**Prompt:**

For System A, students must write a C/C++ program for the following functions.

For safety, please, remember and make sure to unplug your Launchpad from the USB port before making any hardware changes.

Connect jumper wires as explained below in connections section.

Write a program in C/C++ to implement the following function.

Use the ADC module on the MSP430FR5994 MCU, and make sure to use it as a 12-bit resolution (unsigned) as we studied in class.

(Note) You can suspend (pause) the program to check the raw NADC value.

Fill out the following table when measured for 0.5-V and 1.5-V cases. Make sure to include this table in your lab report

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

(Note) By tweaking the potentiometer, you can generate various voltage levels. A measured voltage may not exactly match the desired voltage. You can try to set the value within ±5%.

You may not need to utilize all the connections described below to achieve this.

**Example of a program from a previous lab focuses on ADC test example (Polling based) (That might help):**

#include <msp430.h>

unsigned int adc\_raw;

int main(void) {

WDTCTL = WDTPW | WDTHOLD; // hold watchdog timer

PM5CTL0 &= ~LOCKLPM5; // clear LOCKLPM5 bit

P1DIR |= 0x01; // output direction (P1.0)

P4SEL1 |= BIT1; // alternate function (A9)

P4SEL0 |= BIT1; // alternate function (A9)

ADC12CTL0 = ADC12SHT0\_6 | ADC12ON; // ADC CTL0 set up

ADC12CTL1 = ADC12SHP; // ADC CTL1 set up

ADC12CTL2 = ADC12RES\_2; // 12-bit conversion

ADC12MCTL0 = ADC12INCH\_9; // ADC channel selection

while(1) {

ADC12CTL0 |= ADC12ENC | ADC12SC; // ADC, Start conversion

while ((ADC12IFGR0 & BIT0)==0); // flag check

adc\_raw = ADC12MEM0; // read ADC

P1OUT ^= 0x01; // toggle (P1.0)

\_delay\_cycles(25000); // delay

}

return 0;

}

**Example of a program from a previous lab focuses on ADC test example (Interrupt) (That might help):**

#include <msp430.h>

unsigned int adc\_raw;

int main(void) {

WDTCTL = WDTPW | WDTHOLD; // hold watchdog timer

PM5CTL0 &= ~LOCKLPM5; // clear LOCKLPM5 bit

P1DIR |= 0x01; // output direction (P1.0)

P4SELI |= BIT1; // alternate function (A9)

P4SEL0 |= BIT1; // alternate function (A9)

ADC12CTL0 = ADC12SHT0\_6 | ADC12ON; // ADC CTL0 setup

ADC12CTL1 = ADC12SHP; // ADC CTL1 setup

ADC12CTL2 = ADC12RES\_2; // 12-bit conversion

ADC12MCTL0 = ADC12INCH\_9; // ADC channel selection

ADC12IER0 |= ADC12IE0; // enable interrupt, ADC

\_enable\_interrupt(); // enable general interrupt

while(1) {

ADC12CTL0 |= ADC12ENC | ADC12SC; // ADC, Start conversion

P1OUT ^= 0x01; // toggle (P1.0)

\_delay\_cycles(25000); // delay

}

return 0;

}

#pragma vector = ADC12\_B\_VECTOR

\_\_interrupt void ADC12\_ISR(void) {

if ((ADC12IFGR0 & BIT0) != 0) { // flag check

adc\_raw = ADC12MEM0; // read ADC

}

}

**Example of empty program:**

#include <msp430.h>

/\*\*

\* main.c

\*/

int main(void)

{

WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

return 0;

}

**Connections:**

MSP430FR5994 Launchpad

16x2 LCD screen

P8.3 connects to RS

P8.2 connects to R/W

P8.1 connects to E

P3.0 connects to DB1

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

P3.7 connects to DB7

P6.0 connects to the buzzer.

P4.1 connects to the positive input of the multimeter and to a potentiometer.

GND of the MSP430FR5994 Launchpad connects to the negative input of the multimeter.