



Assignment 1

วงจรเปรียบเทียบค่าจำนวน 2 บิต แสดงผลลัพธ์ทาง LED

(2-bit comparator display on LED 3 state)

จัดทำโดย

นางสาวนาตยา บุญญา รหัส 5910110167

เสนอ

อาจารย์ทวีศักดิ์ เรืองพีระกุล

รายวิชา 242-309 MICROCONTROLLER & INTERFAC

ภาคเรียนที่ 2 ปีการศึกษา 2564

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์

อุปกรณ์ที่ใช้ในการทำงาน

1. บอร์ด Arduino UNO R3	1 บอร์ด
2. สาย USB Port	1 เส้น
3. Dip Switch 4 bit	2 ตัว
4. Resistor 330 Ω	7 ตัว
5. Breadboard	2 บอร์ด
6. LED สีแดง	2 ดวง
7. LED สีฟ้า	2 ดวง
8. LED สีเขียว	2 ดวง
9. โปรแกรม AVR Studio	
10. โปรแกรม Xloader	
11. โปรแกรม tinkercad	

อุปกรณ์ที่ใช้ในการเขียนโปรแกรมและจำลองวงจร

- โปรแกรม AVR Studio4 ใช้ในการเขียนโปรแกรมและรันโปรแกรม
- โปรแกรม Xloader ใช้ในการอัปโหลดไฟล์โปรแกรมที่มีนามสกุล .hex ลงบอร์ด Arduino
- โปรแกรม Proteus ใช้ในการจำลองวงจร
- โปรแกรม tinkercad ใช้ในการจำลองวงจร ก่อนลงบนบอร์ดจริง

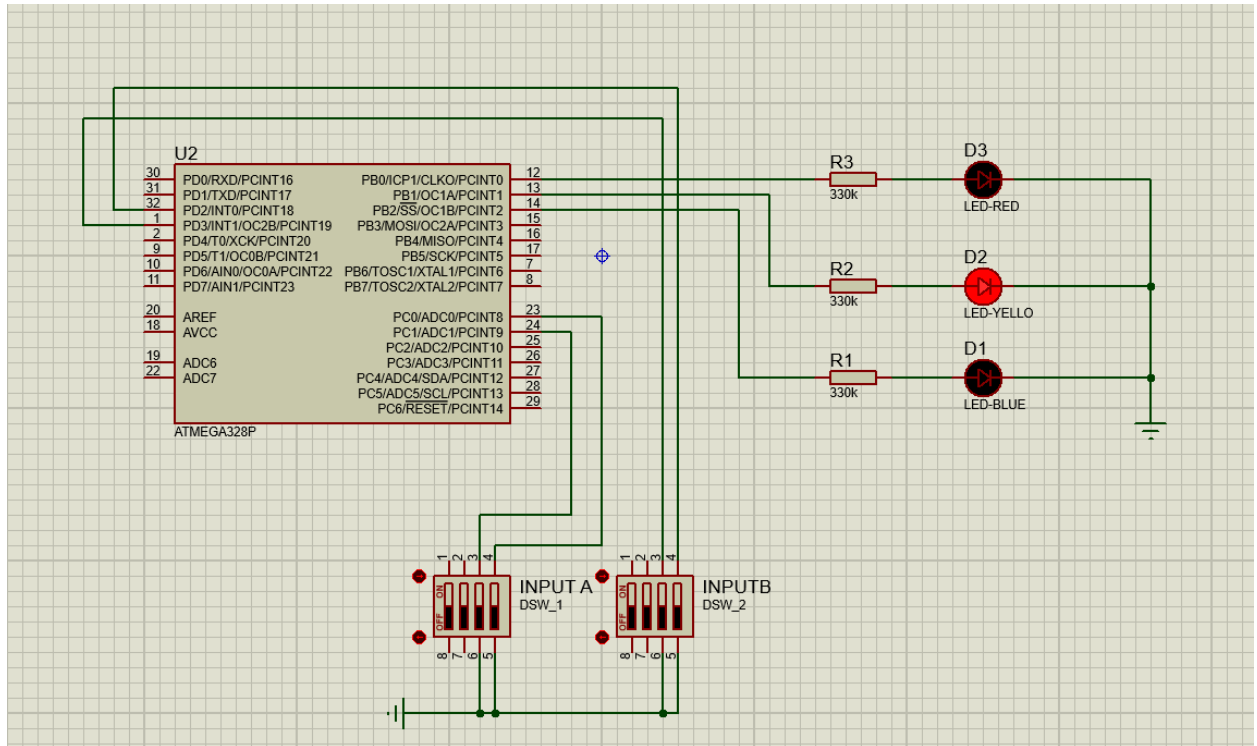
การทำงานของโปรแกรม

1. วงจรทำการรับค่า Input ขนาด 2 บิต จำนวน 2 ชุด ผ่านทาง DIP Switch โดยรับค่า Input A เข้าผ่านพอร์ต C และ Input A พอร์ต D
2. โปรแกรมจะทำการเปรียบเทียบค่าจาก Input ทั้ง 2 ชุด โดยจะแสดงผลลัพธ์ออกมาแตกต่างกัน 3 กรณี ได้แก่ เท่ากับ มากกว่า และน้อยกว่า แสดงผลลัพธ์ออกทางพอร์ต B ออกสู่ LED 3 สี (สีเขียว สีนํ้าเงิน สีแดง ตามลำดับ)

ตารางแสดงการเปรียบเทียบค่า และผลลัพธ์ทุกกรณี

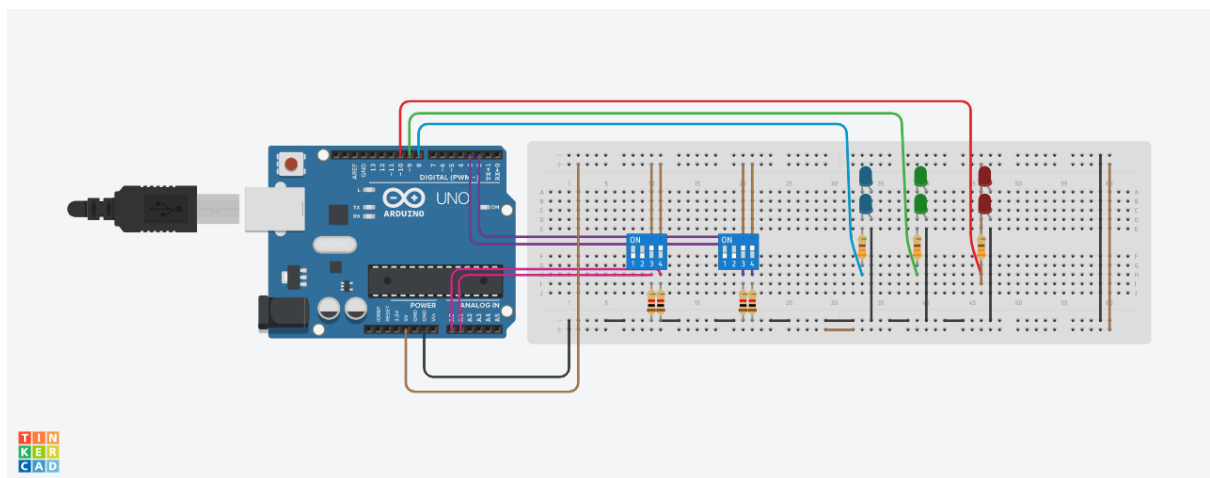
INPUT A	INPUT B	สีของ LED
00	00	เขียว
01	01	เขียว
10	10	เขียว
11	11	เขียว
01	00	น้ำเงิน
10	00	น้ำเงิน
11	00	น้ำเงิน
10	01	น้ำเงิน
11	01	น้ำเงิน
11	10	น้ำเงิน
00	01	แดง
00	10	แดง
01	10	แดง
00	11	แดง
01	11	แดง
10	11	แดง

แบบจำลองวงจรผ่านโปรแกรม Proteus



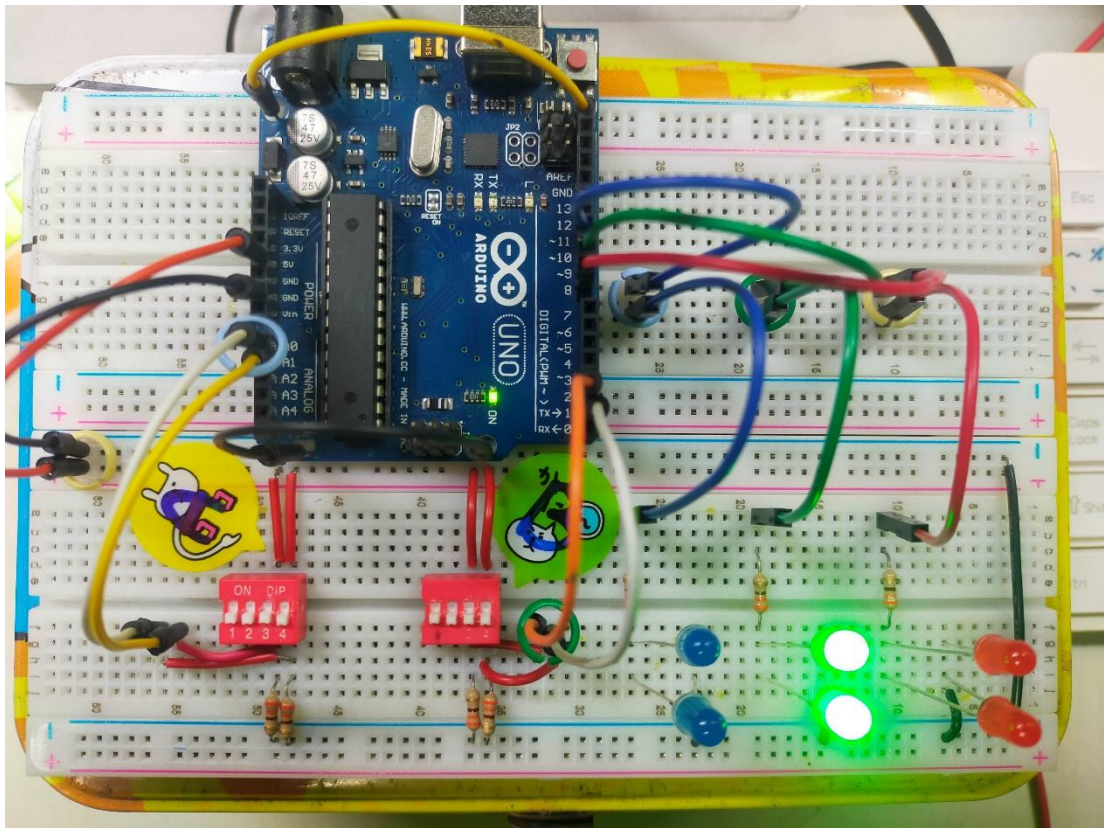
รูปภาพที่ 1 ภาพจำลองวงจรผ่าน โปรแกรม Proteus

แบบจำลองวงจรผ่านโปรแกรม tinkercad

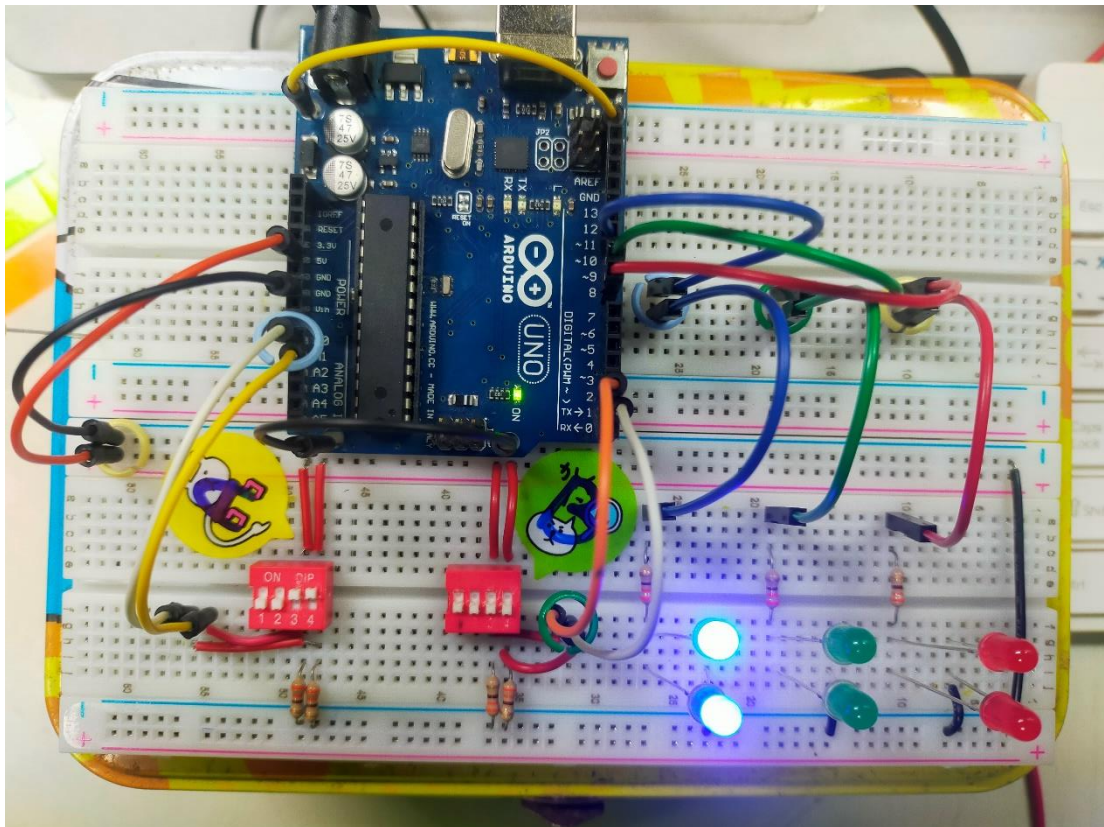


รูปภาพที่ 2 ภาพจำลองวงจรก่อนลงบอร์ดจริง ด้วยโปรแกรม tinkercad

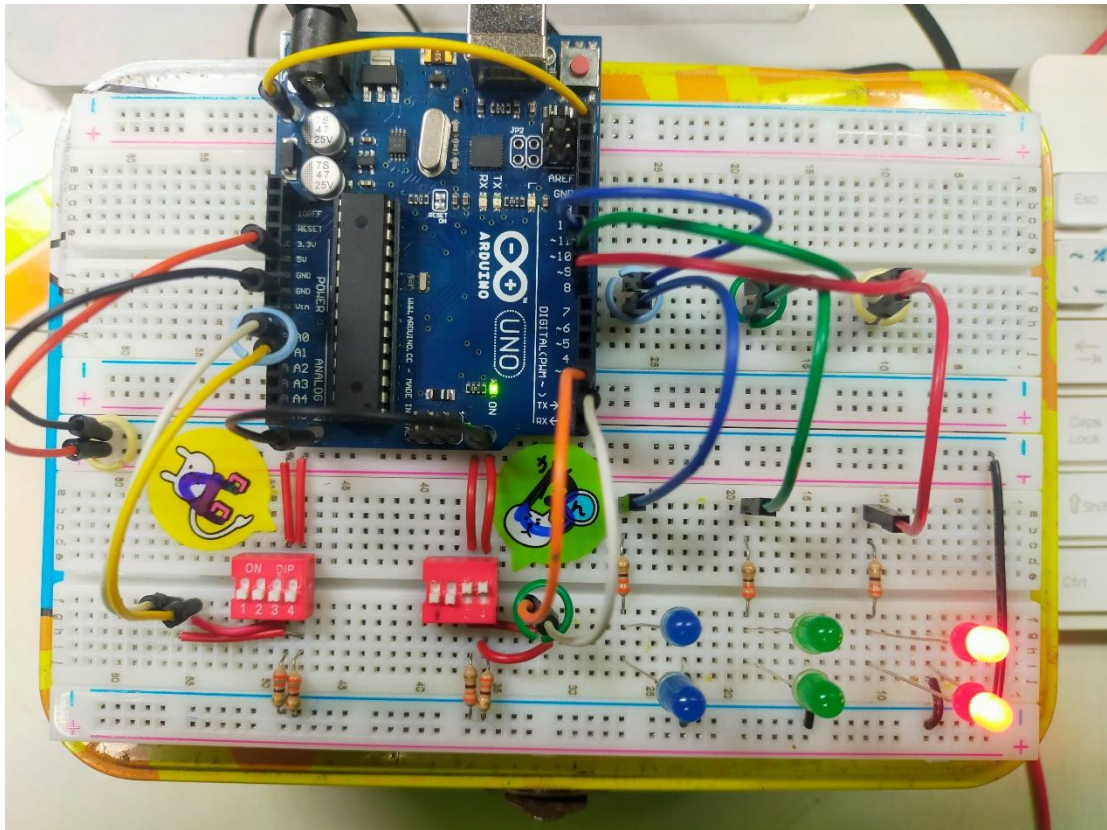
ตัวอย่างวงจรเปรียบเทียบค่า ทั้ง 3 กรณี



รูปภาพที่ 3 เมื่อ $Input A = Input B$ จะแสดงไฟ LED สีเขียว



รูปภาพที่ 4 เมื่อ $Input A > Input B$ จะแสดงไฟ LED สีน้ำเงิน



รูปภาพที่ 5 เมื่อ $Input\ A < Input\ B$ จะแสดงไฟ LED สีแดง

โปรแกรมภาษาแอสเซมบลี

```
.INCLUDE "m328Pdef.inc"
.EQU ALL_PIN_OUT = 0xff
.EQU ALL_PIN_IN = 0x00
.DEF STATUS = R16
.DEF VAR_A = R17
.DEF VAR_B = R18
.CSEG
.ORG 0x0000
    rjmp RESET

RESET:
    ldi STATUS, ALL_PIN_OUT
    out DDRB, STATUS
    ldi VAR_A, ALL_PIN_IN
    out DDRC, VAR_A
    ldi VAR_B, ALL_PIN_IN
    out DDRD, VAR_B

MAIN:
    in     VAR_A, PINC
    ;ldi    VAR_A, 0x03
    andi   VAR_A, 0x03
    in     VAR_B, PIND
    ;ldi    VAR_B, 0x0c
    andi   VAR_B, 0x0c
    lsr    VAR_B
    lsr    VAR_B
    cp     VAR_B, VAR_A
    breq   GREEN

    brge   RED
    jmp    BLUE

// เรียกใช้คลังโปรแกรมชื่อ m328Pdef.inc
// กำหนดให้ ALL_PIN_OUT มีค่าเท่ากับ 0xff
// กำหนดให้ ALL_PIN_OUT มีค่าเท่ากับ 0x00
// กำหนดชื่อสัญลักษณ์ตัวแปร STATUS ให้กับ R16
// กำหนดชื่อสัญลักษณ์ตัวแปร VAR_A ให้กับ R17
// กำหนดชื่อสัญลักษณ์ตัวแปร VAR_B ให้กับ R18
// เริ่มต้นการทำงานในส่วน of code segment
// เริ่มต้นโปรแกรมที่ตำแหน่ง 0x0000
// กระโดดไปยัง RESET

// กำหนดให้ PORT B ทำหน้าที่เป็น Output
// กำหนดให้ PORT C ทำหน้าที่เป็น Input A
// กำหนดให้ PORT D ทำหน้าที่เป็น Input B

// อ่านค่าจากพอร์ต C เก็บในตัวแปร VAR_A
// test
// กรองค่าที่ได้เพื่อให้เหลือเพียง 2 บิตล่างสุด (0b000011)
// อ่านค่าจากพอร์ต C เก็บในตัวแปร VAR_A
// test
// กรองค่าที่ได้เพื่อให้เหลือเพียง 2 บิต (0b00001100)
// shift bit ไปยังตำแหน่งบิตล่างสุด

// เปรียบเทียบค่าใน VAR_B กับ VAR_A
// ถ้า VAR_B เท่ากับ VAR_A กระโดดไปทำงานส่วน GREEN
// ถ้า VAR_B มากกว่า VAR_A กระโดดไปทำงานส่วน RED
// ถ้า VAR_B น้อยกว่า VAR_A กระโดดไปทำงานส่วน BLUE
```

RED:

```
ldi    STATUS,0x01    // กำหนดค่าให้ตัวแปร STATUS เพื่อส่งตรรกะสูงให้ PB0
out    PORTB,STATUS    // นำค่าออกทางพอร์ต B
jmp    END            // กระโดดไปที่ END
```

GREEN:

```
ldi    STATUS,0x02    // กำหนดค่าให้ตัวแปร STATUS เพื่อส่งตรรกะสูงให้ PB1
out    PORTB,STATUS    // นำค่าออกทางพอร์ต B
jmp    END            // กระโดดไปที่ END
```

BLUE:

```
ldi    STATUS,0x04    // กำหนดค่าให้ตัวแปร STATUS เพื่อส่งตรรกะสูงให้ PB2
out    PORTB,STATUS    // นำค่าออกทางพอร์ต B
jmp    END            // กระโดดไปที่ END
```

END:

```
jmp    RESET          // กระโดดไปยัง RESET เพื่อตรวจสอบการทำงานอีกครั้ง
```

.DSEG

.ESEG

วิดีโอการทดลอง : <https://youtu.be/3O7Aox133xw>