LED Learning Activity

Spencer Hernandez - **ECE 3430** - August 23, 2022

I. Intro

The purpose of this lab was to familiarize ourselves with the MSP432 Micro-controller and the Embedded C Programming Language. We did this by generating a square waveform that had a max amplitude of $3.3V$, a duty cycle of 50%, and a frequency of $\sim 2\,Hz$ in the Code Composer Studio IDE.

II. Part A

This part was done by outputting a logic 1 to the LED port (P1.0), waiting for 1/4 of a sec, outputting a logic 0, and waiting for 1/4 of a sec. This process was done in a while loop to keep it continually running. The output waveform is shown below in Figure 1.

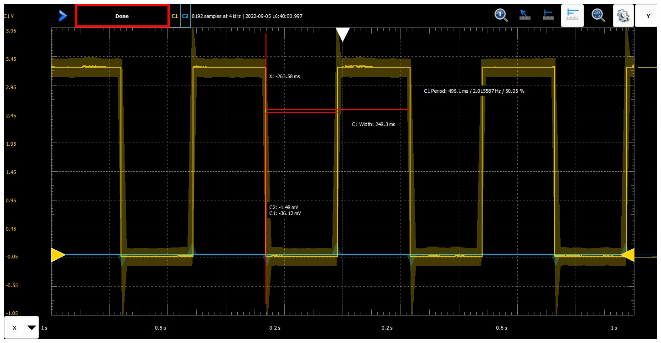


Fig. 1. Square Waveform for Part A

III. Part B

This part was done by calling a function, TimeDelay(), that called the for loops and two other bit-commands from Part A to better familiarize ourselves with C functions. The output waveform is shown below in Figure 2.

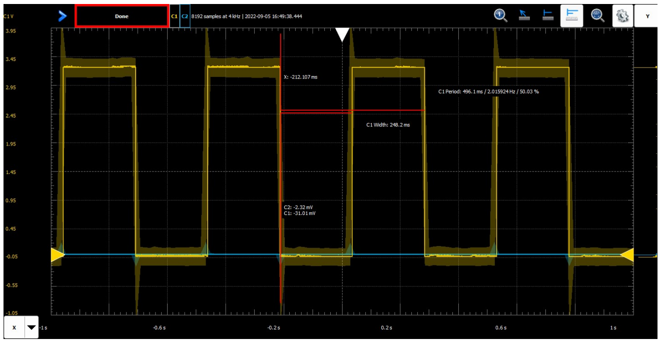


Fig. 2. Square Waveform for Part B

IV. Part C

This part was done by using an enumerated type to control the LED connected to the port. Several header files were used for better design implementation, as well as preprocessor define statements for more readable code. This program enabled not only P1.0, but also P2.1, allowing 2 LED lights to blink and 2 waveforms to be generated. They are shown below in Figures 3 & 4.

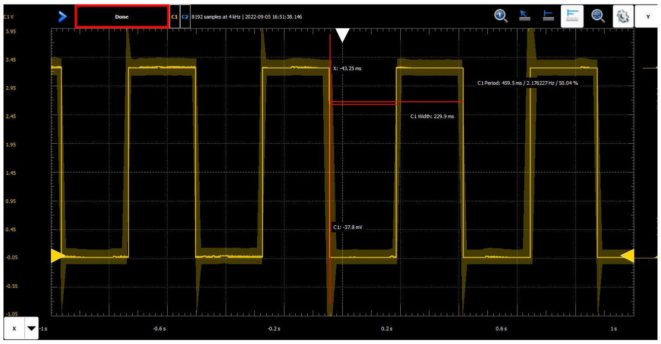


Fig. 3. Square Waveform for Part C P1.0

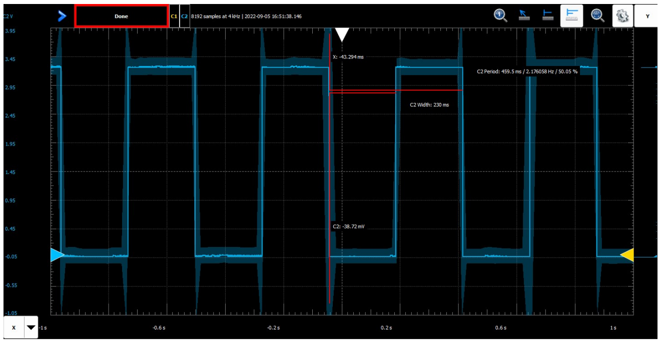
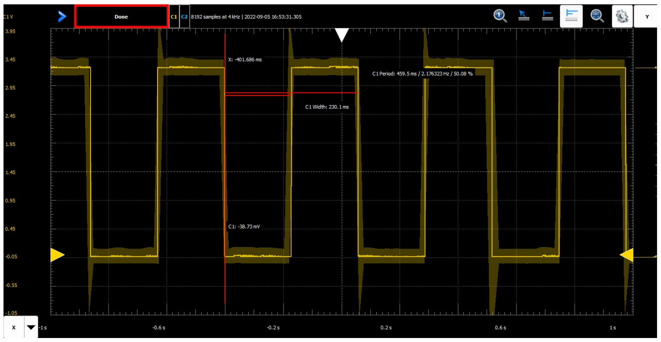


Fig. 4. Square Waveform for Part C P2.1

V. Part D

This part was done by using structs to control the LEDs connected to P1.0 and P2.1. A single function was created to manipulate the On/Off state of the LED and assign a value to a port. These structs utilized pointers that held the output port's register, as well as a bitmask. The output waveforms for P1.0 and P2.1 are shown below in Figure 5 & 6.



8192 samples at 4 kHz | 2022-09-05 16:53:31.305

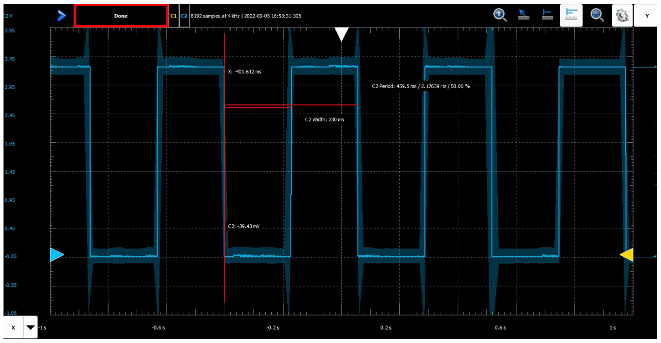
X: -401.686 ms

C1 Period: 459.5 ms / 2.176323 Hz / 50.08 %

C1 Width: 230.1 ms

C1: -38.73 mV

Fig. 5. Square Waveform for Part D P1.0



8192 samples at 4 kHz | 2022-09-05 16:53:31.305

X: -401.612 ms

C2 Period: 459.5 ms / 2.17639 Hz / 50.06 %

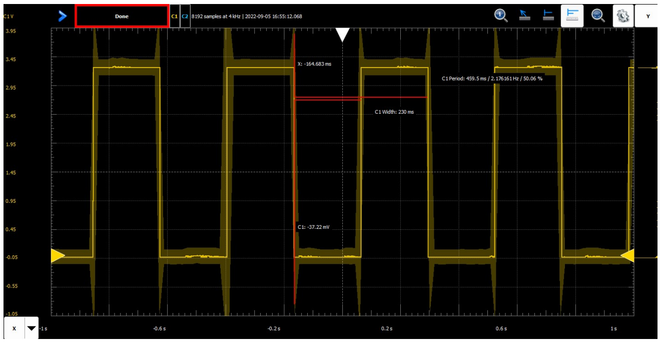
C2 Width: 230 ms

C2: -39.43 mV

Fig. 6. Square Waveform for Part D P2.1

VI. Part E

This part was done by using a function pointer to control the LED connected to P1.0 and P2.1. This function pointer is then called to use the function that turns the LEDs on and off. An array was also used to not only familiarize ourselves with C arrays, but also toggle as many LEDs as desired (2 in our instance). The output waveforms for P1.0 and P2.1 are shown below in Figure 7 & 8.



8192 samples at 4 kHz | 2022-09-05 16:55:12.068

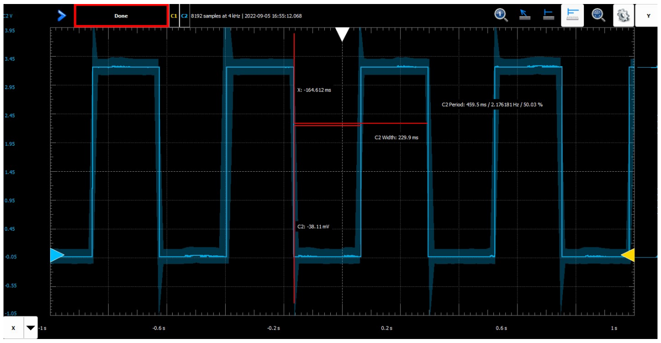
X: -164.683 ms

C1 Period: 459.5 ms / 2.176161 Hz / 50.06 %

C1 Width: 230 ms

C1: -37.22 mV

Fig. 7. Square Waveform for Part E P1.0



8192 samples at 4 kHz | 2022-09-05 16:55:12.068

X: -164.612 ms

C2 Period: 459.5 ms / 2.176181 Hz / 50.03 %

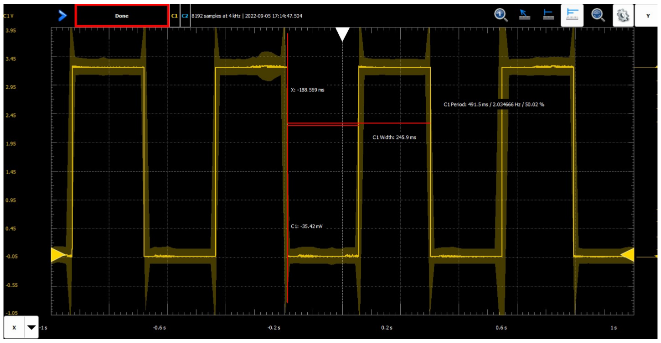
C2 Width: 229.9 ms

C2: -38.11 mV

Fig. 8. Square Waveform for Part E P2.1

VII. Part F

Lastly, this part was done by implementing code that configures the actual clock system of the MSP432. The output waveforms for P1.0 and P2.1 are shown below in Figure 9 & 10.



8192 samples at 4 kHz | 2022-09-05 17:14:47.504

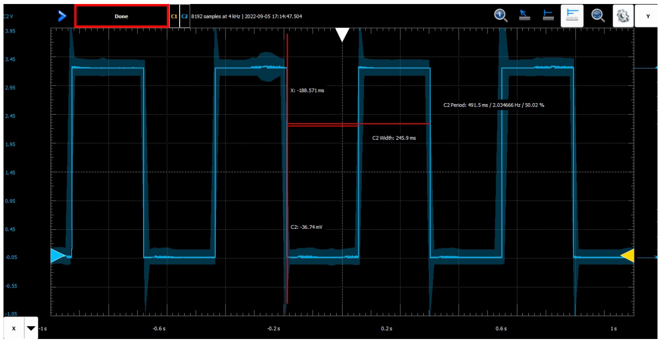
X: -188.569 ms

C1 Period: 491.5 ms / 2.034666 Hz / 50.02 %

C1 Width: 245.9 ms

C1: -35.42 mV

Fig. 9. Square Waveform for Part F P1.0



8192 samples at 4 kHz | 2022-09-05 17:14:47.504

X: -188.571 ms

C2 Period: 491.5 ms / 2.034666 Hz / 50.02 %

C2 Width: 245.9 ms

C2: -36.74 mV

Fig. 10. Square Waveform for Part F P2.1