

**main.c**

/\* ========================================

\*

\* Copyright YOUR COMPANY, THE YEAR

\* All Rights Reserved

\* UNPUBLISHED, LICENSED SOFTWARE.

\*

\* CONFIDENTIAL AND PROPRIETARY INFORMATION

\* WHICH IS THE PROPERTY OF your company.

\*

\* ========================================

\*/

#include "project.h"

#include <stdio.h>

#define PUSHED 0

#include "stdlib.h"

#include <math.h>

#include <string.h>

char mystring[20];

int count=0;

int countFollow=0;

int countDisplay=0;

int angle=0;

//This converts the polar cordinates given to the point to point used for the DrawLine

void GLCD\_Draw\_Line\_Polar(int centerX, int centerY, int angle, int length, int color)

{

GLCD\_DrawLine(centerX,centerY,(length\*cos(((angle\*M\_PI))/180))+centerX,(length\*sin(((angle\*M\_PI))/180))+centerY, color);

}

int main(void)

{

CyGlobalIntEnable; /\* Enable global interrupts. \*/

/\* Place your initialization/startup code here (e.g. MyInst\_Start()) \*/

GLCD\_Start();

A\_Interrupt\_ClearPending();

A\_Interrupt\_Start();

B\_Interrupt\_ClearPending();

B\_Interrupt\_Start();

Backlight\_Write(1);

B\_LED\_Write(1);

G\_LED\_Write(1);

PWM\_1\_Start();

Timer\_1\_Start();

Timer\_1\_WriteCounter(0);

//Prints the ms ending for the time

GLCD\_PrintString("ms", 10, 50, GLCD\_WHITE, GLCD\_BLACK);

//Draw the outer circle and min and max lines

//This is just for looks

GLCD\_DrawCircle(80,65,45,GLCD\_RED);

GLCD\_Draw\_Line\_Polar(80,65,270,45,GLCD\_RED);

GLCD\_Draw\_Line\_Polar(80,65,90,45,GLCD\_RED);

GLCD\_Draw\_Line\_Polar(80,65,180,45,GLCD\_YELLOW);

GLCD\_DrawCircle(80,65,5,GLCD\_YELLOW);

while(1)

{

//This updates the display with the current count number

sprintf(mystring, "%5d", countDisplay);

GLCD\_PrintString(mystring, 10, 10, GLCD\_WHITE, GLCD\_BLACK);

if(count != countFollow)

{

//Takes the count variable and sets to countdisplay

countDisplay=count;

//Draws black over the previous line

GLCD\_Draw\_Line\_Polar(80,65,angle,40,GLCD\_BLACK);

//calculate the angle that the line should be set at

//This is also known as mapping on arduino platform map() is used

angle=(((countDisplay +500) \* (-180)) / (1000)) + 270;

//Draw the line to indicat the time before or after

GLCD\_Draw\_Line\_Polar(80,65,angle,40,GLCD\_YELLOW);

//Sets the counter follower to check if the count has changed

countFollow=count;

//Draws the small center circle

GLCD\_DrawCircle(80,65,5,GLCD\_YELLOW);

//Draws the red lines back if the yellow line is not on them

if(angle !=270)

GLCD\_Draw\_Line\_Polar(80,65,270,45,GLCD\_RED);

if(angle !=90)

GLCD\_Draw\_Line\_Polar(80,65,90,45,GLCD\_RED);

if(angle !=180)

GLCD\_Draw\_Line\_Polar(80,65,180,45,GLCD\_RED);

}

}

}

/\* [] END OF FILE \*/

**Interupt Includes/headers**

#include "project.h"

#include <stdio.h>

#define PUSHED 0

extern int count;

extern char mystring[20];

**A\_Interrupt.c**

count= Timer\_1\_ReadCompare();

if(count>500)

{

//Changes counter number to -500 to 500

count-=1000;

}

Timer\_1\_ClearInterrupt(Timer\_1\_INTR\_MASK\_CC\_MATCH);

A\_Interrupt\_ClearPending(); //Needed to make other interrupts work

**B\_Interrupt.c**

Timer\_1\_WriteCounter(0);

PWM\_1\_ClearInterrupt(PWM\_1\_INTR\_MASK\_TC);

B\_Interrupt\_ClearPending(); //Needed to make other interupts work