```
#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

#define WS2812_PIN 2
#endif

static inline void put_pixel(uint32_t pixel_grb) { 33
    pio_sm_put_blocking(pio0, 0, pixel_grb << 8u);
}

static inline uint32_t urgb_u32(uint8_t r, uint8_t g, uint8_t b) { 34
    return
            ((uint32_t) (r) << 8) |
            ((uint32_t) (g) << 16) |
            (uint32_t) (b);
}

void pattern_snakes(uint len, uint t) { 32
    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);
    }
}

void pattern_random(uint len, uint t) { 32
```

批注 [顾1]: default to pin 2 if the board doesn't have a default WS2812 pin defined

批注 [顾2]: a numeric type that guarantees 32 bits

批注 [顾3]: wait until there is room in the FIFO before pushing data.

批注 [顾4]: amalgamate the red, blue, and green values into a single 32-bit string

批注 [顾5]: call put_pixel helper to output a sequence of snakes-pattern pixel values

```
    if (t % 8)
        return;
    for (int i = 0; i < len; ++i)
        put_pixel(rand());
}

void pattern_sparkle(uint len, uint t) { 32
    if (t % 8)
        return;
    for (int i = 0; i < len; ++i)
        put_pixel(rand() % 16 ? 0 : 0xffffffff);
}

void pattern_greys(uint len, uint t) { 32
    int max = 100;
    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[] = { 24 27 31
        {pattern_snakes,    "Snakes!"},
        {pattern_random,    "Random data"},
        {pattern_sparkle, "Sparkles"},
        {pattern_greys,     "Greys"},
};

int main() {
    //set_sys_clock_48();
    stdio_init_all(); 1
    printf("WS2812 Smoke Test, using pin %d", WS2812_PIN); 2

    // todo get free sm
    PIO pio = pio0; 3
    int sm = 0; 4
    uint offset = pio_add_program(pio, &ws2812_program); 5

    ws2812_program_init(pio, sm, offset, WS2812_PIN, 800000, IS_RGBW); 6
```

批注 [顾6]: call our put_pixel helper to output a sequence of random pixel values

批注 [顾7]: call put_pixel helper to output a sequence of sparkle-pattern pixel values

批注 [顾8]: let's not draw too much current!

批注 [顾9]: call put_pixel helper to output a sequence of greys-pattern pixel values

批注 [顾10]: define the type of the function and member functions

```
    int t = 0; 21
    while (1) { 22
        int pat = rand() % count_of(pattern_table); 23
        int dir = (rand() >> 30) & 1 ? 1 : -1; 25
        puts(pattern_table[pat].name); 26
        puts(dir == 1 ? "(forward)" : "(backward)"); 28
        for (int i = 0; i < 1000; ++i) { 29
            pattern_table[pat].pat(NUM_PIXELS, t); 30
            sleep_ms(10);
            t += dir;
        }
    }
}
```