

3. 3

WS 2812.C Page

ws2812.c 



...



592 Textbook

CIT 5930 NOTES

草稿纸



592 Textbook



CIT 5930 NOTES



WS2812.C Page 2

```

ws2812.c
48
49
50 void pattern_random(uint len, uint t) {
51     if (t % 8)
52         return;
53     for (int i = 0; i < len; ++i)
54         put_pixel(rand());
55 }

56
57 void pattern_sparkle(uint len, uint t) {
58     if (t % 8)
59         return;
60     for (int i = 0; i < len; ++i)
61         put_pixel(rand() % 16 ? 0 : 0xffffffff);
62 }

63
64 void pattern_greys(uint len, uint t) {
65     int max = 100; // let's not draw too much current!
66     t %= max; t = t % max / 67
67     for (int i = 0; i < len; ++i) {
68         put_pixel(t * 0x10101);
69         if (++t >= max) t = 0;
70     }
71 }

72
73 typedef void (*pattern)(uint len, uint t);
74 const struct {
75     pattern pat;
76     const char *name;
77 } pattern_table[] = {
78     {pattern_snakes, "Snakes!"},
79     {pattern_random, "Random data"},
80     {pattern_sparkle, "Sparkles"},
81     {pattern_greys, "Greys"},
82 };
83
84 int main() {
85     //set_sys_clock_48();
86     stdio_init_all(); ①
87     printf("WS2812 Smoke Test, using pin %d", WS2812_PIN); ②
88
89     // todo get free sm
90     PIO pio = pio0; ③
91     int sm = 0; ④
92     uint offset = pio_add_program(pio, &ws2812_program); ⑤ attempt to load program
93     // PIO uint memory uint freq bool
94     ws2812_program_init(pio, sm, offset, WS2812_PIN, 800000, IS_RGBW); ⑥

```

$\frac{1}{8}$ of the times, randomly glow LED with random colors for len period of time.

$\frac{1}{8}$ of the times we glow the LED.

$\frac{1}{16}$ of the times glow white

$\frac{15}{16}$ of the times glow black

Color becomes darker until $t=100$. Then glow white, then color loops again.

Construct all patterns into table.

①②③④⑤⑥

WS 2812. C

```
ws2812.c x

52
53
54
55
56
57
58
59
60
61
62
63
64 void pattern_greys(uint len, uint t) {
65     int max = 100; // let's not draw too much current!
66     t %= max;
67     for (int i = 0; i < len; ++i) {
68         put_pixel(t * 0x10101);
69         if (++t >= max) t = 0;
70     }
71 }
72
73 typedef void (*pattern)(uint len, uint t);
74 const struct {
75     pattern pat;
76     const char *name;
77 } pattern_table[] = {
78     {pattern_snakes, "Snakes!"},
79     {pattern_random, "Random data"},
80     {pattern_sparkle, "Sparkles"},
81     {pattern_greys, "Greys"},
82 };
83
84 int main() {
85     //set_sys_clock_48();
86     stdio_init_all();
87     printf("WS2812 Smoke Test, using pin %d", WS2812_PIN);
88
89     // todo get free sm
90     PIO pio = pio0;
91     int sm = 0;
92     uint offset = pio_add_program(pio, &ws2812_program);
93
94     ws2812_program_init(pio, sm, offset, WS2812_PIN, 800000, TS_RGRW);
95
96     int t = 0; (21)
97     while (1) {
98         int pat = rand() % count_of(pattern_table);
99         int dir = (rand() >> 30) & 1 ? 1 : -1;
100        puts(pattern_table[pat].name);
101        puts(dir == 1 ? "(forward)" : "(backward)");
102        for (int i = 0; i < 1000; ++i) {
103            pattern_table[pat].pat(NUM_PIXELS, t);
104            sleep_ms(10);
105            t += dir;
106        }
107    }
108 }
```

Diagram illustrating the execution flow of the main loop:

- The main loop starts at line 97 with `while (1) {`.
- Inside the loop, it selects a pattern at line 98 using `int pat = rand() % count_of(pattern_table);`.
- It then determines the direction of the pattern at line 99 using `int dir = (rand() >> 30) & 1 ? 1 : -1;`.
- It prints the selected pattern name and direction at lines 100-101.
- It enters a for loop at line 102, which runs 1000 times.
- Inside the for loop, it calls the pattern's `pat` method at line 103, passing `NUM_PIXELS` and the current time `t`.
- After each iteration of the for loop, it sleeps for 10ms at line 104.
- Finally, it updates the time `t` by adding the direction at line 105.
- The loop then repeats from line 97.

Annotations:

- Annotation (21) is a circle around the variable `t` at line 96.
- Annotation (22) is a circle around the closing brace of the for loop at line 106.
- A large curly brace on the right side of the code is labeled "Instruction" above "PIO mode", indicating the context of the code.

↓ back

22
Instruction to
PI0 module

```

1 // -----
2 // This file is autogenerated by pioasm; do not edit! //
3 // -----
4
5 #pragma once
6
7 #if !PICO_NO_HARDWARE
8 #include "hardware/pio.h"
9 #endif
10
11 // -----
12 // ws2812 //
13 // -----
14
15 #define ws2812_wrap_target 0
16 #define ws2812_wrap 3
17
18 #define ws2812_T1 2
19 #define ws2812_T2 5
20 #define ws2812_T3 3
21
22 static const uint16_t ws2812_program_instructions[] = {
23     | | // .wrap_target
24     0x6221, // 0: out    x, 1           side 0 [2]
25     0x1123, // 1: jmp    !x, 3         side 1 [1]
26     0x1400, // 2: jmp    0            side 1 [4]
27     0xa442, // 3: nop           side 0 [4]
28     | | // .wrap
29 };
30
31 #if !PICO_NO_HARDWARE
32 static const struct pio_program ws2812_program = {
33     .instructions = ws2812_program_instructions,
34     .length = 4,
35     .origin = -1,
36 };
37
38 static inline pio_sm_config ws2812_program_get_default_config(uint offset) {
39     pio_sm_config c = pio_get_default_sm_config(); //9
40     sm_config_set_wrap(&c, offset + ws2812_wrap_target, offset + ws2812_wrap); //10  set wrap address
41     sm_config_set_sideset(&c, 1, false, false); // 11 set the 'sideset' options
42     return c;
43 }
44
45 #include "hardware/clocks.h"

```

```

45 #include "hardware/clocks.h"
46 static inline void ws2812_program_init(PIO pio, uint sm, uint offset, uint pin, float freq, bool rgbw) {
47     pio_gpio_init(pio, pin); //7
48     pio_sm_set_consecutive_pindirs(pio, sm, pin, 1, true); //set up pin directions 8
49     pio_sm_config c = ws2812_program_get_default_config(offset); // jump to 9
50     sm_config_set_sideset_pins(&c, pin); //12
51     sm_config_set_out_shift(&c, false, true, rgbw ? 32 : 24); // 13 set out shift parameter
52     sm_config_set_fifo_join(&c, PIO_FIFO_JOIN_TX); // 14
53     int cycles_per_bit = ws2812_T1 + ws2812_T2 + ws2812_T3; //15
54     float div = clock_get_hz(clk_sys) / (freq * cycles_per_bit); //16
55     sm_config_set_clkdiv(&c, div); // 17
56     pio_sm_init(pio, sm, offset, &c); //reset state machine 18
57     pio_sm_set_enabled(pio, sm, true); // 19 enable the state machine 19
58 }
59
60 #endif
61
62 // -----
63 // ws2812_parallel //
64 // -----
65
66 #define ws2812_parallel_wrap_target 0
67 #define ws2812_parallel_wrap 3
68
69 #define ws2812_parallel_T1 2
70 #define ws2812_parallel_T2 5
71 #define ws2812_parallel_T3 3
72
73 static const uint16_t ws2812_parallel_program_instructions[] = {
74     | | // .wrap_target
75     0x6020, // 0: out    x, 32
76     0xa10b, // 1: mov    pins, !null          [1]
77     0xa401, // 2: mov    pins, x             [4]
78     0xa103, // 3: mov    pins, null          [1]
79     | | // .wrap
80 };
81
82 #if !PICO_NO_HARDWARE
83 static const struct pio_program ws2812_parallel_program = {
84     .instructions = ws2812_parallel_program_instructions,
85     .length = 4,
86     .origin = -1,
87 };
88

```