

การทดลองที่ 10 การเชื่อมต่อกับขา GPIO

ไลบรารี wiringPi

4. เรียกคำสั่ง `gpio readall` เพื่อตรวจสอบและบันทึกผลลัพธ์ที่แสดงบนหน้าต่าง Terminal ลงในตารางหน้าถัดไป

```
$ gpio readall
```

5. จงเติมหมายเลขในคอลัมน์wPi (wiringPi) ให้ตรงกับขาเชื่อมต่อ 40 ขาบนบอร์ด Pi ตามที่แสดงบนหน้า
จอลงในตารางต่อไปนี้เพื่อให้ประกอบการต่อวงจรที่ถูกต้อง

ตอบ

```

pi@raspberrypi:~ $ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |      | 3.3v      |      |   | 1  | 2  |      | 5v       |      |      | |
|  2   |  8   | SDA.1     | IN    | 1 | 3  | 4  |      | 5v       |      |      |
|  3   |  9   | SCL.1     | IN    | 1 | 5  | 6  |      | 0v       |      |      |
|  4   |  7   | GPIO. 7   | IN    | 1 | 7  | 8  | 0   | IN       | TxD   | 15   | 14   |
|      |      | 0v        |      |   | 9  | 10 | 1   | IN       | RxD   | 16   | 15   |
| 17   |  0   | GPIO. 0   | IN    | 0 | 11 | 12 | 0   | IN       | GPIO. 1 | 1   | 18   |
| 27   |  2   | GPIO. 2   | IN    | 0 | 13 | 14 |      | 0v       |      |      |
| 22   |  3   | GPIO. 3   | IN    | 0 | 15 | 16 | 0   | IN       | GPIO. 4 | 4   | 23   |
|      |      | 3.3v      |      |   | 17 | 18 | 0   | IN       | GPIO. 5 | 5   | 24   |
| 10   | 12   | MOSI      | IN    | 0 | 19 | 20 |      | 0v       |      |      |
|  9   | 13   | MISO      | IN    | 0 | 21 | 22 | 0   | IN       | GPIO. 6 | 6   | 25   |
| 11   | 14   | SCLK      | IN    | 0 | 23 | 24 | 1   | IN       | CE0    | 10   | 8    |
|      |      | 0v        |      |   | 25 | 26 | 1   | IN       | CE1    | 11   | 7    |
|  0   | 30   | SDA.0     | IN    | 1 | 27 | 28 | 1   | IN       | SCL.0  | 31   | 1    |
|  5   | 21   | GPIO.21   | IN    | 1 | 29 | 30 |      | 0v       |      |      |
|  6   | 22   | GPIO.22   | IN    | 1 | 31 | 32 | 0   | IN       | GPIO.26 | 26   | 12   |
| 13   | 23   | GPIO.23   | IN    | 0 | 33 | 34 |      | 0v       |      |      |
| 19   | 24   | GPIO.24   | IN    | 0 | 35 | 36 | 0   | IN       | GPIO.27 | 27   | 16   |
| 26   | 25   | GPIO.25   | IN    | 0 | 37 | 38 | 0   | IN       | GPIO.28 | 28   | 20   |
|      |      | 0v        |      |   | 39 | 40 | 0   | IN       | GPIO.29 | 29   | 21   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

โปรแกรมไฟ LED กระพริบภาษา C

10. จับเวลาช่วงเวลาที่หลอดสว่างและดับตั้งแต่เริ่มรันโปรแกรมจนเสร็จสิ้น เพื่อหาค่าเฉลี่ยของการสว่างดับ 1 รอบ

ตอบ 1.0019 วินาที (จับเวลาตามใน IDE)
1.0137 วินาที (จับเวลาโดยไม่ใช้ IDE)

โปรแกรมไฟ LED กระพริบภาษาแอสเซมบลี

7. จับเวลาช่วงเวลาที่หลอดสว่างและดับตั้งแต่เริ่มรันโปรแกรมจนเสร็จสิ้น เพื่อหาค่าเฉลี่ยของการสว่างดับ 1 รอบ

ตอบ 0.9629 วินาที (จับเวลาโดยไม่ใช้ IDE)

กิจกรรมท้ายการทดลอง

3. ประโยค PUSH {ip,lr} ทำหน้าที่อะไร เหตุใดจึงต้องเรียกใช้ก่อนประโยคอื่น ๆ

ตอบ ทำการ push ค่าภายใน IP register และค่าใน Link Register ลงใน stack ในส่วนของ stack segment เพื่อที่จะสามารถนำมา pop ออกจาก stack เมื่อจบการทำงานของฟังก์ชันเพื่อคืนค่าให้กับ register นั้น ๆ ได้ สาเหตุที่ต้อง push IP register ทั้งที่ใช้ค่าของ Link Register ก็เพียงพอแล้วคือเพื่อเป็นการทำให้ข้อมูลถูก align ในรูปแบบ 8 bytes ได้อย่างเหมาะสมและป้องกัน segmentation fault

9. ต่อหลอด LED เพิ่มอีก 2 ดวงรวมเป็น 3 ดวงแล้วพัฒนาโปรแกรมภาษา C เดิมให้นับเลข 0-7 และแสดงผลทางหลอด LED เป็นเลขฐานสองวนไปเรื่อย ๆ

ตอบ

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

int main(void) {
    int pin[3] = {0, 2, 3};
    int bi[8][3] = {
        {0,0,0},{0,0,1},{0,1,0},
        {0,1,1},{1,0,0},{1,0,1},
        {1,1,0},{1,1,1}
    };

    printf("LED blinking by wiringPi\n");
    if (wiringPiSetup() == -1) {
        printf("Setting up problem ... Abort!");
        exit(1);
    }
    int i= 0;
    for (i = 0; i < 3; ++i)
        pinMode(pin[i], OUTPUT);

    while(1) {
        for (i = 0; i < 8; ++i) {
            printf("%d\n", i);
            int k;
            for (k = 0; k < 3; ++k)
                digitalWrite(pin[k], bi[i][k]);
            delay(1000);
        }
    }
}
```