

作業報告書（2022 年 6 月 22 日）

J20413 北野正樹

【作業内容】

実際に回路を組んで0～9を表示させる。
アルファベットを表示させる。（自己課題）

【作業項目】

① 回路を組む。

本実験で組んだ回路を図1に示す。

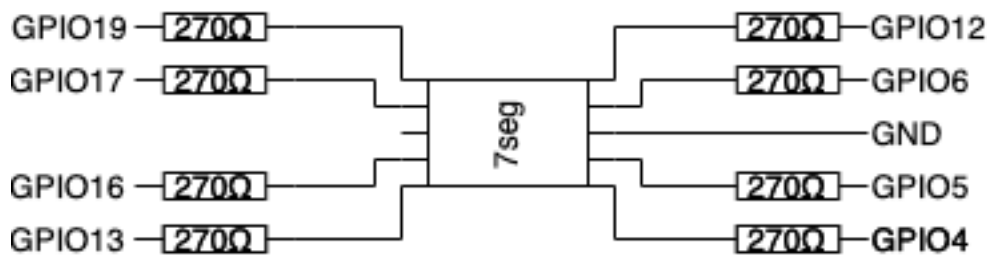


図1：7セグメントディスプレイの回路図

② プログラムを書く

本実験で作成したプログラムを下に示す。

前回と異なり、7セグのデータ保持の形式をいくつか試してみた。

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

int main(void) {

    // ビットを配列に格納している
    const int seg_bit[36][7] = {
        {1, 1, 1, 1, 1, 1, 0}, // 0
        {0, 0, 0, 0, 1, 1, 0}, // 1
        {1, 1, 0, 1, 1, 0, 1}, // 2
        {1, 1, 1, 1, 0, 0, 1}, // 3
        {0, 1, 1, 0, 0, 1, 1}, // 4
        {1, 0, 1, 1, 0, 1, 1}, // 5
        {1, 0, 1, 1, 1, 1, 1}, // 6
        {1, 1, 1, 0, 0, 1, 0}, // 7
        {1, 1, 1, 1, 1, 1, 1}, // 8
        {1, 1, 1, 1, 0, 1, 1}, // 9
        {1, 1, 1, 0, 1, 1, 1}, // A
        {0, 0, 1, 1, 1, 1, 1}, // b
    };
}
```

作業報告書（2022 年 6 月 22 日）

J20413 北野正樹

```
{1, 0, 0, 1, 1, 1, 0}, // C
{0, 1, 1, 1, 1, 0, 1}, // d
{1, 0, 0, 1, 1, 1, 1}, // E
{1, 0, 0, 0, 1, 1, 1}, // F
{1, 0, 1, 1, 1, 1, 0}, // G
{0, 0, 1, 0, 1, 1, 1}, // h
{0, 0, 0, 0, 1, 1, 0}, // I
{0, 1, 1, 1, 1, 0, 0}, // J
{1, 0, 1, 0, 1, 1, 1}, // K
{0, 0, 0, 1, 1, 1, 0}, // L
{1, 1, 1, 0, 1, 1, 0}, // M
{0, 0, 1, 0, 1, 0, 1}, // n
{0, 0, 1, 1, 1, 0, 1}, // o
{1, 1, 0, 0, 1, 1, 1}, // P
{1, 1, 1, 0, 0, 1, 1}, // Q
{0, 0, 0, 0, 1, 0, 1}, // r
{0, 0, 1, 1, 0, 1, 1}, // S
{0, 0, 0, 1, 1, 1, 1}, // t
{0, 0, 1, 1, 1, 0, 0}, // u
{0, 1, 1, 1, 1, 1, 0}, // V
{0, 1, 0, 1, 0, 1, 0}, // W
{0, 1, 1, 0, 1, 1, 1}, // X
{0, 1, 1, 1, 0, 1, 1}, // y
{1, 1, 0, 1, 1, 0, 0}, // z
};
```

// セグの出力データを 10 進数にしてメモリ削減（あんまり意味はないかもしれない）

```
const short seg_dec[36] = {
    126, // 0
    6,   // 1
    109, // 2
    121, // 3
    51,  // 4
    91,  // 5
    95,  // 6
    114, // 7
```

作業報告書（2022 年 6 月 22 日）

J20413 北野正樹

```
127, // 8
123, // 9
119, // A
31,  // b
78,  // C
61,  // d
79,  // E
71,  // F
94,  // G
23,  // h
6,   // I
60,  // J
87,  // K
14,  // L
118, // M
21,  // n
29,  // o
103, // P
115, // q
5,   // r
27,  // S
15,  // t
28,  // u
62,  // V
42,  // w
55,  // X
59,  // y
108, // z

}

const int pin[8] = {4, 5, 6, 12, 13, 16, 17, 19};

int i, j, k, n;
short dot = 0;

/** init wirinpPi */
if (wiringPiSetupGpio() == -1) {
```

作業報告書（2022 年 6 月 22 日）

J20413 北野正樹

```
        return 1;
    }

    /** Set GPIO pins output mode */
    for (i = 1; i < 8; i++) {
        pinMode(pin[i], OUTPUT);
        digitalWrite(pin[i], 0);
    }

    // bit 情報を 7 セグに反映させる
    for (i = 0; i < 36; i++) {
        for (j = 0; j < 7; j++) {
            digitalWrite(pin[j], seg_bit[i][j]);
        }
        if (dot == 1) {
            digitalWrite(pin[7], 1);
        }
        delay(1000);
    }

    // 2 進数に直してデータを反映させる。
    for (i = 0; i < 36; i++) {
        for (j = 7; j >= 0; j--) {
            int n = seg_dec[i];
            for (k = 0; k < 7; k++) {
                digitalWrite(pin[j], n % 2);
                n /= 2;
            }
            if (dot == 1) {
                digitalWrite(pin[7], 1);
            }
            delay(1000);
        }
    }

    return 0;
}
```

作業報告書（2022 年 6 月 22 日）

J20413 北野正樹

③ プログラムをコンパイルし、実行する。

```
gcc -Wall -o PulseLED PluseLED.c -l wiringPi  
sudo ./PulseLED
```

これらを実行すると 7 セグが 1 秒おきに更新される。

【作業時間】

- ・ 作業時間：60 分
- ・ 報告書作成時間：20 分