EE4930 Advanced Embedded Systems

Section 011, Winter 2019/20

Draven Schilling

12-15-19

Lab 2 – MSP432

# Objectives

The objective for this lab was to gain further insight into working with some more of the onboard peripherals the MSP432 has to offer, specifically the Timer and ADC peripherals. Another goal was to familiarize myself with the interrupt structure the MSP432 uses and to gain a bit more experience using the MSOE library and HAL functions.

# Description

In this lab I Initialized two TimerA peripherals, one to be used as a 1 second interrupt and another to be used in PWM mode to produce a square wave output**. I used a period of 1 second because** it’s noticeably intentionally slow but also precise to show it was intentional. I also initialized an ADC in 12 bit mode connected to a potentiometer which would read the pot output and convert it to a percentage 0-99 based on how far the dial was turned to the right. The ADC was setup in single conversion mode where samples were triggered by the 1 second timer. When the ADC was done converting, it would produce its own interrupt which would then read the value and store it for other uses. Simultaneously, whenever the percentage changed, the inverse (100-value), was set to the duty cycle of the Timer running in PWM mode.

Additionally, whenever the pushbutton 1.1 was pressed, the LCD would update the display with the last sampled value of the ADC along with the calculated duty cycle (non-inverted).

# Conclusions

Again, everything in this lab was relatively straightforward. The posted example code provided a good starting point for which I could understand in the documentation why certain bits were being set. For the most part, all of my code worked with little debugging except for the PWM timer which I ran into some problems with some setup bits I missed first time around. I think I am getting familiar enough with the documentation now that I can generally find and interpret what I am looking for.

# Attachments

The first 3 pictures are the PWM output on pin P2.5 at different potentiometer positions. The approximate expected duty cycle is represented in the figure caption.

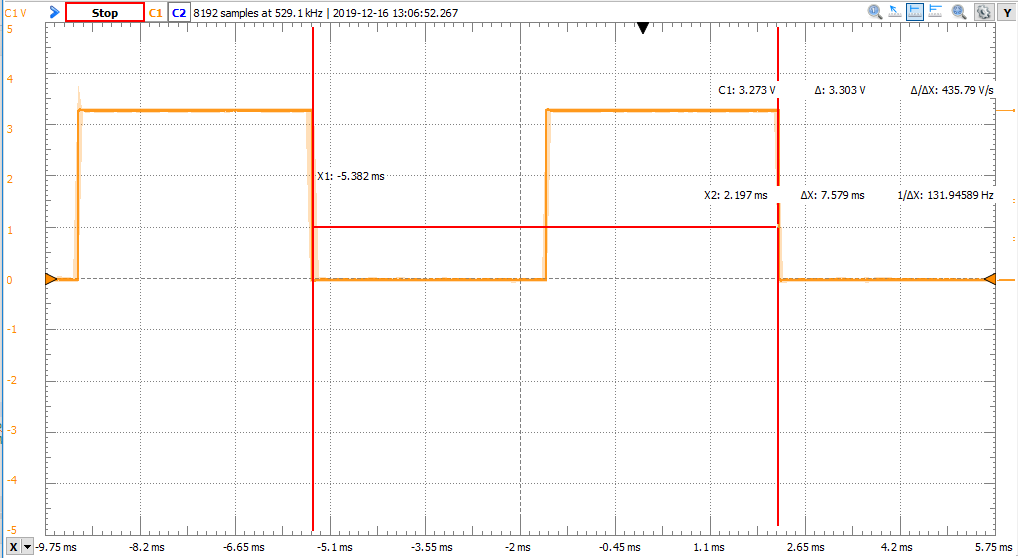


Figure ~50% Duty Cycle

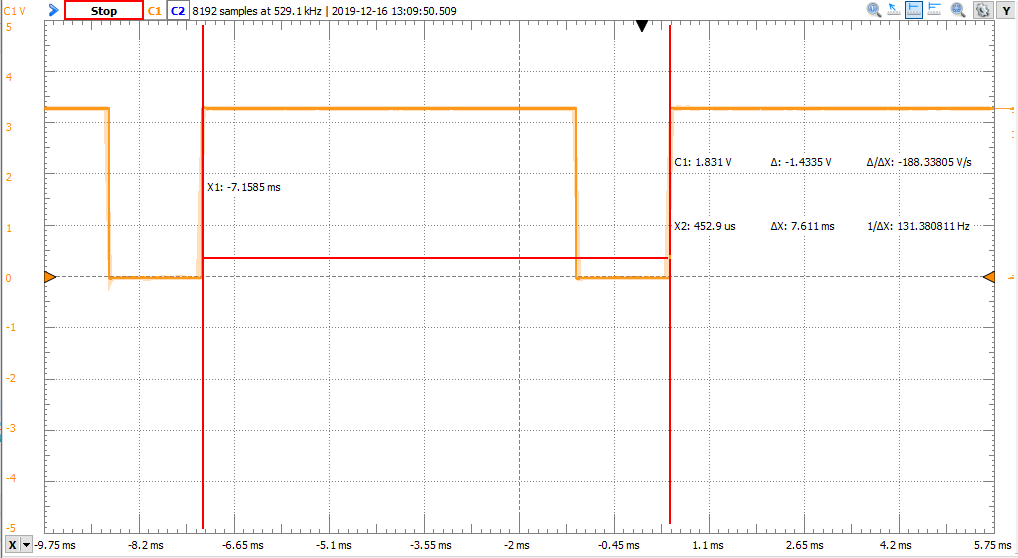


Figure ~80% Duty Cycle

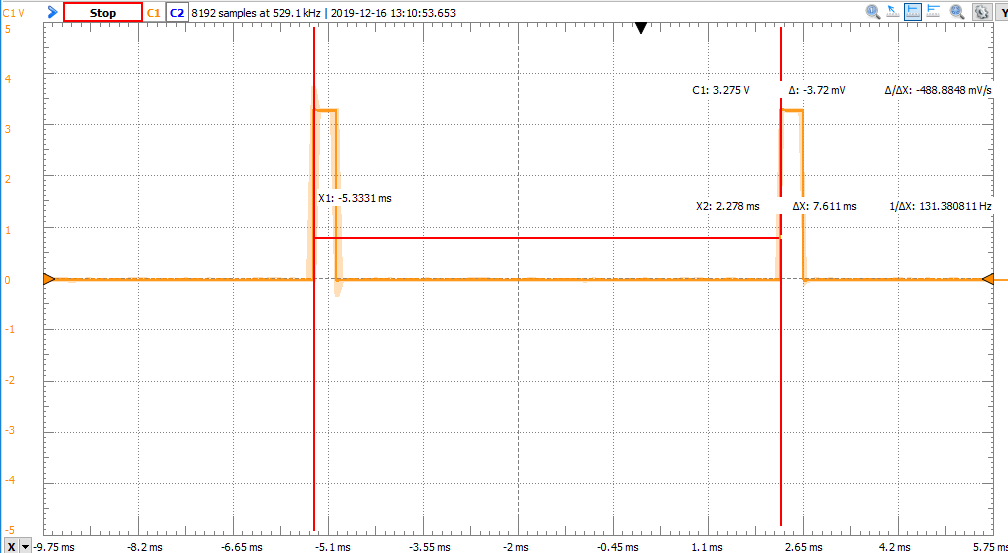
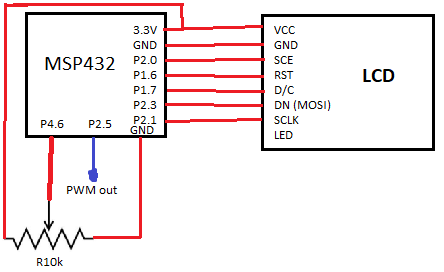


Figure ~5% Duty Cycle

**Wiring Schematic:**



**Source Code**

**#include** "msp.h"

**#include** <stdint.h>

**#include** <string.h>

**#include** <stdio.h>

**#include** "msoe\_lib\_clk.h"

**#include** "msoe\_lib\_lcd.h"

**#include** "msoe\_lib\_delay.h"

/\*\*

\* main.c

\*/

**#define** ONE\_HZ\_PSC 46875 // for timers with 1/64 scale

**#define** PWM\_PER 2865

**#define** ADC\_RANGE 4096 // for 12 bit converstions

//Global Vars

**int** adc\_val = 0;

uint8\_t duty\_cycle = 0;

**void** **main**(**void**)

{

WDT\_A->CTL = WDT\_A\_CTL\_PW | WDT\_A\_CTL\_HOLD; // stop watchdog timer

//Clock\_Init\_48MHz(); // run system at 48 MHz (default is 3 MHz)

// setup lcd

LCD\_Config();

LCD\_clear();

LCD\_home();

LCD\_contrast(10);

//Setup Timers

initTimA1();

initTimA0();

initADC();

initPushbutton();

**char** dc\_display[8];

uint8\_t pushbutton;

**while**(1){

pushbutton = (P1->IN & 0b10) >> 1;

**if**(!pushbutton){

LCD\_goto\_xy(0,0);

LCD\_print\_str("ADC:");

LCD\_print\_udec5(adc\_val);

LCD\_goto\_xy(0,2);

**if**(duty\_cycle > 9)

**snprintf**(dc\_display, 8, "DC: %i%c", duty\_cycle, '%');

**else**

**snprintf**(dc\_display, 8, "DC: 0%i%c", duty\_cycle, '%');

LCD\_print\_str(dc\_display);

}

}

}

// Initialize Output pin 2.5 to Timer A0 PWM

**void** **initTimA0**(){

// set pin 2.5 to TimA0

P2->SEL0 |= 1<<5;

P2->SEL1 &= ~(1<<5);

P2->DIR |= BIT5; // set pin 2.5 to output mode

// Configure TimA0.2 to SMCLOCK, UP mode, Interrupt Enable on max

TIMER\_A0->CTL &= 0xFC00;

TIMER\_A0->CTL |= 0x0212;

TIMER\_A0->CCTL[2] |= TIMER\_A\_CCTLN\_OUTMOD\_7; // Reset/set output mode

TIMER\_A0->CCR[0] = PWM\_PER; // set frequency as 1 kHz

TIMER\_A0->CCR[2] = ( PWM\_PER \* 50 ) / 100; // set duty cycle as 50% to start

TIMER\_A0->EX0 |= TIMER\_A\_EX0\_IDEX\_\_8; // input clock factor of 8

NVIC->ISER[0] |= (1<<9); // enable TA0\_N interrupt

}

//Initialize Timer A1 for 1s interrupt

**void** **initTimA1**(){

// Configure TimA1 to SMCLOCK, UP mode, Interrupt Enable on max

TIMER\_A1->CTL &= 0xFC00;

TIMER\_A1->CTL |= 0x0212;

TIMER\_A1->CTL |= 0b11<<6; // add 1/8 divider

TIMER\_A1->CCR[0] = ONE\_HZ\_PSC; // set frequency as 1 hz

TIMER\_A1->EX0 |= TIMER\_A\_EX0\_IDEX\_\_8; // add another 1/8 divider

NVIC->ISER[0] |= (1<<11); // enable TA1\_N interrupt

}

//Initialize ADC6 on pin 4.7

//Interrupt upon completed conversion

**void** **initADC**(){

// start sampling on SC bit, source a timer, use SMCLK, single-channel single conversion mode

// 96 sample/hold time, turn core on

ADC14->CTL0 &= 0x0;

ADC14->CTL0 |= (1<<26) | (1<<21) | (0b101<<12) | (0b101<<8) | (1<<4);

// 12 bit resolution and use memory location 4 to start

ADC14->CTL1 &= 0xF0000000;

ADC14->CTL1 |= (0b10<<4) | (4<<16);

ADC14->MCTL[4] |= 0x6; // input on A6

ADC14->IER0 |= (1<<4); // enable interrupt for mem[4]

ADC14->CTL0 |= 0b10; // enable

NVIC->ISER[0] |= (1<<24); // enable ADC interrupt in NVIC

}

// initializes pushbutton P1.1 as GPIO input

**void** **initPushbutton**(**void**)

{

P1->SEL0 &= ~0b10; // use GPIO function

P1->SEL1 &= ~0b10;

P1->DIR &= ~0b10; // make input

P1->REN |= 0b10; // allow pull up/down

P1->OUT |= 0b10; // setup as pull up

}

// Interrupt Handler for TA0 (PWM Timer)

**void** **TA0\_N\_IRQHandler**(**void**){

uint16\_t dummy = TIMER\_A0->IV;

//P4->OUT ^= BIT0;

}

// Interrupt Handler for TA1 (Queue ADC Timer)

**void** **TA1\_N\_IRQHandler**(**void**){

uint16\_t dummy = TIMER\_A1->IV;

ADC14->CTL0 |= 1; // start a new ADC conversion

}

// Interrupt Handler for ADC

// Read value and update TimA0.2 duty cycle

**void** **ADC14\_IRQHandler**(**void**)

{

adc\_val = ADC14->MEM[4];

duty\_cycle = (uint8\_t)((((**float**)adc\_val)/ADC\_RANGE)\*100);

TIMER\_A0->CCR[2] = ( PWM\_PER \* (100-duty\_cycle) ) / 100; // invert duty cycle for output

}