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

#include <stdio.h>            ]   import C library
#include <stdlib.h>           ]

#include "pico/stdlib.h"          Pico C library
#include "hardware/pio.h"          Chip library
#include "hardware/clocks.h"       Chip library
#include "ws2812.pio.h"

#define IS_RGBW true
#define NUM_PIXELS 150

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

static inline void put_pixel(uint32_t pixel_grb) {      writes a colored pixel to Fifo
    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) |       Shift left 8 bits
            ((uint32_t) (g) << 16) |      shift left 16 bits
            (uint32_t) (b);                  Changing RGB to GRB
}

void pattern_snakes(uint len, uint t) {       Creating Led pattern function
    for (uint i = 0; i < len; ++i) {
        uint x = (i + (t >> 1)) % 64;
        if (x < 10)
            put_pixel(urgb_u32(0xff, 0, 0));     Turns Red to full Brightness
        else if (x >= 15 && x < 25)
            put_pixel(urgb_u32(0, 0xff, 0));     Turns Green to full Brightness
        else if (x >= 30 && x < 40)
            put_pixel(urgb_u32(0, 0, 0xff));     Turns Blue to full Brightness
        else
            put_pixel(0);
    }
}

void pattern_random(uint len, uint t) {       Create another LED pattern Random
    if (t % 8)
        return;
    for (int i = 0; i < len; ++i)
        put_pixel(rand());            Turn on any random color with random
}                                          brightness
```

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

Create another LED pattern Sparkle

Turn on random color after comparing

```c
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;
    }
}
```

LED pattern grey created

stay brightness at a lowest

```c
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"},
};
```

Pointer to the functions

Function table of above all the patterns

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

Initialize it

Print pin number

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

Initialize module number

initialize sm.

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

Initialize pio

```c
    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;
        }
    }
}
```

chose one pattern function

print the choosen function name

Go to choosen function and [illegible]