

บทที่ 9

เรื่อง การใช้โปรแกรมควบคุมหุ่นยนต์บังคับด้วยมือ

9.1 ลักษณะการทำงานของสวิตช์

บอร์ด Arduino มี Digital I/O PINs สำหรับใช้งาน ซึ่งคำว่า I/O หมายถึง Input และ Output นั้นหมายความว่า นอกจาก Pins เหล่านี้จะสามารถเป็น Output เพื่อไปควบคุมอุปกรณ์ต่างๆได้แล้ว มันยังสามารถใช้เป็นตัว Input หรือตัวรับสัญญาณทางไฟฟ้าต่างๆ เพื่อใช้ควบคุมการทำงานของวงจรได้อีกด้วย

การที่เราใช้ Digital Pins ของ Arduino เป็นตัวรับค่าสัญญาณ คือการที่เราให้ Pins นั้น ๆ เป็นตัว Input โดยค่าที่ Arduino อ่านได้จะมีอยู่ 2 ค่า คือ HIGH และ LOW ขึ้นอยู่กับ Pin นั้น มีสัญญาณหรือกระแสไฟฟ้าเข้ามาหรือไม่

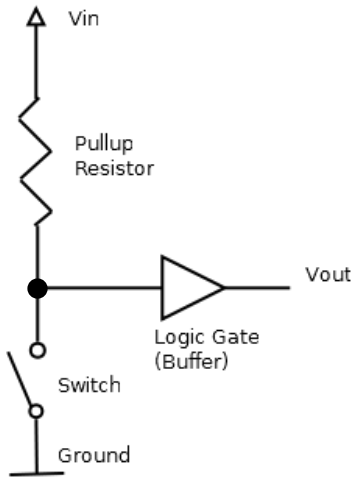
สถานะที่กำหนดให้ขาของอุปกรณ์อิเล็กทรอนิกส์ รอร์บอินพุต (INPUT) ขาพอร์ตจะเป็น High Impedance คือมีความต้านทานสูงมากต่ออยู่ ทำให้ขาพอร์ตนั้นเสมือนถูกปล่อยลอย ค่าอินพุตที่อ่านกลับมาได้ไม่แน่นอน

ดังนั้นในงานขาพอร์ตอินพุต วงจรของสวิตช์จำเป็นมากที่ต้องมี Pull-up Resistor หรือ Pull-down Resistor เพื่อที่จะกำหนดสถานะดิจิทัลที่แน่นอนให้กับอุปกรณ์อิเล็กทรอนิกส์ ตามปกติตัวต้านทานที่ใช้ในวงจร Pull-up หรือ Pull-down จะใช้ประมาณ 5k Ohm - 20k Ohm

9.2 การต่อใช้งานสวิตช์

9.2.1 วงจร Pull-up Resistor

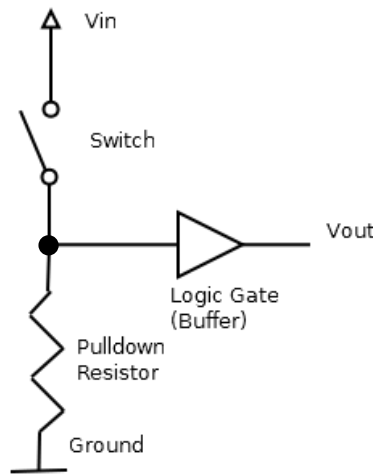
Pull-up Resistor คือการนำตัวต้านทานต่อเข้ากับ Vcc (+5V) เพื่อให้แรงดันอยู่คงที่ ทำให้อยู่ในสถานะ “HIGH” หรือ “1” ตลอดเวลา และเมื่อกดสวิตช์กระแสไฟฟ้าจะไหลลง Ground ทันที ซึ่งทำให้สถานะเป็นลอจิก “LOW” หรือ “0” และ การทำงานลักษณะนี้จะเรียกว่า Active Low เพราะว่าเขียนโปรแกรมที่ทำงาน เมื่อลอจิกเป็น “LOW” ส่วนใหญ่ เราจะเห็นต่อสวิตช์นิยมใช้แบบ Pull-up มากกว่า



รูปที่ 9.1 วงจร Pull-up Resistor

9.2.2 วงจร Pull-down Resistor

Pull-down Resistor โดยใน Pull-down จะมีลักษณะคล้ายกับ Pull-up Resistor แตกต่างตรงที่ สภาวะปกติของ Pull-down จะเป็นลอจิก “LOW” หรือ “0” เมื่อมีการกดปุ่ม กระแสไฟจะไหลเข้าขาอินพุต ทำให้ลอจิกเป็น “HIGH” หรือ “1” ได้ การทำงานในลักษณะนี้จะเรียกว่า Active High



รูปที่ 9.2 วงจร Pull-down Resistor

9.3 คำสั่งที่ใช้สำหรับอ่านค่าจากสวิตช์

เนื่องจากสวิตช์เป็นอุปกรณ์ที่ส่งข้อมูลมายังไมโครคอนโทรลเลอร์ ดังนั้นในการอ่านค่าจากสวิตช์ จะต้องกำหนดสถานะของขาดีจิตอลบนบอร์ด Arduino ที่ต่ออยู่กับสวิตช์และใช้คำสั่งเพื่ออ่านค่าที่ได้ โดยสามารถใช้คำสั่งได้ดังนี้

```
pinMode(PIN,MODE);
```

คำสั่ง pinMode เป็นคำสั่งที่ใช้ในการกำหนดสถานะขาของ Arduino โดยจะมีอยู่ 2 โหมด คือ Input และ Output ซึ่งโหมด Input ก็คือกำหนดให้ค่านั้นใช้สำหรับการรับค่า ส่วน Output คือกำหนดให้ขา นั้นใช้ในการส่งค่า จากตัวอย่างเช่น pinMode(2,INPUT);

```
digitalRead(PIN);
```

คำสั่งนี้จะใช้ในการอ่านค่า Digital จากขาที่เราต้องการ เช่น หากต้องการที่จะอ่านค่าขา D2 จะเขียนได้เป็น digitalRead(2);

โดยค่าที่ได้จากคำสั่งนี้ ก็จะมีแค่ 0 หรือ 1 เท่านั้น การนำไปใช้งานต้องกำหนดตัวแปรสำหรับเก็บค่า ชื่อ buttonState ก็เขียนได้ในลักษณะนี้

```
int buttonState;
```

```
buttonState = digitalRead(2);
```