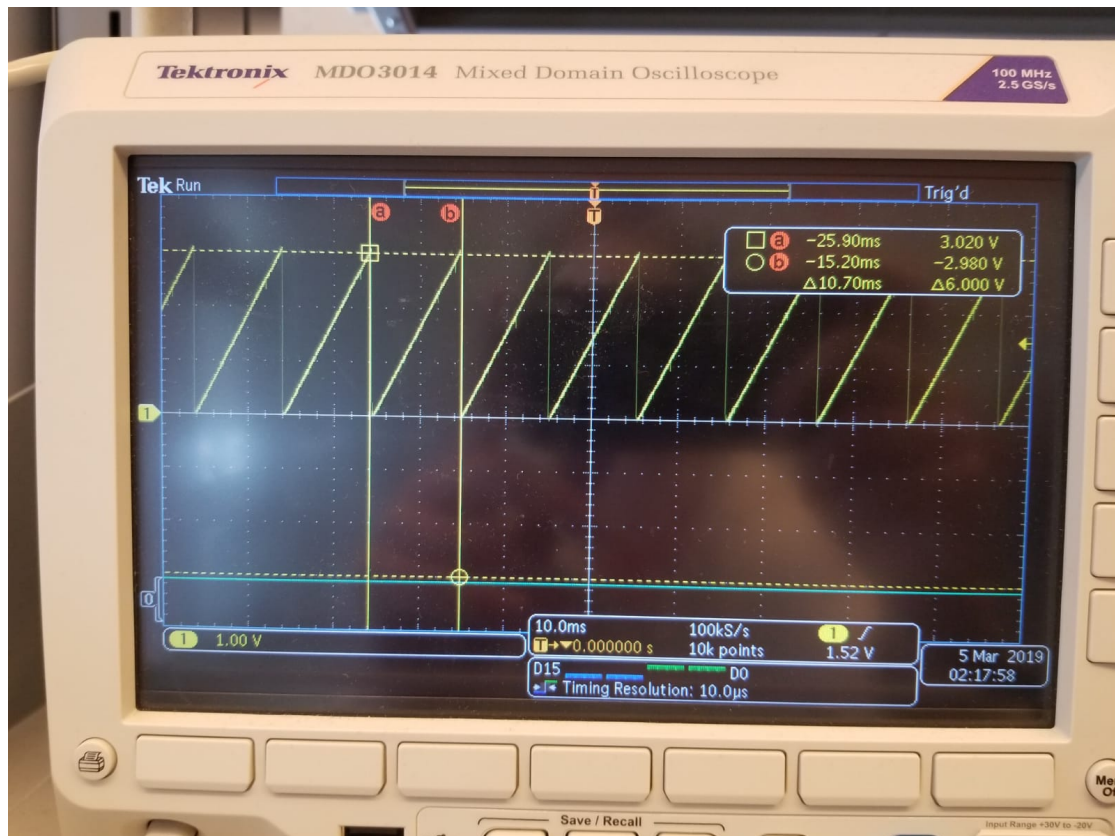


EECS 3215
Digital Logic Design
Lab 5
Michael Founk 213641204
Sean O'Brien 213735741

Problem Statement

_____The goal of this lab is to learn how to use a DAC on the board that we are given and to use it to create a sawtooth function on the oscilloscope. We are to start by simply testing that the DAC works by just having a fixed value shown. Then we create the sawtooth graph by setting the register values to a sequence of numbers and then converting it to analog. The period of the sawtooth waveform is 10.34ms, which gives us a 96.7Hz. Our solution uses



Top-Level Design

In our project, we opted to use the from scratch method and manually enable the DAC and assign our values to the DAC data register. Pin 30, where the DAC is located, is assigned to output. The DAC is enabled, the clock is enabled for the DAC, and the DMA is disabled. Since the buffer is not enabled, the DAT holds the output value. In our design, to utilize all 12 bits available, we used the DACL to hold the eight lower order bits and the DACH to hold the four higher order bits.

Code

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
```

```

#include "clock_config.h"
#include "MKL43Z4.h"
#include "fsl_debug_console.h"
#include <stdlib.h>

int main(void) {

    SysTick->LOAD = 0xFFFFFFFF;
    SysTick->CTRL = 0x05;

    PTE->PDDR |= 1<<30u; /* Configure pin30 as output */

    SIM->SCGC6 |= 1<<31u; /* set DAC clock enable bit */

    DAC0->C0 |= 1<<7; /* set DAC enable bit */
    DAC0->C0 |= 1<<6; /* set DAC enable bit */
    DAC0->C1 |= 0<<7; /* disable DAC buffer */

    int c = 0, d = 0;

    while (1) {

        if (c < 255){
            c++;
            DAC0->DAT[0].DATL = c;
        }
        else if (c == 255 && d < 15){
            d++;
            c = 0;
            DAC0->DAT[0].DATL = c;
            DAC0->DAT[0].DATH = d;
        }
        else {
            c = 0;
            d = 0;
            DAC0->DAT[0].DATL = c;
            DAC0->DAT[0].DATH = d;
        }

    }
}

```