**COMPARATOR**

**Aayahna M. Herbert**

**September 26, 2017**

**ECE 3720 Section 007**

**Microcontroller Interfacing Laboratory**

**Abstract:** A circuit was built in order to send inputs to the microcontroller’s comparators from the potentiometer. As the potentiometer rotated clockwise, the 3 separate comparators were supposed to turn on 3 LEDs sequentially when IVref, CVref, and C3INA were turned on in that order and stay on until the potentiometer was rotated counterclockwise and the voltage was lowered. While the LEDs did turn on, they did not do so in sequential order due to not having enough time to deal with the CPOL of the comparators.

**INTRODUCTION**

The purpose of this lab was to design a circuit that sends three inputs to the microcontroller from the 10k potentiometer and one input to the microcontroller from the voltage power supply (VPS). The microcontroller would then output signals to three LEDs that correspond to the three comparators. When done correctly, as the potentiometer rotates CW, the LEDs should light up, and stay on, in order of when comparator one turns on at IVref, the next one turns on at CVref, and the last turns on at C3INA. They should then turn off when the potentiometer rotates CCW.

**EXPERIMENTAL PROCEDURES**

The equipment used include one PIC32 MC, NI-ELVIS II board, and a 10k potentiometer. The first thing to be done, to avoid forgetting it later, is to turn on the three comparators by using CVRCONbits.ON and CMXCONbits.ON. Next, the registers for CVRCONX should then be set to the correct values; CVRR is set to zero to have 0.75CVRSRC, CVRSS is zero to turn AVDD and AVSS (these are also the 3.3V and ground pins, making the circuit simpler), and CVR is fifteen to get the closest value for 0.75CVref. When setting up the board, the analog inputs that will used are C1INA (pin B3), C2IND (pin B2), C3IND (pin B15), and C3INA (pin B3). C1INA, C2IND, and C3IND will come from the potentiometer’s second pin (pins one and three are power and ground) while C3INA comes from the Voltage Power Supply, which should set to 3.0V. The outputs for the MC will go from pins C2, C1, and C0 to LED0, LED1, and LED2, respectively. In the code using CMXCONbits, the comparators should be set to where IVref, C2IND, and C3IND are the inverting inputs and C1INA, CVref, and C3INA are the non-inverting inputs. Before testing the board, the while loop section of the code should have the LEDs and their corresponding comparator’s COUT equate each other using LATxbits and CMSTATbits. When testing the circuit, if the LEDs are lighting up out of order, the code should checked to make sure the LEDs are with the correct comparator and if the CPOL should be turned on or off.

**RESULTS**

After the circuit was hooked up and ready to be tested, the program detected no compiling errors when ran so the board could then be tested. While running tests, when the potentiometer was rotated, the LED lights lit up out of order. The correct results were never found due to running out of time and, therefore, not being able to the CPOL for each comparator.

**DISCUSSION**

Problems arose when the LEDs didn’t light up with their corresponding comparators in sequential order. The LEDs themselves were tested individually to make sure none of the ones in use were broken and the potentiometer worked because it was changing lights, just not in correct order. The code was second-checked by the TA and the wires all came from a new pack recently purchased to avoid the possibility of dead wires being used and affecting the lab results. The only possible error source left would be not having enough time to go through the code and modify it by turning on or off the comparators’ CPOLs and possibly switching around the LED and COUT pairs.

**CONCLUSIONS**

In general, the take-away of this lab was to get a firm understanding of how a potentiometer can affect a microcontroller’s voltage comparators and to get a better understanding of the inner workings of the PIC32 MC since they will be used more and more as lab progresses.

**FIGURES AND TABLES**

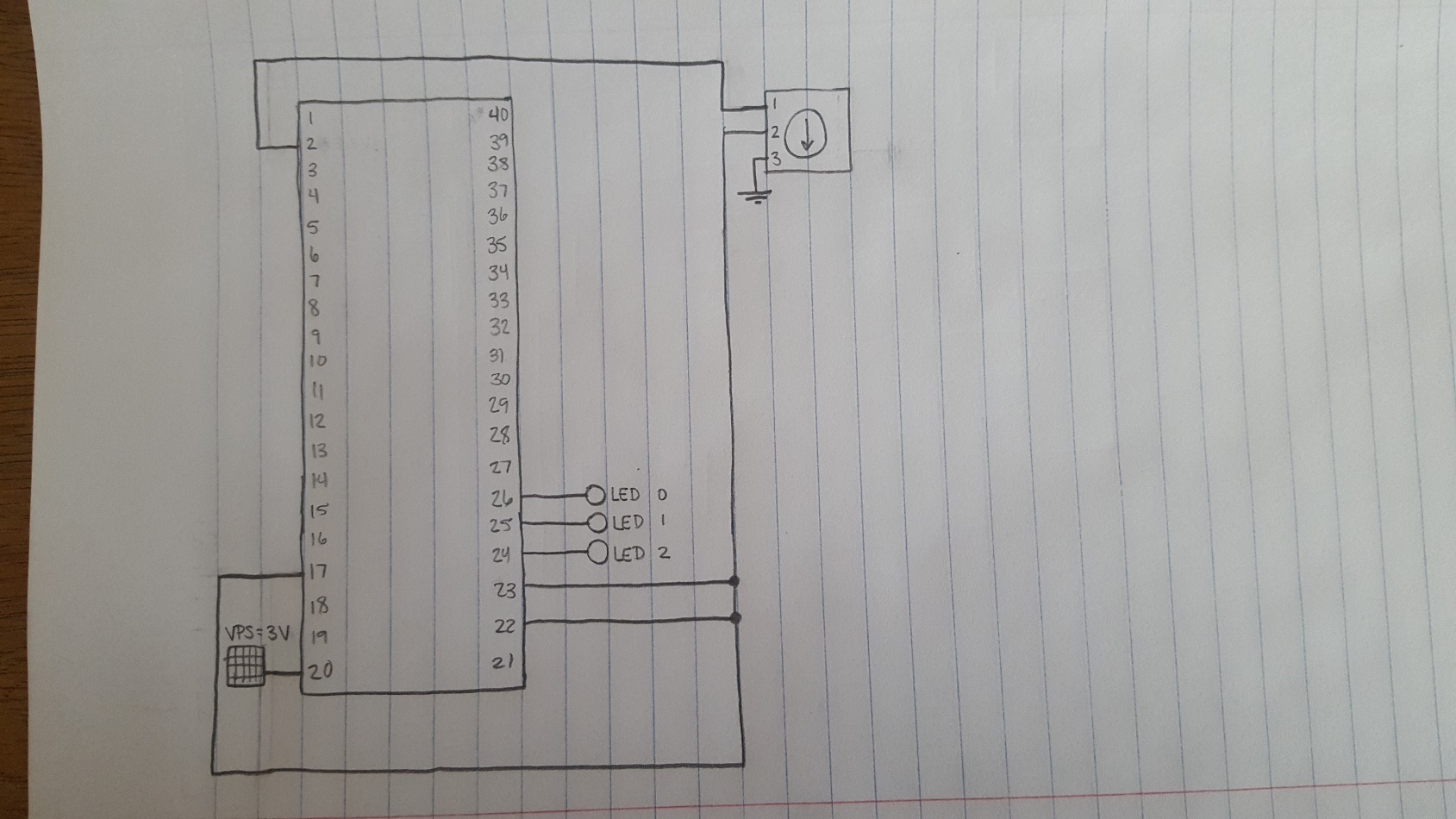


Figure 1: Circuit Schematic

**CODE**

#include<plib.h>

void main (void)

{

CVRCONbits.ON = 1;

CM1CONbits.ON = 1;

CM2CONbits.ON = 1;

CM3CONbits.ON = 1;

CVRCONbits.CVRR = 0;

CVRCONbits.CVRSS = 0;

CVRCONbits.CVR = 15;

TRISBbits.TRISB0 = 1;

TRISBbits.TRISB15 = 1;

TRISBbits.TRISB2 = 1;

TRISBbits.TRISB3 = 1;

TRISCbits.TRISC2 = 0;

TRISCbits.TRISC1 = 0;

TRISCbits.TRISC0 = 0;

CM1CONbits.CCH = 0b11;

CM1CONbits.CREF = 0;

CM2CONbits.CCH = 0b10;

CM2CONbits.CREF = 1;

CM3CONbits.CCH = 0b10;

CM3CONbits.CREF = 0;

while(1)

{

LATBbits.LATC2 = CMSTATbits.C1OUT;

LATBbits.LATC1 = CMSTATbits.C2OUT;

LATBbits.LATC0 = CMSTATbits.C3OUT;

}

}