Resumo do projeto.

Meu nome é Victor Sarrís Silva Santos, estudante do segundo módulo de TADS no Instituto Federal Campus Floriano. Com os conhecimentos adquiridos nas aulas sobre a placa BitDogLab coloquei em prática um projeto simples: A placa toca a abertura tema do Amazing Spiderman de 1990, seriado que passava na televisão daquela época - enquanto exibe seu logotipo no seu display.

De fato, um projeto simples e de baixo calibre. Mas nele conseguimos explorar algumas das variadas funcionalidades da placa BitDogLab.

Código em C do projeto

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include "pico/stdlib.h"
#include "pico/binary_info.h"
#include "inc/ssd1306.h"
#include "hardware/i2c.h"
#include "hardware/pwm.h"
#include "hardware/clocks.h"

#define BUZZER_PIN 21

const uint I2C_SDA = 14;
const uint I2C_SCL = 15;

// Frequências das notas musicais (em Hz)
const uint NOTE_C4 = 262;
const uint NOTE_D4 = 294;
const uint NOTE_E4 = 330;
const uint NOTE_F4 = 349;
const uint NOTE_G4 = 392;
const uint NOTE_A4 = 440;
const uint NOTE_B4 = 494;
```

```
const uint NOTE_C5 = 523;
const uint NOTE_D5 = 587;
const uint NOTE_E5 = 659;
const uint NOTE_F5 = 698;
const uint NOTE_G5 = 784;
const uint NOTE_A5 = 880;
const uint NOTE_B5 = 988;

// Duração das notas (em milissegundos)
const int TEMPO = 200; // Ajuste o tempo para acelerar ou desacelerar a música

// Inicializa o PWM no buzzer
void pwm_init_buzzer(uint pin) {
    gpio_set_function(pin, GPIO_FUNC_PWM);
    uint slice_num = pwm_gpio_to_slice_num(pin);
    pwm_config config = pwm_get_default_config();
    pwm_config_set_clkdiv(&config, 2.5f);
    pwm_init(slice_num, &config, true);
    pwm_set_gpio_level(pin, 0);
}

// Toca uma nota no buzzer
void play_tone(uint pin, uint frequency, uint duration_ms) {
    if (frequency == 0) {
        sleep_ms(duration_ms); // Pausa (nota silenciosa)
        return;
    }

    uint slice_num = pwm_gpio_to_slice_num(pin);
    uint32_t clock_freq = clock_get_hz(clk_sys);
    uint32_t top = clock_freq / frequency - 1;

    pwm_set_wrap(slice_num, top);
    pwm_set_gpio_level(pin, top * 0.5);

    sleep_ms(duration_ms);

    pwm_set_gpio_level(pin, 0);
    sleep_ms(50); // Pequena pausa entre as notas
}

// Toca a música de abertura do Homem-Aranha clássico
void play_spiderman_theme() {
    // Melodia da música
```

```
    int melody[] = {
        NOTE_G4, NOTE_C5, NOTE_E5, NOTE_G5, NOTE_E5, NOTE_C5,
        NOTE_G4, NOTE_C5, NOTE_E5, NOTE_G5, NOTE_E5, NOTE_C5,
        NOTE_G4, NOTE_C5, NOTE_E5, NOTE_G5, NOTE_E5, NOTE_C5,
        NOTE_G4, NOTE_C5, NOTE_E5, NOTE_G5, NOTE_E5, NOTE_C5,
        NOTE_A4, NOTE_D5, NOTE_F5, NOTE_A5, NOTE_F5, NOTE_D5,
        NOTE_A4, NOTE_D5, NOTE_F5, NOTE_A5, NOTE_F5, NOTE_D5,
        NOTE_G4, NOTE_C5, NOTE_E5, NOTE_G5, NOTE_E5, NOTE_C5,
        NOTE_G4, NOTE_C5, NOTE_E5, NOTE_G5, NOTE_E5, NOTE_C5
    };

    // Duração de cada nota (4 = quarto de tempo, 8 = oitavo de tempo, etc.)
    int noteDurations[] = {
        8, 8, 8, 8, 8, 8,
        8, 8, 8, 8, 8, 8,
        8, 8, 8, 8, 8, 8,
        8, 8, 8, 8, 8, 8,
        8, 8, 8, 8, 8, 8,
        8, 8, 8, 8, 8, 8,
        8, 8, 8, 8, 8, 8,
        8, 8, 8, 8, 8, 8
    };

    // Toca a melodia
    for (int i = 0; i < sizeof(melody) / sizeof(int); i++) {
        int noteDuration = TEMPO / noteDurations[i];
        play_tone(BUZZER_PIN, melody[i], noteDuration);
        sleep_ms(noteDuration * 0.3); // Pausa entre as notas
    }
}

int main() {
    stdio_init_all();
    pwm_init_buzzer(BUZZER_PIN);

    // Inicialização do i2c
    i2c_init(i2c1, ssd1306_i2c_clock * 1000);
    gpio_set_function(I2C_SDA, GPIO_FUNC_I2C);
    gpio_set_function(I2C_SCL, GPIO_FUNC_I2C);
    gpio_pull_up(I2C_SDA);
    gpio_pull_up(I2C_SCL);

    // Inicialização do display OLED SSD1306
    ssd1306_init();
```

```
// Configuração da tela
struct render_area frame_area = {
    .start_column = 0,
    .end_column = ssd1306_width - 1,
    .start_page = 0,
    .end_page = ssd1306_n_pages - 1
};
calculate_render_area_buffer_length(&frame_area);

uint8_t ssd[ssd1306_buffer_length];
memset(ssd, 0, ssd1306_buffer_length);
render_on_display(ssd, &frame_area);

char *text[] = {
    "  Bem-vindos!   ",
    "  Embarcatech   "};

int y = 0;
for (uint i = 0; i < count_of(text); i++) {
    ssd1306_draw_string(ssd, 5, y, text[i]);
    y += 8;
}
render_on_display(ssd, &frame_area);

// Exibir o bitmap no display
const uint8_t bitmap_128x64[] = {
    // ... (bitmap data)
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0xff,
      0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x1f, 0xfb, 0xff,
      0xff, 0xff, 0xff, 0xff, 0xff, 0x8f, 0xfc, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xa7, 0xfc, 0xff,
      0xff, 0xff, 0xff, 0xff, 0xff, 0x13, 0xfc, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x09, 0xfe, 0xff,
      0xff, 0xff, 0xff, 0xff, 0xff, 0x0e, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0x07, 0xff, 0xff,
      0xff, 0xff, 0xff, 0xff, 0xbf, 0x07, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xff, 0xdf, 0x83, 0xff, 0xff,
      0xff, 0xff, 0x1f, 0xfc, 0xef, 0x83, 0xff, 0xff, 0xff, 0xff, 0xcf, 0xe0, 0xe7, 0xc1, 0xff, 0xff,
      0xff, 0xff, 0xef, 0xe7, 0xf3, 0xc0, 0xff, 0xff, 0xff, 0xff, 0xef, 0xdf, 0x79, 0xe0, 0xff, 0xff,
      0xff, 0xff, 0xf7, 0x3f, 0x7c, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xf7, 0xff, 0x3f, 0xf8, 0xff, 0xff,
      0xff, 0xff, 0xfb, 0xf8, 0x1f, 0xfc, 0xff, 0xff, 0xff, 0xff, 0x7b, 0xf0, 0x07, 0x7e, 0xfc, 0xff,
      0xff, 0xff, 0x3d, 0xc0, 0x03, 0x1f, 0xfe, 0xff, 0xff, 0xff, 0x3d, 0x18, 0x80, 0x01, 0xf1, 0xff,
      0xff, 0xff, 0x3c, 0x38, 0x00, 0x40, 0xf9, 0xff, 0xff, 0x7f, 0x7e, 0xf0, 0x00, 0x4e, 0xf8, 0xff,
      0xff, 0xff, 0x7e, 0xe0, 0xe0, 0x03, 0xfc, 0xff, 0xff, 0xff, 0x4e, 0x00, 0xff, 0x03, 0xff, 0xff,
      0xff, 0xff, 0x0d, 0xe0, 0xff, 0x81, 0xff, 0xff, 0xff, 0xff, 0x31, 0xfc, 0x3f, 0xc0, 0xff, 0xff,
      0xff, 0xff, 0xfb, 0xff, 0x03, 0xe0, 0xff, 0xff, 0xff, 0xff, 0xfb, 0xdf, 0x03, 0xf8, 0xff, 0xff,
      0xff, 0xff, 0xfb, 0xe1, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7b, 0xe0, 0xc1, 0xff, 0xff, 0xff,
```

0xff, 0xff, 0x77, 0xe0, 0xe1, 0xff, 0xff, 0xff, 0xff, 0xff, 0x77, 0xe0, 0xe0, 0xff, 0xff, 0xff,
0xff, 0xff, 0xf7, 0xf0, 0xe0, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf7, 0xff, 0xe0, 0xff, 0xff, 0xff,
0xff, 0xff, 0xf7, 0x7f, 0xf0, 0xff, 0xff, 0xff, 0xff, 0xff, 0xef, 0x0f, 0x00, 0x80, 0xff, 0xff,
0xff, 0xff, 0x0f, 0x00, 0xf0, 0x1f, 0xfc, 0xff, 0xff, 0xff, 0x2f, 0xf8, 0xff, 0xff, 0xf3, 0xff,
0xff, 0xff, 0x6f, 0xfc, 0xff, 0x3f, 0xe0, 0xff, 0xff, 0xff, 0xef, 0xff, 0x01, 0x00, 0xc0, 0xff,
0xff, 0xff, 0xef, 0x03, 0x00, 0x00, 0x80, 0xff, 0xff, 0xff, 0x0f, 0x00, 0x00, 0x00, 0x00,
0xff,
0xff, 0xff, 0x1f, 0x00, 0x00, 0x00, 0xe0, 0xff, 0xff, 0xff, 0xdf, 0x80, 0xff, 0xff, 0xf7, 0xff,
0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0x03, 0xff, 0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0x83, 0xff,
0xff, 0xff, 0xdf, 0x01, 0x00, 0xe0, 0xc1, 0xff, 0xff, 0xff, 0xdf, 0x01, 0x00, 0xf8, 0xc0, 0xff,
0xff, 0xff, 0xdf, 0x01, 0x00, 0x7c, 0xe0, 0xff, 0xff, 0xff, 0xbf, 0x03, 0x00, 0x3e, 0xf0, 0xff,
0xff, 0xff, 0x3f, 0xff, 0xff, 0x0f, 0xf8, 0xff, 0xff, 0xff, 0x3f, 0xff, 0xff, 0x07, 0xfc, 0xff,
0xff, 0xff, 0x1f, 0xfe, 0xff, 0x01, 0xfe, 0xff, 0xff, 0xff, 0x1f, 0x00, 0x00, 0x00, 0xff, 0xff,
0xff, 0xff, 0xdf, 0x00, 0x00, 0xc0, 0x0f, 0xff, 0xff, 0xff, 0xdf, 0xff, 0x0f, 0x00, 0x20, 0xff,
0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0x3f, 0xfc, 0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0x1f, 0xfc,
0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0x1f, 0xfc, 0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0x0f, 0xfc,
0xff, 0xff, 0xdf, 0x03, 0x3c, 0x00, 0x0f, 0xfe, 0xff, 0xff, 0xdf, 0x01, 0x38, 0x00, 0x07,
0xfe,
0xff, 0xff, 0xdf, 0x01, 0x38, 0x00, 0x07, 0xff, 0xff, 0xff, 0xdf, 0x01, 0x18, 0x00, 0x83,
0xff,
0xff, 0xff, 0x1f, 0xe0, 0x01, 0x7e, 0x81, 0xff, 0xff, 0xff, 0xdf, 0x01, 0x00, 0x7e, 0xc0, 0xff,
0xff, 0xff, 0xdf, 0xff, 0xff, 0x00, 0xf0, 0xff, 0xff, 0xff, 0xdf, 0xff, 0xff, 0x00, 0xf8, 0xff,
0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xf1, 0xff, 0xff, 0xff, 0xdf, 0x03, 0x3e, 0xfc, 0xe3, 0xff,
0xff, 0xff, 0xdf, 0x01, 0x3c, 0xf8, 0xc7, 0xff, 0xff, 0xff, 0xdf, 0x81, 0x7f, 0x00, 0x80, 0xff,
0xff, 0xff, 0xdf, 0xe1, 0xff, 0x03, 0x00, 0xff, 0xff, 0xff, 0xdf, 0xff, 0xe1, 0x0f, 0x00, 0xfe,
0xff, 0xff, 0xdf, 0xff, 0xc0, 0x7f, 0xff, 0xff, 0xff, 0xff, 0x3f, 0x7f, 0x0c, 0x7e, 0xfe, 0xff,
0xff, 0xff, 0x7f, 0x1c, 0x1c, 0xf0, 0xfc, 0xff, 0xff, 0xff, 0xdf, 0x00, 0x1c, 0x80, 0xf9, 0xff,
0xff, 0xff, 0x8f, 0x00, 0x0c, 0x00, 0xf8, 0xff, 0xff, 0xff, 0x6f, 0x00, 0x00, 0x0f, 0xe4, 0xff,
0xff, 0xff, 0xef, 0x81, 0x00, 0x3f, 0xc0, 0xff, 0xff, 0xff, 0xef, 0xff, 0xff, 0xe0, 0x81, 0xff,
0xff, 0xff, 0xcf, 0xff, 0xff, 0x8f, 0x0f, 0xff, 0xff, 0xff, 0x3f, 0xfe, 0xff, 0x7f, 0x7f, 0xfe,
0xff, 0xff, 0x3f, 0xf8, 0x03, 0xfe, 0xfc, 0xfd, 0xff, 0xff, 0xff, 0xe0, 0x07, 0x00, 0xf8, 0xff,
0xff, 0xff, 0xff, 0xe0, 0x1f, 0x00, 0xf0, 0xff, 0xff, 0xff, 0x7f, 0xf8, 0x7f, 0x00, 0xe0, 0xff,
0xff, 0xff, 0x8f, 0xff, 0x87, 0x00, 0xc0, 0xff, 0xff, 0xff, 0xf7, 0xff, 0x03, 0xf0, 0xbf, 0xff,
0xff, 0xff, 0xf7, 0xff, 0x3f, 0xe0, 0xff, 0xff, 0xff, 0xff, 0xf7, 0xff, 0xff, 0xc3, 0xff, 0xff,
0xff, 0xff, 0x07, 0xc0, 0xff, 0x1f, 0xff, 0xff, 0xff, 0xff, 0x07, 0x00, 0xf0, 0xff, 0xfc, 0xff,
0xff, 0xff, 0x3b, 0x00, 0x04, 0xc0, 0xf9, 0xff, 0xff, 0xff, 0x3b, 0x00, 0x0e, 0x00, 0xf0, 0xff,
0xff, 0xff, 0xfb, 0xff, 0x7f, 0x00, 0xe0, 0xff, 0xff, 0xff, 0xfb, 0xff, 0xff, 0x3f, 0xc0, 0xff,
0xff, 0xff, 0xfb, 0xf9, 0xdf, 0x7e, 0xc0, 0xff, 0xff, 0xff, 0xfb, 0x00, 0x1c, 0xfe, 0xf3, 0xff,
0xff, 0xff, 0xfb, 0x0f, 0x1c, 0x00, 0xc0, 0xff, 0xff, 0xff, 0xfb, 0x3f, 0x07, 0x00, 0x80, 0xff,
0xff, 0xff, 0xfb, 0xff, 0x07, 0x07, 0x00, 0xff, 0xff, 0xff, 0xc1, 0xff, 0x7f, 0xfc, 0xff, 0xff,
0xff, 0xff, 0x01, 0xf8, 0x7f, 0xf0, 0xff, 0xff, 0xff, 0xff, 0x01, 0xc0, 0xff, 0xc3, 0xff, 0xff,
0xff, 0xff, 0x0d, 0x00, 0x80, 0x8f, 0xff, 0xff, 0xff, 0x7f, 0x3e, 0x00, 0x80, 0x78, 0xff, 0xff,
0xff, 0xff, 0xfe, 0x0f, 0x00, 0xf0, 0xfe, 0xff, 0xff, 0xff, 0xfc, 0xff, 0xff, 0x80, 0xfd, 0xff,
0xff, 0xff, 0xf0, 0xff, 0xff, 0x07, 0xf8, 0xff, 0xff, 0xff, 0x02, 0xff, 0xe1, 0x0f, 0xf0, 0xff,

```
        0xff, 0x3f, 0x07, 0xfc, 0x03, 0x9f, 0xe1, 0xff, 0xff, 0x3f, 0x7f, 0xf0, 0x07, 0x70, 0xff, 0xff,
        0xff, 0x3f, 0x7f, 0xe0, 0x0f, 0x00, 0xfe, 0xff, 0xff, 0x3f, 0xff, 0xef, 0x3f, 0x00, 0xfe, 0xff,
        0xff, 0x1f, 0xc0, 0xff, 0xff, 0x06, 0xfc, 0xff, 0xff, 0x1f, 0x80, 0xff, 0xff, 0x3d, 0xf8, 0xff,
        0xff, 0xff, 0x01, 0xf8, 0xff, 0xf3, 0xff, 0xff, 0xff, 0xff, 0x01, 0x00, 0xff, 0xe7, 0xff, 0xff,
        0xff, 0xff, 0xff, 0x00, 0xe0, 0xcf, 0xff, 0xff, 0xff, 0xff, 0xff, 0x1f, 0x00, 0x80, 0xff, 0xff,
        0xff, 0xff, 0xff, 0xff, 0x03, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0x00, 0xfe, 0xff,
        0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff
    };

    ssd1306_t ssd_bm;
    ssd1306_init_bm(&ssd_bm, 128, 64, false, 0x3C, i2c1);
    ssd1306_config(&ssd_bm);

    ssd1306_draw_bitmap(&ssd_bm, bitmap_128x64);

    while (true) {
        play_spiderman_theme();  // Toca a música de abertura do Homem-Aranha
        sleep_ms(1000);          // Aguarda 1 segundo antes de tocar novamente
    }

    return 0;
}
```

Código do CMake List

```
# Generated Cmake Pico project file

cmake_minimum_required(VERSION 3.13)

set(CMAKE_C_STANDARD 11)
set(CMAKE_CXX_STANDARD 17)
set(CMAKE_EXPORT_COMPILE_COMMANDS ON)

# Initialise pico_sdk from installed location
# (note this can come from environment, CMake cache etc)

# == DO NOT EDIT THE FOLLOWING LINES for the Raspberry Pi Pico VS Code Extension to
work ==
if(WIN32)
```

```
    set(USERHOME $ENV{USERPROFILE})
else()
    set(USERHOME $ENV{HOME})
endif()
set(sdkVersion 1.5.1)
set(toolchainVersion 13_2_Rel1)
set(picotoolVersion 2.0.0)
set(picoVscode ${USERHOME}/.pico-sdk/cmake/pico-vscode.cmake)
if (EXISTS ${picoVscode})
    include(${picoVscode})
endif()
# ====================================================================================
set(PICO_BOARD pico_w CACHE STRING "Board type")

# Pull in Raspberry Pi Pico SDK (must be before project)
include(pico_sdk_import.cmake)

project(tt3 C CXX ASM)

# Initialise the Raspberry Pi Pico SDK
pico_sdk_init()

# Add executable. Default name is the project name, version 0.1

add_executable(tt3 tt3.c inc/ssd1306_i2c.c )

pico_set_program_name(tt3 "tt3")
pico_set_program_version(tt3 "0.1")

# Modify the below lines to enable/disable output over UART/USB
pico_enable_stdio_uart(tt3 0)
pico_enable_stdio_usb(tt3 0)

# Add the standard library to the build
target_link_libraries(tt3
        pico_stdlib)

# Add the standard include files to the build
target_include_directories(tt3 PRIVATE
        ${CMAKE_CURRENT_LIST_DIR}
)
pico_generate_pio_header(tt3 ${CMAKE_CURRENT_LIST_DIR}/ws2818b.pio)
```

```
# Add any user requested libraries
target_link_libraries(tt3

        pico_stdlib
        pico_binary_info
        hardware_i2c
        hardware_pwm
        hardware_clocks


        )

pico_add_extra_outputs(tt3)
```