Application Note: AN00168

How to use the LCD library

This application note demonstrates how to use the lib_lcd library to write to an LCD screen via a parallel bus. The LCD library is used to display on a single graphics LCD module of upto 800 * 600 pixels with a maximum pixel clock of 62.5MHz. This application note uses a 480 * 272 pixel LCD module with a pixel clock of 16.67MHz which is available on XA-SK-SCR480 sliceCARD.

Features

- Standard component to support different LCD displays with an RGB interface.
- Different color depths 32 bpp, 16 bpp, etc. based on user configuration.
- Resolution of up to 800 * 600 pixels.
- Configurability of LCD pixel dimensions, clock rate, horizontal and vertical timing requirements.
- The function lcd_server requires just one core. The client functions, located in lcd.h are very low overhead and are called from the application.

Required tools and libraries

- xTIMEcomposer Tools - Version >=14.0
- XMOS LCD library - Version 3.0.0

Required hardware

This application note is designed to run on an XMOS xCORE-Analogue sliceKIT. The application provided has been implemented and tested on the A16 sliceKIT (XP-SKC-A16 1V0) with an LCD sliceCARD(XA-SK-SCR480 1V0). This application can be modified to work on any XMOS XCORE device as long as the control signals,

- are connected to one bit ports
- do not overlap with any other used ports
- are all on the same tile

Prerequisites

- This document assumes familiarity with the XMOS xCORE architecture, LCD basics, the XMOS tool chain and the xC language. Documentation related to these aspects which are not specific to this application note are linked to in the references appendix.
- For descriptions of XMOS related terms found in this document please see the XMOS Glossary[1].
- For the full API listing of the XMOS LCD Library please see the library user guide[2].
- For the datasheet of the LCD module attached with the LCD sliceCARD[3], KENTEC K430WQA-V4-F[4].

[1] http://www.xmos.com/published/glossary
[2] http://www.xmos.com/support/libraries/lib_lcd
[3] https://www.xmos.com/support/boards?product=15832
[4] http://www.kentec.com.hk/images/UploadFile/20111115190908-6.pdf

1 Overview

1.1 Introduction

LCD Demo app note demonstrates how the LCD library is used to write image data to the LCD screen. The purpose of this application is to show how data is passed to the lcd_server.

1.2 Block diagram

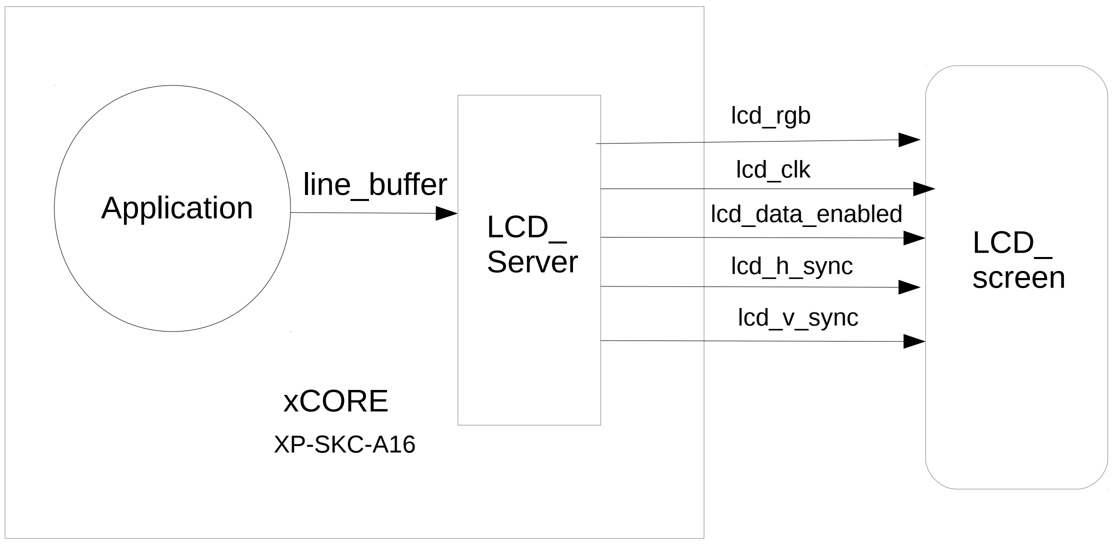

Figure 1: Block diagram of LCD application example

2 LCD demo

This demo application has two major tasks:

- A task to run the lcd server
- A task to run the lcd demo

These tasks communicate via the use of xC interfaces. The following diagram shows the task and communication structure of the application.

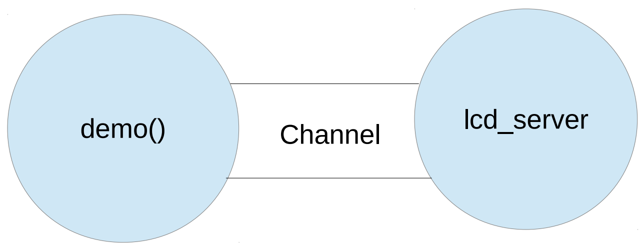

Figure 2: Task Diagram

2.1 Makefile additions for this example

To start using the LCD library, add lib_lcd to the Makefile:

```
USED_MODULES = ... lib_lcd
```

Access the LCD functions in the source code via the header files:

```
#include "lcd.h"
```

2.2 Declaring ports

The LCD library connects to external pins via xCORE ports. In lcd_demo.xc these are declared as variables of type port in the beginning of the file:

```
on tile[1] : out buffered port:32   lcd_rgb                      = XS1_PORT_16B;
on tile[1] : out port               lcd_clk                      = XS1_PORT_1I;
on tile[1] : out port               lcd_data_enabled             = XS1_PORT_1L;
on tile[1] : out buffered port:32   lcd_h_sync                   = XS1_PORT_1J;
on tile[1] : out port               lcd_v_sync                   = XS1_PORT_1K;
on tile[1] : clock                  lcd_cb                       = XS1_CLKBLK_1;
```

lcd_rgb has the pixel data in the standard RGB565 format supplied to the LCD over a 16 bit parallel bus. lcd_clk is the clock line to clock all other signals. lcd_data_enabled is to indicate that the data on the 16 bit paralell bus is valid. lcd_h_sync is a signal which sends a pulse to indicate the start of the horizontal scan. lcd_v_sync is a signal which sends a pulse to indicate the start of the vertical scan. lcd_cb is a CLOCK BLOCK which provides a regular clock signal to which ever port it is configured to and in this case it is lcd_clk.

2.3 Implementation specific defines

The defines available are:

```
#define LCD_CLOCK_DIVIDER 3
#define LCD_H_FRONT_PORCH 5
#define LCD_H_BACK_PORCH 40
#define LCD_H_PULSE_WIDTH 1
#define LCD_V_FRONT_PORCH 8
#define LCD_V_BACK_PORCH 8
#define LCD_V_PULSE_WIDTH 1
#define LCD_HEIGHT 272
#define LCD_WIDTH 480
#define LCD_BYTES_PER_PIXEL 2
#define LCD_OUTPUT_MODE data16_port16
#define LCD_ROW_WORDS (LCD_WIDTH/2)
```

- LCD_CLOCK_DIVIDER: This define is used to divide the frequency of the clock block.
- LCD_HOR_FRONT_PORCH: The horizontal front porch timing requirement given in pixel clocks.
- LCD_HOR_BACK_PORCH: The horizontal back porch timing requirement given in pixel clocks.
- LCD_HOR_PULSE_WIDTH: The horizontal pulse width timing requirement given in pixel clocks. This is the duration that the hsync signal should go low to denote the start of the horizontal frame. Set to 0 when hsync is not necessary.
- LCD_VERT_FRONT_PORCH: The vertical front porch timing requirement given in horizontal time periods.
- LCD_VERT_BACK_PORCH: The vertical back porch timing requirement given in horizontal time peri-

ods.

- LCD_VERT_PULSE_WIDTH: The vertical pulse width timing requirement given in vertical time periods. This is the duration that the vsync signal should go low to denote the start of the vertical frame. Set to 0 when vsync is not necessary.
- LCD_HEIGHT: This define is used to represent the height of the LCD panel in pixels.
- LCD_WIDTH: This define is used to represent the width of the LCD panel in pixels.
- LCD_BYTES_PER_PIXEL: Count of bytes used to set the pixels RGB value, i.e. if the screen was wired for rgb565 then the LCD_BYTES_PER_PIXEL would be 2, rgb888 would be 3. This is independant of the actual bit depth of the lcd.
- LCD_OUTPUT_MODE: Depending upon the port selected to send the pixels data, it is either selected as `data16_port16` or `data16_port32`.
- LCD_ROW_WORDS: This define indicates the total number of words present.

2.4 The header file

The header `sprite.h` has the defines and data required for the image of logo X to be displayed on the LCD screen. The variable `logo[SPRITE_WIDTH_WORDS * SPRITE_HEIGHT_PX]` has all the pixel data of the image X that is displayed on the screen. lib_lcd library keeps on generating a frame with the image to be displayed embedded in it depending on the co-ordinates of the image provided.

2.5 The application main() function

Below is the source code for the main function of this application, which is taken from the source file `lcd_demo.xc`

```
int main() {
    streaming chan c_lcd;
  par {
      on tile[1]:lcd_server(
              c_lcd,
              lcd_rgb,
              lcd_clk,
              lcd_data_enabled,
              lcd_h_sync,
              lcd_v_sync,
              lcd_cb,
              LCD_WIDTH,
              LCD_HEIGHT,
              LCD_H_FRONT_PORCH,
              LCD_H_BACK_PORCH,
              LCD_H_PULSE_WIDTH,
              LCD_V_FRONT_PORCH,
              LCD_V_BACK_PORCH,
              LCD_V_PULSE_WIDTH,
              LCD_OUTPUT_MODE,
              LCD_CLOCK_DIVIDER);
    on tile[1]: demo(c_lcd);
  }
  return 0;
}
```

Looking at this in more detail the following can be observed:

- `lcd_server()` is run on one core of tile 1. This send all the pre-defined parameters to the lib_lcd.
- demo() which is also run on tile 1 passes the buffer pointer to lib_lcd.

2.6 The demo() function

The demo() function is given below:

```
void demo(streaming chanend c_lcd) {
    unsigned buffer[2][LCD_ROW_WORDS];
    int x = 20, y = 0, vx = 1, vy = 2;
    unsigned index = 1;

    for(unsigned i=0;i<LCD_ROW_WORDS;i++){
        buffer[0][i] = BACK_COLOUR;
        buffer[1][i] = BACK_COLOUR;
    }

    unsafe {

        add(x, y, 0, buffer[0]);

        lcd_init(c_lcd, buffer[0]);

        unsigned line = 1;

        while(1){
            while(line < LCD_HEIGHT) {
                add(x, y, line, buffer[index]);
                lcd_req(c_lcd);
                lcd_update(c_lcd, buffer[index]);
                index = 1 - index;
                if(line)
                    sub(x, y, line - 1, buffer[index]);
                else
                    sub(x, y, LCD_HEIGHT - 1, buffer[index]);

                line++;
            }
            line = 0;

            move_sprite(x, y, vx, vy);
        }
  }
}
```

`lcd_init` must be called before `lcd_update` is called. This puts the LCD's server into a ready state to accept the data. A double line buffer is maintained to store two successive lines. While one line buffer is being sent by the lcd_server, the demo creates the other line buffer. `lcd_update` is used to send a buffer of pixel data to the LCD server. There is a real-time requirement that this function is called often enough to maintain the display. `lcd_req` is a function (also a select handler) that acknowledges the LCDs request for the next line of pixel data. The LCD server does no buffering of pixel line arrays, therefore, for every `lcd_req` there must be only one `lcd_update`. Likewise for every `lcd_update` there must be only one `lcd_req`.

The pixel array must be on the same tile as the `lcd_server`. `x` and `y` variables present in the demo indicate the co-ardinates of the image being displayed on the screen. The image keeps moving randomly on the screen when the lcd_demo is run. `move_sprite(x, y, vx, vy)` function is responsible for the random movement of the logo and whenever the logo touches the borders the logo bounces back. add() function available in the demo pushes the pixel data value into one of the two buffers available in the demo and sub() function is used to remove any residue of the `logo[]` present in the previous buffer and fill it with the white pixel data. An overview of the the communication between demo and `lcd_server` is shown below:

2.7 Setting up the hardware

The XP-SKC-A16 sliceKIT Core board(1V0) has four slots with edge connectors: SQUARE, CIRCLE, TRIANGLE and STAR.

To setup up the system:

- Connect XA-SK-SCR480 Slice Card(1V0) to the XP-SKC-A16 sliceKIT Core board using the connector marked with the CIRCLE.
- Connect the XTAG Adapter(1V1) to sliceKIT Core board, and connect XTAG-2(1V0) to the Adapter.
- Connect the XTAG-2 to host PC using a USB extension cable if necessary.
- Set the XMOS LINK to OFF on the XTAG Adapter.
- Ensure the jumper P1 on the XA-SK-SCR480 is bridged if the back light is required. Switch on the 12V DC power supply to the sliceKIT Core board.

2.8 Running the application

Next step is to run it on the sliceKIT Core Board using the xTIMEComposer tool to load the application over JTAG (via the XTAG2 and XTAG Adapter card) into the xCORE multicore microcontroller.

- Select the file lcd_demo.xc in the AN00168_lcd_demo project from the Project Explorer.
- Click on the Run icon (the white arrow in the green circle).
- At the Select Device dialog select XMOS XTAG-2 connect to A16[0..1] and click OK. The output on the LCD screen should be a "X" in the centre of the screen.

2.9 Next steps

- Trying changing the `int x=20, y=0, vx=1, vy=2` variables on line 58, they represent: initial x coord, initial y coord, x velocity and y velocity respectively.
- In the sub() observe what happens when the line: `buffer[i] = BACK_COLOUR;` is changed to `buffer[i] = 0x12345678;` !

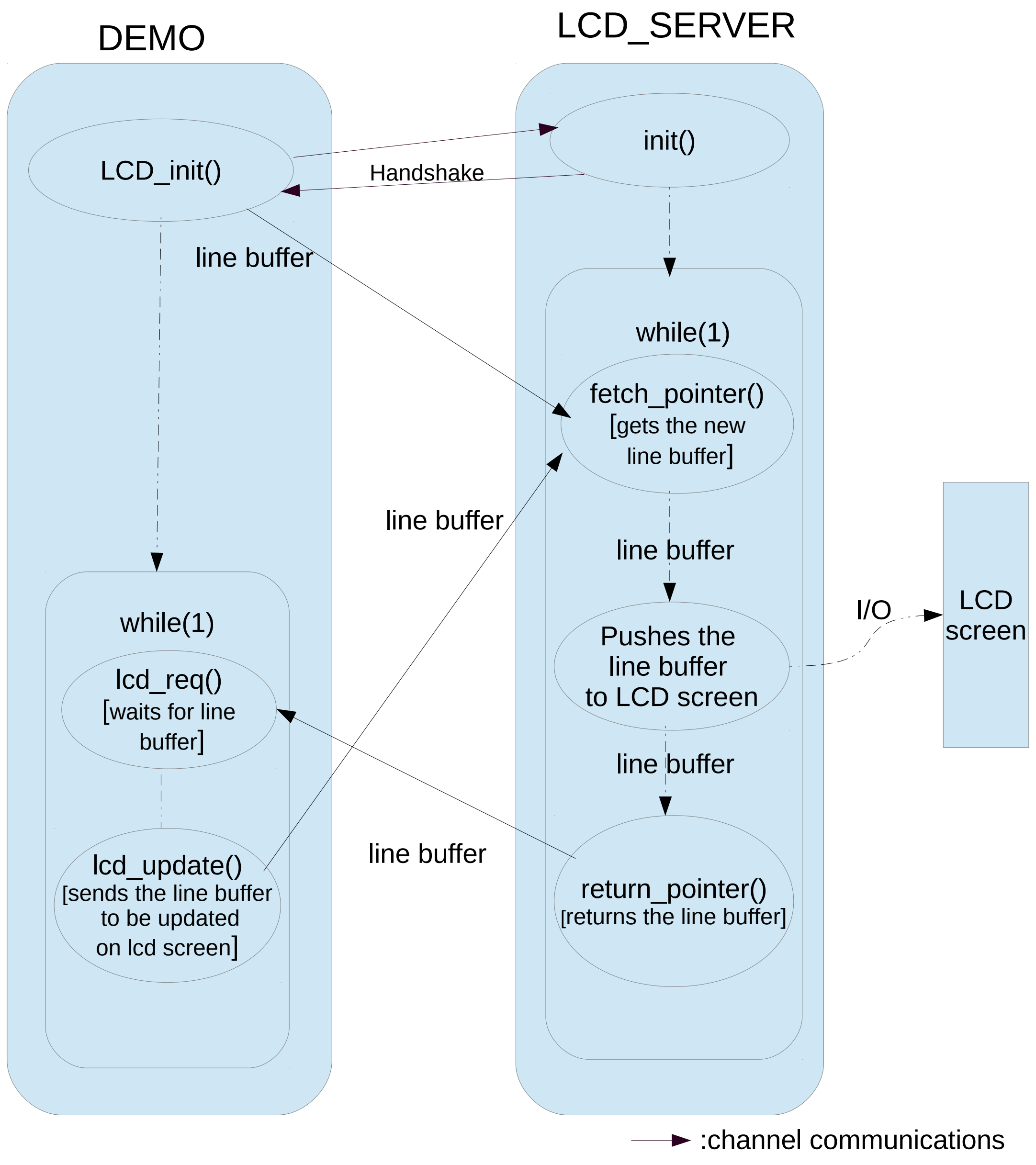
DEMO
LCD_SERVER
LCD_init()
init()
Handshake
line buffer
while(1)
fetch_pointer() [gets the new line buffer]
line buffer
line buffer
while(1)
lcd_req() [waits for line buffer]
Pushes the line buffer to LCD screen
I/O
LCD screen
line buffer
line buffer
lcd_update() [sends the line buffer to be updated on lcd screen]
return_pointer() [returns the line buffer]
:channel communications

Figure 3: Overview of the LCD demo

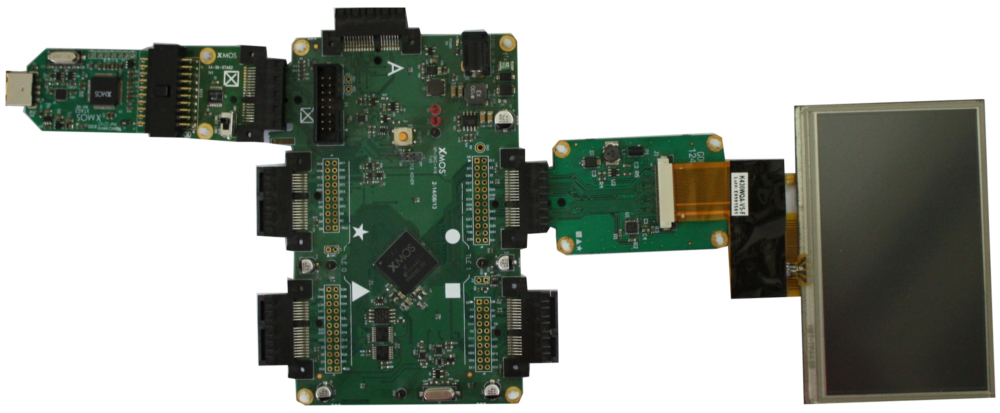

Figure 4: Hardware setup for LCD demo

3 References

XMOS Tools User Guide

http://www.xmos.com/published/xtimecomposer-user-guide

XMOS xCORE Programming Guide

http://www.xmos.com/published/xmos-programming-guide

XMOS LCD Library

https://www.xmos.com/support/libraries/lib_lcd

4 Full source code listing

4.1 lcd_demo.xc

```
// Copyright (c) 2016, XMOS Ltd, All rights reserved
#include <platform.h>
#include "lcd.h"
#include "sprite.h"

//define start
#define LCD_CLOCK_DIVIDER 3
#define LCD_H_FRONT_PORCH 5
#define LCD_H_BACK_PORCH 40
#define LCD_H_PULSE_WIDTH 1
#define LCD_V_FRONT_PORCH 8
#define LCD_V_BACK_PORCH 8
#define LCD_V_PULSE_WIDTH 1
#define LCD_HEIGHT 272
#define LCD_WIDTH 480
#define LCD_BYTES_PER_PIXEL 2
#define LCD_OUTPUT_MODE data16_port16
#define LCD_ROW_WORDS (LCD_WIDTH/2)
//define stop
/*
 * Put an lcd into circle slot of A16 board.
 * You should see a bouncing XMOS logo.
 */
static unsafe void add(unsigned x, unsigned y, unsigned line, unsigned * unsafe buffer) {
  if (line >= x && line < x + SPRITE_HEIGHT_PX)
    for (unsigned i = y; i < y + SPRITE_WIDTH_WORDS; i++)
      buffer[i] = logo[(line - x) * SPRITE_WIDTH_WORDS + (i - y)];
}

static unsafe void sub(unsigned x, unsigned y, unsigned line, unsigned * unsafe buffer) {
  if (line >= x && line < x + SPRITE_HEIGHT_PX)
    for (unsigned i = y; i < y + SPRITE_WIDTH_WORDS; i++)
      buffer[i] = BACK_COLOUR;
}

static void move_sprite(int &x, int &y, int &vx, int &vy){
    x += vx;
    y += vy;
    if (y <= 0) {
        vy = -vy;
        y = 0;
    }
    if (y >= LCD_ROW_WORDS - SPRITE_WIDTH_WORDS) {
        vy = -vy;
        y = LCD_ROW_WORDS - SPRITE_WIDTH_WORDS - 1;
    }
    if (x <= 0) {
        vx = -vx;
        x = 0;
    }
    if (x >= LCD_HEIGHT - SPRITE_HEIGHT_PX) {
        vx = -vx;
        x = LCD_HEIGHT - SPRITE_HEIGHT_PX - 1;
    }
}

void demo(streaming chanend c_lcd) {
    unsigned buffer[2][LCD_ROW_WORDS];
    int x = 20, y = 0, vx = 1, vy = 2;
    unsigned index = 1;

    for(unsigned i=0;i<LCD_ROW_WORDS;i++){
        buffer[0][i] = BACK_COLOUR;
        buffer[1][i] = BACK_COLOUR;
    }

    unsafe {

        add(x, y, 0, buffer[0]);
        lcd_init(c_lcd, buffer[0]);

        unsigned line = 1;

        while(1){
            while(line < LCD_HEIGHT) {
                add(x, y, line, buffer[index]);
                lcd_req(c_lcd);
                lcd_update(c_lcd, buffer[index]);
                index = 1 - index;
                if(line)
                    sub(x, y, line - 1, buffer[index]);
                else
                    sub(x, y, LCD_HEIGHT - 1, buffer[index]);

                line++;
            }
            line = 0;

            move_sprite(x, y, vx, vy);
        }
  }
}


//port start
on tile[1] : out buffered port:32   lcd_rgb                     = XS1_PORT_16B;
on tile[1] : out port               lcd_clk                     = XS1_PORT_1I;
on tile[1] : out port               lcd_data_enabled            = XS1_PORT_1L;
on tile[1] : out buffered port:32   lcd_h_sync                  = XS1_PORT_1J;
on tile[1] : out port               lcd_v_sync                  = XS1_PORT_1K;
on tile[1] : clock                  lcd_cb                      = XS1_CLKBLK_1;
//port stop

int main() {
    streaming chan c_lcd;
  par {
      on tile[1]:lcd_server(
              c_lcd,
              lcd_rgb,
              lcd_clk,
              lcd_data_enabled,
              lcd_h_sync,
              lcd_v_sync,
              lcd_cb,
              LCD_WIDTH,
              LCD_HEIGHT,
              LCD_H_FRONT_PORCH,
              LCD_H_BACK_PORCH,
              LCD_H_PULSE_WIDTH,
              LCD_V_FRONT_PORCH,
              LCD_V_BACK_PORCH,
              LCD_V_PULSE_WIDTH,
              LCD_OUTPUT_MODE,
              LCD_CLOCK_DIVIDER);
    on tile[1]: demo(c_lcd);
  }
  return 0;
}
```

4.2 sprite.h

```
// Copyright (c) 2016, XMOS Ltd, All rights reserved
#ifndef SPRITE_H_
#define SPRITE_H_
#define SPRITE_WIDTH_PX  64
#define SPRITE_HEIGHT_PX 64

#define BACK_COLOUR (0xffffffff)
#define SPRITE_WIDTH_WORDS  (SPRITE_WIDTH_PX/2)
```

```
unsigned logo[SPRITE_WIDTH_WORDS * SPRITE_HEIGHT_PX] = { 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xee83f79a,
                0xee81f681, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xf7baffff, 0xee81ee81, 0xee81f681, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xee81f776, 0xf661f682, 0xee81ee81, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xf6ecef31, 0xee81eea7,
                0xee60ee40, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xef32f7bc, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xe7df4f7b, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffdfffff, 0x477a975b,
                0x477a477a, 0x477a477a, 0x477a477a, 0x3f7a477a, 0x377a3f7a, 0xcfde377a,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x0f591759, 0xffffaf9d,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xef5dffff, 0x1f59e71c, 0x07590759, 0x07590759, 0x07590759, 0x07590759,
                0x07590759, 0xffff5f9b, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0x07591759, 0x475a0738, 0xffffefff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xe71cf79e, 0xcf3ce71c, 0x07590f39, 0x07590759,
                0x07590759, 0x07590759, 0x375a0759, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0x07381f59, 0x07590739, 0xafbd0f39, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffdfffff, 0xe71ce71c, 0xe71ce71c,
                0x0738973b, 0x07390759, 0x07390739, 0x07390739, 0xdfde0758, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x07382759, 0x07380738,
                0x0f380738, 0xffff7f9c, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xe71cffff,
                0xe71ce71c, 0xe71ce71c, 0x673aef1c, 0x07580738, 0x07380758, 0x07380738,
                0xffffb7dd, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0x07382759, 0x07380758, 0xafbd0738, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xe71ce73c, 0xe71ce71c, 0xe71ce71c, 0xdf1be71c, 0x07581f38,
                0x07380758, 0x577a0738, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0x07382f59, 0x3f5a0738, 0xffffdfde, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xef7dffff, 0xe71cdefb, 0xe71ce71c, 0xe71ce71c,
                0xe71bdf1c, 0x0f38d71b, 0x07380738, 0xffff2f59, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x0f382f59, 0xffff779b,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xdefbf7be, 0xdefbdefb,
                0xdefbdefb, 0xdefbdefb, 0xdefbdefb, 0x9f1adefb, 0x07370738, 0xffffd7de,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xdfdf4f5a, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xdefcdefc, 0xdefcdefc, 0xdefcdefc, 0xdefcdefc, 0xdefcdefc, 0xe6fcdefc,
                0xb7bd6f1a, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xe6879ef5, 0xee85f665, 0xee65ee65, 0xee65ee65, 0xee45ee65,
                0xee44ee44, 0xf712ee43, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0x9f39ffff, 0xe604f604, 0xee04ee04, 0xee04ee04, 0xee04ee04,
                0xee04ee04, 0xee03ee04, 0xffffe6af, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xf7beffff, 0x175acf3b, 0xe6428eac, 0xee41ee41,
                0xee21ee21, 0xee21ee21, 0xee01ee01, 0xe5e0ee01, 0xffffffbc, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0x5f1affdf, 0xede06eae, 0xede0ede0,
                0xede0ede0, 0xede0ede0, 0xede1ede0, 0xee24ede0, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xef7dffff, 0xa73be71c,
                0x0f390758, 0xf6409eab, 0xee21ee22, 0xee21ee21, 0xee21ee21, 0xee01ee01,
                0xffffee89, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x6f19defb,
                0xc5e40717, 0xedc0ede0, 0xedc0edc0, 0xedc0edc0, 0xede0edc0, 0xf778eda0,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xe73cffff, 0xe73ce71c, 0x07595f3a, 0x27370759, 0xf62096ac, 0xee21ee21,
                0xee01ee21, 0xee00ee01, 0xf779ede0, 0xffffffff, 0xffffffff, 0xffffffff,
                0xf79effff, 0x6ef9dedb, 0x17160717, 0xe5c0dde2, 0xe5c0e5c0, 0xe5c0e5c0,
                0xe5c0e5c0, 0xffffeef2, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xe71cffdf, 0xe71ce71c, 0x1f59df3c, 0x07590759,
                0x2f350759, 0xee21ae89, 0xee21ee01, 0xede1ee01, 0xede2ee00, 0xffffffde,
                0xffffffff, 0xffffffff, 0xdedbffff, 0x6ef8dedb, 0x07170717, 0xf5a056b0,
                0xedc0edc0, 0xedc0edc0, 0xe605edc0, 0xffffffdf, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xe73cef5d, 0xe73ce73c,
                0xc73ce73c, 0x07590f59, 0x07390759, 0x2f360759, 0xee21c645, 0xee01ee00,
                0xede1ede0, 0xffffff712, 0xffffffff, 0xffffffff, 0xdedbe73c, 0x6ef8dedb,
                0x07170716, 0x8e2a0717, 0xe5a0eda0, 0xedc0e5a0, 0xffdde5a1, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xe73ce73c, 0xe73ce73c, 0xef3ce73c, 0x07596f5a, 0x07590759, 0x07390759,
                0x36f40738, 0xf5e0d625, 0xe5e1e5e1, 0xf7bce5c0, 0xffffffff, 0xffdfffff,
                0xdedbdedb, 0x6ef8dedb, 0x07160716, 0x17150716, 0xe5a0d5a2, 0xeda0e5a0,
                0xfffff6f3, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffdfffff, 0xe73ce73c, 0xe73ce73c, 0xe73ce73c, 0x3f59ef3c,
                0x07590759, 0x07380759, 0x07380758, 0x56d10739, 0xf5e0c625, 0xee49edc0,
                0xffffffdf, 0xdedbffff, 0xdedbd6ba, 0x6ed8deda, 0x07160716, 0x07160716,
                0xe5a126f3, 0xe629e580, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xf79effff, 0xe73ce73c, 0xe73ce73c,
                0xe73ce73c, 0xc73ce73c, 0x07590759, 0x07380759, 0x07380738, 0x07370737,
                0x66900738, 0xeda0d603, 0xfffff779, 0xd6baf79e, 0xd6bad6ba, 0x7ed8deba,
                0x06f606f5, 0x06f606f6, 0x6e4c06f6, 0xf7deeda2, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xe73cffff,
                0xe73ce73c, 0xe73ce73c, 0xe73ce73c, 0xe73ce73c, 0x0759af3b, 0x07590759,
                0x07380758, 0x07370738, 0x07370737, 0x5eaf0717, 0xf7dde5a2, 0xd6bad6ba,
                0xd6bad6ba, 0x7ed8deba, 0x06f60715, 0x071606f6, 0x06f506f6, 0xffffe759,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xf7beffff, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0x4f7af7be, 0x07590759, 0x07380758, 0x07370738, 0x07170737, 0x07170716,
                0x86d71715, 0xd6badebb, 0xd6bad6ba, 0x7ed8d6ba, 0x06f506f5, 0x06f506f5,
                0x6f5906f5, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0x07591f59, 0x07380738, 0x07370738,
                0x07170737, 0x07160716, 0x07160716, 0xd6ba9ed9, 0xd6bad6ba, 0x7ed8d6ba,
                0x06f506f5, 0x06f506f5, 0xf7ff1f16, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x0739bfbd,
                0x07380758, 0x07170737, 0x07160737, 0x07160716, 0x06f60716, 0xd6ba16f6,
                0xd69ad69a, 0x86b8d69a, 0x06f506f4, 0x0ef506f5, 0xffffdfde, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0x879cffff, 0x07380738, 0x07170737, 0x07160737, 0x07160716,
                0x06f506f6, 0x2ed60715, 0xd69ade9a, 0x86b8d69a, 0x06f406f4, 0x877a06f4,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x07382f59, 0x07170737,
                0x07160717, 0x06f60716, 0x06f506f6, 0x06f406f5, 0xde9a76b7, 0x86b8d699,
                0x06f406f4, 0xffff4f17, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0x0737dfdf, 0x07370717, 0x07160717, 0x06f60716, 0x06f506f5, 0x06f406f5,
                0xae9806f4, 0x8e98ce99, 0x06d406d4, 0xffffe7de, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0x979cffff, 0x07170737, 0x07160716, 0x06f60716,
                0x06f506f5, 0x06f406f4, 0x16d406f4, 0x8e97ce99, 0xc7bd06d4, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x07174739,
                0x07160716, 0x06f506f6, 0x06f406f5, 0x06d406f4, 0x06d406d4, 0x8e9836b5,
                0xffff5737, 0xffffffff, 0xffffffff, 0xffffffff, 0xee0cf738, 0xffffd565,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0x2f17f7ff, 0x06f60f16, 0x06f506f5, 0x06d406f5, 0x06d406d4,
                0x06d306d4, 0x4eb606d3, 0xffffffff, 0xffffffff, 0xf79bffff, 0xdd22eeb2,
                0xdd00dce0, 0xffffe62f, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xd6dbef7d, 0xa6d9d6ba, 0x06f546d6,
                0x06f406f4, 0x06d406d4, 0x06d306d3, 0x06d306d3, 0xffffbf7c, 0xffbeffff,
                0xe500dd24, 0xdd00dce0, 0xdce0dd00, 0xfffff759, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xf7beffff, 0xd6badebb,
                0xd6badeba, 0xce99de9a, 0x7eb7b699, 0x06d32eb5, 0x06d306d3, 0x06d306d3,
                0xf7ff1ed4, 0xffffffff, 0xe60cf7bd, 0xdd00dcc0, 0xe501e500, 0xffffffbd,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffdfffff, 0xffdfffff, 0xffdfffdf,
                0xdedbffdf, 0xd6bad6ba, 0xd69ad6ba, 0xd69ad69a, 0xd679ce79, 0xce79de79,
                0x4e958e77, 0x06d31eb3, 0x46f606d2, 0xffffffdf, 0xffffffdf, 0xdd02eef5,
                0xdd23dcc0, 0xfffffffe, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffdfffff, 0xffdfffdf, 0xffdfffdf, 0xffffffdf, 0xffdfffff, 0xffdfffdf,
                0xffdfffdf, 0xf7dfffdf, 0xd6badefb, 0xd6bad6ba, 0xce99ceba, 0xce79ce9a,
                0xce79ce79, 0xce59ce59, 0xce59ce58, 0xc658d639, 0x1eb27e76, 0xffdf9f39,
                0xffdfffde, 0xffffffde, 0xe5cae568, 0xffffffdf, 0xffdfffdf, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xf7beffff, 0xffdff7be, 0xffdfffdf, 0xffdfffdf,
                0xffdfffdf, 0xf79ef7be, 0xef3cef7d, 0xdeb2df1c, 0x16f61716, 0x16f516f6,
                0x16f416f5, 0x16d416d4, 0x16d316d3, 0x1eb316d3, 0xdce2d548, 0xdce2dce2,
                0xdce2dce2, 0xc79d9587, 0xffdfffde, 0xf7dfffdf, 0xf79dffdf, 0xffdfffdf,
                0xffdfffdf, 0xffdfffdf, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0xf7bef7be, 0xef5df7be, 0xdedbdefb, 0xdedbdedb, 0xdedbdedb, 0xde6ae6b5,
                0x06f506f5, 0x06f406f5, 0x06d406d4, 0x06d306d4, 0x06d306d3, 0xae5806b2,
                0xdcc0d56c, 0xdcc0dcc0, 0xe4a0dca0, 0x26d365cb, 0xf7bef7de, 0xf7bef7be,
                0xffdef7de, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf,
                0xffffffdf, 0xffffffff, 0xffffffff, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e,
                0xf7bef79e, 0xf7bef7be, 0xf7bef7be, 0xef7df7be, 0xdedbdedb, 0xdedbdedb,
                0xdebadedb, 0xe66af649, 0x06f506f5, 0x06d406f4, 0x06d306d4, 0x06d306d3,
                0x06b206d2, 0xc6596675, 0xd4a0d5f3, 0xdca0dca0, 0xdca0d4a0, 0x06b2362e,
                0xf7be6717, 0xf7beffbe, 0xf7bef7be, 0xf7bef7be, 0xffdff7be, 0xffdfffdf,
                0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffffffff, 0xf79ef79e,
                0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e,
                0xdedbd6ba, 0xdebad6ba, 0xee4cd6b9, 0xe649f649, 0x06f406f5, 0x06d406d4,
                0x06d306d3, 0x06d206d3, 0x369306d2, 0xce38d639, 0xdc80ce5a, 0xdca0dca0,
                0xe480dca0, 0x06b20e91, 0xaf5a06b1, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0xffdff7be, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf,
                0xffffffdf, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0xf7bef7be, 0xf7bef7be, 0xd6bad6ba, 0xd6bbd6ba, 0xf628ee6f, 0xe629f628,
                0x06d406f4, 0x06d306d3, 0x06b206d3, 0x06b206b2, 0xcf1b06b2, 0xc638c618,
                0xdd07c659, 0xdc80dc80, 0xb4e3dc80, 0x06b106b2, 0x0eb206b1, 0xf7dfefbe,
                0xf7bef7be, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf,
                0xffdfffdf, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xd69adedb, 0xe674d69a,
                0xf628fe28, 0xe629f628, 0x06d306d3, 0x06d306d3, 0x06b206b2, 0x06b106b2,
                0xffffb77b, 0xc638defb, 0xcd90c638, 0xdc80dc80, 0x5dcbdc80, 0x06b106b1,
                0x06b106b1, 0xffff36d4, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xd69adefb, 0xf628de99, 0xf608f608, 0xe628f608, 0x06d306d3, 0x06b206d2,
                0x06b206b2, 0x5ef606b1, 0xffffffff, 0xc638ffdf, 0xcdf4c638, 0xdc60dc80,
                0x1e70dc80, 0x06b106b1, 0x06b106b1, 0x975a06b1, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xce79e73c, 0xf607ee0a, 0xf608f608, 0xe608f5e7,
                0x06b206b2, 0x06b206b2, 0x06b106b1, 0xffff26b2, 0xffffffff, 0xf79effff,
                0xc617c618, 0xdc60dc60, 0x0e91cca1, 0x06910691, 0x06910691, 0x06910691,
                0xffffcf9c, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xe64fef5e, 0xf607f5e7,
                0xf5e7f607, 0xdde8f5e7, 0x06b206b2, 0x06b106b1, 0x06b106b1, 0xffffd7bd,
                0xffffffff, 0xffffffff, 0xc618d6ba, 0xdc60d4a4, 0x0690b4c3, 0x06910691,
                0x06910691, 0x06900691, 0xffff26b3, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffdfffff,
                0xfdc7f737, 0xf5e7f5e7, 0xf5c7f5e7, 0xe5c7f5c7, 0x06b106b1, 0x06b106b1,
                0x87380690, 0xffffffff, 0xffffffff, 0xffffffff, 0xc618ffff, 0xdc40cd6e,
                0x06907548, 0x06900690, 0x06900690, 0x06900690, 0x4ef50690, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xf7ddffff, 0xf5c7e544, 0xf5c6f5c7, 0xf5c6f5c6, 0xe5c7f5c6,
                0x06b10eb1, 0x06900691, 0xffff46d4, 0xffffffff, 0xffffffff, 0xffffffff,
                0xef7dffff, 0xe420ce17, 0x06b02e2e, 0x06b006b0, 0x06b006b0, 0x06b006b0,
                0x06900690, 0xffffc79c, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xdce4ffff, 0xe523d460, 0xf5a6f5c6,
                0xf5a6f5a6, 0xe5c7f5a6, 0x06900e90, 0x0e900690, 0xffffefde, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xcc63d6ba, 0x069006b1, 0x06900690,
                0x06900690, 0x06900690, 0x06900690, 0xe7dd0670, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xdc60ee0e,
                0xd460dc60, 0xeda6e4e2, 0xf586f5a6, 0xe5a7f5a6, 0x06900e90, 0xc79b0690,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0x958dffff,
                0x069006b0, 0x06900690, 0x06900690, 0x06900690, 0x06900690, 0x46d40690,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xee94ffff, 0xdc40d440, 0xd440dc40, 0xe481d440, 0xed85f586, 0xe586f585,
                0x066f0e90, 0xffff66f6, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xdf5cffff, 0x46529615, 0x068f0690, 0x0690068f, 0x06900690,
                0x06900690, 0x066f0670, 0xffff7f17, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xd441f7bd, 0xd420dc20, 0xdc20d420, 0xd420d420,
                0xed65dc61, 0xe586ed85, 0x2eb20e90, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xc5f8debb, 0xa616c5f7,
                0x4e528635, 0x066f0e70, 0x068f068f, 0x0690068f, 0xd7bd0690, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xdc20d463, 0xd420d420,
                0xd420d420, 0xd420d420, 0xd420d420, 0xe586ed44, 0xd7bc0e6f, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xc617ffdf, 0xbdf7bdf7, 0xcdf7bdf7, 0xa616cdf8, 0x36526e34, 0x066f1670,
                0x1e90068f, 0xfffff7ff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xe5ceffff,
                0xd420d420, 0xd400d400, 0xd400d400, 0xd400d400, 0xd400d400, 0xe545d400,
                0xffffb77a, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xef7effff, 0xbdf7bdf7, 0xbdf7bdf7, 0xc5f7bdf7,
                0xbdf7bdf7, 0xb617c5f7, 0x1e716e34, 0xffff6ef5, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xf77cffff, 0xf79cf79c, 0xf79cf79c, 0xf79cf79c, 0xf79cf79c,
                0xf79cf79c, 0xf79cf79c, 0xfffff7ff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xd442dd8e,
                0xd442d442, 0xd442d442, 0xd443d442, 0xf7bfef7c, 0xffbff7bf, 0xffffefde,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xd422ffdf, 0xd400d3e0, 0xcc00d400, 0xe632d3e0, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffdfffdf, 0xffffffdf, 0xffdfffdf,
                0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf,
                0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xef3affdf, 0xd3e0d3e0, 0xd3e0d400,
                0xffffdd4b, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffdfffdf,
                0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0xf7bef7be, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e,
                0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf7bef79e,
                0xd3e0dd6d, 0xcbe1d3e0, 0xf7bef79e, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffffffdf, 0xffffffdf, 0xffdfffdf,
                0xf7beffdf, 0xf7bef7be, 0xf7bef7be, 0xf79ef7be, 0xf79ef79e, 0xf79ef79e,
                0xef7df79e, 0xef7def7d, 0xef7def7d, 0xef5def5d, 0xef5def5d, 0xef5def5d,
                0xef5def5d, 0xef5def5d, 0xef5def5d, 0xef5def5d, 0xef5def5d, 0xef5def5d,
                0xef5def5d, 0xef5def5d, 0xd465ef7e, 0xe6d9d3e0, 0xef7df77d, 0xef7def7d,
                0xf79ef79e, 0xf79ef79e, 0xf7bef79e, 0xf7bef7be, 0xf7bef7be, 0xffdff7be,
                0xffdfffdf, 0xffdfffdf, 0xf7beffdf, 0xf7bef7be, 0xf7bef7be, 0xf79ef7be,
                0xf79ef79e, 0xf79ef79e, 0xef7def7d, 0xef7def7d, 0xef7def7d, 0xef5def5d,
                0xef5def5d, 0xef5def5d, 0xef5def5d, 0xe73cef5d, 0xe73ce73c, 0xe73ce73c,
                0xe73ce73c, 0xef5def5d, 0xef5def5d, 0xef5def5d, 0xe697ef5d, 0xef5edd4c,
                0xef7def7d, 0xef7def7d, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf7bef7be,
                0xf7bef7be, 0xffdff7be, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf,
                0xffdfffdf, 0xffdfffdf, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0xf79ef7be, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e,
                0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e, 0xf79ef79e,
                0xf79ef79e, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be, 0xf7bef7be,
                0xffdff7be, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffdfffff, 0xffdfffdf, 0xffffffdf,
                0xffffffff, 0xffdfffff, 0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffffffff,
                0xffdfffdf, 0xffdfffdf, 0xffdfffdf, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
                0xffffffff};


#endif /* SPRITE_H_ */
```