```
/* File Name:    fixed.c
 * Authors:      Kevin Yee (kjy252), Stefan Bordovsky ()
 * Created:      01/19/2017
 * Description:  Changes integers to strings that can be outputs in an LCD
 *               Contains function to plot points to draw shapes
 *
 * Lab Number: MW 330-500
 * TA: Mahesh
 *
 * Hardware Configurations:
 * ST7735R LCD:
 *     Backlight    (pin 10) connected to +3.3 V
 *     MISO         (pin 9) unconnected
 *     SCK          (pin 8) connected to PA2 (SSI0Clk)
 *     MOSI         (pin 7) connected to PA5 (SSI0Tx)
 *     TFT_CS       (pin 6) connected to PA3 (SSI0Fss)
 *     CARD_CS      (pin 5) unconnected
 *     Data/Command (pin 4) connected to PA6 (GPIO)
 *     RESET        (pin 3) connected to PA7 (GPIO)
 *     VCC          (pin 2) connected to +3.3 V
 *     Gnd          (pin 1) connected to ground
 */

#include <stdlib.h>
#include <stdint.h>
#include "fixed.h"
#include "ST7735.h"

#define TRUE 1
```

```
#define FALSE 0



/***************numDigits**************
 counts the number of digits in an integer
 */

 int num_Digits(int32_t n)
{
	int numDigit =0;
	while(n!=0)
	{
		n=n/10;
		numDigit++;
	}
	return numDigit;
}

/***************change_To_Output***************
 changes a string to an appropriate LCD ST7735 format
 dependant on maximum digits and decimal positions
 */

 char* change_To_Output(int32_t n, int MAX_DIGITS, int DECIMAL_POSITION, char buffer[], int
numDigits, int signedInt)
{
	int i =0;
	//create buffer translating integer to string
```

```
		for(i=MAX_DIGITS-1; i > 0; i--)
		{
			if(i != DECIMAL_POSITION)
			{
			buffer[i] = n%10 + '0'; //inserts values
			n = n/10;
				numDigits --;
			}
			else
			{
				buffer[i] = '.'; //places decimal point
			}
			if(i > numDigits && i < DECIMAL_POSITION -1  && buffer[i] == '0')
			{
				buffer[i] = ' '; //removes leading zeroes
			}
		}
	if(!signedInt && n != 0)
	{
		 buffer[0] = n%10 + '0';
	}

	return buffer;
}

/***************ST7735_sDecOut3***************
 converts fixed point number to LCD
 format signed 32-bit with resolution 0.001
 range -9.999 to +9.999
```

```
 Inputs:  signed 32-bit integer part of fixed-point number
 Outputs: none
 send exactly 6 characters to the LCD
Parameter LCD display
 12345    " *.***"
  2345    " 2.345"
 -8100    "-8.100"
  -102    "-0.102"
    31    " 0.031"
-12345    " *.***"
 */
void ST7735_sDecOut3(int32_t n)
        {
                int MAX = 9999; //max upper bound
                int MIN = -9999; //min upper bound

                int MAX_DIGITS = 6; //Maximum digit of fixed point number
                int DECIMAL_POSITION = 2; //Position of Decimal Point to be placed
                char buffer [6] = {' '}; //Initialize array to empty string
                int numDigit;


                //check if within bounds
                if(n > MAX || n < MIN)
                {
                        ST7735_OutString("*.***");
                        return;
                }
                //check if negative
```

```
        if(n < 0)
        {
                buffer[0] = '-';
                n *= -1;
        }
        else
        {
            buffer[0] = ' ';
        }

        numDigit = num_Digits(n); //count number of digits
        change_To_Output(n, MAX_DIGITS, DECIMAL_POSITION, buffer, numDigit,TRUE);

        ST7735_OutString(buffer);


    }

    /**************ST7735_uBinOut8***************
unsigned 32-bit binary fixed-point with a resolution of 1/256.
The full-scale range is from 0 to 999.99.
If the integer part is larger than 256000, it signifies an error.
The ST7735_uBinOut8 function takes an unsigned 32-bit integer part
of the binary fixed-point number and outputs the fixed-point value on the LCD
Inputs:  unsigned 32-bit integer part of binary fixed-point number
Outputs: none
send exactly 6 characters to the LCD
Parameter LCD display
   0       "  0.00"
```

```
   2       "  0.01"
  64       "  0.25"
 100       "  0.39"
 500       "  1.95"
 512       "  2.00"
5000       " 19.53"
30000      "117.19"
255997     "999.99"
256000     "***.**"
*/
void ST7735_uBinOut8(uint32_t n){

        int MAX = 255999;
        int MIN = 0;

        double RESOLUTION = 256;
        double num = (double) n;
        int MAX_DIGITS = 6;
        int DECIMAL_POSITION = 3;
        char buffer [6] = {' '};
        double roundedResult;
        int scaledResult =0;
        int numDigit;

        //check if value is within bounds
        if( n > MAX || n < MIN)
        {
                ST7735_OutString("***.**");
                return;
```

```
        }
        //calculate value based on Resolution: I * RESOLUTION
        roundedResult = ((double)((double) num / RESOLUTION*100)/100);



        scaledResult = roundedResult * 100;
        numDigit = num_Digits(n); //count number of digits
        change_To_Output( scaledResult, MAX_DIGITS, DECIMAL_POSITION,  buffer,numDigit,FALSE);

        ST7735_OutString(buffer);









}

/**************ST7735_XYplotInit***************
 Specify the X and Y axes for an x-y scatter plot
 Draw the title and clear the plot area
 Inputs:  title  ASCII string to label the plot, null-termination
        minX   smallest X data value allowed, resolution= 0.001
        maxX   largest X data value allowed, resolution= 0.001
        minY   smallest Y data value allowed, resolution= 0.001
        maxY   largest Y data value allowed, resolution= 0.001
 Outputs: none
```

```
 assumes minX < maxX, and miny < maxY
*/
static int32_t MinX, MaxX, MinY, MaxY;
void ST7735_XYplotInit(char *title, int32_t minX, int32_t maxX, int32_t minY, int32_t maxY){
        //sets static variable
        MinX = minX; MaxX = maxX;
  MinY = minY; MaxY = maxY;

        //clear screen and output title
        ST7735_FillScreen(0);
  ST7735_OutString(title);
}

/**************ST7735_XYplot***************
 Plot an array of (x,y) data
 Inputs:  num    number of data points in the two arrays
       bufX   array of 32-bit fixed-point data, resolution= 0.001
       bufY   array of 32-bit fixed-point data, resolution= 0.001
 Outputs: none
 assumes ST7735_XYplotInit has been previously called
 neglect any points outside the minX maxY minY maxY bounds
*/
void ST7735_XYplot(uint32_t num, int32_t bufX[], int32_t bufY[]){

        /*
        In this code, the plotting area is a square on the bottom (0,32) to (127,159)
   i goes from 0 to 127
   x=MaxX maps to i=0
   x=MaxX maps to i=127
```

```
  i = (127*(x-MinX))/(MaxX-MinX);
   y=MaxY maps to j=32
   y=MinY maps to j=159
  j = 32+(127*(MaxY-y))/(MaxY-MinY);
*/



for (uint32_t i=0 ; i < num; ++i) {
     int32_t x = (127*(bufX[i] - MinX)) / (MaxX - MinX);
     int32_t y = 32 + (127*((MaxY - bufY[i]))  / (MaxY - MinY));


     ST7735_DrawPixel(x,   y,   ST7735_CYAN);
     ST7735_DrawPixel(x+1, y,   ST7735_CYAN);
     ST7735_DrawPixel(x,   y+1, ST7735_CYAN);
     ST7735_DrawPixel(x+1, y+1, ST7735_CYAN);


  }
}
```