CPE 3500

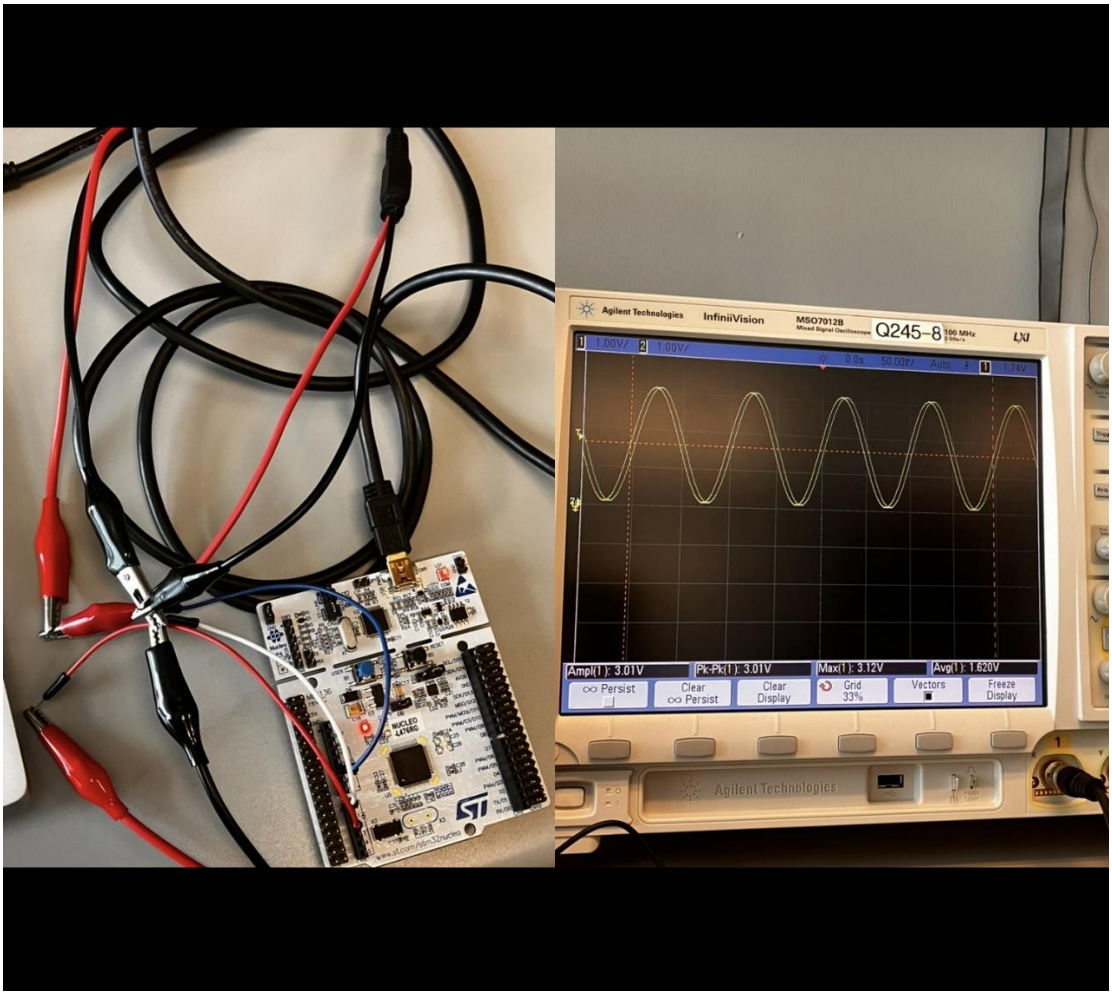
Embedded Digital Signal Processing

Lab 5: Digital to Analog Conversion through ADC and DMA

Anindita Deb

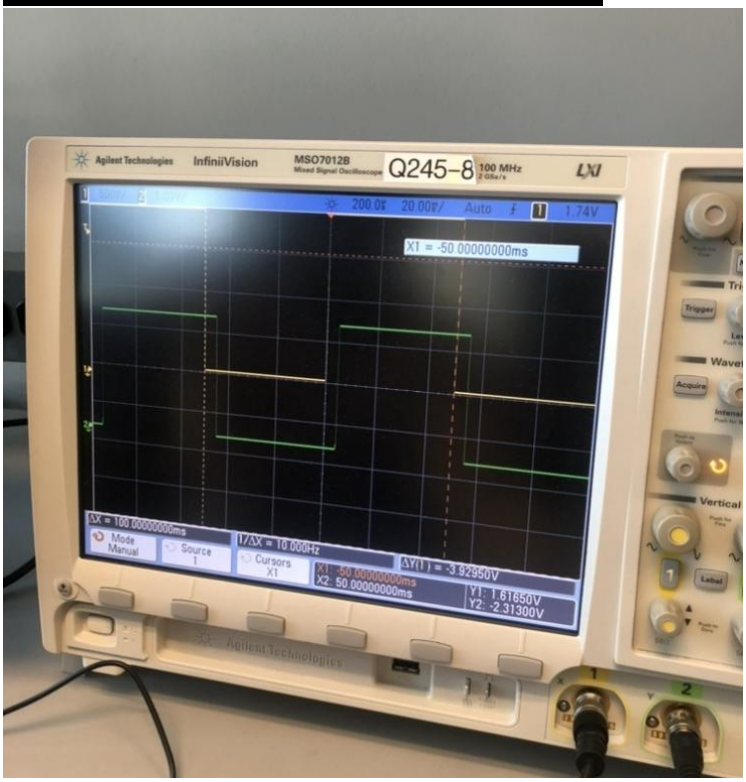
10/15/2024

- Task-1: Insert the picture of the oscilloscope display with label and explanation.



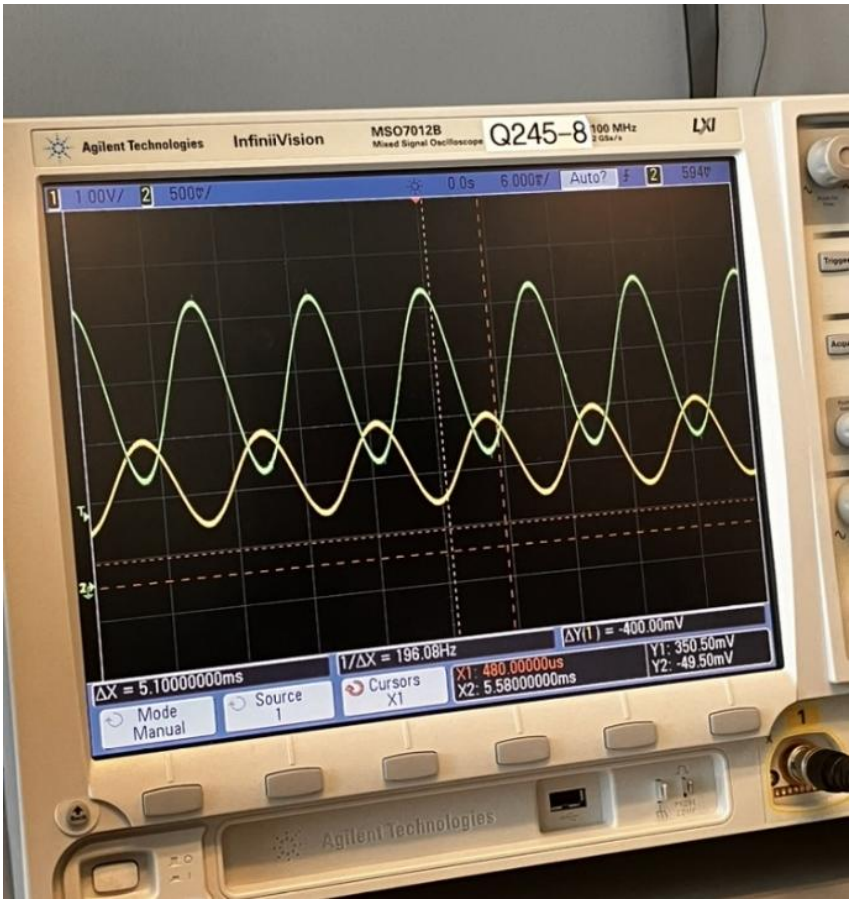
When converting from digital to analog the information being read then stored processed and spit out took up a tiny amount of time. This is the delay we see here on the sine waves between the signal generator wave and the converted one from the microcontroller.

- Task-2: Insert the picture of the oscilloscope display with label and explanation. Report the measured delay and comment on the result.



It's hard to see where exactly the peak of each wave is on the sine so when we change it to square we can more definitely see the start and end of the waves. We got 5 microseconds delay when measuring using cursors on the square waves.

- Task-3: Insert the picture of the new signals. Comment on the DC value differences.



The sine waves are flipped and as mentioned in class there is a delay, but not significant to the eye.

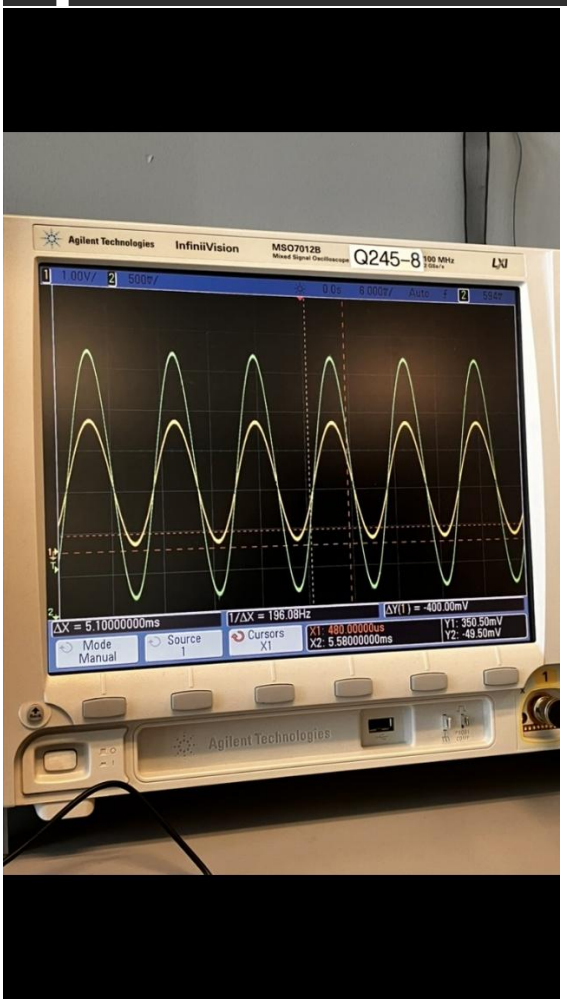
DC Difference=2.4535V−(−0.3465V)=2.8V

- Task-4: Include the modified code in callback functions. Insert the picture of the oscilloscope display.

```
/* USER CODE BEGIN 4 */
#define MIDPOINT 2048  // 12-bit ADC midpoint for full range (0 to 4095)

void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef *hadc)
{
    for (int n = 0; n < BUFFER_HALFSIZE; n++)
    {
        uint32_t adc_value = adc_buffer[n];
        // Invert the sine wave value around the midpoint
        dac_buffer[n] = MIDPOINT + (MIDPOINT - adc_value);
    }
}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    for (int n = BUFFER_HALFSIZE; n < BUFFER_SIZE; n++)
    {
        uint32_t adc_value = adc_buffer[n];
        // Invert the sine wave value around the midpoint
        dac_buffer[n] = MIDPOINT + (MIDPOINT - adc_value);
    }
}
/* USER CODE END 4 */
```



- Conclusion (1-paragraph)
  The overall lab was not difficult. The logical portion of the sine way flipping took a moment but, overall was doable within 10 minutes of starting the coding process for it. The cursors portion was a bit confusing as we didn't realize we could change them in 4 different ways (2 for x-axis and 2 for y-axis). We just ended up taking multiple pictures for each orientation and calculated the difference between the measurements. We needed to be careful with the signal generator connection to the microcontroller and vice versa (he had wires the wrong way and weren't getting a clean signal). With those issues aside, the overall lab was far easier than the previous (built from on top of it).