

list6.txt

入力:ラズベリーパイPicoを使って外部に接続したフラッシュメモリW25Q32からデータを読み出すプログラムを作ってください。データは2バイトのバイナリ形式で入っています。C言語、Pico-SDKを使ってください。この2バイトのデータは、16ビット、16kHzサンプリング、モノラル(1チャンネル)の音声です。このデータをPWMに変換して出力してください。PWMの繰り返し周期は16kHzとしてください。

GPT: 以下は、ラズベリーパイ Pico で外部フラッシュメモリ W25Q32 からデータを読み取り、PWM で音声として出力するための基本的なコード例です。

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/spi.h"
#include "hardware/pwm.h"

// W25Q32のSPIコマンド
#define READ_DATA 0x03

// SPIの設定
#define SPI_PORT spi0
#define CS_PIN 5
#define SCK_PIN 2
#define MISO_PIN 3
#define MOSI_PIN 4

// PWMの設定
#define PWM_PIN 6
#define PWM_FREQ_HZ 16000

void setup_spi() {
    spi_init(SPI_PORT, 500 * 1000); // 500kHz
    gpio_set_function(SCK_PIN, GPIO_FUNC_SPI);
    gpio_set_function(MISO_PIN, GPIO_FUNC_SPI);
    gpio_set_function(MOSI_PIN, GPIO_FUNC_SPI);
    gpio_init(CS_PIN);
    gpio_set_dir(CS_PIN, GPIO_OUT);
    gpio_put(CS_PIN, 1); // デイセーブル
}

uint16_t read_flash_memory(uint32_t address) {
    uint8_t rx_buf[2];
    uint8_t tx_buf[4] = {
        READ_DATA,
        (address >> 16) & 0xFF,
        (address >> 8) & 0xFF,
        address & 0xFF
    };

    gpio_put(CS_PIN, 0); // イネーブル
    spi_write_read_blocking(SPI_PORT, tx_buf, rx_buf, sizeof(tx_buf) +
sizeof(rx_buf));
    gpio_put(CS_PIN, 1); // デイセーブル

    return ((uint16_t)rx_buf[0] << 8) | rx_buf[1];
}
```

list6.txt

```
void setup_pwm() {
    gpio_set_function(PWM_PIN, GPIO_FUNC_PWM);
    uint slice_num = pwm_gpio_to_slice_num(PWM_PIN);
    pwm_set_wrap(slice_num, 0xffff);
    pwm_set_clkdiv(slice_num, clock_get_hz(clk_sys) / (PWM_FREQ_HZ *
0x10000ull));
}

void set_pwm_duty_cycle(uint16_t sample) {
    uint slice_num = pwm_gpio_to_slice_num(PWM_PIN);
    pwm_set_chan_level(slice_num, PWM_CHAN_A, sample);
}

int main() {
    stdio_init_all();
    setup_spi();
    setup_pwm();

    uint32_t address = 0x000000;

    while (1) {
        uint16_t sample = read_flash_memory(address);
        set_pwm_duty_cycle(sample);
        sleep_us(1000000 / PWM_FREQ_HZ);
        address += 2;
    }

    return 0;
}
```

このコードは、W25Q32から音声データを読み取り、PWM出力として再生するための基本的な実装を提供しています。しかし、実際のハードウェア環境や他の要件によって、追加の設定や最適化が必要になるかもしれません。上記のコードを基に、実際の環境に合わせて適切に調整してください。