CMPE 460 Laboratory Exercise 5

MSP432 Timers, Interrupts, Analog-to-Digital Converter and SPI DAC

Hunter Culverhouse, Gloria MBaka
Performed: 02/14/24
Submitted: 03/06/24

Lab Section: L1

Instructor: Prof. Ketout

TA: Jack Sipley
Andrew Tevebaugh
Colin Vo

Lecture Section: 01
Professor: Prof. Ketout

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: ____________ 03/06/2024

Abstract

An interrupt is a signal to the processor from hardware or software indicating an event that requires immediate attention. The Texas Instruments MSP432 uses multiple modules with interrupts to communicate with the microprocessor. This exercise aims to improve understanding of using interrupts in various modules, employing timers for time generation and measurement, digitizing analog signals with a photo-electronic device, and reading analog input from a line scan camera. The exercise was divided into three parts. Firstly, Timer32-1 and Timer32-2 modules were configured to generate interrupts at half-second and millisecond intervals, activated by two switches for different functions: Timer32-1 to toggle an LED, and Timer32-2 to measure the time between presses of the second switch, displaying this duration in milliseconds on a terminal. The next phase involved setting up an Analog-to-Digital Converter (ADC) to digitize signals from circuits with a photocell resistor and a TMP36 temperature sensor, displaying outputs on a terminal, including temperature in Fahrenheit and Celsius. The task also included programming to interface with a TSL1401-DB line scan camera and developing MatLab scripts for edge detection in test patterns. Additionally, the exercise included initializing a Serial Peripheral Interface (SPI) and conducting Digital-to-Analog Conversion (DAC) by writing to a MCP4901 DAC, with the results observed on an oscilloscope. The successful initialization of Timer32 modules, ADC integration, camera interfacing, and SPI and DAC operations confirmed the efficacy of the coding and circuit configurations, demonstrating the expected outputs and waveforms.

Design Methodology

The exercise involved configuring Switch 1 and Switch 2 of the MSP432 to trigger interrupts when pressed, which in turn activated the Timer32-1 and Timer32-2 modules, respectively. The first step was to develop timer initialization functions, which required inputs specifying the task, the timer period, and a prescaler value for division. The specified period was set in the LOAD register, while any existing interrupts were cleared through the INTCLR register. The CONTROL register was then configured to enable the timer, set its mode to periodic, activate interrupts, and define its operation as 32-bit. Interrupt service routines (ISRs) were crafted to acknowledge and clear the interrupt, carry out the specified task, and reset the timer period. Switch initialization involved enabling pull-up resistors and configuring the IE (Interrupt Enable) and IES (Interrupt Edge Select) registers to detect falling-edge triggers, ensuring no pending interrupts by clearing the IFG (Interrupt Flag) register. Timer32-1 was set to interrupt every half second, and Timer32-2 every millisecond, with the Timer32-1 ISR toggling LED1, and the Timer32-2 ISR incrementing a counter to track the duration between presses of Switch 2.

Additionally, code was developed for ADC initialization, setting the reference voltage to 2.5V and configuring the CTL0 register for sample-and-hold pulse mode sampling, a prescaler of 1, and the SMCLK as the clock source. The CTL1 register was set for 14-bit resolution. The ADC was configured to disable interrupt generation, with the analog input pin designated as 4.7, and enabled via the CTL0 register. An "ADC In" function was designed to initiate conversion and, upon

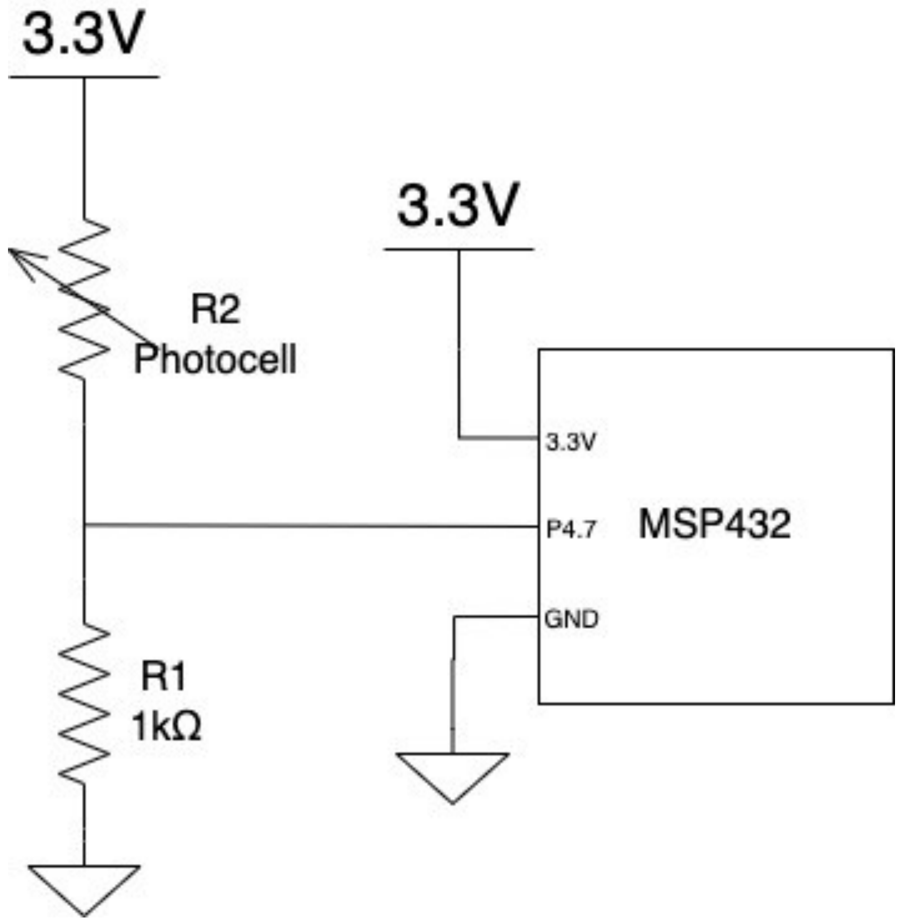

Figure 1: Photocell Voltage Divider Circuit Diagram

completion, read and return the value from the MEM register. This ADC functionality was integrated with the UART0 module to measure and display the voltage across a photocell resistor in a voltage divider circuit, as illustrated in Figure 1. The voltage divider circuit was constructed using a 3.3V power source connected in series with a photocell resistor and a 1kΩ fixed resistor. The point of connection between the photocell and the fixed resistor, where the voltage varies depending on the light intensity, was interfaced with the ADC to measure the voltage at this junction. Additionally, a separate circuit incorporating a TMP36 temperature sensor was designed, as depicted in Figure 2.

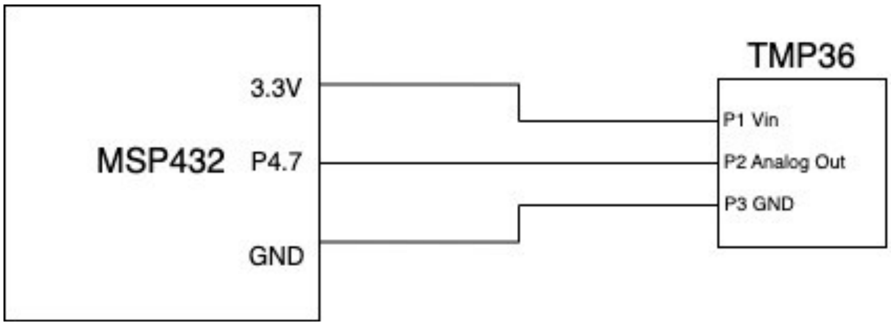

Figure 2: TMP36 Circuit Diagram

Using Equation 1, the ADC value was converted to a temperature reading in Celsius, which was then further converted into Fahrenheit.

$$Temperature = ((AnalogIn * (2.5/16384)) - 0.5) * 100 \quad (1)$$

Code was developed to interface the MSP432 with a TSL1401-DB Line Scan Camera. The connection between the MSP432 and the line scan camera was established using Serial Input (SI), Clock (CLK), and Analog Output (A0), as illustrated in Figure 3.

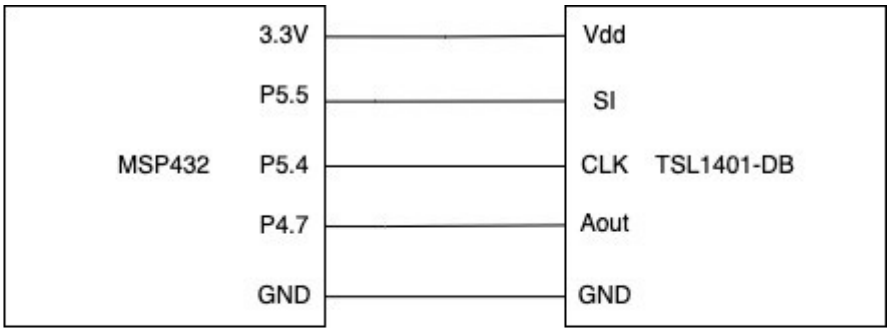

Figure 3: TSL1401-DB Circuit Diagram

Code was developed to initialize the SI by configuring pin 5.5 as an output and setting a 50Hz clock frequency. The CLK was initialized to use output pin 5.4, operating at a 48MHz frequency. Handler functions for both SI and CLK were implemented. The SI handler function is designed to first set the CLK low, and then sequence both SI and CLK from low to high and back to low, which activates the camera. The CLK handler function alternates the CLK signal and employs the ADC to capture the camera's output, storing it in an array.

Additionally, MATLAB code was crafted to read the camera's output via the COM port and visualize the data. The MATLAB script plots the raw data, applies a low pass filter to the data for smoother results, and converts the data into a binary format for analysis.

A separate circuit was also designed to facilitate the connection of an MCP4901 DAC to the

MSP432 board, as depicted in

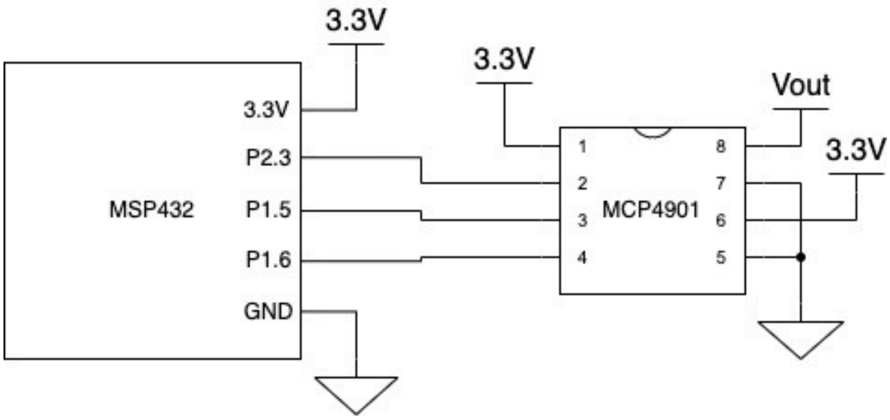

Figure 4: MCP4901 Circuit Diagram

Code was developed to interface with an MCP4901 DAC for the purpose of generating a 1Hz sine wave. This involved creating functions for SPI initialization and DAC writing. The SPI was initialized by configuring the EUSCI B0 CTLW0 register, and the BRW (Baud Rate Word) register was set to achieve a baud rate of 3MHz. Pins 1.5, 1.6, and 2.3 were designated for SI, CLK, and Chip Select (CS) functionalities, respectively. The DAC Write function was designed to accept a write value, which was initially shifted right by 4 bits to facilitate chip selection. Subsequently, the value was shifted back to the left by 4 bits to convey the actual data intended for the DAC. Utilizing these functions and applying Equation 2, the required values were calculated and sequentially written to the DAC to produce the desired 1Hz sine wave over time.

$$Vout = 120 * sin(2 * pi * t/100) + 128 \tag{2}$$

Since the DAC can only read values from 0 to 255, 128 had to be added to Vout to normalize the data.

Results and Analysis

The code for the timers was uploaded to the MSP432 board. When switch 1 was pressed, it activated the Timer32-1 module, which resulted in LED1 toggling on and off at a frequency of 2Hz. Pressing switch 2 triggered the Timer32-2 module, which then measured the time interval between the presses of switch 2 and displayed this information on the terminal through PuTTy over UART, as illustrated in Figure 5. Additionally, switch 2's activation also caused LED2 to cycle through a sequence of colors: red, green, blue, cyan, magenta, yellow, and white.

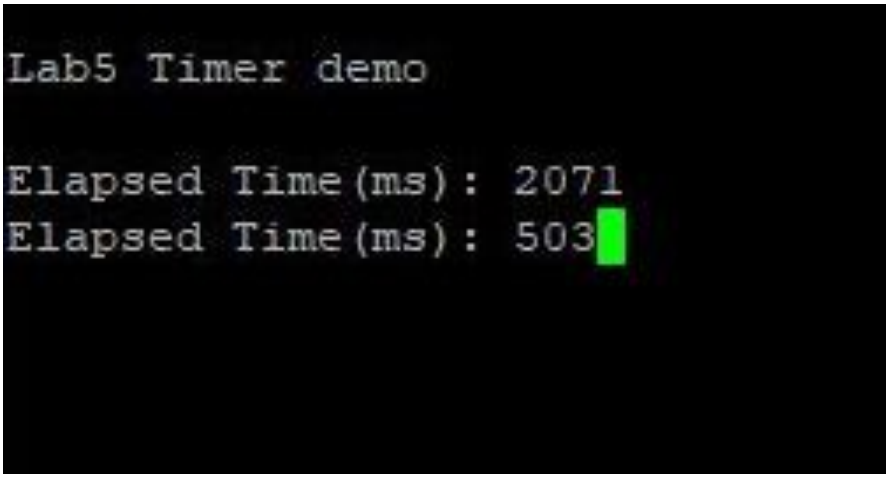

Figure 5: PuTTy Terminal of Time Between Switch2 Presses

Upon pressing switch 2 twice, the terminal displayed the elapsed time between presses in milliseconds. The minimum measurable time was approximately 150ms, and the maximum was theoretically 4294967295ms, dictated by the counter variable's capacity as an unsigned long, which represents its largest possible value.

The ADC code was uploaded to the board, and the photocell voltage divider circuit, as shown in Figure 1, was assembled. Observations revealed that increased illumination on the photocell resistor led to a decrease in resistance, thereby raising the voltage detected by the ADC. Conversely, covering the photocell resistor increased its resistance, which in turn reduced the ADC-perceived voltage. The TMP36 temperature sensor was connected as depicted in Figure 2, with the corresponding temperature readings in Fahrenheit and Celsius displayed on PuTTy via UART, as illustrated in Figure 6.

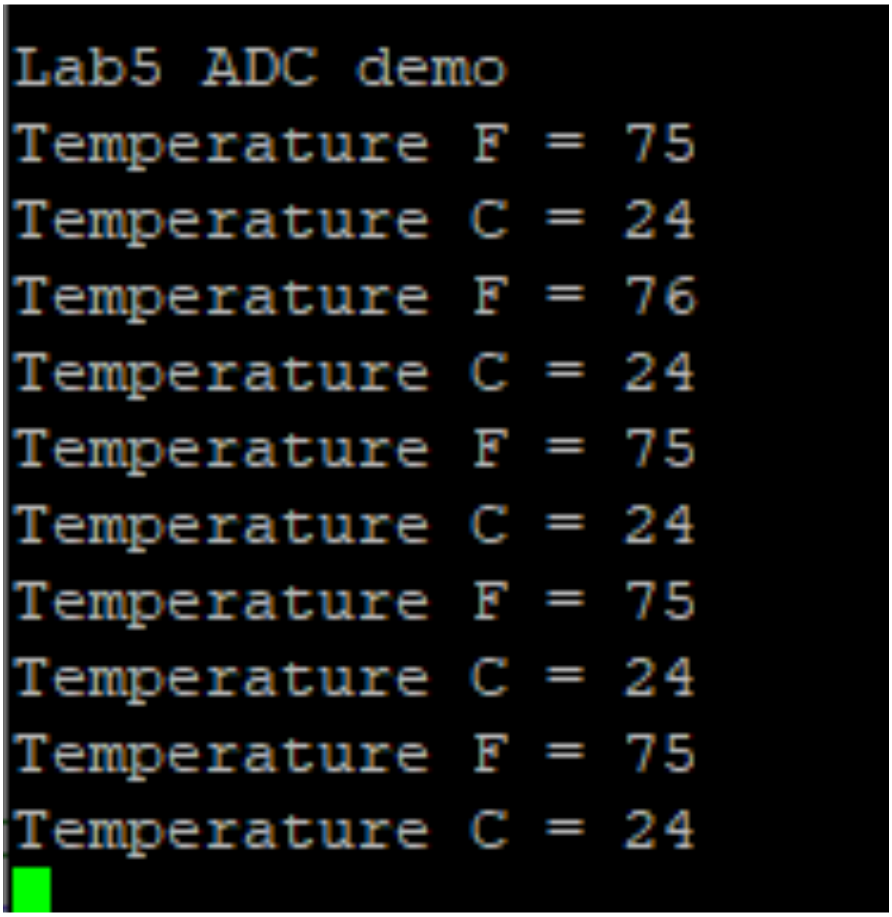

Figure 6: PuTTy Terminal of TMP36 Temperature Output

The terminal output successfully displayed the room temperature every second, indicating correct functionality.

The code for the camera was uploaded to the board, and the camera was connected as depicted in Figure 3. An oscilloscope was used to focus the camera on a specific test pattern, and the resulting oscilloscope capture is presented in Figure 7.

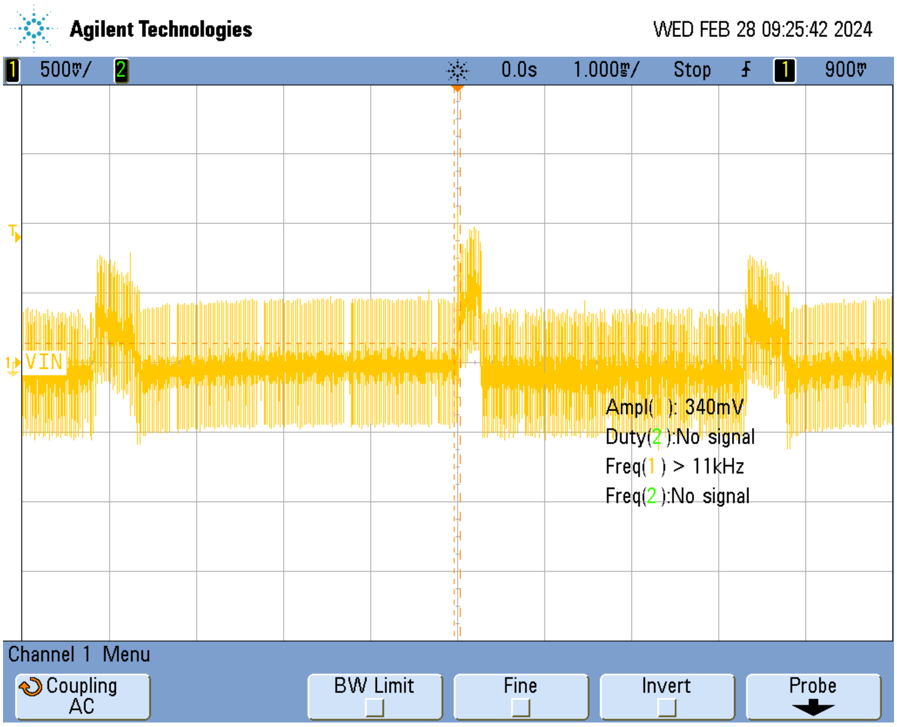

Figure 7: Oscilloscope Camera Focus

The oscilloscope display revealed the outline of the test pattern. Subsequently, the MATLAB code was executed to generate the waveforms, which are illustrated in Figure 8.

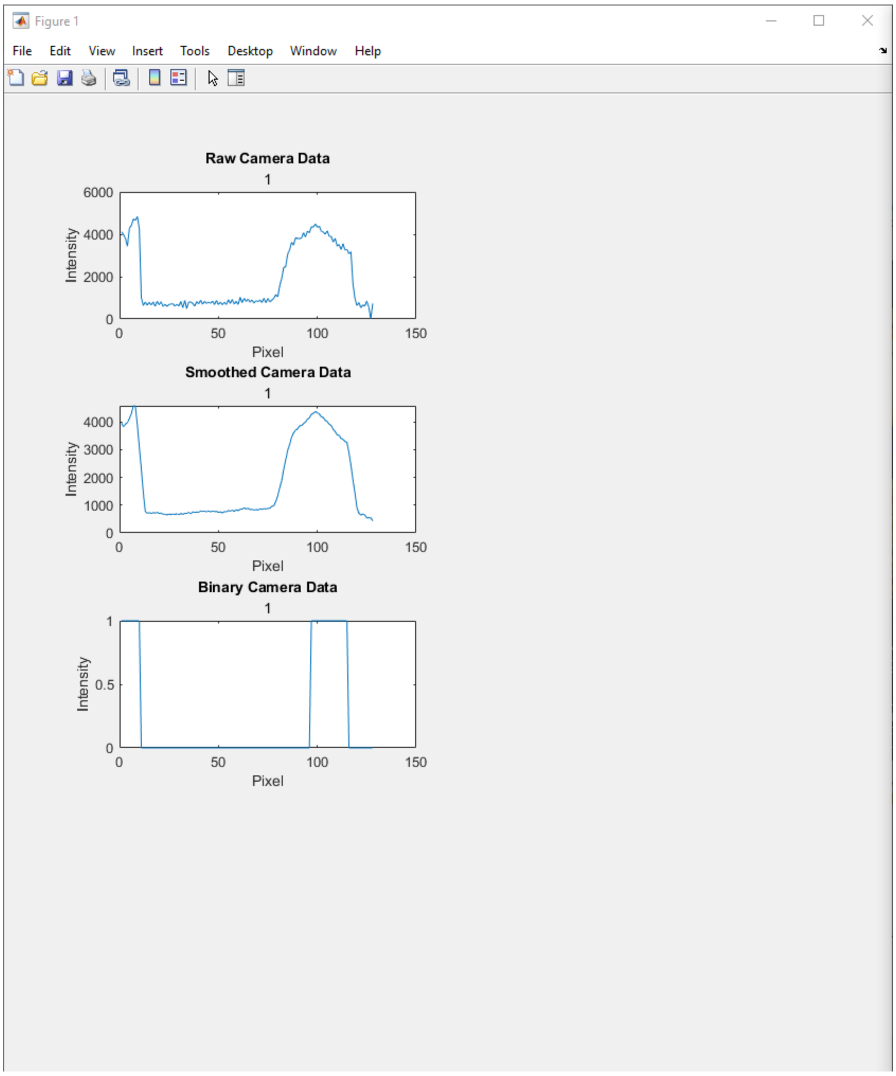

Figure 8: MatLab Camera Waveforms

The waveform processed through the low pass filter appears significantly smoother compared to the raw data. In the binary graph, the edges of the test pattern are distinctly visible, with the camera producing higher values for lighter areas and lower values for darker areas. This correlation is also reflected in the oscilloscope capture, which, when focused through the camera, aligns with the binary graph produced by MATLAB.

The code for SPI DAC was uploaded to the board, and the MCP4901 DAC was set up as depicted in Figure 4. The DAC's output was then monitored using an oscilloscope, which plotted the resulting waveform, as shown in Figure 9.

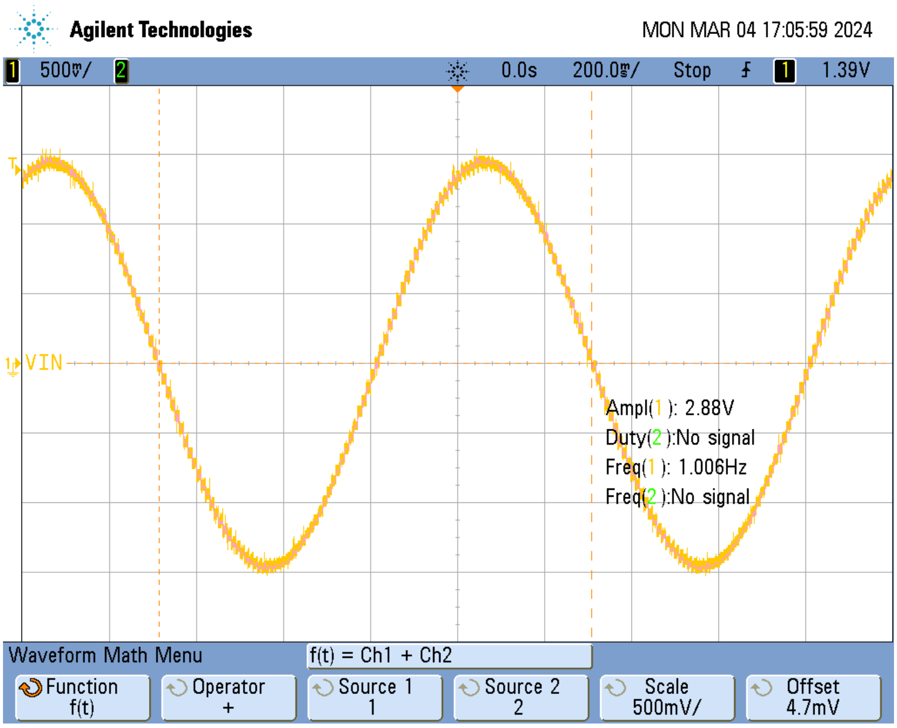

Figure 9: DAC Oscilloscope Waveform

The oscilloscope waveform displays a frequency of 1Hz and an amplitude of 2.88V, consistent with the specifications outlined in Equation 2.

Questions

1. Clearing IRQ flags in interrupt service routines is a common practice to prevent the same interrupt from being serviced repeatedly. When an interrupt occurs, the corresponding flag in the interrupt flag register is set. The ISR must clear this flag to indicate that the interrupt has been handled, ensuring the system is ready to respond to subsequent interrupts. Without clearing the flag, the system could interpret the still-set flag as a new interrupt, leading to repeated handling of the same event and potentially causing a software lockup or other unintended behavior.

2. When an interrupt occurs, the specific pin that caused it can be identified by checking the interrupt flag register (IFG) associated with the pins or the port that generated the interrupt. Each pin that can generate an interrupt typically has a corresponding bit in an interrupt flag register. When an interrupt is triggered, the corresponding bit for that pin is set in the register. By examining the IFG register and identifying which bit(s) is set, you can determine which pin(s) caused the interrupt. Furthermore, the interrupt service routine (ISR) that executes in response to the interrupt can include code to check this register and act accordingly based on which pin triggered the interrupt.

3. To initiate data transfer from a camera, the startup sequence typically involves first setting the SI high, followed by setting the CLK high. Then, SI is set low, and a clock pulse is provided (CLK goes high then low). This sequence primes the camera to start capturing and shifting out pixel data on subsequent clock pulses. The sequence ensures the camera's internal register is set

to read the first pixel data, readying it for the sequential readout of pixel information synchronized with the clock pulses.

Conclusion

This exercise demonstrated the MSP432's capabilities with Timer, Interrupt, ADC, and SPI modules by developing initialization codes and interfacing with various sensors. The Timer32 modules were set to trigger interrupts at specific intervals, with switches activating these timers, leading to LED toggling and time counting between switch activations. The ADC was used to process signals from a photocell resistor and a TMP36 temperature sensor, with outputs displayed via UART. Additionally, interfacing with a TSL1401-DB line scan camera and a MCP4901 DAC was achieved, facilitating edge detection through MATLAB processing and sine wave plotting on an oscilloscope, respectively. All implemented codes and scripts successfully produced the anticipated outputs and waveforms.

Exercise 5: MSP432 Timers, Interrupts, and Analog-to-Digital Converter

<table>
<tr><th colspan="2">Report</th><th>Point Value</th><th>Points Earned</th><th>Comments</th></tr>
<tr><td colspan="2">Abstract</td><td>10</td><td></td><td></td></tr>
<tr><td>Design Methodology</td><td>Discussion</td><td>10</td><td></td><td></td></tr>
<tr><td rowspan="4">Results and Analysis</td><td>Part 1 Screenshot</td><td>5</td><td></td><td rowspan="4"></td></tr>
<tr><td>Part 2 Screenshot</td><td>5</td><td></td></tr>
<tr><td>Camera Plot</td><td>5</td><td></td></tr>
<tr><td>Timing Explanations</td><td>5</td><td></td></tr>
<tr><td colspan="2">Conclusion</td><td>5</td><td></td><td></td></tr>
<tr><td rowspan="3">Questions</td><td>Question a</td><td>5</td><td></td><td rowspan="3"></td></tr>
<tr><td>Question b</td><td>5</td><td></td></tr>
<tr><td>Question c</td><td>5</td><td></td></tr>
<tr><td colspan="2">Total for prelab, demo, and report</td><td>100</td><td></td><td></td></tr>
</table>

Exercise 5: MSP432 Timers, Interrupts, and Analog-to-Digital Converter

Student's Name: Hunter Culverhouse Gloria MBaka Section: 01

PreLab		Point Value	Points Earned	Comments
PreLab	Prelab C Code	10	10	AJT 2/14

Demo		Point Value	Points Earned	Date
Demo	Functionality with SW1	5	5	AJT 2/20
	Functionality with SW2	5	5	
	ADC Functionality with CdS	5	5	AJT 2/21
	ADC Functionality with Temp Sensor	5	5	CV 2/28
	Linescan Camera MATLAB and Oscilloscope	5	5	AJT 2/28
	DAC SPI Interface Oscilloscope	5	5	AJT 3/4

To receive any grading credit students must earn points for both the demonstration and the report.