

list5.txt

入力: 次のプログラムをrasberry pi PICOで作ってください。ツールは、PICO-SDKを使い、C言語でプログラムしてください。デジタルGPIOは、HでHigh=3V, LでLow=0Vとします。ポート名称は、GP26(A0)がアナログ入力ピンSOLAR、GP10が入力ピンALW_L、GP11が入力ピン5MIN_L、GP13が出力ピンAMP_ONです。AMP_ONは通常Lです。アルゴリズムは、電源ON後、ALW_LがLか、または、ALW_LがHかつSOLAR>0.9VならばAMP_ONを20秒間Hにします。その後、5MIN_LがLならば、5分、5MIN_LがHならば30分待機(wait)します。その後、プログラムの先頭(AMP_ON動作)に戻ります。

GPT: 以下は、指定されたアルゴリズムをRaspberry Pi PICOで実行するためのC言語のプログラムです。

```
#include "pico/stdlib.h"
#include "hardware/gpio.h"
#include "hardware/adc.h"
```

// ピンの定義

```
#define PIN_SOLAR GP26_A0
#define PIN_ALW_L GP10
#define PIN_5MIN_L GP11
#define PIN_AMP_ON GP13
```

// タイムアウトの定義 (ミリ秒単位)

```
#define AMP_ON_DURATION 20000 // 20秒
#define SHORT_WAIT_DURATION 300000 // 5分
#define LONG_WAIT_DURATION 1800000 // 30分
```

// アナログ読み取りのためのしきい値

```
#define SOLAR_THRESHOLD (0.9 * 4095.0 / 3.3) // ADCは12bitなので、4095が最大値。0.9Vの値を計算
```

```
int main() {
    // セットアップ
    stdio_init_all();
    adc_init();
    adc_select_input(0); // A0 = 0
    gpio_init(PIN_ALW_L);
    gpio_set_dir(PIN_ALW_L, GPIO_IN);
    gpio_init(PIN_5MIN_L);
    gpio_set_dir(PIN_5MIN_L, GPIO_IN);
    gpio_init(PIN_AMP_ON);
    gpio_set_dir(PIN_AMP_ON, GPIO_OUT);
    gpio_put(PIN_AMP_ON, 0); // 初期状態をLOWに

    while (1) {
        // ALW_LがLか、ALW_LがHかつSOLAR>0.9Vの場合
        if (gpio_get(PIN_ALW_L) == 0 || (gpio_get(PIN_ALW_L) == 1 && adc_read()
> SOLAR_THRESHOLD)) {
            gpio_put(PIN_AMP_ON, 1); // AMP_ONをHに
            sleep_ms(AMP_ON_DURATION); // 20秒待つ
            gpio_put(PIN_AMP_ON, 0); // AMP_ONをLに
        }

        // 5MIN_LがLならば、5分、5MIN_LがHならば30分待機
        if (gpio_get(PIN_5MIN_L) == 0) {
            sleep_ms(SHORT_WAIT_DURATION); // 5分待つ
        } else {

```

```
list5.txt
sleep_ms(LONG_WAIT_DURATION); // 30分待つ
}
}

return 0;
}
```

このプログラムは、指定された条件に基づいて動作します。'pico/stdlib.h'、'hardware/gpio.h'、および'hardware/adc.h'を使って、必要なライブラリとヘッダーファイルをインクルードしています。