**ET420 Microcontroller Applications**

Matt Alanskas

**Assignment #1**

**35 points**

**Part 1:** Create a C application using a 16F1823 to strobe the outside LEDs in a seven-segment display to give the illusion of a rotating circle. Use a pushbutton to control the direction of the rotation. Initially the rotation should be clockwise (segment order: a-b-c-d-e-f-a …). Each segment should be displayed for about 100ms. Holding down a pushbutton should reverse the rotation order. We have Liteon LTS-4301JR and LTS-4801JR seven segment displays available in the lab.

You will need to turn in this cover sheet, your design, your properly commented code, a CAD-drawn schematic diagram, test plan, and test data. I will check your functioning systems off in lab. Your design should consist of a paragraph describing your approach to the problem, and a flowchart or pseudocode. Your test plan should consist of how you will test your program, the sample data used, and the results obtained.

**Part 2:**

You are programming a PIC16F1823. Port A is to be configured as RA0 – analog input, RA1, RA2, RA3 – digital input, RA4 – analog input, RA5 – digital output, RC0 – digital output, RC1,RC2 – digital input, RC3 analog input, RC4, RC5 – digital output. All digital inputs need pull-up resistors. Fill in the following configuration registers (using hex) for this application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port A** |  |  | **Port C** |  |
| PORTA = | 0x00 |  | PORTC = | 0x00 |
| LATA = | 0x00 |  | LATC = | 0x00 |
| ANSELA= | 0x00 |  | ANSELC = | 0x00 |
| TRISA = | 0x0f |  | TRISC = | 0x01 |
| WPUA = | 0x04 |  | WPUC = | 0x00 |

|  |  |  |  |
| --- | --- | --- | --- |
| Part 1 | Functionality: |  | /10 |
| Part 1 | Documentation: |  | /10 |
| Part 1 | Code Analysis: |  | /5 |
| Part 2 | Code Analysis: |  | /10 |

**Design Approach:**

To begin the software side of this assignment, I tried to follow along with the given examples from the recorded zoom lectures and between those and pulling information off of the data sheets I was able to get the code together. Moving onto the wiring, it was pretty simple to put together going off the wiring diagrams from the pic and seven segment display data sheets. The trickiest part with the wiring was ensuring that the correct pins were wired to the correct destination on the seven-segment display.

**Flowchart or Pseudo Code:**

Set all the configuration bits

Include xc.h

Set frequency to 1000000

Start main function

Set port a and port c values according to assigned pin values above

PORTA = 0x00;

LATA = 0x00;

ANSELA = 0x00;

TRISA = 0x0f;

WPUA = 0x04;

PORTC = 0x00;

LATC = 0x00;

ANSELC = 0x00;

TRISC = 0x01;

WPUC = 0x00;

Define hex array for diePa and diePc

Enable pull-up resistors

Set clock to 1Mhz

Start while loop

if(RA2)

for(int i=0; i<6; i++)

LATA = diePa[i];

LATC = diePc[i];

Delay 100ms

else

for(int i=5; i>-1; i--)

LATA = diePa[i];

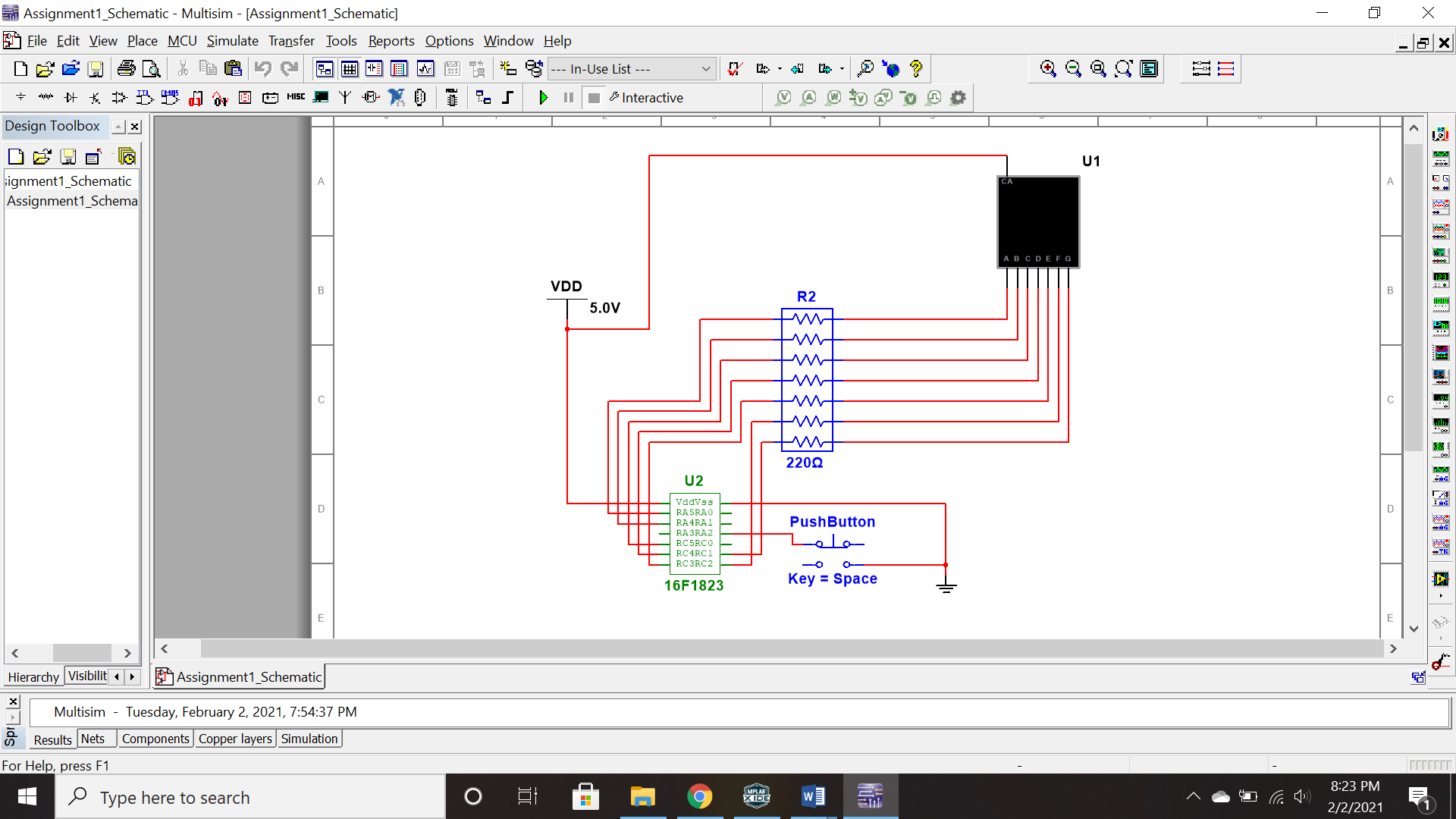
LATC = diePc[i];

Delay 100ms

End while loop

Return zero

**Schematic:**



**Test Plan:**

Once my program was finally uploading onto the pic, I ran into the issue of all of my segments on the seven-segment display staying on at all times, rather than staying off and only lighting up when the racetrack pattern came around. I discovered that inside of my for loops I had LATA=diePa[i] and LATA=diePc[i], when it should have been LATA=diePa[i] and LATC=diePc[i]. Once the segments were properly functioning, staying low and allowing the racetrack pattern to go around, I noticed that the pattern was skipping segment A while spinning clockwise and segment F while spinning counter clockwise. After picking through my code for awhile I found the simple fix to this problem, I had i<5 in my for loop instead of i<6. Switching this to a six fixed the issue and the program worked perfectly!