```c
#include "address_map_arm.h"
#include <stdbool.h>
#include <stdlib.h>
#include <stdio.h>

volatile int pixel_buffer_start; // global variable

void plot_pixel(int x, int y, short int line_color)
{
    *(short int *)(pixel_buffer_start + (y << 10) + (x << 1)) = line_color;
}

// Swaps 2 numbers using the XOR operation
void swap(int * x, int * y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

void draw_line(int x1, int y1, int x2, int y2, short int colour)
{
    // Check steepness of the line, if it is steep, it's better
    // to move along the y-axis when drawing
    bool is_steep = abs(y2-y1) > abs(x2-x1);
    // If it is steep switch the x and y values
    // the drawing loop will decide how the drawing will occur
    if(is_steep) {
        swap(&x1,&y1);
        swap(&x2,&y2);
    }

    // We are going to increment from x1 to x2 so
    // swap the endpoints if x1 > x2
    if(x1 > x2) {
        swap(&x1,&x2);
        swap(&y1,&y2);
    }

    int deltax = x2-x1;
    int deltay = abs(y2-y1);
    int error = -(deltax/2);
    int x,y,y_step;

    // Figure out how y will be incremented
    if(y1<y2) y_step = 1;
    else y_step = -1;

    for(x=x1,y=y1; x<=x2; x++) {
        // If the line is steep the x and y values are swapped
        if(is_steep) plot_pixel(y,x,colour);
        else plot_pixel(x,y,colour);

        // Check margin of error
        error += deltay;
        if(error>=0) {
            y += y_step; // Increment y val
            error -= deltax; // Reset error
        }
    }
}

// Draw black to every pixel on the screen
void clear_screen()
{
    int x,y;
    // The screen is 320x240
    for(x=0; x<320; x++) {
        for(y=0; y<240; y++) {
```

```
            plot_pixel(x,y,0x0000);
        }
    }
}

// Synchronizes the display with the VGA timing
void wait_for_vsync()
{
    volatile int * pixel_ctrl_ptr = (int *)PIXEL_BUF_CTRL_BASE;
    register int status;

    *pixel_ctrl_ptr = 1; // Start synchronization process

    // Keep waiting until the whole screen ahs been drawn
    do {
        status = *(pixel_ctrl_ptr + 3);
    } while((status & 0x01) != 0);
}

int main()
{
    volatile int * pixel_ctrl_ptr = (int *)PIXEL_BUF_CTRL_BASE;
    /* Read location of the pixel buffer from the pixel buffer controller */
    pixel_buffer_start = *pixel_ctrl_ptr;

    clear_screen();

    // Infinitely loop
    int y = 0; // We are only moving the line's y-coordinate
    int y_step = 1; // Start by moving down
    while(1) {
        draw_line(0,y,319,y,0x001F); // Draw a blue line at new y coordinate

        wait_for_vsync(); // Draw the line at a rate of 60 pixels/second

        draw_line(0,y,319,y,0x0000); // Black line to "erase" previous line
        y += y_step; // Increment y
        // Bounce the line when it gets to the ends
        if(y==239) y_step = -1;
        else if(y==0) y_step = 1;
    }

    return 0;
}
```