

## หน่วยที่ 5

### Arduino กับพอร์ตดิจิตอลอินพุต

#### สาระการเรียนรู้

1. การต่อสวิตช์กับไมโครคอนโทรลเลอร์ Arduino
2. การแก้ปัญหาด้วยฮาร์ดแวร์ ดีเบอซ์ (Hardware Debouncing)
3. การแก้ปัญหาด้วยซอฟต์แวร์ ดีเบอซ์ (Software Debouncing)

#### จุดประสงค์การเรียนรู้

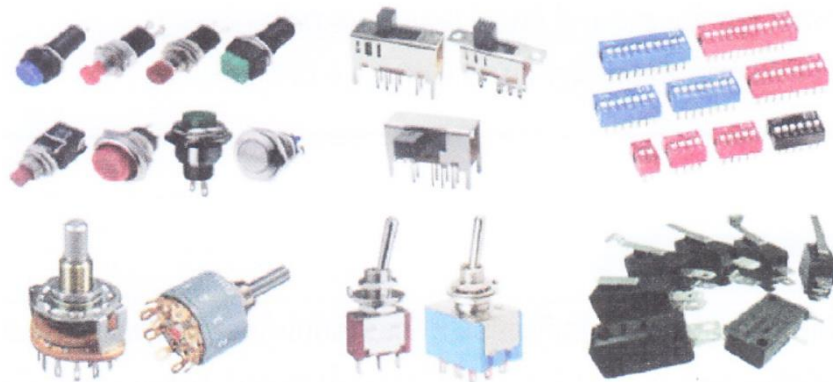
1. เพื่อให้มีความรู้ความเข้าใจเกี่ยวกับการต่อสวิตช์กับไมโครคอนโทรลเลอร์ Arduino
2. เพื่อให้มีทักษะในการเขียนโปรแกรมไมโครคอนโทรลเลอร์ Arduino รับค่าสวิตช์อินพุต
3. เพื่อให้มีกิจนิสัยในการใช้วัสดุ อุปกรณ์ชุดทดลองไมโครคอนโทรลเลอร์ Arduino

#### จุดประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการต่อสวิตช์กับไมโครคอนโทรลเลอร์ Arduino
2. สามารถเขียนโปรแกรมไมโครคอนโทรลเลอร์ Arduino รับค่าสวิตช์อินพุต
3. เตรียมความพร้อมด้านวัสดุ อุปกรณ์สอดคล้องกับงานได้อย่างถูกต้อง

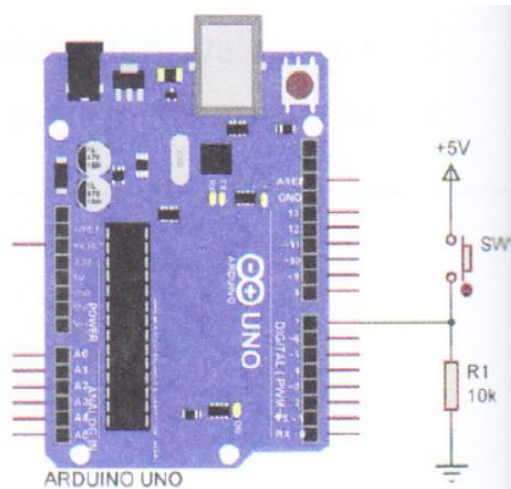
#### การต่อสวิตช์กับไมโครคอนโทรลเลอร์ Arduino

จากคุณสมบัติของพอร์ตดิจิตอลในบอร์ด Arduino ซึ่งมีพอร์ตดิจิตอลทั้งหมด 14 พอร์ตนั้น สามารถกำหนดให้เป็นได้ทั้งพอร์ตเอาต์พุตและอินพุต เมื่อกำหนดให้เป็นพอร์ตอินพุตแล้วสามารถใช้ สำหรับรับค่าสัญญาณไฟฟ้าจากอุปกรณ์อินพุตต่าง ๆ ได้เพื่อนำไปใช้ในการควบคุมการทำงานของ วงจรเอาต์พุตที่ออกแบบไว้ ตัวอย่างอุปกรณ์อินพุต ได้แก่ สวิตช์หรือปุ่มกด สัญญาณอินพุตที่ต่อเข้า พอร์ตไมโครคอนโทรลเลอร์ Arduino มีการเปลี่ยนแปลงสถานะอยู่ 2 สถานะ คือ ลอจิก “1” กับ ลอจิก “0” ตัวอย่างสวิตช์อินพุตแสดงในรูปที่ 5.1



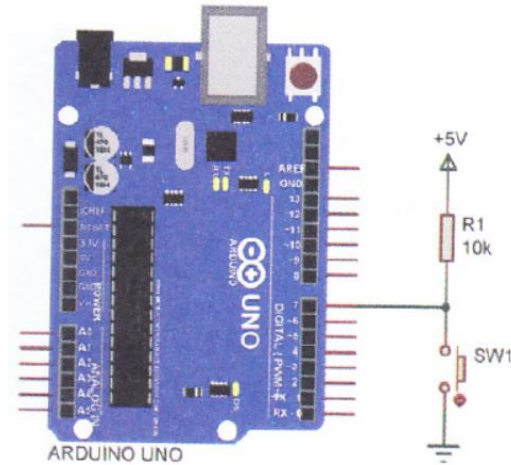
รูปที่ 5.1 สวิตช์อินพุตรูปร่างต่าง ๆ

การต่อสวิตช์เข้าพอร์ตไมโครคอนโทรลเลอร์ Arduino สามารถต่อใช้งานได้ 2 แบบ ได้แก่ การต่อสวิตช์แบบแอคทีฟ ไฮท์ (Active high) และการต่อสวิตช์แบบแอคทีฟ โลว์ (Active low) สำหรับการต่อสวิตช์แบบแอคทีฟ ไฮท์ (Active high) เป็นการต่อจากแหล่งจ่ายไฟผ่านสวิตช์และตัวต้านทานลงกราวด์จุดต่อระหว่างขาสวิตช์กับตัวต้านทานต่อ เข้าที่พอร์ตไมโครคอนโทรลเลอร์ Arduino ลักษณะการต่อ ตัวต้านทานแบบนี้เรียกว่า การต่อตัวต้านทานแบบพูลดาวน์ (Pull - down resistor) แสดงในรูปที่ 5.2 ลักษณะการ ทำงานของวงจรนี้ ขณะยังไม่กดสวิตช์ที่ขาอินพุตของพอร์ต ไมโครคอนโทรลเลอร์มีค่าเป็นลอจิก “0” เมื่อใดที่กดสวิตช์ ที่พอร์ตของไมโครคอนโทรลเลอร์มีค่าเป็นลอจิก “1” ซึ่ง ลักษณะการเปลี่ยนแปลงที่ขาอินพุตนี้นำไปใช้สำหรับการ เขียนโปรแกรมรับค่าอินพุตทางดิจิทัลต่อไป



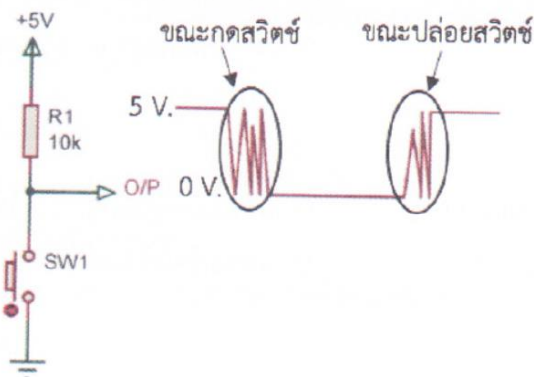
รูปที่ 5.2 การต่อสวิตช์แบบแอคทีฟ ไฮท์ (Active high)

ส่วนการต่อสวิตช์แบบแอคทีฟ โลว์ (Active low) เป็นการต่อจากแหล่งจ่ายไฟผ่านตัวต้านทาน และผ่านสวิตช์ลงกราวด์ จุดต่อระหว่างขาตัวต้านทานกับสวิตช์ต่อเข้าพอร์ตดิจิทัลของไมโคร คอนโทรลเลอร์ Arduino ลักษณะการต่อตัวต้านทาน แบบนี้เรียกว่า การต่อตัวต้านทานแบบพูลอัพ แสดงในรูปที่ 5.3 ลักษณะการทำงาน ของวงจรนี้ ขณะยังไม่กดสวิตช์ที่ขา อินพุตของพอร์ตไมโครคอนโทรลเลอร์มีค่าเป็นลอจิก “1” แต่เมื่อกดสวิตช์ที่ พอร์ตของไมโครคอนโทรลเลอร์มีค่าเป็น ลอจิก “0” ซึ่งลักษณะการเปลี่ยนแปลงที่ขาอินพุตนี้นำไป ใช้สำหรับการ เขียนโปรแกรมรับค่าอินพุตทางดิจิทัลต่อไป จากวงจรทั้ง 2 แบบนี้วงจรที่นิยมนำมาออกแบบสำหรับการ ขอบวงจรร่วมกับไมโครคอนโทรลเลอร์คือวงจรการต่อสวิตช์แบบแอคทีฟ โลว์ (Active low) เนื่องจากช่วยในการป้องกัน สัญญาณรบกวนจากภายนอกได้ดีกว่าการต่อสวิตช์แบบแอคทีฟ ไฮท์ (Active high)



รูปที่ 5.3 การต่อสวิตช์แบบแอกทีฟ โลว์ (Active low)

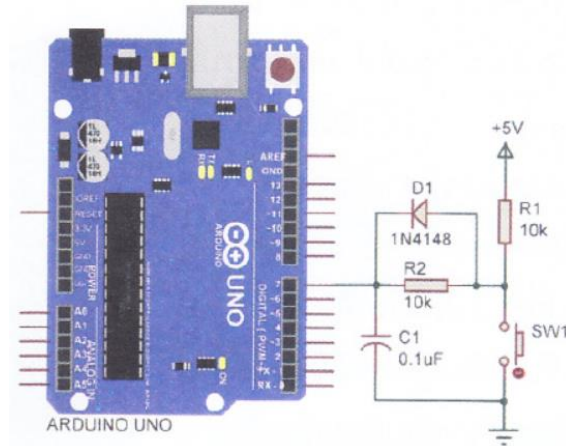
จากการต่อสวิตช์แบบแอกทีฟ ไฮท์ (Active high) หรือแบบแอกทีฟ โลว์ (Active low) มีปัญหาที่เกิดขึ้นในการใช้สวิตช์แบบหน้าสัมผัสคือ ขณะกดสวิตช์และปล่อยสวิตช์นั้นเกิดการแตะกัน ของหน้าสัมผัสหลาย ๆ ครั้งเกิดขึ้น จากปัญหาดังกล่าวทำให้ไมโครคอนโทรลเลอร์สามารถอ่านค่าที่ได้จากสัมผัสหลาย ๆ ครั้งนั้นได้ ซึ่งเป็นสาเหตุที่ทำให้ไมโครคอนโทรลเลอร์ทำงานผิดพลาดตาม โปรแกรมที่เขียนขึ้นได้ แสดงในรูปที่ 5.4



รูปที่ 5.4 การเกิดสัญญาณรบกวน ขณะกดและปล่อยสวิตช์

วิธีการแก้ปัญหาดังกล่าวสามารถแก้ไขได้ดังนี้

1. การแก้ปัญหาด้วยฮาร์ดแวร์ ดีเบานซ์ (Hardware Debouncing)



รูปที่ 5.5 การแก้ปัญหาด้วยฮาร์ดแวร์ ดีเบอซ์

ขณะยังไม่กดสวิตช์ SW1 ตัวเก็บประจุ C1 ทำการชาร์จประจุไว้ ทำให้พอร์ตอินพุตของไมโครคอนโทรลเลอร์มีค่าเป็นลอจิก “1” แต่เมื่อทำการกดสวิตช์ SW1 ตัวเก็บประจุ C1 ทำการคายประจุ ผ่านตัวต้านทาน R2 ซึ่งทำให้แรงดันค่อย ๆ ลดลงเรื่อย ๆ เมื่อแรงดันไฟฟ้ามีค่าต่ำกว่า 1.8 โวลต์ ไมโครคอนโทรลเลอร์จึงมีค่าเป็นลอจิก “0” แต่ถ้าแรงดันไฟฟ้ามีค่ามากกว่า 3.1 โวลต์ ไมโครคอนโทรลเลอร์ จึงมีค่าเป็นลอจิก “1”

ตัวอย่างการเขียนโปรแกรมทดสอบวงจรฮาร์ดแวร์ ดีเบอซ์ (Hardware Debouncing)

```
#define buttonPin 2 // กำหนดตัวแปร buttonPin ต่อที่พอร์ตดิจิตอลขา 2
#define ledPin 13 // กำหนดตัวแปร ledPin ต่อที่พอร์ตดิจิตอลขา 13
void setup() {
    pinMode(buttonPin, INPUT); // เซตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตเอาต์พุต
    pinMode(ledPin, OUTPUT); // เซตพอร์ตดิจิตอลขา 13 ให้เป็นพอร์ตเอาต์พุต
}
void loop() {
    digitalWrite(ledPin, digitalRead(buttonPin)); // อ่านค่าจากอินพุต และนำค่าที่อ่านได้
    // แสดงทางเอาต์พุต
}

```

## 2. การแก้ปัญหาด้วยซอฟต์แวร์ ดีเบอซ์ (Software Debouncing)

การแก้ปัญหาด้วยวิธีนี้สามารถทำได้ง่ายที่สุด คือ เมื่อโปรแกรมทำการตรวจสอบว่าสวิตช์ ถูกกดแล้ว ให้ทำการหน่วงเวลาออกไปประมาณ 10 มิลลิวินาที

ตัวอย่างการเขียนโปรแกรมทดสอบวงจรซอฟต์แวร์ ดีเบานซ์ (Software Debouncing)

```
#define buttonPin 2 // กำหนดตัวแปร buttonPin ต่อที่พอร์ตดิจิตอลขา 2
#define ledPin 13 // กำหนดตัวแปร ledPin ต่อที่พอร์ตดิจิตอลขา 13

void setup() {
    pinMode(buttonPin, INPUT); // เซตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตเอาต์พุต
    pinMode(ledPin, OUTPUT); // เซตพอร์ตดิจิตอลขา 13 ให้เป็นพอร์ตเอาต์พุต
}

void loop() {
    digitalWrite(ledPin, digitalRead(buttonPin)); // อ่านค่าจากอินพุต และนำค่าที่อ่านได้
                                                // แสดงทางเอาต์พุต
    delay(10); // หน่วงเวลา 0.01 วินาที
}
```

การเขียนโปรแกรมเพื่อใช้งานเป็นพอร์ตดิจิตอลอินพุต ต้องใช้คำสั่งรับค่าอินพุตเข้ามายังพอร์ต ของไมโครคอนโทรลเลอร์ Arduino ดังนี้

คำสั่ง `digitalRead()`;

รูปแบบคำสั่ง

`digitalRead (ตำแหน่งพอร์ต)`

เป็นการอ่านค่าสถานะของตำแหน่งพอร์ตดิจิตอลที่ต้องการอ่าน ตั้งแต่พอร์ต 0 ถึง 13 ว่ามีค่าเป็น HIGH หรือ LOW

ตัวอย่างการเขียนโปรแกรม

```
int ledPin = 13; // กำหนดตัวแปร ledPin ต่อที่พอร์ตดิจิตอลขา 13
int inPin = 7; // กำหนดตัวแปร inPin ต่อที่พอร์ตดิจิตอลขา 7
int val = 0; // กำหนดตัวแปร val มีค่าเท่ากับ 0

void setup() {
    pinMode(ledPin, OUTPUT); // เซตพอร์ตดิจิตอลขา 13 ให้เป็นพอร์ตเอาต์พุต
    pinMode (inPin, INPUT); // เซตพอร์ตดิจิตอลขา 7 ให้เป็นพอร์ตอินพุต
}

void loop() {
    val = digitalRead(inPin); // อ่านค่าจากสัญญาณอินพุต
    digitalWrite(ledPin, val); // นำค่าที่อ่านได้แสดงผลที่พอร์ตเอาต์พุต
}
```

## สรุป

ไมโครคอนโทรลเลอร์ Arduino มีพอร์ตดิจิตอลทั้งหมด 14 พอร์ต สามารถกำหนดให้พอร์ต เอาต์พุตและอินพุตได้ขึ้นอยู่กับผู้พัฒนาโปรแกรม เมื่อกำหนดให้เป็นพอร์ตอินพุตสามารถรับ ค่าสัญญาณไฟฟ้าจากอุปกรณ์อินพุตต่าง ๆ ได้แก่ สวิตช์ หรือปุ่มกด สัญญาณอินพุตที่ต่อเข้ามายัง ไมโครคอนโทรลเลอร์มีการเปลี่ยนแปลงสถานะ 2 สถานะคือ ลอจิก “0” และลอจิก “1” การต่อใช้งานสวิตช์นิยมต่อแบบแอคทีฟ โลว์ (Active low) คำสั่งสำหรับการอ่านค่าดิจิตอลอินพุตคือ คำสั่ง `digitalRead()` ;