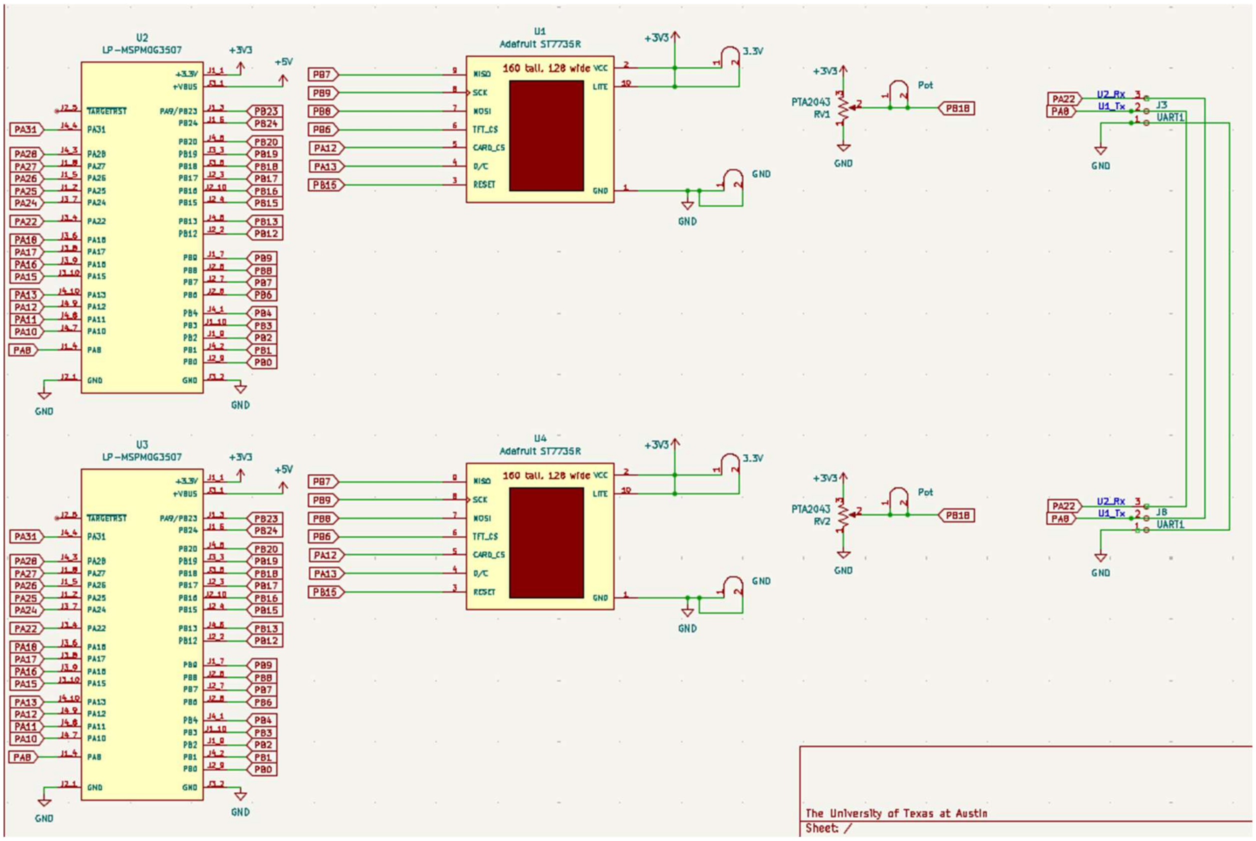
Travis Beach and Jonathan Esqueda

tmb3956 and je28736 Dr.Y

1.



2.

Question 1: Why does it take less than a 1us to call UART1_OutChar four times?

All UART1_OutChar does is write a character into the FIFO, which only takes a few bus cycles

Question 2: We know it takes 5ms to output a frame (Figure 8.5). If your UART1_OutChar does not wait for the last character to be sent (it is blind), why is data not lost if one outputs to UART1->TXDATA four times in a row?

Because the data is saved immediately in the FIFO before the TXDATA register receives a new value

Question 3: What condition triggers the receive timeout interrupt?

Setting bit 0 of the CPU_INT.IMASK triggers the receive timeout interrupt

UART2->CPU_INT.IMASK = 0x0001;

Question 4: The interrupt trigger flag is bit 0 in the RIS register. How is this bit cleared (acknowledged)?

Reading the UART2->CPU_INT.IIDX register also clears bit 0 in the register

status = UART2->CPU_INT.IIDX; // reading clears bit in RIS

Question 5: Use the triple toggle to measure how long it takes to run the UART2 ISR (t1) and the time between running UART2 ISRs (t2). Calculate the overhead of the receive tasks (t1/t2).

t1 time for 3 toggles to happen 340 us

t2 time to next toggle is 33.5 ms

t1/t2 =0.0101

Question 6: The TimerG12ISR runs every 33.3 ms, and each ISR calls UART1_OutChar 4 times. Does this transmit hardware FIFO ever fill? In other words, does the blind UART1_OutChar ever lose data?

No because we send 4 characters every 33.3 ms but we can pull off of the FIFO at a faster rate than the 33.3 ms we push onto.

Question 7: Use the triple toggle to measure how long it takes to run the TimerG12 ISR (t3) and the time between running TimerG12 ISRs (t4). Calculate the overhead of the transmit task (t3/t4).

t3 time for 3 toggles to happen 198 us

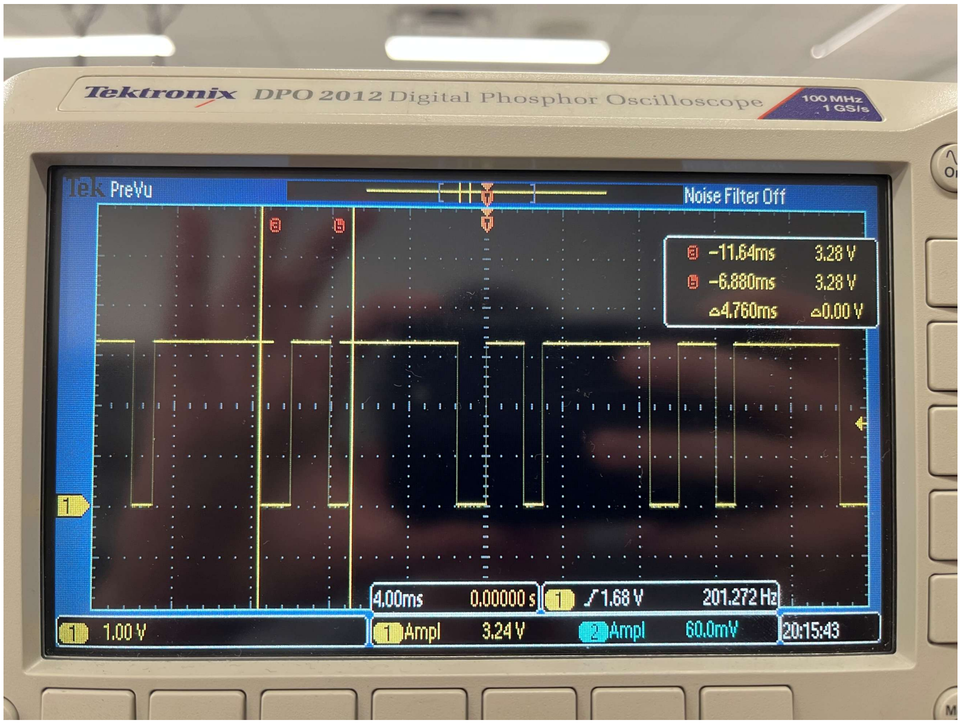
t4 time to next toggle is 18.1ms

t3/t4 =0.0109

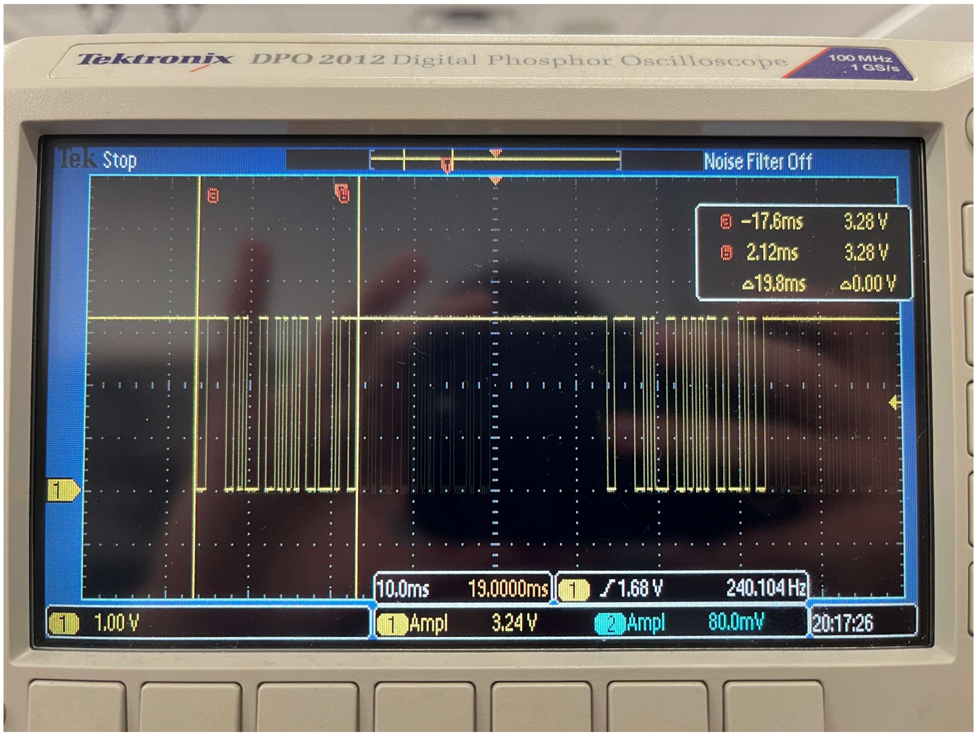
Question 8: Use the two triple toggles to measure how long it takes between sampling the ADC on the producer and putting the data in FIFO1 on the consumer. This time is the network latency.

22.3 ms network latency

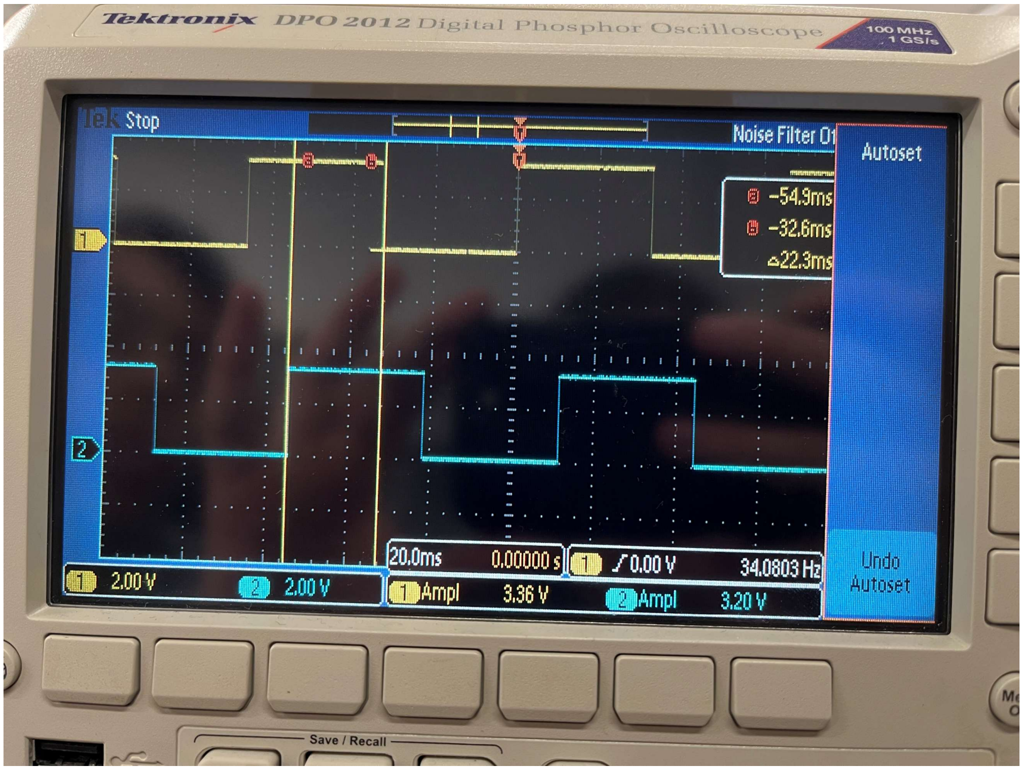
3. Capture a waveform like Figure 8.5 zoomed to see one frame.



4. Capture a waveform like Figure 8.6 or Figure 8.10 zoomed to see one message



5. Capture a waveform like Figure 8.11, zoomed to see both the G12 ISR and the UART2 ISR



6. The sampling rate is 30 Hz, what changes would you have to make to sample 33 times faster, at 1000 Hz?

TimerG12 arm at 80M/1000 instead of 80M/30

(instead of using 2,666,667 for Timer G12’s period, use 80,000)