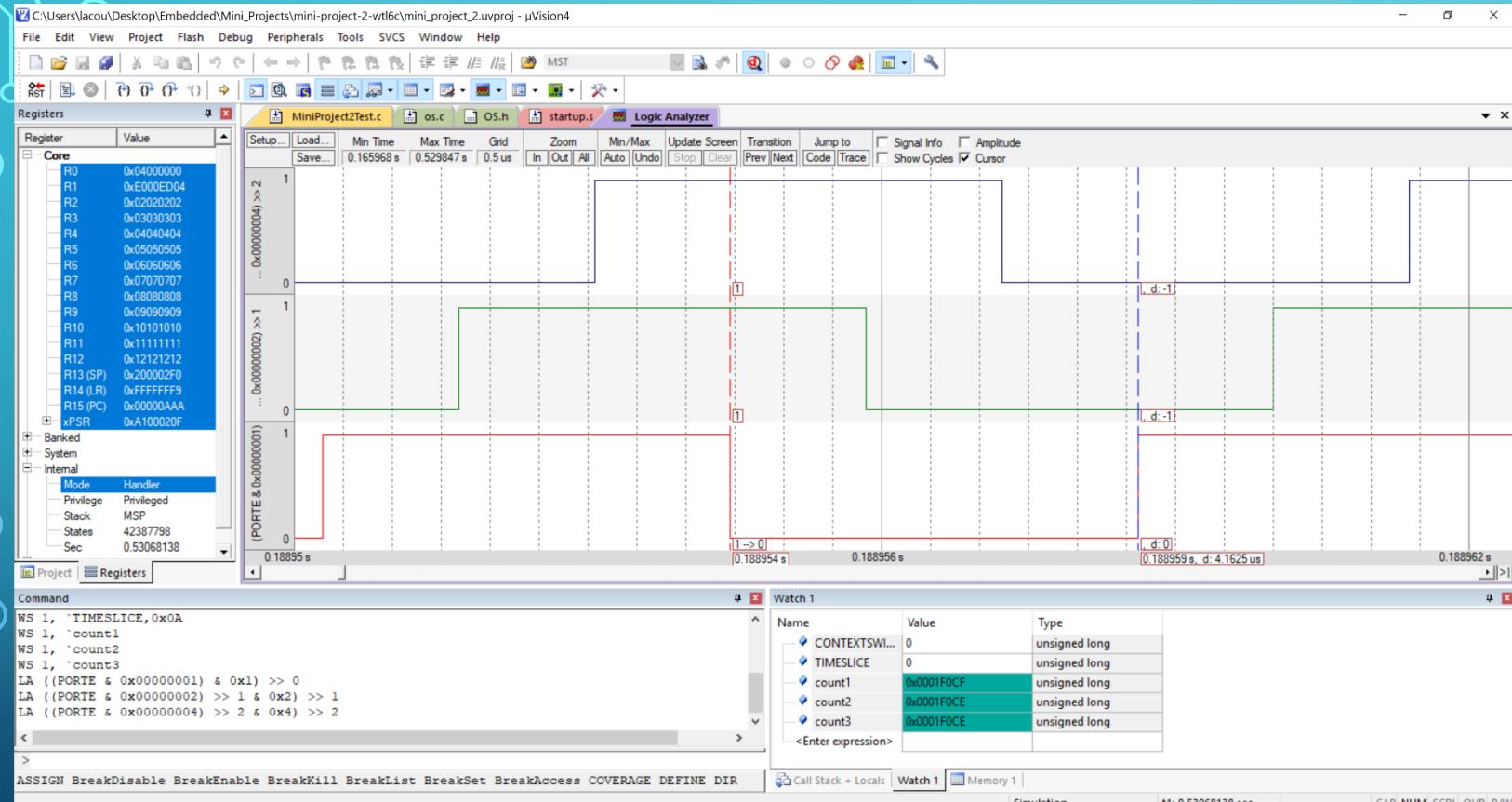
DELIVERABLES FOR PART 2A – TIME SLICE



$$1.3875 \mu\text{s} = \text{Time Slice} + \text{Context Switch Time}$$

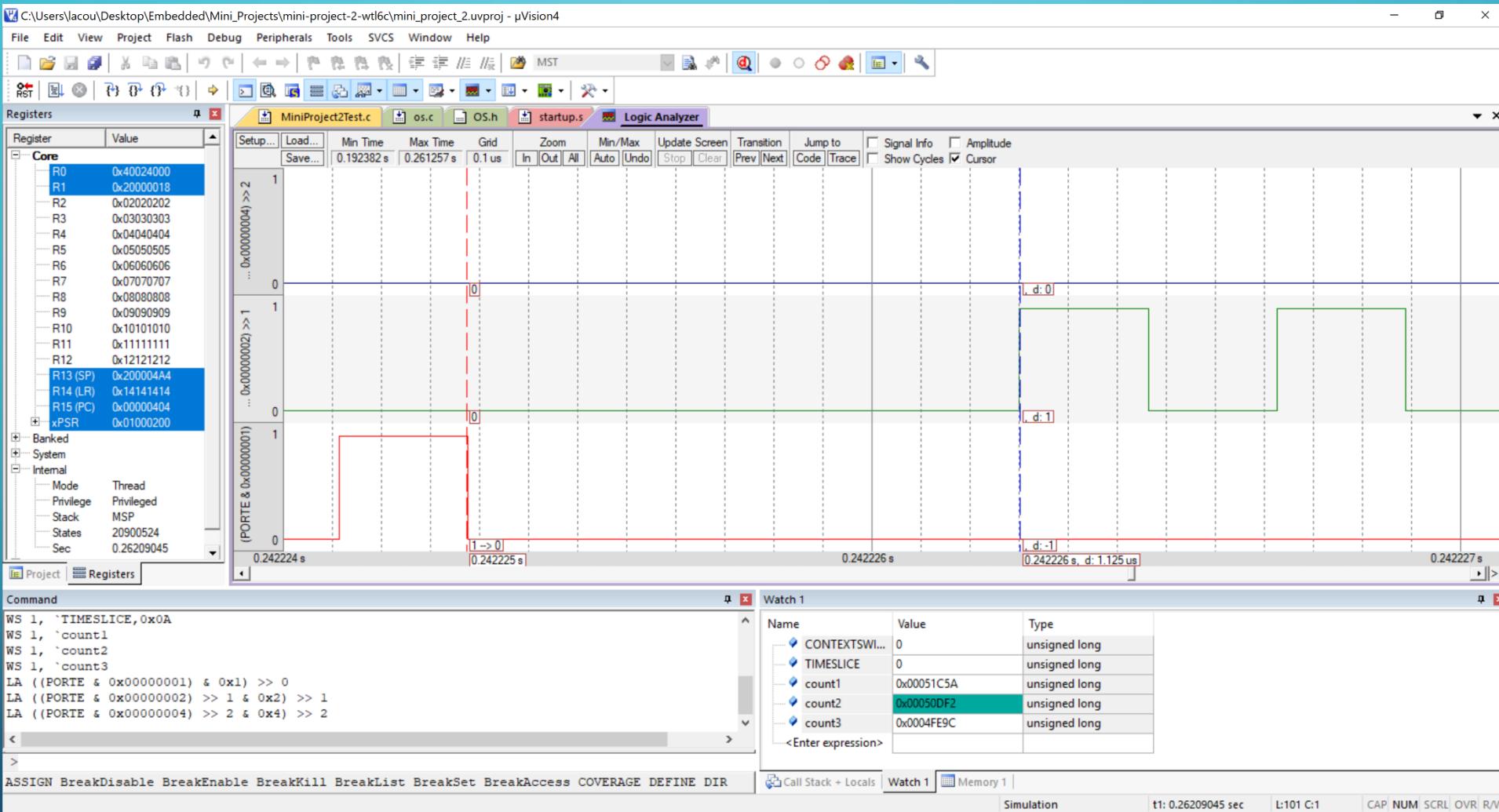
In the following slides, I will explain how to isolate the time slice. It requires that you know the context switch time, in this case I had to assume the context switch is the same as the one measured for TestMain0. Without changing the code you could not find the context switch time for TestMain1 but you could for TestMain2.

DELIVERABLES FOR PART 2A – FREQUENCY



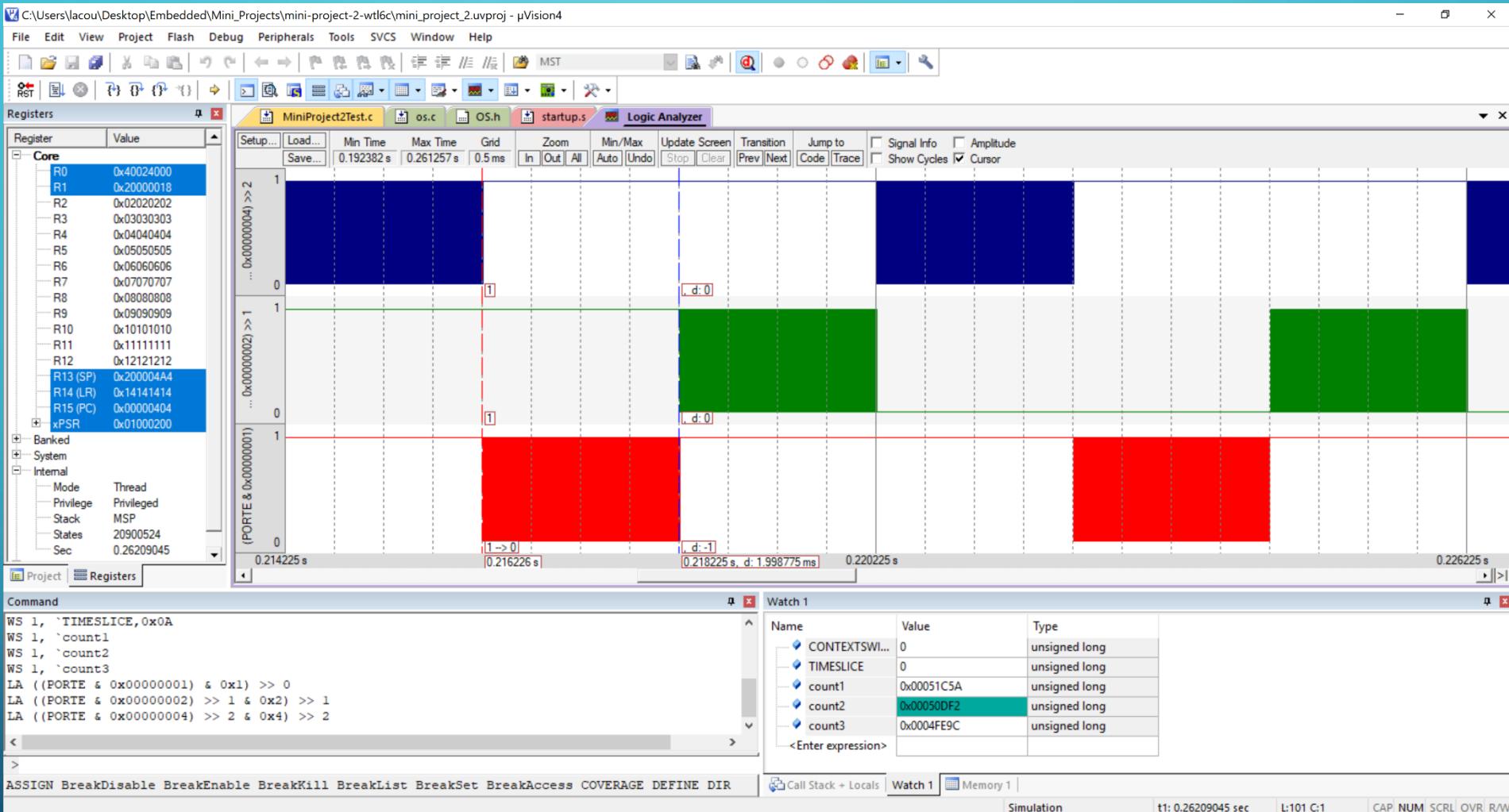
$$\text{Frequency} (\text{TestMain1}) = \frac{1}{4.1625 \mu\text{s}} = 240.240 \text{ kHz}$$

DELIVERABLES FOR PART 2B – CONTEXT SWITCH TIME



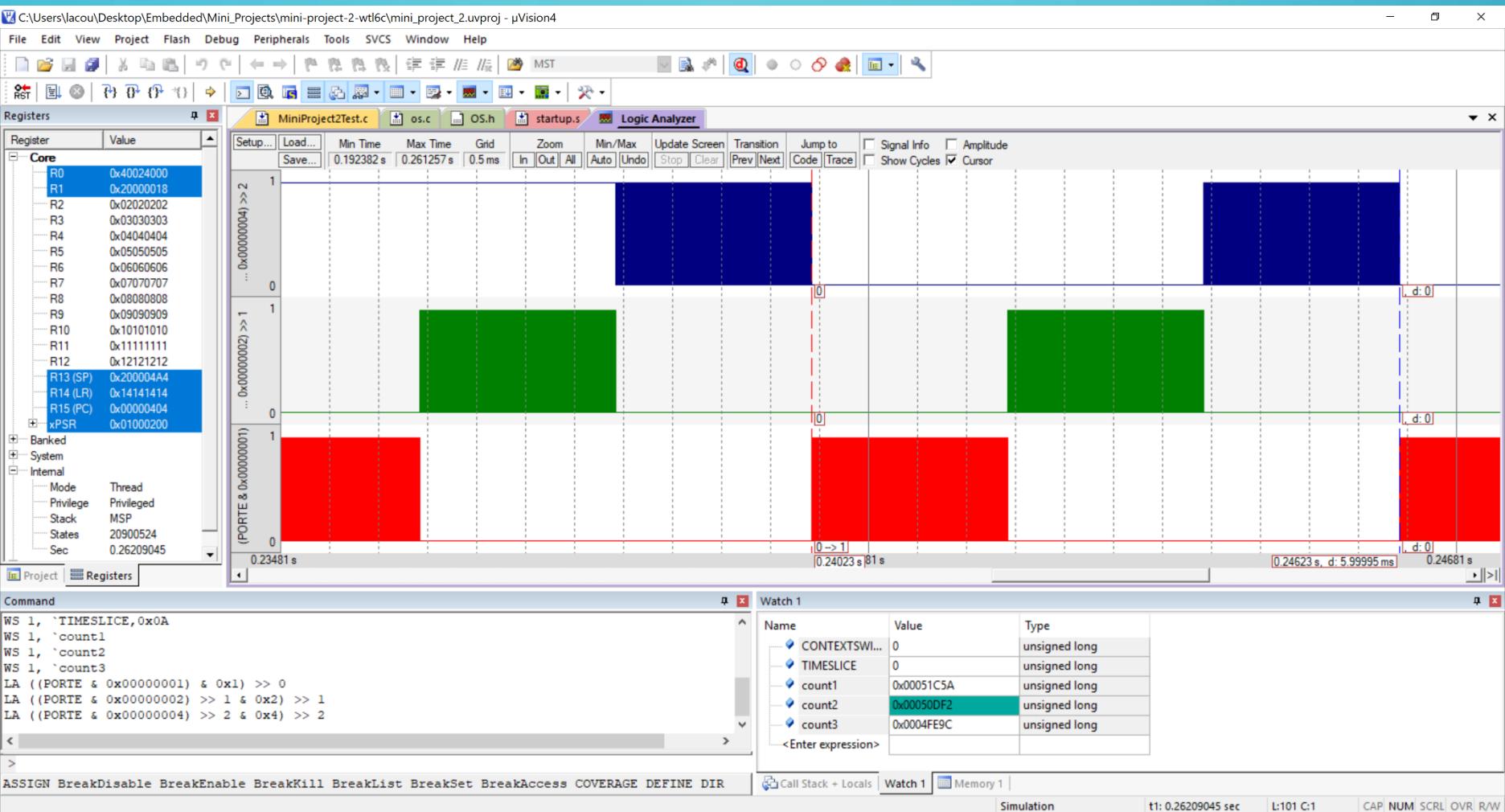
Context Switch Time = 1.125 μs

DELIVERABLES FOR PART 2B – TIME SLICE

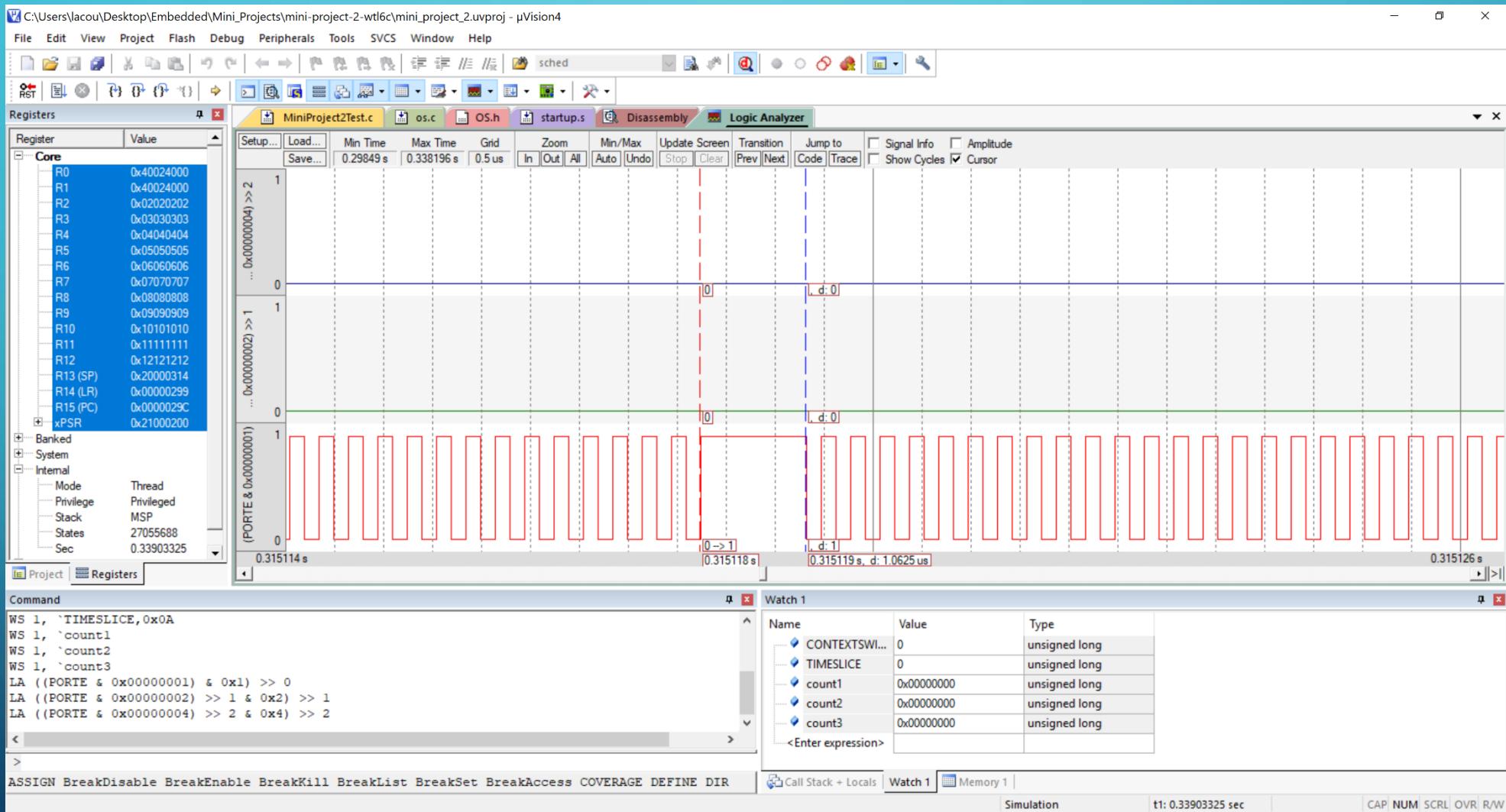


I zoomed in and found the last cycle before the context switch and measured that as my Time Slice. Adding the context switch time and the Time Slice should be roughly 2ms.

DELIVERABLES FOR PART 2B – FREQUENCY



DELIVERABLES FOR PART 2C – CONTEXT SWITCH TIME



Context Switch Time(*TestMain0*) = 1.0625 us

DELIVERABLES FOR PART 2 – SUMMARY

1. We found that the context switch time, the time it takes from leaving one thread and going to another, was about :
 - $1.0625 \mu\text{s}$ (TestMain0)
2. We also found that the total time to run a thread and context switch was about :
 - $1.3875 \mu\text{s}$ for TestMain1 (one time through the thread)
3. Knowing these both, the time slice for TestMain1 can be found by the following :

$$\text{Time Slice (TestMain1)} = 1.3875 \mu\text{s} - 1.0625 \mu\text{s} = 325 \text{ ns}$$

TestMain1

Frequency = 240.240 kHz
Time Slice = 325 ns

TestMain2

Frequency = 166.67 Hz
Time Slice = 1.998775 ms

TestMain0

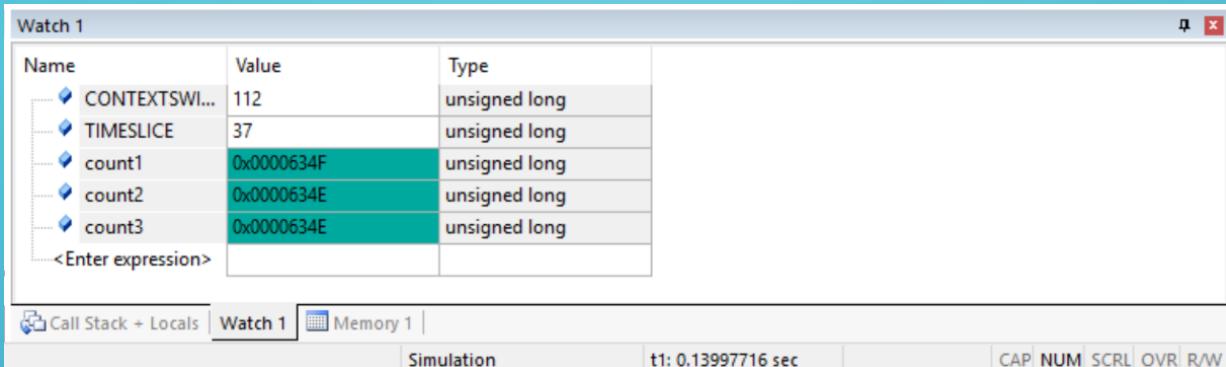
Context Switch Time = $1.0625 \mu\text{s}$

DELIVERABLES FOR PART 2D – QUESTION

Explain why the values of the counters will be much higher in TestMain2 than they were in TestMain1.

- The counter values were higher in TestMain2 because TestMain2 was allowed to run for 2ms where a interrupt would then switch to the next thread. TestMain1 was only allowed to run 1 time before manually switching to the next thread via the OS_Suspend() function. The fact that the TestMain2 got to run a lot longer before switching allowed for the values of the counters to increase to a higher value in a shorter amount of time.

DELIVERABLES FOR PART 4A – TESTMAIN1



TestMain1

Context Switch Time = $112 * 12.5 \text{ ns} = 1.4 \mu\text{s}$

Time Slice = $37 * 12.5 \text{ ns} = 462.5 \text{ ns}$

Overall values were close to that of Part 2,
just more instructions, making the overall time
longer

DELIVERABLES FOR PART 4A – TESTMAIN2

Name	Value	Type
CONTEXTSWI...	130	unsigned long
TIMESLICE	159842	unsigned long
count1	0x00001DBE	unsigned long
count2	0x00000F9C	unsigned long
count3	0x00000B8F	unsigned long
<Enter expression>		

TestMain2

Context Switch Time = $130 * 12.5 \text{ ns} = 1.625 \mu\text{s}$

Time Slice = $159842 * 12.5 \text{ ns} = 1.998025 \text{ ms}$

Overall values were close to that of Part 2, just more instructions, making the overall time longer

DELIVERABLES FOR PART 4B – QUESTION #1

Are there any differences between the values you measured using logic analyzer versus the measurements using OS_Time functions? Explain why.

- Yes, the reason being that if you were to look at the difference in the assembly breakdown there is simply more instructions that need to be run now that we have function calls to OS_Time and OS_Timedifference. Granted it won't be much of a difference, a couple microseconds at most.

DELIVERABLES FOR PART 4B – QUESTION #2

Are there any differences between the values you measured in Testmain1 versus Testmain2 function? Explain why

- Yes, TestMain1 had a smaller Time Slice than TestMain2 because it was only allowed to run once versus that of TestMain2 which had the entirety of the 2 ms. The context switch value was roughly the same, with only a few microseconds difference. Overall the context switch should be about the same but the Time Slice, the time a thread is allowed to run, is different because one uses OS_Suspend and one simply waits for the interrupt.

DELIVERABLES FOR PART 4B – QUESTION #3

Explain the purpose of using Timer1A, Timer2A, Timer3A, and Timer4A in this mini project.

1. Timer1A – used to run periodic tasks (runs in the background)
2. Timer2A – used when OS_Suspend triggers SysTick Interrupt as well as the when OS_Launch sets the interrupts and timers.
3. Timer3A – used to keep the system time via OS_Time
4. Timer4A – used to run periodic tasks (runs in the background)

SURVEY SCREENSHOT

