

ນາຍ ດົກທິກີ່ ຜົມມະໄກນ໌ 62010684

ນາຍ ດົກນິພາກ ເນັ້ນສູນ 62010710

ภาคผนวก K

การทดลองที่ 11 การเชื่อมต่อกับสัญญาณอินเทอร์ร์พท์

การทดลองนี้ค่าดูว่าผู้อ่านเคยเรียนการเขียนหรือพัฒนาโปรแกรมด้วยภาษา C และแ Olsen เชมบลีจากการทดลองก่อนหน้า ดังนั้น การทดลองมีวัตถุประสงค์เหล่านี้

- เพื่อพัฒนาการทำงานของอินเทอร์รัฟท์ร่วมโปรแกรมภาษา C และแอสเซมบลี ตามเนื้อหาในหัวข้อที่ 6.12
 - เพื่อศึกษาการทำงานของอินเทอร์รัฟท์ร่วมกับขา GPIO ตามเนื้อหาใน หัวข้อที่ 6.11

K.1 การจัดการอินเทอร์รัพท์ของ WiringPi

ไลบรารี wiringPi รองรับการทำอินเทอร์รัพท์ของ GPIO ได้ ทำให้โปรแกรมหลักสามารถทำงาน หลักได้ตามปกติ เมื่อเกิดสัญญาณอินเทอร์รัพท์ขึ้น ไม่ว่าจะเป็นสัญญาณจากการกดปุ่ม ทำให้เกิดขอบขาขึ้นหรือขอบขาลง หรือทั้งสองขอบ โดยการเรียกใช้คำสั่ง

```
wiringPiISR(pin, edgeType, &callback)
```

โดย pin หมายเลขที่ wiringPi กำหนด edgeType กำหนดจากค่าคงที่ 4 ค่านี้

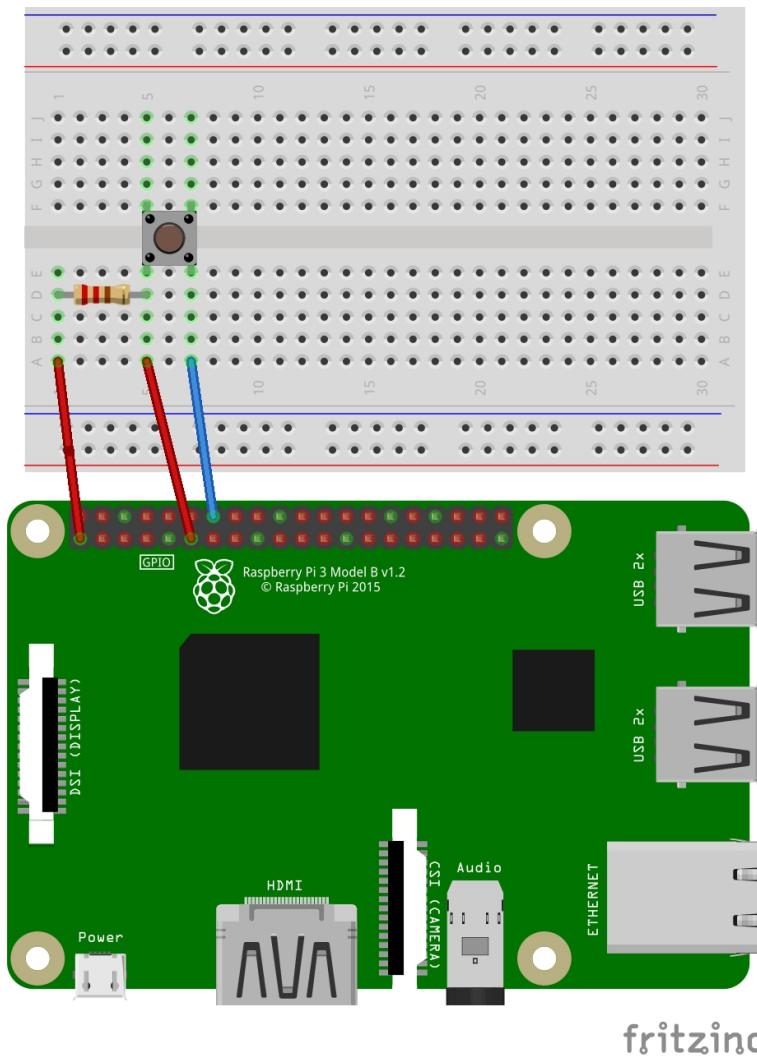
- INT_EDGE_FALLING,
 - INT_EDGE_RISING,
 - INT_EDGE_BOTH
 - INT_EDGE_SETUP.

การกำหนดชนิดขอบขาเป็น 3 ชนิดแรก ไลบรารีจะตั้งค่าเริ่มต้น (Initialization) ให้โดยอัตโนมัติ หากกำหนดชนิดขอบเป็น INT_EDGE_SETUP ไลบรารีจะไม่ตั้งค่าเริ่มต้น (Initialization) ให้ เนื่องจากโปรแกรมเมอร์จะต้องกำหนดเอง

พารามิเตอร์ **callback** คือ ชื่อฟังก์ชันที่จะทำหน้าที่เป็น ISR สัญลักษณ์ & หมายถึง แอดเดรสของฟังก์ชัน callback ฟังก์ชัน callback นี้จะเริ่มต้นทำงานโดยแจ้งต่อว่างจร Dispatcher ในหัวข้อที่ [6.12](#) ก่อนจะเริ่มต้นทำงาน โดยฟังก์ชัน callback จะสามารถอ่าน หรือเขียนค่าของตัวแปรโกลบอลในโปรแกรมได้ ซึ่งต้องย่างการทำงานจะได้กล่าวในหัวข้อถัดไป

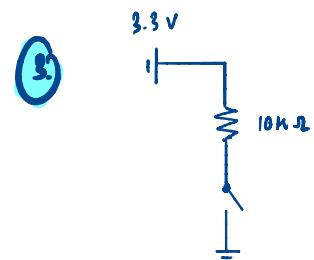
K.2 วงจรปุ่มกด Push Button เชื่อมผ่านขา GPIO

1. ซักดาวน์และตัดไฟเลี้ยงออกจากบอร์ด Pi3 เพื่อความปลอดภัยในการต่อวงจร
2. ต่อวงจรตามรูปที่ K.1



รูปที่ K.1: วงจรกดปุ่มสำหรับทดลองการเขียนโปรแกรมอินเทอร์รัพท์ในการทดลองที่ 11 ที่มา: fritzing.org

3. จรวจวงจรที่ต่อในรูปที่ K.1 ประกอบด้วย สวิตช์กดปุ่ม ตัวต้านทาน ไฟเลี้ยง 3.3 โวลท์ ขา BUT-TON_PIN และกราวด์ (0 โวลท์)
4. ตรวจสอบความถูกต้อง โดยให้ผู้ควบคุมการทดลองตรวจสอบ
5. สร้าง project ใหม่ชื่อ Lab11 ภายใต้เดเรกทอรี /home/pi/asm/Lab11

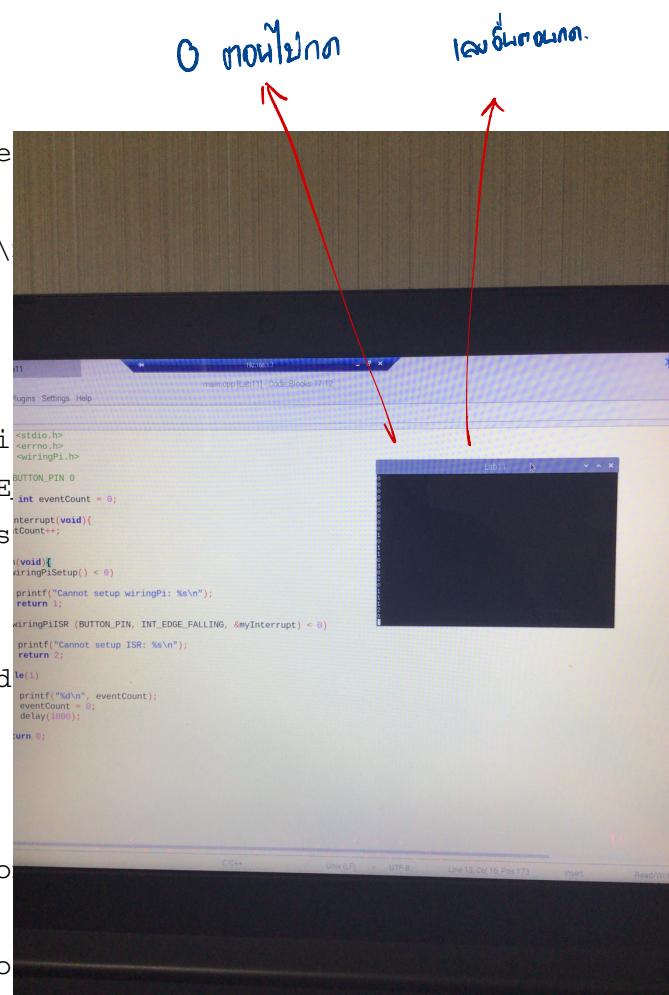


K.3 โปรแกรมภาษา C สำหรับทดลองอินเทอร์รัพท์

ผู้อ่านต้องทำความเข้าใจกับตัวโปรแกรมก่อนคอมไพล์หรือรันโปรแกรม เพื่อความเข้าใจสูงสุด โดยเฉพาะชื่อตัวแปร ชนิดของตัวแปร evenCounter การติดตั้งฟังค์ชัน wiringPiISR เพื่อเชื่อมโยงกับขา GPIO ชนิดของ

การตรวจสอบ และข้อผิดพลาดของ ISR หรือ ฟังค์ชัน callback

```
#include <stdio.h>
#include <errno.h>
#include <wiringPi.h>
#define BUTTON_PIN 0
// Use GPIO Pin 17 = Pin 0 of wiringPi library
volatile int eventCount = 0;
void myInterrupt(void) { // called every time an event occurs
    eventCount++; // the event counter
}
int main(void) {
    if (wiringPiSetup() < 0) // check the
    {
        printf ("Cannot setup wiringPi: %s\n");
        return 1; // error code = 1
    }
    // set wiringPi Pin 0 to generate an
    // myInterrupt() = my Interrupt Servi
    if (wiringPiISR (BUTTON_PIN, INT_EDGE_RISING, &myInterrupt) < 0)
        printf ("Cannot setup ISR: %s\n", s
        return 2; // error code = 2
    }
    // display counter value every second
    while(1) {
        printf ("%d\n", eventCount);
        eventCount = 0;
        delay(1000); // wait 1000 milliseco
    }
    return 0; // error code = 0 (No Errro
}
```



1. ป้อนโปรแกรมด้านบนใน main.c โดยใช้โปรแกรม Text Editor ที่ว่าไป

2. สร้าง makefile สำหรับคอมไพล์และลิงค์โปรแกรมจากการทดลองก่อนหน้านี้ จนไม่เกิดข้อผิดพลาด

3. รันโปรแกรม ทดสอบการทำงานด้วยการกดปุ่มที่ต่อไว้ สังเกตผลลัพธ์ทางหน้าจอ Terminal ที่รัน

```
$ sudo ./Lab11
```

K.4 กิจกรรมท้ายการทดลอง

1. จงวัดสัญญาณที่ขา BUTTON_PIN ก่อนกดปุ่ม ระหว่างกดปุ่ม และปล่อยมือจากปุ่มกด โดยให้แก่นอนเป็นแกนเวลา แกนตั้งเป็นค่าโวลเตจ หรือ ค่าลอจิกของขาสัญญาณ BUTTON_PIN
2. จงบอกความหมายและการประยุกต์ใช้งานตัวแปรชนิด volatile
3. ปรับแก้ volatile ออกเหลือแค่ int eventCount = 0; make แล้วจึงรันโปรแกรมทดสอบการทำงานด้วยการกดปุ่มที่ต่อໄว้ สังเกตผลลัพธ์ทางหน้าจอ Terminal ที่รัน เปรียบเทียบการทำงานของโปรแกรมก่อนและหลังการปรับแก้ และหาเหตุผล
4. จงปรับแก้โปรแกรมที่ทดลองตามประโยชน์ต่อไปนี้

```
if (wiringPiISR (BUTTON_PIN, INT_EDGE_RISING, &myInterrupt) < 0) {
    ...
}
```

ทำ make ใหม่และทดลองกดปุ่ม สังเกตการเปลี่ยนแปลงและอธิบาย

5. จงตอบคำถามจากประโยชน์ต่อไปนี้

```
if (wiringPiISR (BUTTON_PIN, INT_EDGE_FALLING, &myInterrupt) < 0) {
    ...
}
```

- พังค์ชัน wiringPiISR ทำหน้าที่อะไร เหตุใดอยู่ในประโยชน์เนื่องจาก if
- ตัวแปร &myInterrupt คืออะไร เหตุใดจึงมีสัญลักษณ์ & นำหน้า
- พังค์ชันนี้เชื่อมโยงกับตารางที่ 6.6 อย่างไร

6. จงใช้วงจรหลอด LED 3 ดวงและโปรแกรมจากการทดลองที่ 10 นับขึ้นจาก 0-7-0 โดยเพิ่มปุ่มกดในการทดลองนี้ และเพิ่มพังค์ชันการอินเทอร์รัพท์จากโปรแกรม Lab11.2 นี้ เมื่อกดปุ่มแต่ละครั้งจะทำให้ความเร็วในการนับเพิ่มขึ้น หรือ Delay สั้นลงครึ่งหนึ่ง เมื่อกดครั้งที่ 2 จะสั้นลงอีกครึ่งหนึ่ง เมื่อกดครั้งที่ 3 จะทำให้ Delay กลับไปเป็นค่าเริ่มต้น
7. จงใช้วงจรหลอด LED 3 ดวงและโปรแกรมจากการทดลองที่ 10 แต่นับลงจาก 7-0-7 โดยเพิ่มปุ่มกดในการทดลองนี้ และเพิ่มพังค์ชันการอินเทอร์รัพท์จากโปรแกรม Lab11.2 นี้ เมื่อกดปุ่มแต่ละครั้งจะทำให้ความเร็วในการนับลดลง หรือ Delay เพิ่มขึ้นเท่าตัว เมื่อกดครั้งที่ 2 Delay เพิ่มขึ้นอีกเท่าตัว เมื่อกดครั้งที่ 3 จะทำให้ Delay กลับไปเป็นค่าเริ่มต้น

หน้าต่อไป →

6.

main.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <wiringPi.h>
4 #include <errno.h>
5 #define BUTTON_PIN 0

6 volatile int DL = 1000;

7 void myInterrupt(void)
8 {
9     if(DL == 1000)
10    {
11        DL = 500;
12    }
13    else if(DL == 500)
14    {
15        DL = 250;
16    }
17    else if(DL == 250)
18    {
19        DL = 1000;
20    }
21    printf("Delay decreased x2 !\n");
22 }
23
24 int main (void){
25     int pin0 = 7, pin1 = 0, pin2 = 2;
26     if(wiringPiSetup() < 0)
27     {
28         printf("Cannot setup wiring Pi\n");
29         return 1;
30     }
31     if(wiringPiISR(BUTTON_PIN, INT_EDGE_FALLING, &myInterrupt) < 0)
32     {
33         printf("Cannot setup ISR\n");
34         return 2;
35     }
36
37     pinMode(pin0,OUTPUT);
38     pinMode(pin1,OUTPUT);
39     pinMode(pin2,OUTPUT);

40     while(1)
41     {
42         // upward 0 -> 7
43         for(int i = 0;i < 8; i++)
44         {
45             printf("LED dec is %d\n",i);
46             if(i == 0)
47             {
48                 digitalWrite(pin0,0);
49                 digitalWrite(pin1,0);
50                 digitalWrite(pin2,0);
51                 delay(DL);
52             }
53         }
54     }
55 }
```

Delay initially = 1000 ms

function on delay init

0 → 7

[0-7-0]

main.c

```

56     if(i == 1)
57     {
58         digitalWrite(pin0,1);
59         digitalWrite(pin1,0);
60         digitalWrite(pin2,0);
61         delay(DL);
62     }
63     if(i == 2)
64     {
65         digitalWrite(pin0,0);
66         digitalWrite(pin1,1);
67         digitalWrite(pin2,0);
68         delay(DL);
69     }
70     if(i == 3)
71     {
72         digitalWrite(pin0,1);
73         digitalWrite(pin1,1);
74         digitalWrite(pin2,0);
75         delay(DL);
76     }
77     if(i == 4)
78     {
79         digitalWrite(pin0,0);
80         digitalWrite(pin1,0);
81         digitalWrite(pin2,1);
82         delay(DL);
83     }
84     if(i == 5)
85     {
86         digitalWrite(pin0,1);
87         digitalWrite(pin1,0);
88         digitalWrite(pin2,1);
89         delay(DL);
90     }
91     if(i == 6)
92     {
93         digitalWrite(pin0,0);
94         digitalWrite(pin1,1);
95         digitalWrite(pin2,1);
96         delay(DL);
97     }
98     if(i == 7)
99     {
100        digitalWrite(pin0,1);
101        digitalWrite(pin1,1);
102        digitalWrite(pin2,1);
103        delay(DL);
104    }
105    // downward 7 -> 0 LED
106    for(int i = 6; i > 1; i--)
107    {
108        printf("LED dec is %d\n",i);
109        if(i == 6)
110    }
```

7 → 0

main.c

```

110    if(i == 6)
111    {
112        digitalWrite(pin0,0);
113        digitalWrite(pin1,1);
114        digitalWrite(pin2,1);
115        delay(DL);
116    }
117    if(i == 5)
118    {
119        digitalWrite(pin0,1);
120        digitalWrite(pin1,0);
121        digitalWrite(pin2,1);
122        delay(DL);
123    }
124    if(i == 4)
125    {
126        digitalWrite(pin0,0);
127        digitalWrite(pin1,0);
128        digitalWrite(pin2,1);
129        delay(DL);
130    }
131    if(i == 3)
132    {
133        digitalWrite(pin0,1);
134        digitalWrite(pin1,1);
135        digitalWrite(pin2,0);
136        delay(DL);
137    }
138    if(i == 2)
139    {
140        digitalWrite(pin0,0);
141        digitalWrite(pin1,1);
142        digitalWrite(pin2,0);
143        delay(DL);
144    }
145    if(i == 1)
146    {
147        digitalWrite(pin0,1);
148        digitalWrite(pin1,0);
149        digitalWrite(pin2,0);
150        delay(DL);
151    }
152    if(i == 0)
153    {
154        digitalWrite(pin0,0);
155        digitalWrite(pin1,0);
156        digitalWrite(pin2,0);
157        delay(DL);
158    }
159    }
160    }
161    }
162    }
163 }
```

return 0;

7

main.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <wiringPi.h>
4 #include <errno.h>
5 #define BUTTON_PIN 0
6
7 volatile int DL = 250;
8
9 void myInterrupt(void)
10 {
11     if(DL == 250)
12     {
13         DL = 500;
14     }
15     else if(DL == 500)
16     {
17         DL = 1000;
18     }
19     else if(DL == 1000)
20     {
21         DL = 250;
22     }
23     printf("Delay decreased x2 !\n");
24 }
25
26 int main (void){
27     int pin0 = 7, pin1 = 0, pin2 = 2;
28     if(wiringPiSetup() < 0)
29     {
30         printf("Cannot setup wiring Pi\n");
31         return 1;
32     }
33     if(wiringPiISR(BUTTON_PIN, INT_EDGE_FALLING, &myInterrupt) < 0)
34     {
35         printf("Cannot setup ISR\n");
36         return 2;
37     }
38
39     pinMode(pin0,OUTPUT);
40     pinMode(pin1,OUTPUT);
41     pinMode(pin2,OUTPUT);
42
43     while(1)
44     {
45         // downward 7 -> 0 LED
46         for(int i = 6; i > 1; i--)
47         {
48             printf("LED dec is %d\n",i);
49             if(i == 6)
50             {
51                 digitalWrite(pin0,0);
52                 digitalWrite(pin1,1);
53                 digitalWrite(pin2,1);
54                 delay(DL);
55             }
56         }
57     }
58 }
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110

```

Delay is now = 250 ms

function in delay intro.

mn 7-70

[7-0-7]

main.c

```

56     if(i == 5)
57     {
58         digitalWrite(pin0,1);
59         digitalWrite(pin1,0);
60         digitalWrite(pin2,1);
61         delay(DL);
62     }
63     if(i == 4)
64     {
65         digitalWrite(pin0,0);
66         digitalWrite(pin1,0);
67         digitalWrite(pin2,1);
68         delay(DL);
69     }
70     if(i == 3)
71     {
72         digitalWrite(pin0,1);
73         digitalWrite(pin1,1);
74         digitalWrite(pin2,0);
75         delay(DL);
76     }
77     if(i == 2)
78     {
79         digitalWrite(pin0,0);
80         digitalWrite(pin1,1);
81         digitalWrite(pin2,0);
82         delay(DL);
83     }
84     if(i == 1)
85     {
86         digitalWrite(pin0,1);
87         digitalWrite(pin1,0);
88         digitalWrite(pin2,0);
89         delay(DL);
90     }
91     if(i == 0)
92     {
93         digitalWrite(pin0,0);
94         digitalWrite(pin1,0);
95         digitalWrite(pin2,0);
96         delay(DL);
97     }
98 }
99 // upward 0 -> 7
100 for(int i = 0;i < 8; i++)
101 {
102     printf("LED dec is %d\n",i);
103     if(i == 0)
104     {
105         digitalWrite(pin0,0);
106         digitalWrite(pin1,0);
107         digitalWrite(pin2,1);
108         delay(DL);
109     }
110     if(i == 1)
111     {
112         digitalWrite(pin0,1);
113         digitalWrite(pin1,0);
114         digitalWrite(pin2,0);
115         delay(DL);
116     }
117     if(i == 2)
118     {
119         digitalWrite(pin0,0);
120         digitalWrite(pin1,1);
121         digitalWrite(pin2,0);
122         delay(DL);
123     }
124     if(i == 3)
125     {
126         digitalWrite(pin0,1);
127         digitalWrite(pin1,1);
128         digitalWrite(pin2,0);
129         delay(DL);
130     }
131     if(i == 4)
132     {
133         digitalWrite(pin0,0);
134         digitalWrite(pin1,0);
135         digitalWrite(pin2,1);
136         delay(DL);
137     }
138     if(i == 5)
139     {
140         digitalWrite(pin0,1);
141         digitalWrite(pin1,0);
142         digitalWrite(pin2,1);
143         delay(DL);
144     }
145     if(i == 6)
146     {
147         digitalWrite(pin0,0);
148         digitalWrite(pin1,1);
149         digitalWrite(pin2,1);
150         delay(DL);
151     }
152     if(i == 7)
153     {
154         digitalWrite(pin0,1);
155         digitalWrite(pin1,1);
156         digitalWrite(pin2,1);
157         delay(DL);
158     }
159 }
160 }
161
162
163

```

mn 0-7

```

110 if(i == 1)
111 {
112     digitalWrite(pin0,1);
113     digitalWrite(pin1,0);
114     digitalWrite(pin2,0);
115     delay(DL);
116 }
117 if(i == 2)
118 {
119     digitalWrite(pin0,0);
120     digitalWrite(pin1,1);
121     digitalWrite(pin2,0);
122     delay(DL);
123 }
124 if(i == 3)
125 {
126     digitalWrite(pin0,1);
127     digitalWrite(pin1,1);
128     digitalWrite(pin2,0);
129     delay(DL);
130 }
131 if(i == 4)
132 {
133     digitalWrite(pin0,0);
134     digitalWrite(pin1,0);
135     digitalWrite(pin2,1);
136     delay(DL);
137 }
138 if(i == 5)
139 {
140     digitalWrite(pin0,1);
141     digitalWrite(pin1,0);
142     digitalWrite(pin2,1);
143     delay(DL);
144 }
145 if(i == 6)
146 {
147     digitalWrite(pin0,0);
148     digitalWrite(pin1,1);
149     digitalWrite(pin2,1);
150     delay(DL);
151 }
152 if(i == 7)
153 {
154     digitalWrite(pin0,1);
155     digitalWrite(pin1,1);
156     digitalWrite(pin2,1);
157     delay(DL);
158 }
159 }
160 }
161
162
163

```

} return 0;