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

#include "pico/stdlib.h"
#include "hardware/pio.h"
#include "hardware/clocks.h"
#include "ws2812.pio.h"
#include "adafruit_qtpy_rp2040.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
#define PICO_DEFAULT_WS2812_POWER_PIN 11

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
```

→ Include libraries

→ Assignment of Pins

→ Put_Pixel function definition

→ Defining the wait for FIFO Pull

```c
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);
    }
}

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);
}
```

Calling the "Put_Pixel()" function with RGB values as Parameters.

→ Pull-down of LED (Value to ground)

→ Defining a "Pattern_random()" function to Set LED to random Colour Pattern

→ The `rand()` function sets the `Put_Pixel` RGB colour to a random value.

```
72
73  void
74  pattern_greys (uint len, uint t)
75  {
76    int max = 100;         // let's not draw too much current!
77    t %= max;
78    for (int i = 0; i < len; ++i)
79      {
80        put_pixel (t * 0x10101);
81        if (++t >= max)
82      t = 0;
83      }
84  }
85
86  typedef void (*pattern) (uint len, uint t);
87  const struct
88  {
89    pattern pat;
90    const char *name;
91  } pattern_table[] =
92  {
93    {
94    pattern_snakes, "Snakes!"},
95    {
96    pattern_random, "Random data"},
97    {
98    pattern_sparkle, "Sparkles"},
99    {
100 pattern_greys, "Greys"},};
101
102 int
103 main ()
104 {
105   //set_sys_clock_48();
106   const uint POW_PIN = PICO_DEFAULT_WS2812_POWER_PIN;
107   gpio_init (POW_PIN);
108   gpio_set_dir (POW_PIN, GPIO_OUT);
109   gpio_put (POW_PIN, 1);
110   stdio_init_all ();
```

→ 4th Pattern

→ Defining (creating) a table with all the "4" above defined Patterns.

→ Initializing the Pow_Pin & defining direction.

```
110   stdio_init_all ();
111   printf ("WS2812 Smoke Test, using pin %d", WS2812_PIN);
112
113   // todo get free sm
114   PIO pio = pio0;
115   int sm = 0;
116   uint offset = pio_add_program (pio, &ws2812_program);
117
118   ws2812_program_init (pio, sm, offset, WS2812_PIN, 800000, IS_RGBW);
119
120   int t = 0;
121   while (1)
122     {
123       int pat = rand() % count_of(pattern_table);
124       int dir = (rand() >> 30) & 1 ? 1 : -1;
125       puts(pattern_table[pat].name);
126       puts(dir == 1 ? "forward") : "(backward)");
127       for (int i = 0; i < 1000; ++i) {
128           pattern_table[pat].pat(NUM_PIXELS, t);
129           sleep_ms(10);
130           t+ = dir;
131       }
132
133
134     }
135 }
136
```

→ Printf ("WS2812 Smoke Test, using Pin%d", WS2812_PIN); → Prints the Pin in action.

→ selecting a random Pattern from the table

→ setting the direction of Pattern