ECE 3720

Microcomputer Interfacing Laboratory

Section 6

Timothy Driscoll

Date Performed: September / 19 / 2019

Lab 3

ABSTRACT:

A lab designed to show the application of the PIC32MX150F128D microcontroller’s internal comparators when combined with inputs from a single potentiometer.

**INTRODUCTION:**

In this lab, we created a new project that involved software, internal functions of the microcontroller and external hardware. The internal components of the PIC32MX150F128D microcontroller that was used were the three comparators. The external hardware that was used was a 10k potentiometer. These components were then combined with the software to create a project that would have three separate comparators turn on three separate LEDs in order as the potentiometer is rotated.

**EXPERIMENTAL PROCEDURES:**

In this lab, we started by unpacking the microcontroller and attaching the chip kit in the proper orientation to ensure that the pins wouldn’t have a flipped result. Then we could start wiring the circuit to complete the project. This circuit is represented by figure 1 below. The circuit includes the necessary hardware including the potentiometer (trim pot) and LEDs. It is also specifically wired to the pins that represent C1INA, C2IND, C3IND, C3INA and the VREF on the PIC32MX150F128D. Also, in order to achieve the 3.0 V input into the microcontroller the variable power supply on the NI-ELVIS was used. A more detailed description of the pin connections can be found in the discussion section. After the completion of the circuit wiring, we created a new project using the MPLAB X IDE software. After creating the project, we added a new C type source file where we wrote the code in order to have the microcontroller function properly. The code can be viewed below in the figures and tables section. The flow of the code includes setting the proper pins as inputs and outputs, then the needed comparator inputs were all set to be analog inputs. After this initial setup the 3 comparators were all set up with the desired parameters. Lastly the three comparator status registers were set to equal the three LED output registers within the infinite while loop so they would constantly be updated. The program could then be loaded to the microcontroller and the circuit could be tested using the NI-ELVIS Board and software.

**RESULTS and DISCUSSION:**

In this lab, we were successfully able to get the desired output from the project and achieve all the desired goals described in the slides. Which involved having the LEDs turn on sequentially as the trim pot was adjusted resulting in a varying voltage output. The first LED was powered on after the trim pot’s output hit 1.2 V. This corresponds to when the first comparator compares the trim pot output directly to the internal reference of the microcontroller which was 1.2 V. The second LED was then turned on when the output of the trim pot hit about 2.48 V. This corresponds to the microcontrollers 3.3 V output being inputted as the CVREF to comparator 2 and then being multiplied by (15/24) due to the value of CVR that was set. We deemed this fraction to be close to 0.75 thus giving us the 2.48 V threshold. Lastly LED 3 would be turned on when the trim pot output would reach a voltage of 3.0 V. This 3.0 V threshold was created by our external voltage reference, this voltage was supplied using the function generator and was compared using comparator 3. In order to understand why all the LEDs lit up sequentially at the different thresholds we had to learn how two key components functioned. The first component we needed to understand was the trim pot which simply acts as a variable resistor. As the trim pot is turned the resistance increases thus increasing the output voltage. The next component that we had to understand was how the comparators worked. We learned that the comparator functioned as an op-amp but differed by outputting a digital signal instead of an analog single. The comparator will compare the two inputted voltages and was setup to determine when the trim pot output voltage hit the defined threshold voltage. The only error that I encountered in this lab were multiple careless human errors. These errors included not checking if I had my outputs going to working LEDs and making some simple coding errors that were a result of copy and paste.

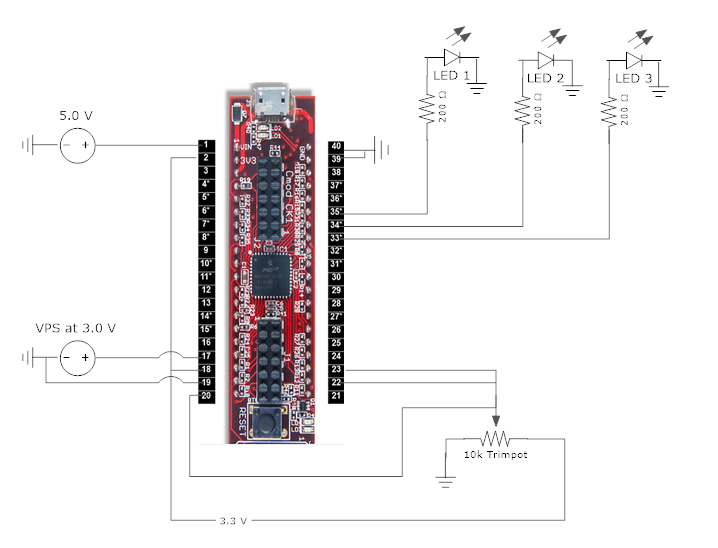
**CONCLUSSION:**

We have come to the conclusion that in this lab there were some key procedural points that needed to be learned in order to achieve all the goals. To start we first needed to learn how to assign the specific registers to be analog active. This allowed us to provide all the comparator inputs as analog as the comparator expected. The next key point that we needed to understand was that the trim pot we were using was not entirely precise. Since the trim pot was not the most precise, we learned that as it was turned the rising resistance was not linearly proportional. The effects of this trim pot on our experiment was that LED 1 would turn on at about a half turn then LED 2 and 3 would turn on very quickly after. These results reflect the fact that rising resistance relationship was not linear with the turn amount. The next thing we had to learn was the function on the comparator which I described in the above section. In order to put the lab all together It was critical that we learned how to determine the proper registers that corresponded to the needed pins as described in the Lab 3 PowerPoint. Lastly, we needed to rely on the PIC32 Datasheet in order to determine how to setup each comparator with the correct code to achieve the desired outputs. By working through the lab, I was able to determine all the necessary procedural points to understand and achieve the desired outputs for the lab.

**REFERENCES:**

Clemson University’s ECE 372 Lab 3 PowerPoint.

**FIGURES AND TABLES:**

****

**Figure 1: Wiring for lab 3 (Pin connections described in experimental procedures)**

**CODE:**

#include <plib.h>

main(){

// Set three LEDs as Output

TRISBbits.TRISB5 = 0;

TRISCbits.TRISC5 = 0;

TRISCbits.TRISC4 = 0;

// Set input pins

TRISBbits.TRISB3 = 1;

TRISBbits.TRISB2 = 1;

TRISBbits.TRISB0 = 1;

TRISBbits.TRISB15 = 1;

TRISAbits.TRISA0 = 1;

TRISAbits.TRISA1 = 1;

ANSELBbits.ANSB3 = 1;

ANSELBbits.ANSB2 = 1;

ANSELBbits.ANSB0 = 1;

ANSELBbits.ANSB15 = 1;

ANSELAbits.ANSA0 = 1;

ANSELAbits.ANSA1 = 1;

// Turn on the comparator

CM1CONbits.ON = 1;

CM2CONbits.ON = 1;

CM3CONbits.ON = 1;

CM1CONbits.CCH = 0b11;

CM1CONbits.CREF = 0;

CM1CONbits.CPOL = 0;

CM2CONbits.CCH = 0b10;

CM2CONbits.CREF = 1;

CM2CONbits.CPOL = 1;

CM3CONbits.CCH = 0b10;

CM3CONbits.CREF = 0;

CM3CONbits.CPOL = 1;

CVRCONbits.ON = 1;

CVRCONbits.CVRSS = 1;

CVRCONbits.CVRR = 0;

CVRCONbits.CVR = 15;

while(1){

LATBbits.LATB5 = CMSTATbits.C1OUT;

LATCbits.LATC5 = CMSTATbits.C2OUT;

LATCbits.LATC4 = CMSTATbits.C3OUT;

}

}