ECE 376 - Homework #5

Keypads in C, Stepper Motors, NeoPixels in C

NeoPixel Flashlight

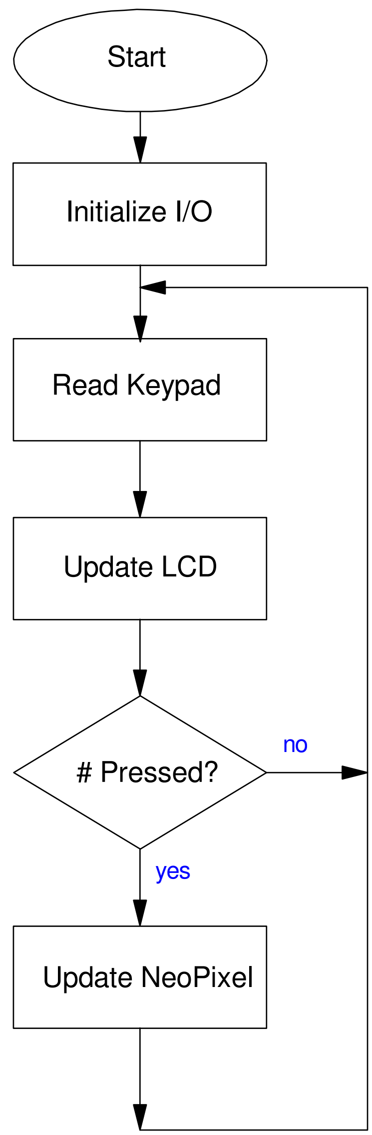
1) Requirements: Specify the inputs / outputs / how they relate.

- Input a number from 0..255 using the keypad
- Press RB0
- The NeoPixel then lights up with a white light at that brightness level (0..255)

2) C code, flow chart, and resulting number of lines of assembler

Code: Main Loop

```
//   NeoPixel Flashlight
//
//   Input a number from 000 to 255
//   Press RB0
//   NeoPixel turns on at that brightness

// Global Variables

unsigned char PIXEL @ 0x000;

const unsigned char MSG0[21] = "Flashlight
       ";
const unsigned char MSG1[21] = "Level:
       ";


// Subroutine Declarations
#include <pic18.h>

// Subroutines
#include        "lcd_portd.c"
```



Compiler Results

```
Memory Summary:
    Program space        used   10E6h (   4326) of 10000h bytes   (  6.6%)
    Data space           used     2Ch (     44) of   F80h bytes   (  1.1%)
    EEPROM space         used      0h (      0) of   400h bytes   (  0.0%)
    ID Location space    used      0h (      0) of      8h nibbles (  0.0%)
    Configuration bits   used      0h (      0) of      7h words   (  0.0%)
```

3) Validation: Collect data in lab to verify you met the requirements.

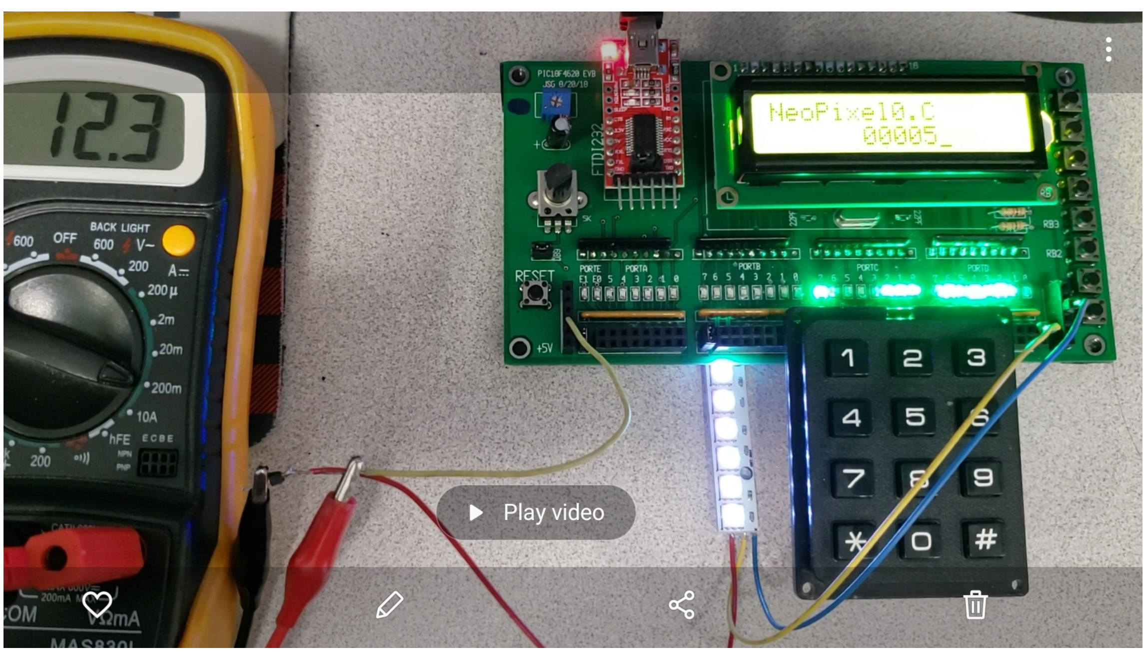
Requirement: Input a number from 000 to 255 using the keypad

- Input 000 (works)
- Input 255 (works)
- Input 123 (works)

Requirement: Press #. The NeoPixel goes to that brightness (255 = 100%)

Input Number	NeoPixels	Current (mA)	% Full Scale theory	% Full Scale measured
0	off	7.1	0%	0.0%
5	dim	12.0	1.9%	1.9%
50		58.9	19.6%	20.48%
100		110.0	39.2%	40.69%
255	really bright	260	100%	100.0%

4) Demo. Video or in person.



Analog Inputs

5) Determine how long it takes to do an A/D conversion with a PIC processor

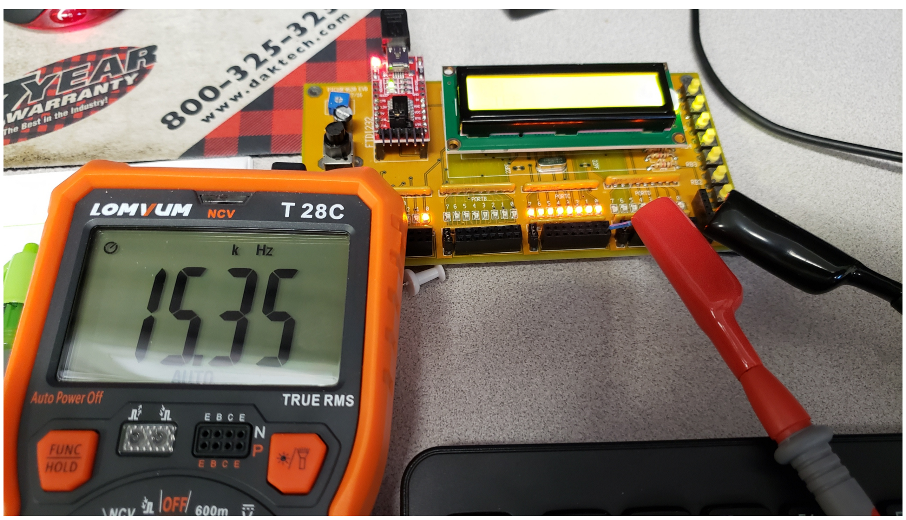
```
void main(void)
{
   TRISC = 0;
   ADCON1 = 0x0F;

// Turn on the A/D input
   TRISA = 0xFF;
   TRISE = 0x0F;
   ADCON2 = 0x95;
   ADCON1 = 0x07;
   ADCON0 = 0x01;

   while(1) {
      A2D = A2D_Read(0);
      PORTC = PORTC + 1;
      }
   }
```

f = 15.35kHz

$$N = \left(\frac{10{,}000{,}000}{2 \cdot Hz} \right) = 325.73$$

It takes about 32.57us to do an A/D read

(a little less if you take the time to count into account)



6) Assume the A/D reads 875 for the following circuit.

- What is the voltage, Vx?
- What is the resitance, Rt?
- What is the temperature?

Assume

$$R_t = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

Vx is proportional to the A/D reading

$$V_x = \left(\frac{875}{1023}\right)5.00V$$

$$\boxed{V_x = 4.2766V}$$

Rt comes from voltage division:

$$V_x = \left(\frac{R_t}{R_t+1000}\right)5V$$

$$R_t = \left(\frac{V_x}{5-V_x}\right)1000\Omega$$

$$\boxed{R_t = 5912\Omega}$$

Tempeature comes from the thermistor equation

$$5912.16\Omega = 1000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right)\Omega$$

$$\boxed{T = -10.58C}$$

Stepper Motor Angle Control

7) Requirements: Specify the inputs / outputs / how they relate.

Input:

- Analog Input: 0..255

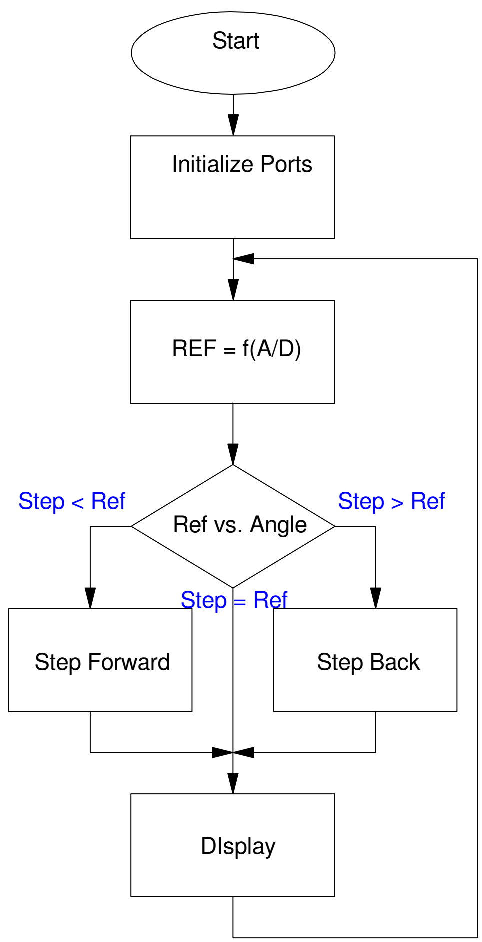
Output:

- Stepper Motor

Relationship

- Input a number from 000 to 255 using the analog input
- The stepper motor then moves to that number of steps
- At a rate of 30ms / step, +/- 5ms

8) C code, flow chart, and resulting number of lines of assembler



```c
// Stepper3.C
// Position control of a stepper motor

// Global Variables
const unsigned char MSG1[16] = "REF            ";
const unsigned char MSG2[16] = "STEP           ";

unsigned char TABLE[4] = {1, 2, 4, 8};

// Subroutine Declarations
#include <pic18.h>
#include "LCD_PortD.C"



unsigned int A2D_Read(unsigned char c)
{
   unsigned int result;
   unsigned char i;
   c = c & 0x0F;
   ADCON0 = (c << 2) + 0x01; // set Channel Select
   for (i=0; i<20; i++); // wait 2.4us (approx)
   GODONE = 1; // start the A/D conversion
   while(GODONE); // wait until complete (approx 8us)
   return(ADRES);
   }


// main routine
void main(void) {
   unsigned int i, REF, STEP ;

   TRISA = 0;
   TRISB = 0xFF;
   TRISC = 0;
   ADCON1 = 0x0F;

   STEP = 0;
   REF   = 100;

// Initialize the A/D port
   TRISA = 0xFF;
   TRISE = 0x0F;
   ADCON2 = 0x85;
   ADCON1 = 0x07;
   ADCON0 = 0x01;

   LCD_Init();
   LCD_Move(0,0);  for (i=0; i<16; i++) LCD_Write(MSG1[i]);
   LCD_Move(1,0);  for (i=0; i<16; i++) LCD_Write(MSG2[i]);
   Wait_ms(100);

   while(1) {

      REF = A2D_Read(0) / 4;

      if (STEP < REF) STEP = STEP + 1;
      if (STEP > REF) STEP = STEP - 1;

      PORTC = TABLE[STEP % 4];
      LCD_Move(0,8);  LCD_Out(REF, 5, 0);
      LCD_Move(1,8);  LCD_Out(STEP, 5, 0);
      Wait_ms(1);
      }
   }
```

Compilation Results

```
Memory Summary:
    Program space        used   B1Ah (  2842) of 10000h bytes   (  4.3%)
    Data space           used    2Dh (    45) of   F80h bytes   (  1.1%)
    EEPROM space         used     0h (     0) of   400h bytes   (  0.0%)
    ID Location space    used     0h (     0) of     8h nibbles (  0.0%)
    Configuration bits   used     0h (     0) of     7h words   (  0.0%)
```

9) Validation: Collect data in lab to verify you met the requirements.

Requirement: Input numbers 000 to 255 with the analog input

- Knob all the way left: 000 (works)
- Knob all the way right (255) (works)
- Knob in the middle (126) (works)

Requirement: The motor goes to that angle

Input	Went To..
0	0
50	50 steps (90 degrees)
100	100 steps (180 degrees)
200	200 steps (360 degrees)

Requirement: At a rate of 30ms / step, +/- 5ms

- 255 steps took 7.47 seconds (using stopwatch)
- Time = 29.3ms / step

10) Demo. Video or in person.