ESET 369 LAB 5 REPORT

Student name	Kyle Rex and Nikesh Nair
Group number	38
Lab Section	501
Lab session date	March 03, 2025
Lab instructor	Xin Zhuang

INTRODUCTION

The objective of this lab is to configure and test the MSP430FR5944 Launchpad board to interface with external components such as a potentiometer, an LCD display, and a buzzer. The lab is divided into two systems. System A focuses on configuring pin P4.1 as an output voltage controlled by a potentiometer and measuring the resulting voltage using a digital multimeter. System B extends this functionality by displaying the measured voltage on an LCD screen and producing a corresponding sound through a buzzer. The implementation of these systems involves programming the microcontroller using C/C++, utilizing the ADC module for voltage readings, and employing pulse-width modulation (PWM) for sound generation. The purpose of this lab is to gain hands-on experience in microcontroller interfacing, ADC configuration, and peripheral control while verifying the accuracy of the implemented system through measurements and testing.

SYSTEM A

The purpose of System A is to correctly configure pin P4.1 of the MSP430FR5944 Launchpad board to output a voltage that is controlled with the potentiometer on the BH board using a C/C++ code sequence while measuring that voltage using a digital multimeter. By adjusting the potentiometer, various volage levels can be generated. A measured voltage may not exactly match the desired voltage. The ADC module on MCU must be used to perform this function using a 12-bit unsigned resolution. The first step in addressing this problem is making the physical connections between the multimeter, BH board, and MSP430FR5994 microcontroller. This can be done by connecting three specific pins, P4.1 of the MCU, the BH potentiometer pin, and GND of the MCU, to the multimeter's positive voltage probe, positive voltage probe, and negative voltage probe respectively, using male-to-female jumper wires and male-to-male jumper wires. This setup is depicted in Figure 1.

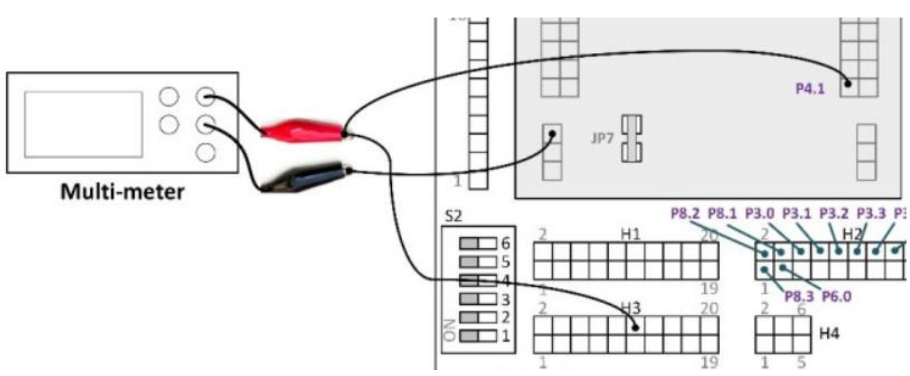

Figure 1: Physical Connection Configuration

Using the code sequences that focus on operating the ADC in single channel selection labeled 10.1 (Polling Method) and 10.2 (Interrupt Method), found in: *Learning Embedded Systems with MSP430 FRAM microcontrollers* by B. Hur, as references the code sequence used for this system was created. The program is designed to perform analog-to-digital conversion (ADC) using the ADC12_B module. It begins by stopping the watchdog timer with WDTCTL = WDTPW | WDTHOLD; to prevent an unintended system reset. The PM5CTL0 &= ~LOCKLPM5; statement clears the LOCKLPM5 bit, enabling the use of GPIO pins. The direction of pin P1.0 is set as output using P1DIR |= 0x01;. The alternate function for pin P4.1 is enabled by setting P4SEL1 |= BIT1; and P4SEL0 |= BIT1;, configuring it for ADC input (A9). The ADC12_B module is configured by setting ADC12CTL0 = ADC12SHT0_6 | ADC12ON;, which enables the ADC and sets the sample-and-hold time. The control register ADC12CTL1 is set to ADC12SHP mode, allowing the ADC to sample using an internal timer. A 12-bit resolution is specified by ADC12CTL2 = ADC12RES_2;. The input channel is selected by ADC12MCTL0 = ADC12INCH_9;, designating A9 as the input channel. Interrupts for ADC12_B are enabled using ADC12IER0 |= ADC12IE0;, followed by __enable_interrupt(); to enable general interrupts. Within the infinite while(1) loop, the ADC conversion is initiated by setting ADC12CTL0 |= ADC12ENC | ADC12SC;. A delay is introduced with __delay_cycles(25000); before the loop repeats. The ADC interrupt service routine ADC12_ISR(void) handles ADC conversion completion. When an interrupt occurs, it checks if the ADC12IFGR0 flag is set for BIT0, indicating that the conversion result is available. If so, the ADC result is read from ADC12MEM0 into the variable adc_raw. This ensures that the program processes ADC conversions as they complete.

By suspending (pausing) the program the raw NADC value, the digital value resulting from the ADC conversion, stored in the adc_raw global variable, can be checked at target voltages of 0.5 V and 1.5 V. The resulting values can be seen in Table 1. The actual measurements of the multimeter and NADC values themselves can be seen in Figure 2, Figure 3, Figure 4, and Figure 5.

Table 1: System A Measurement Values

Target Voltage	Measured Voltage (Multi-meter)	NADC (Launchpad)
0.5 V	0.52 V	1113
1.5 V	1.502 V	1865

Figure 2: Measured Voltage (Multi-meter) Example Value for ~0.5V

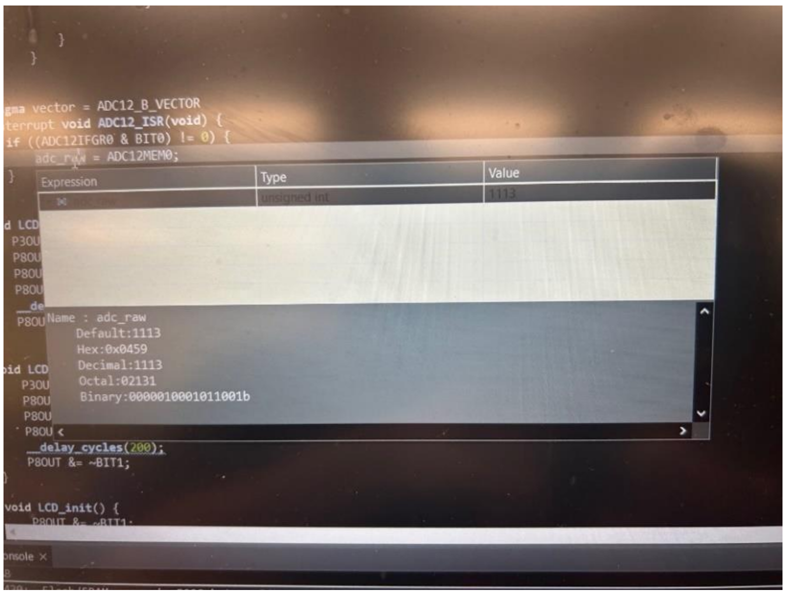

Figure 3: NADC (Launchpad) Example Value for ~0.5V

Figure 4: Measured Voltage (Multi-meter) Example Value for ~1.5V

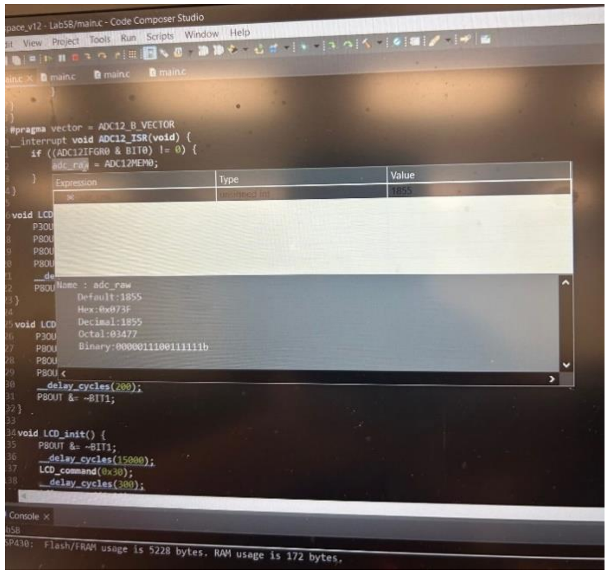

Figure 5: NADC (Launchpad) Example Value for ~1.5V

SYSTEM B

The purpose of System B is to correctly configure the MSP430FR5944 Launchpad board to display the name of a lab group member and the current voltage of pin P4.1, controlled by the potentiometer, on the BH board 16x2 LCD screen while playing a specific note on the buzzer based on the voltage at P4.1 using a C/C++ code sequence. By adjusting the potentiometer, various volage levels can be generated and should be continuously displayed on the LCD screen. The ADC module on MCU must be used to perform this function using a 12-bit unsigned resolution. The first line of the LCD will display the name of the group member, and the second line will display the converted voltage, to two decimal places, that can be read through pin P4.1. The voltage range from 0 V to 3.3 V on the LCD display should be matched with the actual full range of the rotation of the potentiometer knob. Based on the voltage a different note should be played on the buzzer. The values and corresponding notes and voltage values can be seen in Table 2.

Table 2: System A Measurement Values

Voltage < 1V	NADC < 1241	Stop playing a note
1V <= Voltage < 2V	1241 <= NADC < 2482	C note (CCR0 value of 956)
2V <= Voltage < 3V	2482 <= NADC < 3723	D note (CCR0 value of 851)
Voltage >= 3V	NADC >= 3723	E note (CCR0 value of 758)

The first step in addressing this problem is making the physical board connections between the BH board and the MSP430FR5994 microcontroller. This can be done by connecting thirteen specific pins together using male-to-female 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. For the 16x2 LCD screen pin P8.3 connects to RS, P8.2 connects to R/W, P8.1 connects to E, P3.0 connects to DB0, P3.1 connects to DB1, P3.2 connects to DB2, P3.3 connects to DB3, P3.4 connects to DB4, P3.5 connects to DB5, P3.6 connects to DB6, and P3.7 connects to DB7. Pin P4.1 connects to a potentiometer on the BH board. Pin P6.0 controls the buzzer and must be provided with a periodic signal from the microcontroller to function. This signal is modified by changing the TA1CCR0 value. Different TA1CCR0 values result in different frequencies and each frequency corresponds to a unique musical note that the buzzer will play. The notes and their corresponding TA1CCR0 values can be seen in Table 2, previously displayed.

Using the code sequence created for System A and the code sequence labeled 9.1 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 developed correctly configures the MSP430FR5944 Launchpad board to read the voltage from a potentiometer connected to pin P4.1, display the voltage and a lab group member's name on a 16x2 LCD screen, and control a buzzer to play specific notes based on the voltage level. The ADC module is set up to perform a 12-bit unsigned conversion, ensuring the full 0V to 3.3V range of the potentiometer is mapped accurately. At the start, the watchdog timer is disabled, and the necessary configurations for ADC, timer, and GPIO are performed. The ADC is set up to sample from channel 9 (corresponding to P4.1) and trigger an interrupt when a conversion is complete. Timer_A is configured to generate a periodic signal, which will be used to control the buzzer. The LCD is initialized and prepared for displaying text. Inside the main loop, an ADC conversion is started, and the raw ADC value is converted into a voltage. The LCD is then reinitialized and cleared before displaying the lab group member's name on the first line and the converted voltage to two decimal places on the second line. The buzzer output is controlled by modifying the Timer_A CCR0 register, which adjusts the frequency based on the measured voltage. If the voltage is below 1V (ADC value < 1241), the buzzer remains off. If the voltage falls within the 1V-2V range (1241 ≤ ADC value < 2482), the Timer_A CCR0 register is set to 956, producing a C note. For a voltage between 2V and 3V (2482 ≤ ADC value < 3723), the CCR0 value is set to 851, playing a D note. If the voltage is at or above 3V (ADC value ≥ 3723), the CCR0 value is set to 758, producing an E note. The LED on P6.0 is toggled based on the Timer_A flag to indicate the buzzer's activity. An ADC interrupt service routine updates the global variable storing the ADC result whenever a conversion completes, ensuring the latest voltage reading is used. The LCD functions handle command transmission, data writing, and initialization to ensure proper communication with the display. The buzzer's frequency is modified dynamically based on the real-time voltage readings, allowing the system to produce the correct musical note as specified in the requirements.

CONCLUSION

The lab successfully demonstrated the ability to interface the MSP430FR5944 Launchpad board with external components, achieving the intended objectives. System A allowed for accurate voltage measurement through the ADC module, while System B successfully displayed real-time voltage readings on the LCD screen and played corresponding notes through the buzzer. The results confirmed that the ADC module correctly converted the analog voltage into a digital value, with minor discrepancies observed due to resolution limitations. The implementation of the LCD display and buzzer interaction provided insight into real-time embedded system programming. Overall, the lab reinforced key concepts in microcontroller-based system design, including ADC operation, peripheral communication, and signal processing, contributing to a deeper understanding of embedded systems.

REFERENCES

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

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.