

作業報告書（2023 年 5 月 18 日）

J20413 北野正樹

【作業内容】

プログラムの概要の理解

前へ進むプログラムの作成

【作業項目】

① 本日作成したプログラムを下記に示す

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <wiringPiI2C.h>

// PWM ユニットの I2C アドレス
#define PWMI2CADR 0x40

// PWM 制御に使う。（1 でブリッジ動作、0 はブリッジオフ）
#define ENA_PWM 8

// IN1 と IN2 は右車輪の回転方向を決める（後進 : 0, 1, 前進 : 1, 0）（0, 0 と 1, 1 はブレーキ）
#define IN1_PWM 9
#define IN2_PWM 10

// 左側のモーター：パワーユニットの K3 または K4 に接続
// ENB は PWM 駆動に使う（1 でブリッジ動作、0 はブリッジオフ）
#define ENB_PWM 13

// IN3 と IN4 は左車輪の回転方向を決める（後進 : 0, 1, 前進 : 1, 0）（0, 0 と 1, 1 はブレーキ）
#define IN3_PWM 11
#define IN4_PWM 12

// PWM モジュールのレジスタ番号
#define PWM_MODE1 0
#define PWM_MODE2 1
#define PWM_SUBADR1 2
#define PWM_SUBADR2 3
#define PWM_SUBADR3 4
```

作業報告書 (2023 年 5 月 18 日)

J20413 北野正樹

```
#define PWM_ALLCALL 5

// PWM 番号 * 4 + PWM_0_??_? でレジスタ番号は求まる
#define PWM_0_ON_L 6
#define PWM_0_ON_H 7
#define PWM_0_OFF_L 8
#define PWM_0_OFF_H 9

// PWM 出力定数
#define PWMFULLON 16
#define PWMFULLOFF 0

// プリスケーラのレジスタ番号
// PWM 周波数を決めるレジスタ番号、100Hz なら 61 をセット
#define PWM_PRESCALE 254

// 光センサーのピン番号
#define GPIO_1 5
#define GPIO_2 6
#define GPIO_3 13
#define GPIO_4 19
#define GPIO_5 26

// motor_drive() から呼ばれる関数、PWM ユニットへの書き込みをする。
int set_pwm_output(int fd, int pwmch, int outval) {
    int ef = 0;
    int regno;

    if ((pwmch < 0) || (pwmch > 15)) ef = 1; // チャンネルの指定違反チェック
    if ((outval < 0) || (outval > 16)) ef = ef + 2; // 出力値の指定違反チェック
    if (ef == 0) {
        regno = PWM_0_ON_L + pwmch * 4; // 1ch あたり 4 レジスタで 16ch 分あるので
        if (outval == 16) {
            wiringPiI2CWriteReg8(fd, regno + 3, 0);
            wiringPiI2CWriteReg8(fd, regno + 1, 0x10);
        } else {
            wiringPiI2CWriteReg8(fd, regno + 1, 0);
        }
    }
}
```

作業報告書 (2023 年 5 月 18 日)

J20413 北野正樹

```
wiringPiI2CWriteReg8(fd, regno + 3, outval);
    }
}
return ef; // エラーがなければ 0 が返る
}
```

// モーターを制御するための関数。
// fd は I2C 初期化時のファイルディスクリプタ (デバイス番号のようなもの)
// lm は左モーター、rm は右モーターの駆動数値で、-16~+16 の範囲で指定
// 負の場合は後ろ方向に回転、正の場合は前方向に回転
// 全体値が大きいほど、パワーが大きくなる
// PWM ユニット自体は 12 ビット制度だが、上位 4 ビット分を制御

```
int motor_drive(int fd, int lm, int rm) {
    set_pwm_output(fd, ENA_PWM, 0); // 右のモーター無効化
    set_pwm_output(fd, ENB_PWM, 0); // 左のモーター有効化
    // 右モーターの制御
    if (rm < 0) {
        set_pwm_output(fd, IN1_PWM, 0); // OUT -> GND
        set_pwm_output(fd, IN2_PWM, rm); // OUT2 -> +Vs
        rm = abs(rm);
    } else {
        set_pwm_output(fd, IN1_PWM, lm); // OUT1 -> +Vs
        set_pwm_output(fd, IN2_PWM, 0); // OUT2 -> GND
    }

    // 左モーターの制御
    if (lm < 0) {
        set_pwm_output(fd, IN3_PWM, 0); // OUT3 -> GND
        set_pwm_output(fd, IN4_PWM, lm); // OUT -> +Vs
        lm = abs(lm);
    } else {
        set_pwm_output(fd, IN3_PWM, lm); // OUT3 -> +Vs
        set_pwm_output(fd, IN4_PWM, 0); // OUT4 -> GND
    }

    if (lm > 16) lm = 16;
    if (rm > 16) rm = 16;
}
```

作業報告書（2023 年 5 月 18 日）

J20413 北野正樹

```
set_pwm_output(fd, ENA_PWM, rm); // 右モータースタート
set_pwm_output(fd, ENB_PWM, lm); // 左モータースタート
return 0;
}

int main() {
    int fd;
    wiringPiSetupGpio();
    fd = wiringPiI2CSetup(PWMI2CADR);
    if (fd < 0) {
        printf("I2C の初期化に失敗しました。終了します。¥n");
        exit(EXIT_FAILURE);
    }
    wiringPiI2CWriteReg8(fd, PWM_PRESCALE, 61);
    wiringPiI2CWriteReg8(fd, PWM_MODE1, 0x10);
    wiringPiI2CWriteReg8(fd, PWM_MODE1, 0);
    delay(1);
    wiringPiI2CWriteReg8(fd, PWM_MODE1, 0x80);

    while (1) {
        if (digitalRead(GPIO_1) == LOW) {
            motor_drive(fd, 0, 0);
            return 0;
        }
        if (digitalRead(GPIO_3) == LOW) {
            motor_drive(fd, 10, 10);
        } else {
            motor_drive(fd, 0, 0);
        }
    }
    return 0;
}
```

以上のプログラムで前へ前進させることができる。Motor_drive 関数に出力を入れると左右のモータを制御できる。

作業報告書（2023 年 5 月 18 日）

J20413 北野正樹

【作業時間】

- ・ 作業時間：90 分
- ・ 報告書作成時間：10 分