//Evan Garcia

//Professor Bolding

//EE 3280

//April 12, 2017

#include <project.h>

#include <stdlib.h>

void createStarfield(int16, int16[], int16[]);

void twinkle(int16, int16[], int16[]);

void createCrosshair(int16, int16);

void deleteCrosshair(int16, int16);

void normalBlasterExploson(int16, int16, int16);

void largeBlasterExploson(int16, int16, int16);

void deleteBlasterExploson(int16, int16, int16);

int blasterCharged();

int main()

{

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

uint32 MasterCounter; //Counter for Master Loop

int16 i; //Counter for smaller loops

int16 x = 62; //X and Y coordinates for center of circle

int16 y = 62;

int16 randNumX; //X coordinates for stars

int16 randNumY; //Y coordinates for stars

int16 starCoordinateX[150]; //Keep track of star x location

int16 starCoordinateY[150]; //Keep track of star y location

uint8 blasterReady = 2; //Keep track of blaster state

uint32 blasterCounter; //Counter for blaster time delay

GLCD\_Start();

Backlight\_Write(1); //Turn on backlight

LED\_Blue\_Write(1); LED\_Green\_Write(1); LED\_Red\_Write(1); //Turn off all LEDs (1 means off)

GLCD\_Clear(GLCD\_BLACK);

LED\_Blue\_Write(0); //Blaster light for Hyper Charged state

//Find Coordinates for starfield

for(i = 0; i < 150; i++)

{

randNumX = rand() % 121;

randNumY = rand() % 121;

starCoordinateX[i] = randNumX;

starCoordinateY[i] = randNumY;

}

//Draw Starfield

createStarfield(i, starCoordinateX, starCoordinateY);

//Master Loop

for(MasterCounter = 0; MasterCounter < UINT32\_MAX; MasterCounter++)

{

//Draw Crosshair

createCrosshair(x,y);

//Make stars twinkle

if(MasterCounter % 150 == 0)

{

createStarfield(i, starCoordinateX, starCoordinateY);

twinkle(i, starCoordinateX, starCoordinateY);

}

//Update blaster state to 1 (Charged), and set Green LED

if(blasterCounter % 1300 == 0 && blasterReady == 0)

{

LED\_Red\_Write(1);

LED\_Green\_Write(0);

blasterReady = 1;

}

//Update blaster state to 2 (Hyper-Charged), and set Blue LED

if(blasterCounter % 1300 == 1299 && blasterReady == 1)

{

LED\_Green\_Write(1);

LED\_Blue\_Write(0);

blasterReady = 2;

}

//Scroll Crosshair Up

if (A\_Read()==0 && x > 10) {

deleteCrosshair(x,y);

x = x - 1;

createCrosshair(x,y);

createStarfield(i, starCoordinateX, starCoordinateY);

}

//Scroll Crosshair Right

if (B\_Read()==0 && y < 120) {

deleteCrosshair(x,y);

y = y + 1;

createCrosshair(x,y);

createStarfield(i, starCoordinateX, starCoordinateY);

}

//Scroll Crosshair Down

if (C\_Read()==0 && x < 121) {

deleteCrosshair(x,y);

x = x + 1;

createCrosshair(x,y);

createStarfield(i, starCoordinateX, starCoordinateY);

}

//Scroll Crosshair Left

if (D\_Read()==0 && y > 10) {

deleteCrosshair(x,y);

y = y - 1;

createCrosshair(x,y);

createStarfield(i, starCoordinateX, starCoordinateY);

}

//Call explosion functions, set red LED

if(Joy\_Center\_Read()==0 && (blasterReady == 1 || blasterReady == 2)) {

if(blasterReady == 1)

{

normalBlasterExploson(i,x,y);

}

else if (blasterReady == 2)

{

largeBlasterExploson(i,x,y);

}

createStarfield(i, starCoordinateX, starCoordinateY);

LED\_Green\_Write(1);

LED\_Blue\_Write(1);

LED\_Red\_Write(0);

blasterReady = 0;

blasterCounter = 0;

}

blasterCounter++;

}

//Reset Master Counter

if(MasterCounter == UINT32\_MAX - 1)

{

MasterCounter = 0;

}

}

//Draw Starfield

void createStarfield(int16 i, int16 starCoordinateX[], int16 starCoordinateY[])

{

for(i = 0; i < 150; i++)

{

GLCD\_Pixel(starCoordinateX[i], starCoordinateY[i], GLCD\_WHITE);

}

}

//Make random stars go black to simulate twinkling

void twinkle(int16 i, int16 starCoordinateX[], int16 starCoordinateY[])

{

for(i = rand() % 151; i < 150; i+= 80 )

{

GLCD\_Pixel(starCoordinateX[i], starCoordinateY[i], GLCD\_BLACK);

}

}

//Draw Crosshair

void createCrosshair(int16 x, int16 y)

{

GLCD\_DrawCircle(x, y, 8, GLCD\_YELLOW);

GLCD\_DrawLine(x, y-12, x, y + 12, GLCD\_YELLOW);

GLCD\_DrawLine(x-12, y, x + 12, y, GLCD\_YELLOW);

}

//Delete old crosshairs

void deleteCrosshair(int16 x, int16 y)

{

GLCD\_DrawCircle(x, y, 8, GLCD\_BLACK);

GLCD\_DrawLine(x, y-12, x, y + 12, GLCD\_BLACK);

GLCD\_DrawLine(x-12, y, x + 12, y, GLCD\_BLACK);

}

//Normal Size explosion

void normalBlasterExploson(int16 i ,int16 x, int16 y)

{

for(i = 0; i < 25 ; i++)

{

GLCD\_DrawCircle(x, y, i, GLCD\_CRIMSON);

}

for(i = 0; i < 26 ; i++)

{

GLCD\_DrawCircle(x, y, 25 - i, GLCD\_BLACK);

}

}

//Hyper Charged explosion

void largeBlasterExploson(int16 i ,int16 x, int16 y)

{

for(i = 0; i < 50 ; i++)

{

GLCD\_DrawCircle(x, y, i, GLCD\_CRIMSON);

}

for(i = 0; i < 51 ; i++)

{

GLCD\_DrawCircle(x, y, 50 - i, GLCD\_BLACK);

}

}

Top Level Schematic – Star Blasters

