

作業報告書（2019 年 10 月 6 日）

j16434 福原 誠也

【作業内容】

GPIO を操作して LED を点灯・消灯させる。

【作業項目】

インストール : `sudo apt install wiringpi`

ピンの確認 : `gpio readall`

ピンの設定 : `gpio -g mode 4 out`

4 : ピンの番号

out : ピンの入出力の設定 (out or in)

ピンの値の変更 : `gpio -g write 4 1` か `gpio -g write 4 0`

4 : ピンの番号

1 or 0 : 値の設定

1 の場合 : 点灯 `gpio -g write 4 1`

消灯 `gpio -g write 4 0`

2 の場合 : 点灯 `gpio -g write 4 0`

消灯 `gpio -g write 4 1`

【作業時間】

- ・ 作業時間 : 60 分
- ・ 報告書作成時間 : 5 分

作業報告書（2019年10月16日）

j16434 福原 誠也

【作業内容】

C 言語で Lチカ

【作業項目】

- Cでプログラムを書く
ヘッダファイルで wiringPi.h というものが用意されているので使う。
wiringPiSetup() を使い setup されているかを調べ異常がなければ pinMode(LED, OUTPUT) を使い GPIO4 を OUTPUT の設定にする。
その後 digitalWrite(LED, 1 or 0) を使いピンの値を決め、出力する。
delay() を使い一度スリープさせる。※()内は ms となっているので注意。
- 実行
コンパイルを行う際に -I でヘッダファイルの追加、-L でライブラリの追加をする。
実行する際はこの様になる。
gcc test.c -o test -I/usr/local/include -L/usr/local/lib -lwiringPi
ちなみに -o は好きな名前で作成ファイルを作成できる。
コンパイルが終了したら実行して完了！！
- ソース

```
#include <wiringPi.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define LED 7 // GPIO4
```

```
int main()
```

```
{
```

```
    int i;
```

```
    if (wiringPiSetup() == -1)
```

```
        exit(1);
```

```
    pinMode(LED, OUTPUT);
```

```
    for (i=0; i<10; i++) {
```

```
        digitalWrite(LED, 1);
```

```
        delay(500);
```

```
        digitalWrite(LED, 0);
```

作業報告書（2019 年 10 月 16 日）

j16434 福原 誠也

```
    delay(500);  
}  
  
return 0;  
}
```

【作業時間】

- ・ 作業時間 : 20 分
- ・ 報告書作成時間 : 30 分

作業報告書（2019 年 10 月 21 日）

j16434 福原 誠也

【作業内容】

- プルアップ、プルダウンについて調べる。
- SW が押されたら LED が光るプログラムを作成する。

【作業項目】

- プルアップについて
「何も入力がないとき、アップ（ON）状態」にするプルアップ
スイッチが OFF の時に浮いた状態になる
その状態を避けるために使われる
スイッチが ON の時は OFF
スイッチが OFF の時は ON
- プルダウンについて
「何も入力がないとき、ダウン（OFF）状態」にするプルアップ
プルダウンはプルアップとは逆でスイッチが ON の時は ON
スイッチが OFF の時は OFF
- ラズパイ内蔵のプルアップ抵抗、プルダウン抵抗について
ラズパイの GPIO はプルアップ抵抗、プルダウン抵抗を内蔵している。
この抵抗はソフトウェアで ON、OFF の切り替えができる
- ソースコード

```
#include <wiringPi.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define GPIO17 0 // GPIO17
```

```
#define GPIO18 1 // GPIO18
```

```
int main()
```

```
{
```

```
    int i;
```

```
    if (wiringPiSetup() == -1)
```

```
        exit(1);
```

```
    pinMode(GPIO17, OUTPUT);
```

```
    pinMode(GPIO18, INPUT);
```

```
    pullUpDnControl(GPIO18, PUD_UP);
```

作業報告書（2019 年 10 月 21 日）

j16434 福原 誠也

```
while(1){  
  
    if (digitalRead(GPIO18) == 0){  
  
        digitalWrite(GPIO17, 1);  
  
    }else{  
  
        digitalWrite(GPIO17, 0);  
  
    }  
}  
  
return 0;  
}
```

- ソースについて
 前回の部分は省略
 pullUpDnControl(GPIO18, PUD_UP) : プルアップ・プルダウン抵抗の設定
 digitalRead(GPIO18) : ピンの値の読み込み
 gpio readall で一度 wiringPi でのピン番号を確認した方がよい。
- 参考
<https://voltechno.com/blog/pullup-pulldown/>
<https://hnw.hatenablog.com/entry/20150607>
<http://www.greensignal.jp/blog/?p=625>

【作業時間】

- ・ 作業時間 : 60 分
- ・ 報告書作成時間 : 60 分

作業報告書（2019 年 10 月 28 日）

j16434 福原 誠也

【作業内容】

スイッチを押すたびにLEDチカさせる。

チャタリングの問題を解決する。

【作業項目】

ソースコード

```
#include <wiringPi.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define GPIO17 0 // GPIO17
```

```
#define GPIO18 1 // GPIO18
```

```
int main()
```

```
{
```

```
    int i;
```

```
    if (wiringPiSetup() == -1) exit(1);
```

```
    pinMode(GPIO17, OUTPUT);
```

```
    pinMode(GPIO18, INPUT);
```

```
    pullUpDnControl(GPIO18, PUD_UP);
```

```
    while(1){
```

```
        if (digitalRead(GPIO18) == 0){
```

```
            i++;
```

```
            digitalWrite(GPIO17, i % 2);
```

```
            while (digitalRead(GPIO18) == 0) {
```

```
                delay(1);
```

```
            }
```

```
        }
```

```
    }
```

作業報告書（2019 年 10 月 28 日）

j16434 福原 誠也

```
    return 0;  
}
```

ソースの解説

```
while (digitalRead(GPIO18) == 0) {delay(1);} 
```

この部分はスイッチが押されたとき一定期間の間スイッチからの入力を切り、チャタリングが起きても誤作動しないようにしている。

【作業時間】

- ・ 作業時間 : 10 分
- ・ 報告書作成時間 : 10 分

作業報告書（2019 年 11 月 7 日）

j16434 福原 誠也

【作業内容】

前回の続き

【作業項目】

前回、チャタリング除去をしていたが色々いじっていたら誤作動があったのでその解決をする。

ソースの編集

```
while (digitalRead(GPIO18) == 0) {delay(10);}
```

前は delay の値が 1 だったが 10 にすることで解決した。

【作業時間】

- ・ 作業時間 : 10 分
- ・ 報告書作成時間 : 5 分

作業報告書（2019 年 11 月 11 日）

j16434 福原 誠也

【作業内容】

前回のプログラムを pthread を用いて実装

【作業項目】

前回のプログラムを pthread を用いて実装するために調べていたが、pthead.h の関数の使い方が分からなかった。

【作業時間】

- ・ 作業時間 : 70 分
- ・ 報告書作成時間 : 5 分

作業報告書（2019 年 11 月 18 日）

j16434 福原 誠也

【作業内容】

I2C を使い LCD の表示

【作業項目】

調べ作業

成果なし

【作業時間】

- ・ 作業時間 : 80 分
- ・ 報告書作成時間 : 1 分

作業報告書（2019 年 12 月 2 日）

j16434 福原 誠也

【作業内容】

前回の続き

【作業項目】

初期設定ができたようにみえた

【作業時間】

- ・ 作業時間 : 89 分
- ・ 報告書作成時間 : 1 分

作業報告書（2019 年 12 月 9 日）

j16434 福原 誠也

【作業内容】

前回の続き

【作業項目】

進捗なし

【作業時間】

- ・ 作業時間 : 89 分
- ・ 報告書作成時間 : 1 分

作業報告書（2019 年 1 月 6 日）

j16434 福原 誠也

【作業内容】

指導書の解説

【作業項目】

指導書を読んで回路を組み、ソースをコピー

【作業時間】

- ・ 作業時間 : 40 分
- ・ 報告書作成時間 : 3 分

作業報告書（2019 年 1 月 20 日）

j16434 福原 誠也

【作業内容】

サーボモータを使うために指導書を読み、ソースを書く

【作業項目】

$N=2.02 \times 10^5$

PWM=2

center が分からん

【作業時間】

- ・ 作業時間 : 70 分
- ・ 報告書作成時間 : 2 分

作業報告書（2019 年 1 月 27 日）

j16434 福原 誠也

【作業内容】

前回の続き

【作業項目】

モーターを動かす

T, N, PWMrange を考えたが分からない

【作業時間】

- ・ 作業時間 : 80 分
- ・ 報告書作成時間 : 2 分

実験実習参 A 作業報告書

提出日 : 02/10

16-434 福原 誠也