ESET 369 LAB 7 REPORT

Student name	Kyle Rex, Shane Moe, and Denton Zuercher
Group number	41 & 31
Lab Section	501 & 508
Lab session date	April 4, 2025
Lab instructor	Xin Zhuang

INTRODUCTION

The objective of this lab was to implement UART-based communication and servo motor control using the MSP430FR5944 microcontroller. The lab was divided into two systems, each with distinct functionalities. System A focused on developing a UART-driven command interface to display predefined text and read the internal temperature of the microcontroller. The temperature data was converted from raw ADC values into Fahrenheit using calibration constants and linear interpolation. System B involved configuring a servo motor to respond to serial input commands, rotating to preset positions using PWM signals generated by the microcontroller. The lab required integrating code sequences from previous labs and adapting them to create a functional, responsive system that met the specific requirements for both text output and motor control. Emphasis was placed on correct hardware connections, UART communication setup, and precise PWM signal generation.

SYSTEM A

The purpose of System A is to correctly configure the MSP430FR5944 Launchpad board to display a list on the serial terminal when a certain key is pressed. The list includes: 1. List, 2. Name, 3. Temp, 4. Center, 5. Left, and 6. Right. Additionally, the first 3 prompts on the list (List, Name, and Temp) must be created using a C/C++ code sequence and displayed on the serial terminal when their respective key is pressed. When the '1' key is pressed the list should be displayed. If the '2' key is pressed the group members names should be displayed in this form: "ESET 369: Denton, Shane, Kyle". If the '3' key is pressed the internal temperature of the microcontroller should be read using the ADC value and displayed in terms of Fahrenheit.

The first step in addressing this problem is simply connecting to the MSP430FR5994 microcontroller itself. There are no physical board connections between the BH board and the MSP430FR5994 microcontroller required for system A.

Using the code sequences created for previous labs that were based on code sequences found in: *Learning Embedded Systems with MSP430 FRAM microcontrollers* by B. Hur, as references the code sequence used for this system was created. One part of the system was coded and tested at a time to ensure that there were no issues with the code or the hardware. The code implements a UART-driven command system for the MSP430 microcontroller, specifically tailored to meet the requirements outlined previously. It listens for character input through UART and uses an interrupt service routine (ISR) to parse each command. When the user sends the character '1' over the serial interface, the ISR responds by printing a menu of six options to the terminal using the uart_send_string function. This fulfills the requirement to display the list of available commands when '1' is pressed. The code sends each line individually, ensuring correct formatting and serial timing, thus matching the specified output format in the lab instructions. For the '2' key input, the code prints the string "Eset 369: Denton, Shane, Kyle\r\n" to the terminal. This matches the requirement to display the course code and lab group member names when prompted. The use of \r\n ensures proper line termination for terminal compatibility. The most complex logic is found in the response to the '3' key, which initiates a temperature reading using the microcontroller's internal temperature sensor. The code first retrieves calibration constants for 30°C and 85°C from the TLV memory section, which are unique to each chip. These values are used in a linear interpolation formula to convert the raw ADC result (adc_raw) into degrees Celsius. The Celsius value is then converted to Fahrenheit using the standard formula: Tempf = (TempDegC * 1.8f) + 32.0f. The temperature is split into whole and fractional parts for serial output. These are transmitted over UART using custom functions (uart_send_number and uart_send_int), which manually convert integers to ASCII characters, ensuring the correct display format on the terminal. This implementation meets the lab requirement to read and output a temperature value in degrees Fahrenheit upon pressing '3'. The temperature output in the serial terminal can be seen in Table 1 and Figure 1.

Table 1: System A Measurement Values

	Temperature (Degrees Fahrenheit)
Case 1	77.86
Case 2	78.31

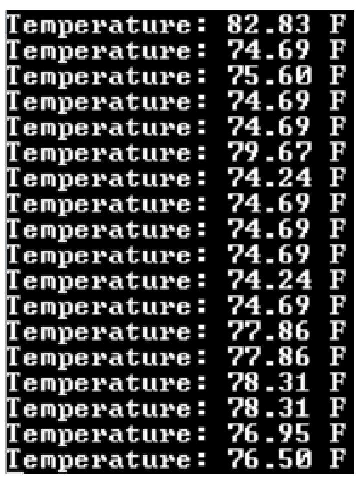

Figure 1: Temperature Values

SYSTEM B

The purpose of System B is to correctly configure the MSP430FR5944 Launchpad board to move the servo motor arm in certain orientations when a certain key is pressed using a C/C++ code sequence. If the '4' key is pressed the servo motor arm should move to its neutral position (0 degrees). If the '5' key is pressed the servo motor arm should move to its left position (-45 degrees).

If the ‘6’ key is pressed the servo motor arm should move to its right position (+45 degrees). An example of the servo motor that was used in the lab can be seen in Figure 2.

Figure 2: Servo Motor

The first step in addressing this problem is making the physical board connections between the BH board, the DC power supply, and the MSP430FR5994 microcontroller. This can be done by connecting three specific pins together using male-to-female/male-to-male jumper wires, checking that all the DIP switches on the BH board are in the correct configuration, and ensuring the microcontroller itself is properly seated on the BH board. Pin H4.3 of the BH board must be connected to the brown servo motor connection port. Pin H4.5 of the BH board must be connected to the red servo motor connection port. Pin P1.2 of the microcontroller must be connected to the orange servo motor connection port. Finally, the DC power supply must be connected to the BH board DC barrel jack itself and set to provide 9V of DC power.

Using the code sequences created for previous labs that were based on code sequences found in: *Learning Embedded Systems with MSP430 FRAM microcontrollers* by B. Hur, as references the code sequence used for this system was created. One part of the system was coded and tested at a time to ensure that there were no issues with the code or the hardware. The servo motor control code is designed to interpret user input from the serial terminal and adjust the orientation of the servo arm accordingly through PWM (Pulse Width Modulation) signals. The MSP430FR5944 LaunchPad is configured to output these PWM signals on pin P1.2, which is connected to the control input of the standard hobby servo motor. The servo motor responds to the width of the PWM pulse to position its arm at specific angles. This portion of the code handles key presses '4', '5', and '6' within the UART receive interrupt. When the '4' key is detected, the instruction TA1CCR1 = 1500; is executed. This sets the PWM duty cycle to 1.5 ms, which corresponds to the servo’s neutral or center position (0 degrees). Similarly, pressing '5' sets TA1CCR1 = 2000, which increases the pulse width to 2.0 ms, commanding the servo to rotate to its leftmost position, -45 degrees. Conversely, pressing '6' sets TA1CCR1 = 1000, reducing the pulse width to 1.0 ms, causing the servo to move to its rightmost position, +45 degrees. The timer module used is Timer1_A, configured in up mode with TA1CCR0 = 20000, corresponding to a 20 ms period or 50 Hz signal is standard for hobby servos. The PWM signal is generated on the TA1.1 channel, and its duty cycle is adjusted by changing the value in TA1CCR1. The direction of the servo arm movement is controlled entirely by the width of the pulse set through this compare register. This implementation directly meets the requirements of System B by enabling the servo to move to the specified positions based on serial input. The ISR captures the user input and updates the PWM signal in real-time, resulting in immediate changes in servo orientation. Thus, the code successfully configures and utilizes the MSP430FR5944 to control a servo motor via UART commands.

CONCLUSION

The lab successfully demonstrated the capabilities of the MSP430FR5944 microcontroller in handling UART-based user interactions and PWM-driven servo motor control. Through System A, the UART interface was effectively used to display predefined text and measure internal temperature, showcasing both data handling and serial communication. System B further expanded the microcontroller's functionality by implementing accurate PWM control of a servo motor in response to user input. Both systems operated reliably, fulfilling the lab objectives and reinforcing foundational concepts in embedded systems design such as interrupt-driven programming, ADC utilization, and timer configuration. The results validate the practical applications of embedded control in real-time systems.

REFERENCES

B. Hur, *Learning Embedded Systems with MSP430 FRAM microcontrollers*, 2nd ed., 2023.

K. Rex, *ESET 369 Lab Report 6*, 2025.

K. Rex, *ESET 369 Lab Report 5*, 2025.

K. Rex, *ESET 369 Lab Report 4*, 2025.

K. Rex, *ESET 369 Lab Report 3*, 2025.

K. Rex, *ESET 369 Lab Report 2*, 2025.

K. Rex, *ESET 369 Lab Report 1*, 2025.