//Michael Park(mjp2853), Jack Zhao
//SP16 01/26/16

Fixed.C

```
#include <stdint.h>
#include <stdio.h>
#include "fixed.h"
#include "ST7735.h"

//global variables shared by the plotting methods (3,4)
int32_t Y_min, X_min , Y_max, X_max;
char *graphtitle;

//The purpose of this technique is to use integer arithmetic (int, long...) while being able to represent fractions
void ST7735_sDecOut3(int32_t n)
{
	uint8_t flag = 0;
	char ch[6];
	for (int i=0; i<6; i++)
	{
		ch[i] = '0';
	}

	if (n > 9999 || n < -9999)
	{
		printf(" *.*** ");
	}
	else
	{
		sprintf(ch, "%.5d", n);
		for (int i=0; i<6; i++)
		{
			//negatives
			if(ch[0] == '-')
			{
				if(i==3)
				{
					fputc('.', 0);
					fputc(ch[i], 0);
				}
				else if (i==1 && ch[i]=='0')
				{
					flag = 1;
					continue;
				}
				else
					fputc(ch[i], 0);
```

```
                        }

                        //positives
                        else
                        {
                                if(i==2)
                                {
                                        fputc('.', 0);
                                        fputc(ch[i], 0);
                                }
                                else if(ch[i] == '0' && i==0)  //replacing the most significant digit with
space
                                {
                                        fputc(' ' , 0);
                                        ch[i] = ' ';
                                }
                                else
                                        fputc(ch[i], 0);
                        }
                }

                if(flag) //if skipped one digit for negative values
                        fputc(' ', 0);
        }
}

void ST7735_uBinOut8(uint32_t n)
{
        uint32_t n1 = n*1000/256;               //integer part
        uint32_t n2 = n1%10;
        char ch[6];
        for (int i=0; i<6; i++)
        {
                ch[i] = '0';
        }
        /*
        if (n2 >= 5)
        {
                n1 = n1/10;
                n1++;
        }
        */

        n1 = n1/10;

        //build string and then output
        if (n1 > 99999) //error case
        {
                printf("***.** ");
```

```
        }
        else
        {
                sprintf(ch, "%.5d", n1);
                for (int i=0; i<6; i++)
                {
                        if (i == 3)       //putting a dot
                        {
                                fputc('.', 0);
                                fputc(ch[i], 0);
                        }
                        else if(ch[i] == '0' && i==0)  //replacing the most significant digit with space
                        {
                                fputc(' ' , 0);
                                ch[i] = ' ';
                        }
                        else if((ch[i-1]==' ') && (ch[i]=='0')) //replacing second most significant digit
with space
                        {
                                fputc(' ' , 0);
                        }
                        else
                                fputc(ch[i], 0);
                }
        }
}

void ST7735_XYplotInit(char *title, int32_t minX, int32_t maxX, int32_t minY, int32_t maxY)
{
        Output_Clear();
        double res = 0.001;
        graphtitle  = title;
        Y_min = minY*res;
        Y_max = maxY*res;
        X_min = minX*res;
        X_max = maxX*res;

        ST7735_DrawString(0, Y_max, graphtitle, ST7735_BLUE);
        ST7735_PlotClear(Y_min, Y_max);
}


/**************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[])
{
	double res = 0.001;
	uint32_t Yrange = Y_max - Y_min;
	uint32_t Xrange = X_max - X_min;
	double x;
	double y;
	for(int i=0;i<num;i++){
		x=(double)bufX[i]*res;
		y=(double)bufY[i]*res;

		if(y<Y_min) y=Y_min;
		if(y>Y_max) y=Y_max;

		if(x<X_min) x=X_min;
		if(x>X_max) x=X_max;
		// X goes from 0 to 127
		// j goes from 159 to 32
		// y=Y_max maps to j=32
		// y=Y_min maps to j=159
		uint32_t h =(127*(X_max-x))/Xrange;//X_min+ (X_max-x)*8*Xrange;//+
(X_min*(x))/Xrange;
		uint32_t j = 32+(127*(Y_max-y))/Yrange;

		if(h>127) h=127;
		if(j<32) j = 32;
		if(j>159) j = 159;
		ST7735_DrawPixel(h,   j,   ST7735_BLUE);
		ST7735_DrawPixel(h+1, j,   ST7735_BLUE);
		ST7735_DrawPixel(h,   j+1, ST7735_BLUE);
		ST7735_DrawPixel(h+1, j+1, ST7735_BLUE);
	}
}
```