```c
/**
 * Copyright (c) 2020 Raspberry Pi (Trading) Ltd.
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */

#include <stdio.h>
#include <stdlib.h>

#include "pico/stdlib.h"
#include "hardware/pio.h"
#include "hardware/clocks.h"
#include "ws2812.pio.h"

#define IS_RGBW true
#define NUM_PIXELS 150

#ifdef PICO_DEFAULT_WS2812_PIN
#define WS2812_PIN PICO_DEFAULT_WS2812_PIN
#else
// default to pin 2 if the board doesn't have a default WS2812 pin defined
#define WS2812_PIN 2
#endif

static inline void put_pixel(uint32_t pixel_grb) {
    pio_sm_put_blocking(pio0, 0, pixel_grb << 8u);
}

static inline uint32_t urgb_u32(uint8_t r, uint8_t g, uint8_t b) {
    return
            ((uint32_t) (r) << 8) |
            ((uint32_t) (g) << 16) |
            (uint32_t) (b);
}

void pattern_snakes(uint len, uint t) {
    for (uint i = 0; i < len; ++i) {
        uint x = (i + (t >> 1)) % 64;
        if (x < 10)
            put_pixel(urgb_u32(0xff, 0, 0));
        else if (x >= 15 && x < 25)
            put_pixel(urgb_u32(0, 0xff, 0));
        else if (x >= 30 && x < 40)
            put_pixel(urgb_u32(0, 0, 0xff));
        else
            put_pixel(0);
    }
}
```

Handwritten annotations:

- standard C lib.
- standard C from Pico
- chip lib.
- chip lib.
- Autogenerated lib from Pioasm.
- Signal pin (default)
- We define pin2 (default).
- (3i) from each LED function.
- put GRB to OSR (blocking mode)
- (32) used in pattern-snakes.
- left shift 8 bit
- left shift 16 bit
- ⇒ result: change RGB to GRB by changing bits
- (30)
- Create a LED function called Snake
- (31) (32) (31) (31) (51)

```c
void pattern_random(uint len, uint t) {
    if (t % 8)
        return;
    for (int i = 0; i < len; ++i)
        put_pixel(rand());
}

void pattern_sparkle(uint len, uint t) {
    if (t % 8)
        return;
    for (int i = 0; i < len; ++i)
        put_pixel(rand() % 16 ? 0 : 0xffffffff);
}

void pattern_greys(uint len, uint t) {
    int max = 100; // let's not draw too much current!
    t %= max;
    for (int i = 0; i < len; ++i) {
        put_pixel(t * 0x10101);
        if (++t >= max) t = 0;
    }
}

typedef void (*pattern)(uint len, uint t);
const struct {
    pattern pat;
    const char *name;
} pattern_table[] = {
        {pattern_snakes,  "Snakes!"},
        {pattern_random,  "Random data"},
        {pattern_sparkle, "Sparkles"},
        {pattern_greys,   "Greys"},
};

int main() {
    //set_sys_clock_48();
    stdio_init_all();
    printf("WS2812 Smoke Test, using pin %d", WS2812_PIN);

    // todo get free sm
    PIO pio = pio0;
    int sm = 0;
    uint offset = pio_add_program(pio, &ws2812_program);

    ws2812_program_init(pio, sm, offset, WS2812_PIN, 800000, IS_RGBW);

    int t = 0;
```

Handwritten annotations:
- "is include in table." (near pattern_random)
- "Create a LED function called random"
- ㉛ (put_pixel(rand());)
- ㉚ "is include in table."
- "Create a LED function called Sparkle."
- ㉛ (put_pixel line in sparkle)
- ㉚ "is included in table."
- "Create a LED function called greys."
- ㉛ (put_pixel(t * 0x10101);)
- "a pointer function pointing to LED functions above."
- ㉙ (pattern_table[])
- "from the main function"
- ㉚ (pattern_sparkle line)
- "LED functions table. Including all four LED functions above."
- ① "Initialize the board." (stdio_init_all())
- ② "print pin #" (printf line)
- ③ "Initialize PIO module # (used for sending color to WS2812" (PIO pio = pio0)
- ④ "Initialize sm (State machine)" (int sm = 0)
- ⑤ "Initialize PiO" (uint offset line)
- ⑥ "Initialize WS2812 program." (ws2812_program_init line)
- ㉒ (int t = 0)

```
while (1) {
    int pat = rand() % count_of(pattern_table);
    int dir = (rand() >> 30) & 1 ? 1 : -1;
    puts(pattern_table[pat].name);
    puts(dir == 1 ? "(forward)" : "(backward)");
    for (int i = 0; i < 1000; ++i) {
        pattern_table[pat].pat(NUM_PIXELS, t);
        sleep_ms(10);
        t += dir;
    }
}
}
```

Annotations:
- 23 while (1) { — while true.
- 24 — randomly choose one function
- 25
- 26 — print mode info.
- 27
- 28
- 29 — choose one. LED functions
- 35
- 30 — t = t + dir