

## หน่วยที่ 9 การควบคุมเซอร์โวมอเตอร์ด้วย ARDUINO

### สาระสำคัญ

Servo เป็นคำศัพท์ที่ใช้กันทั่วไปในระบบควบคุมอัตโนมัติ มาจากภาษาละตินคำว่า Sevus หมายถึง “ทาส” (Slave) ในเชิงความหมายของ Servo Motor ก็คือ Motor ที่เราสามารถสั่งงานหรือตั้งค่า แล้วตัว Motor จะหมุนไปยังตำแหน่งองศาที่เราสั่งได้เองอย่างถูกต้อง โดยใช้การควบคุมแบบป้อนกลับ (Feedback Control) ในหน่วยนี้ จะกล่าวถึง RC Servo Motor ซึ่งนิยมนำมาใช้ในเครื่องบินที่บังคับด้วยคลื่นวิทยุ (RC = Radio - Controlled) เช่น เรือบังคับวิทยุ รถบังคับวิทยุ เฮลิคอปเตอร์บังคับวิทยุ เป็นต้น

### เนื้อหาสาระการเรียนรู้

- 9.1 ความรู้เบื้องต้นเกี่ยวกับ Servo Motor
- 9.2 ส่วนประกอบภายใน RC Servo Motor
- 9.3 บล็อกไดอะแกรมของ Servo Motor
- 9.4 หลักการทำงานของ RC Servo Motor
- 9.5 วิธีควบคุม RC Servo Motor ด้วย Arduino
- 9.6 การเชื่อมต่อ RC Servo Motor เข้ากับบอร์ด Arduino

### จุดประสงค์การเรียนรู้

#### จุดประสงค์ทั่วไป

1. เพื่อให้มีความรู้ความเข้าใจเกี่ยวกับลักษณะการควบคุมเซอร์โวมอเตอร์ด้วย Arduino
2. เพื่อให้สามารถนำความรู้ไปประยุกต์ใช้ในการเขียนโปรแกรมกำหนดการทำงานพื้นฐานด้วย Arduino
3. เพื่อให้ตระหนักถึงความสำคัญของลักษณะการควบคุมเซอร์โวมอเตอร์ด้วย Arduino

## จุดประสงค์เชิงพฤติกรรม

1. อธิบายความรู้เบื้องต้นเกี่ยวกับ Servo Motor ได้
2. อธิบายส่วนประกอบภายใน RC Servo Motor ได้
3. อธิบายบล็อกไดอะแกรมของ RC Servo Motor ได้
4. บอกหลักการทำงานของ RC Servo Motor ได้
5. ควบคุม RC Servo Motor ด้วย Arduino ได้
6. ควบคุมตำแหน่ง RC Servo Motor โดยใช้ Potentiometer ได้

## แบบทดสอบก่อนเรียน/หลังเรียน หน่วยที่ 9

### เรื่อง การคุมเซอร์โวมอเตอร์ด้วย Arduino

เรื่อง การคุมเซอร์โวมอเตอร์ด้วย Arduino

ใช้เวลา 20 นาที

วิชา ไมโครคอนโทรลเลอร์เบื้องต้น

รหัสวิชา (2127-2107)

ระดับชั้น ประกาศนียบัตรวิชาชีพ (ปวช.)

สาขาวิชา เมคคาทรอนิกส์

\*\*\*\*\*

คำชี้แจง 1. แบบทดสอบมีทั้งหมด 10 ข้อ (10 คะแนน)

2. ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ลงในกระดาษคำตอบ

1. Servo มาจากภาษาละตินจากคำว่า Sevus หมายถึงอะไร

- ก. อัตโนมติ
- ข. มอเตอร์
- ค. เจ้านาย
- ง. ทาส

2. RC Servo Motor ซึ่งนิยมนำมาใช้กับอะไร

- ก. เครื่องเล่นที่บังคับด้วยคลื่นโทรศัพท์
- ข. เครื่องเล่นที่บังคับด้วยคลื่นโทรทัศน์
- ค. เครื่องเล่นที่บังคับด้วยคลื่นวิทยุ
- ง. เครื่องเล่นที่บังคับด้วยคลื่นไมโครเวฟ

3. Gear Train หรือ Gearbox คือ

- ก. เป็นชุดเกียร์เพิ่มแรง
- ข. เป็นชุดเกียร์ทดแรง
- ค. เป็นส่วนที่ควบคุมและประมวลผล
- ง. เป็นส่วนของตัวมอเตอร์

4. Electronic Control System คือ

- ก. เป็นส่วนที่ควบคุมและประมวลผล
- ข. เป็นส่วนของตัวมอเตอร์
- ค. จุดเชื่อมต่อสายไฟ
- ง. เซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน

5. Position Sensor เป็นตัวเซ็นเซอร์ที่วัดค่ามุมที่กำลังหมุนส่งกลับมา เพื่ออะไร
  - ก. ประมวลผลค่าความกว้างของสัญญาณพัลส์
  - ข. ขับ Gear Train
  - ค. ขับ Connector
  - ง. ให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการ
6. กำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor หมุนแบบใด
  - ก. ไปทางซ้ายจนสุด
  - ข. ไปทางซ้ายจนสุด
  - ค. อยู่ที่ตำแหน่งตรงกลาง
  - ง. เดินหน้าจนสุด
7. ที่ 1 องศาความกว้างพัลส์เท่ากับ 5.55 us ที่มุม 45 องศา หาค่าความกว้างพัลส์ที่ต้องการได้เท่าใด
  - ก. 274.75 us
  - ข. 249.75 us
  - ค. 222.75 us
  - ง. 107.75 us
8. ฟังก์ชันของ Arduino ภายใน Servo Library ชื่อ attach() คือ
  - ก. ฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ต้องการ
  - ข. ฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ให้ Servo Motor หมุนไปยังตำแหน่งองศาที่กำหนด
  - ค. ฟังก์ชันอ่านค่าองศาที่ส่งเข้าไปด้วยฟังก์ชัน write()
  - ง. ฟังก์ชันที่ใช้ในการกำหนดขาสัญญาณ
9. ฟังก์ชันของ Arduino ภายใน Servo Library ชื่อ read() คือ
  - ก. ฟังก์ชันชะลอการทำงานในหน่วยวินาที
  - ข. ฟังก์ชันชะลอการทำงานในหน่วยไมโครวินาที
  - ค. ฟังก์ชันอ่านค่าองศาที่ส่งเข้าไปด้วยฟังก์ชัน write()
  - ง. ฟังก์ชันชะลอการทำงานในหน่วยนาฬิกา
10. Servo Library ของ Arduino สามารถสั่งงาน RC Servo Motor ได้แบบใดบ้าง
  - ก. 0-180 องศา และ หมุนครบรอบ
  - ข. 0-120 องศา และ หมุนครบรอบ
  - ค. 0-90 องศา และ หมุนครบรอบ
  - ง. 0-45 องศา และ หมุนครบรอบ

## หน่วยที่ 9

### การคุมเซอร์โวมอเตอร์ด้วย Arduino

#### 9.1 ความรู้เบื้องต้นเกี่ยวกับ Servo Motor

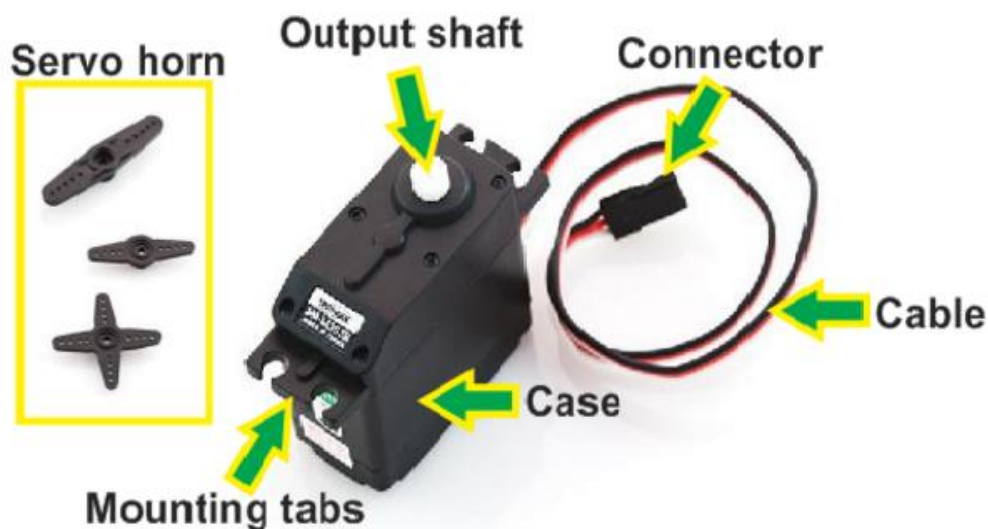
Servo เป็นคำศัพท์ที่ใช้กันทั่วไปในระบบควบคุมอัตโนมัติ มาจากภาษาละตินคำว่า Sevus หมายถึง “ทาส” (Slave) ในเชิงความหมายของ Servo Motor ก็คือ Motor ที่เราสามารถสั่งงานหรือตั้งค่า แล้วตัว Motor จะหมุนไปยังตำแหน่งองศาที่เราสั่งได้เองอย่างถูกต้อง โดยใช้การควบคุมแบบป้อนกลับ (Feedback Control) ในหน่วยนี้จะกล่าวถึง RC Servo Motor ซึ่งนิยมนำมาใช้ในเครื่องเล่นที่บังคับด้วยคลื่นวิทยุ (RC = Radio - Controlled) เช่น เรือบังคับวิทยุ รถบังคับวิทยุ เฮลิคอปเตอร์บังคับวิทยุ เป็นต้น



#### รูปที่ 9.1 ลักษณะโดยทั่วไปของเซอร์โวมอเตอร์

(ที่มา [www.Pyroelectro.com/](http://www.Pyroelectro.com/))

Feedback Control คือระบบควบคุมที่มีการนำค่าเอาต์พุตของระบบนำมาเปรียบเทียบกับค่าอินพุต เพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่าเท่ากับหรือใกล้เคียงกับค่าอินพุต ส่วนประกอบภายนอก RC Servo Motor



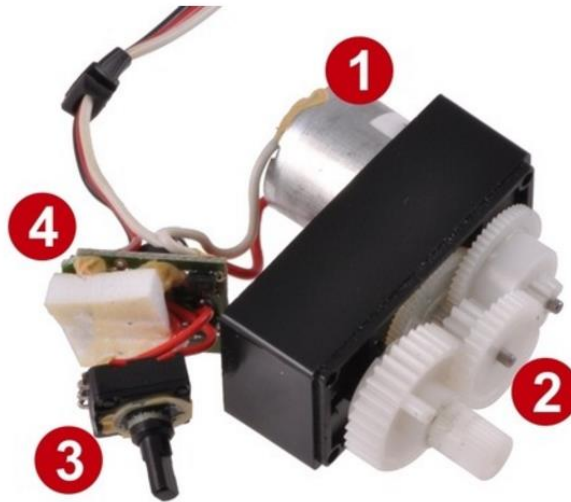
รูปที่ 9.2 ส่วนประกอบของเซอร์โวมอเตอร์  
(ที่มา [www.Arduino.cc/en/reference/servo](http://www.Arduino.cc/en/reference/servo))

- Case ตัวถัง หรือ กรอบของตัว Servo Motor
- Mounting Tab ส่วนจับยึดตัว Servo กับชิ้นงาน
- Output Shaft เฟลาส่งกำลัง
- Servo Horns ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไก
- Cable สายเชื่อมต่อเพื่อ จ่ายไฟฟ้า และ ควบคุม Servo Motor จะประกอบด้วยสายไฟ 3 เส้น และ

ใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้

- สายสีแดง คือ ไฟเลี้ยง (4.8-6V)
- สายสีดำ หรือ น้ำตาล คือ กราวด์
- สายสีเหลือง (ส้ม ขาว หรือฟ้า) คือ สายส่งสัญญาณพัลส์ควบคุม (3-5V)
- Connector จุดเชื่อมต่อสายไฟ

## 9.2 ส่วนประกอบภายใน RC Servo Motor



รูปที่ 9.3 ส่วนประกอบภายใน RC Servo Motor  
(ที่มา [www.Pololu.com](http://www.Pololu.com))

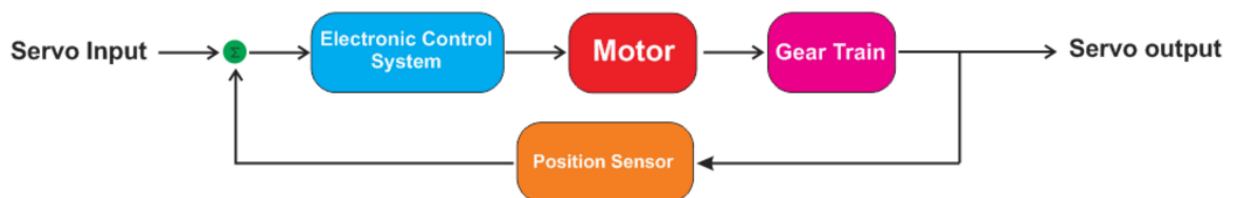
หมายเลข 1 คือ ดีซีมอเตอร์

หมายเลข 2 คือ ชุดเฟืองเกียร์ทดแรงมอเตอร์

หมายเลข 3 คือ ชุดควบคุมและปรับแต่งให้ค่าเอาต์พุตมีค่าเท่ากับ หรือใกล้เคียงกับค่าอินพุต

หมายเลข 4 คือ ส่วนที่ควบคุมและประมวลผล

## 9.3 บล็อกไดอะแกรมของ Servo Motor



รูปที่ 9.5 Block Diagram Servo Motor  
(ที่มา [www.Ebldc.com/?p=48](http://www.Ebldc.com/?p=48))

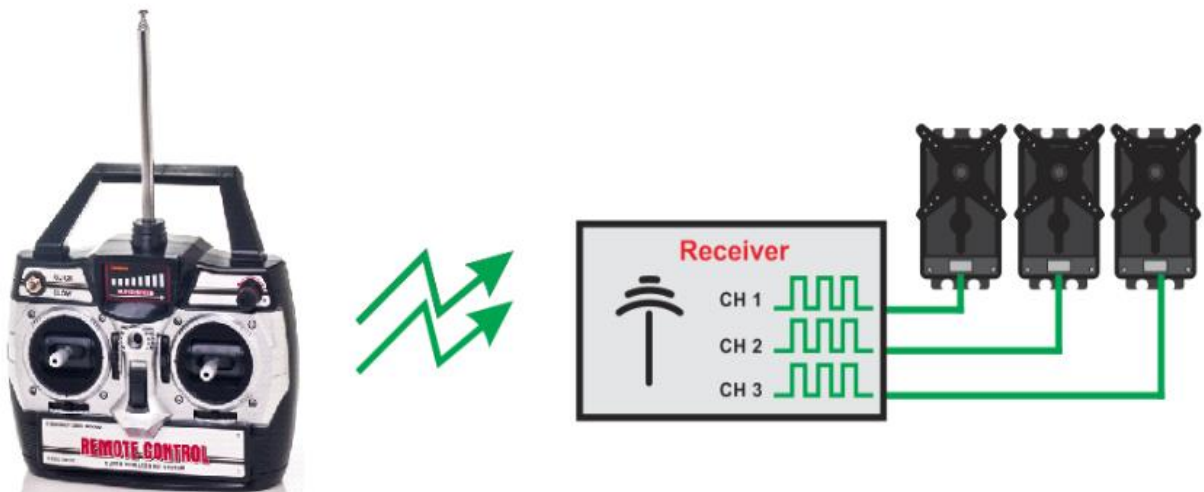
- Motor เป็นส่วนของตัวมอเตอร์
- Gear Train หรือ Gearbox เป็นชุดเกียร์ทดแรง
- Position Sensor เป็นเซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
- Electronic Control System เป็นส่วนที่ควบคุมและประมวลผล

#### 9.4 หลักการทำงานของ RC Servo Motor

เมื่อจ่ายสัญญาณพัลส์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุนเป็น Feedback กลับมาให่วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ

สัญญาณ RC ในรูปแบบ PWM

ตัว RC Servo Motor ออกแบบมาใช้สำหรับรับคำสั่งจาก Remote Control ที่ใช้ควบคุมของเล่นด้วยสัญญาณวิทยุต่างๆ เช่นเครื่องบินบังคับ รถบังคับ เรือบังคับ เป็นต้น ซึ่ง Remote จำพวกนี้ที่ภาครับจะแปลงความถี่วิทยุออกมาในรูปแบบสัญญาณ PWM (Pulse Width Modulation)



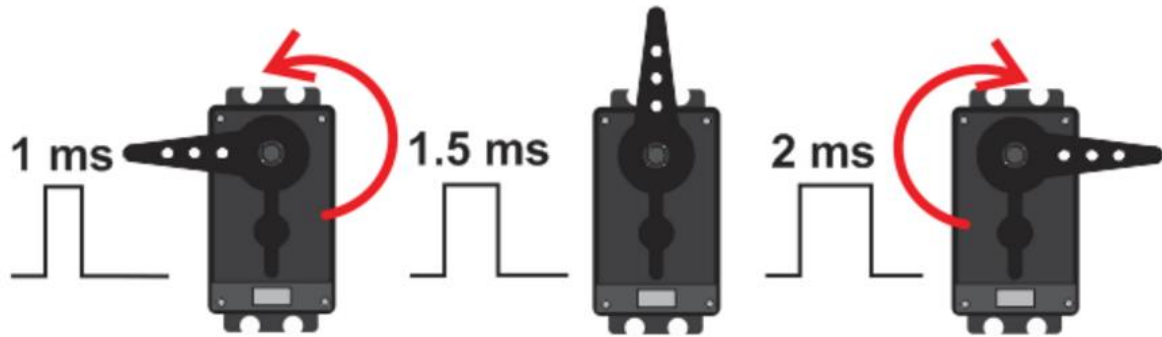
รูปที่ 9.6 Remote Control RC Servo Motor

(ที่มา [www.Learn.parallax.com/KickStart/900-00008](http://www.Learn.parallax.com/KickStart/900-00008))

มุมหรือองศาจะขึ้นอยู่กับความกว้างของสัญญาณพัลส์ ซึ่งโดยส่วนมากความกว้างของพัลส์ที่ใช้ใน RC Servo Motor จะอยู่ในช่วง 1-2 ms หรือ 0.5-2.5 ms



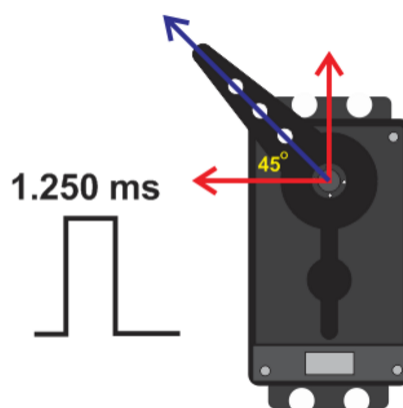
ตัวอย่างเช่นหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms ตัว Servo Motor จะหมุนไปทางซ้ายสุด ในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor ก็จะหมุนมาอยู่ที่ตำแหน่งตรงกลางพอดี



รูปที่ 9.7 มุมหรือองศาหมุนขึ้นอยู่กับความกว้างของสัญญาณพัลส์  
(ที่มา [www.Learn.parallax.com/KickStart/900-00008](http://www.Learn.parallax.com/KickStart/900-00008))

ดังนั้นสามารถกำหนดองศาการหมุนของ RC Servo Motor ได้โดยการเทียบค่า เช่น RC Servo Motor สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ความกว้างพัลส์เท่ากับ 1000 us ที่ 180 องศาความกว้างพัลส์เท่ากับ 2000 us เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ความกว้างพัลส์ต่างกัน  $(2000-1000)/180$  เท่ากับ 5.55 us

จากการหาค่าความกว้างพัลส์ที่มุม 1 องศาข้างต้น หากต้องกำหนดให้ RC Servo Motor หมุนไปที่มุม 45 องศาจะหาค่าพัลส์ที่ต้องการได้จาก  $5.55 \times 45$  เท่ากับ 249.75 us แต่ที่มุม 0 องศาเราเริ่มที่ความกว้างพัลส์ 1ms หรือ 1000 us เพราะฉะนั้นความกว้างพัลส์ที่ใช้กำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา คือ  $1000 + 249.75$  เท่ากับประมาณ 1250 us



รูปที่ 9.8 RC Servo Motor หมุนไปที่ 45 องศา  
(ที่มา [www.Learn.parallax.com/KickStart/900-00008](http://www.Learn.parallax.com/KickStart/900-00008))

## 9.5 วิธีควบคุม RC Servo Motor ด้วย Arduino

Arduino มีไลบรารีสำหรับสั่งงาน RC Servo Motor มาให้ใช้งานอยู่แล้ว เป็นฟังก์ชันสำเร็จรูปและใช้งานได้ง่าย ในหน้าเว็บไซต์ <http://arduino.cc/en/reference/servo> ได้ให้ข้อมูลไว้ว่า Servo Library ของ Arduino สามารถสั่งงาน RC Servo Motor ได้ทั้งแบบหมุนไป-กลับได้ 0-180 องศา (ที่กล่าวถึงตามตัวอย่างข้างต้น) และแบบต่อเนื่องที่หมุนครบรอบได้เรียกว่าเป็น Continuous Rotation Servo

โดยสามารถรองรับการเชื่อมต่อ RC Servo Motor ได้ถึง 12 ตัวกับบอร์ด Arduino UNO และรองรับสูงสุดถึง 48 ตัวหากใช้บอร์ด Arduino Mega

### ฟังก์ชันภายใน Servo Library

- attach()
- write()
- writeMicroseconds()
- read()
- attached()
- detach()

#### 9.5.1 attach()

##### Description

คือฟังก์ชันที่ใช้ในการกำหนดขาสัญญาณที่ Servo Motor ต่อกับ Arduino และกำหนดความกว้างของพัลส์ที่ 0 องศาและ 180 องศา

##### Syntax

```
Servo.attach(pin)
```

```
Servo.attach(pin,min,max)
```

##### Parameters

Pin: คือ ขาสัญญาณของ Arduino ที่ใช้เชื่อมต่อกับ Servo Motor

Min: คือ ความกว้างของพัลส์ที่ 0 องศาของ Servo ตัวที่ใช้ในหน่วยไมโครวินาที (us) โดยปกติแล้วหากไม่มีการตั้งค่าโปรแกรมจะกำหนดค่าไว้ที่ 544 us

Max: คือ ความกว้างของพัลส์ที่ 180 องศาของ Servo ตัวที่ใช้ในหน่วยไมโครวินาที (us) โดยปกติแล้วหากไม่มีการตั้งค่าโปรแกรมจะกำหนดค่าไว้ที่ 2400 us

### 9.5.2 Write()

#### Description

คือฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ต้องการให้ Servo Motor หมุนไปยังองศาที่กำหนดสามารถกำหนดเป็นค่าองศาได้ คือ 0-180 องศา แต่ใน Servo Motor ที่เป็น Full Rotation คำสั่ง write จะเป็นการกำหนดความเร็วในการหมุน โดย

ค่าเท่ากับ 90 คือคำสั่งให้ Servo Motor หยุดหมุน

ค่าเท่ากับ 0 คือการหมุนด้วยความเร็วสูงสุดในทิศทางหนึ่ง

ค่าเท่ากับ 180 คือการหมุนด้วยความเร็วสูงสุดในทิศทางตรงกันข้าม

#### Syntax

```
servo.write(angle)
```

#### Parameters

Angle: คือมุมที่ต้องการให้ RC Servo Motor แบบ 0-180 องศาหมุนไป แต่หากเป็น RC Servo Motor แบบ Full Rotation ค่า Angle คือ การกำหนดความเร็วและทิศทางในการหมุน

### 9.5.3 writeMicroseconds()

#### Description

คือฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ให้ Servo Motor หมุนไปยังตำแหน่งองศาที่กำหนดโดยกำหนดเป็นค่าความกว้างของพัลส์ในหน่วย us ปกติแล้ว RC Servo Motor จะใช้ความกว้างของพัลส์อยู่ที่ 1000-2000 us ตามที่ได้กล่าวไปข้างต้นแล้ว แต่ RC Servo Motor บางรุ่นหรือบางยี่ห้อไม่ได้ใช้ช่วงความกว้างของพัลส์ตามที่ได้กล่าวเอาไว้ อาจจะใช้ช่วง 700-2300 แทนก็สามารถใช้ฟังก์ชัน writeMicroseconds นี้เพื่อกำหนดความกว้างพัลส์ได้เอง

การใช้ฟังก์ชัน writeMicroseconds สามารถกำหนดค่าได้อิสระ ต้องระวังในการใช้งาน หากสั่งงาน RC Servo Motor (แบบ 0 - 180 องศา) จนหมุนไปเกินจุดสิ้นสุดคือเกินทั้งฝั่ง 0 หรือ 180 องศา จะทำให้เกิดเสียงครางดังจากการหมุนไปต่อไม่ได้และมอเตอร์จะกินกระแสสูงขึ้นด้วยในเวลาเดียวกันนั้น ซึ่งอาจทำให้ RC Servo Motor เกิดความเสียหายได้

#### Syntax

```
servo.writeMicroseconds(uS)
```

#### Parameters

uS: คือค่าความกว้างของพัลส์ที่ต้องการกำหนดในหน่วยไมโครวินาที (โดยตัวแปร int)

### 9.5.4 read()

#### Description

คือฟังก์ชันอ่านค่าองศาที่ส่งเข้าไปด้วยฟังก์ชัน write() เพื่อให้รู้ว่าตำแหน่งองศาสุดท้าย ที่ส่งเข้าไปนั้นมีค่าเท่าไร ซึ่งค่าที่อ่านออกมานั้นจะมีค่าอยู่ในช่วง 0 - 180

#### Syntax

```
servo.read()
```

#### Parameters

ไม่มี: จะ Return ค่า 0-180

### 9.5.5 attached()

#### Description

คือฟังก์ชันตรวจสอบว่า Servo ที่ต้องการใช้กำลังต่ออยู่กับขาสัญญาณของ Arduino หรือไม่

#### Syntax

```
servo.attached()
```

#### Parameters

ไม่มี: จะ Return ค่า True ออกมาหาก Servo Motor เชื่อมต่ออยู่กับ Arduino แต่ถ้าหาก Return ออกมาเป็นค่าอื่นถือว่าไม่เชื่อมต่อ

### 9.5.6 detach()

#### Description

คือฟังก์ชันคืนสถานะของขาที่เรากำหนดให้เป็นขาควบคุม Servo Motor ด้วยคำสั่ง attached() ใ้กลับคือสู่การใช้งานปกติ

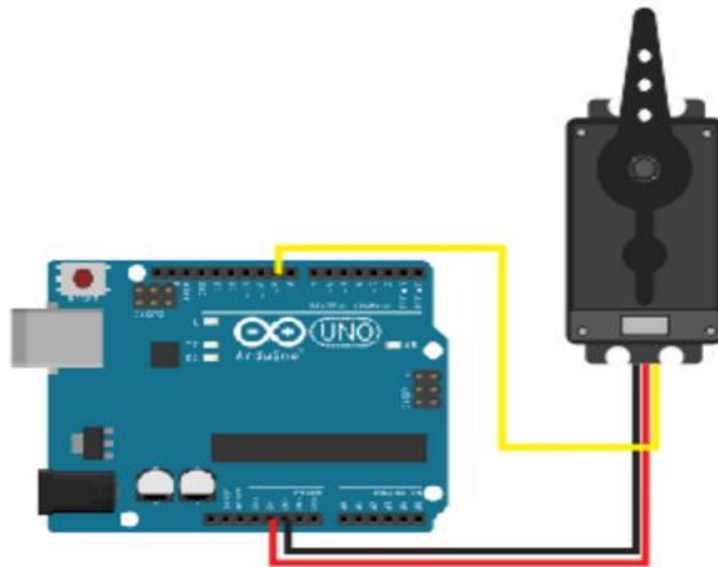
#### Syntax

```
servo.detach()
```

#### Parameters

ไม่มี

## 9.6 การเชื่อมต่อ RC Servo Motor เข้ากับบอร์ด Arduino



รูปที่ 9.9 EFDV245 Futaba S3003 Servo Motor เชื่อมต่อกับบอร์ด Arduino  
(ที่มา [sparkfun.com/tutorials/283](http://sparkfun.com/tutorials/283))

### โปรแกรมที่ 9.1 การควบคุมตำแหน่ง RC Servo Motor

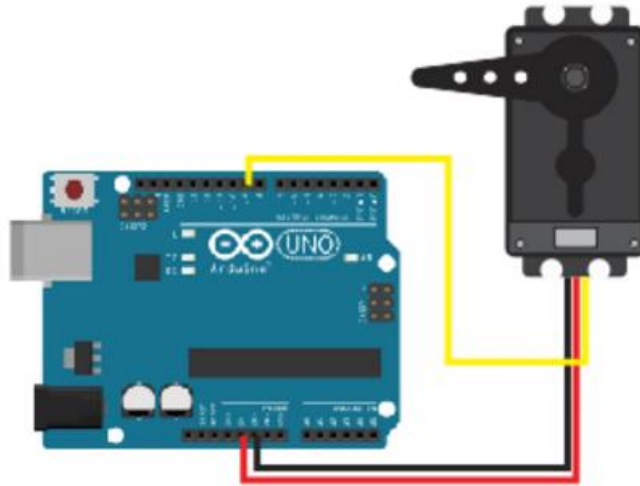
```
#include <Servo.h>

Servo myservo;

void setup()
{
  myservo.attach(9);
}

void loop() {
  myservo.write(0);
  delay(1000);
  myservo.write(90);
  delay(1000);
  myservo.write(180);
  delay(1000); }
```

เมื่ออัปโหลดโปรแกรมเรียบร้อยแล้ว ผลการควบคุมตำแหน่ง RC Servo Motor เป็นดังนี้

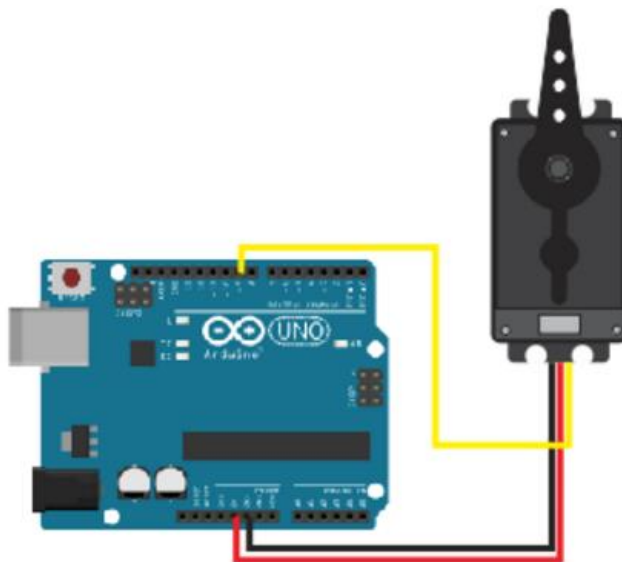


รูปที่ 9.10 การควบคุมตำแหน่ง RC Servo Motor  
(ที่มา [www.Sparkfun.com/tutorials/283](http://www.Sparkfun.com/tutorials/283))

```
myservo.write(0);
```

```
delay(1000);
```

Servo Motor จะหมุนไปที่ตำแหน่ง 0 องศา และ หยุดเป็นเวลา 1 วินาที

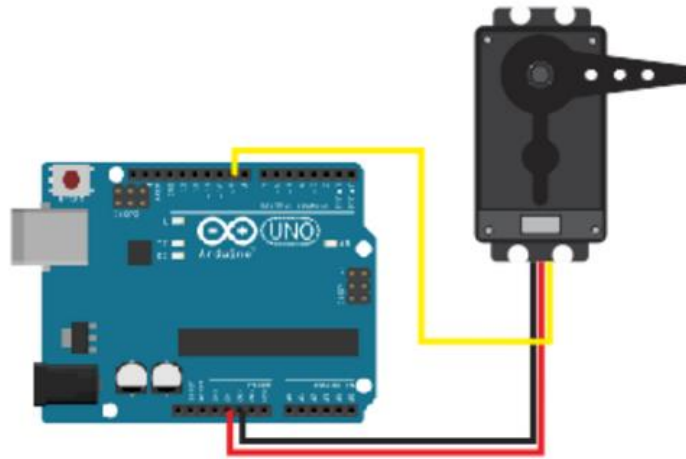


รูปที่ 9.11 Servo Motor จะหมุนไปที่ตำแหน่ง 0 องศา  
(ที่มา [www.Sparkfun.com/tutorials/283](http://www.Sparkfun.com/tutorials/283))

```
myservo.write(90);
```

```
delay(1000);
```

Servo Motor จะหมุนไปที่ตำแหน่ง 90 องศา และ หยุดเป็นเวลา 1 วินาที



รูปที่ 9.12 Servo Motor จะหมุนไปที่ตำแหน่ง 90 องศา

<https://www.sparkfun.com/tutorials/283>

```
myservo.write(90);
```

```
delay(1000);
```

Servo Motor จะหมุนไปที่ตำแหน่ง 180 องศา และ หยุดเป็นเวลา 1 วินาที  
จากนั้นจะหมุนกลับไปที่ตำแหน่ง 0 องศา และวนรอบไปเช่นนี้เรื่อยๆ

### โปรแกรมที่ 9.2 การควบคุมตำแหน่ง RC Servo Motor แบบ Sweep

```
#include <Servo.h>
```

```
Servo myservo; //create servo object to control a servo
```

```
// a maximum of eight servo objects can be created
```

```
int pos = 0; // variable to store the servo position
```

```
void setup(){
```

```
myservo.attach(9); // attaches the servo on pin 9 to the servo object
```

```
}
```

```
void loop(){
```

```
for(pos = 0; pos < 180; pos += 1) // goes from 0 degrees to 180 degrees
```

```
{
```

```
myservo.write(pos); // tell servo to go to position in variable 'pos'
delay(15); // waits 15ms for the servo to reach the position
}
for(pos = 180; pos>=1; pos-=1)// goes from 180 degrees to 0 degrees
{
myservo.write(pos); // tell servo to go to position in variable 'pos'
delay(15); // waits 15ms for the servo to reach the position
}
}
```

ผลการทำงานของโค้ดการควบคุมตำแหน่ง RC Servo Motor แบบ Sweep

```
for(pos = 0; pos < 180; pos += 1){
myservo.write(pos);
delay(15);
}
```

ลูป for กำหนดให้ค่า position มีค่าเท่ากับ 0 และทุกๆ การทำงานคำสั่งภายใน for loop ค่า position จะเพิ่มค่าขึ้น 1 ค่าจนถึง 180 ก็จะหลุดออกจาก loop

ภายใน loop for คำสั่ง myservo.write(pos); ก็คือการกำหนดให้ Servo Motor หมุนไปยังตำแหน่งมุมตามค่าในตัวแปร pos และหน่วงเวลา 15ms ด้วยคำสั่ง delay(15); ดังนั้น Servo Motor จะค่อยๆ หมุนอย่างช้าๆ จากตำแหน่ง 0 องศาไปที่ 180 องศา

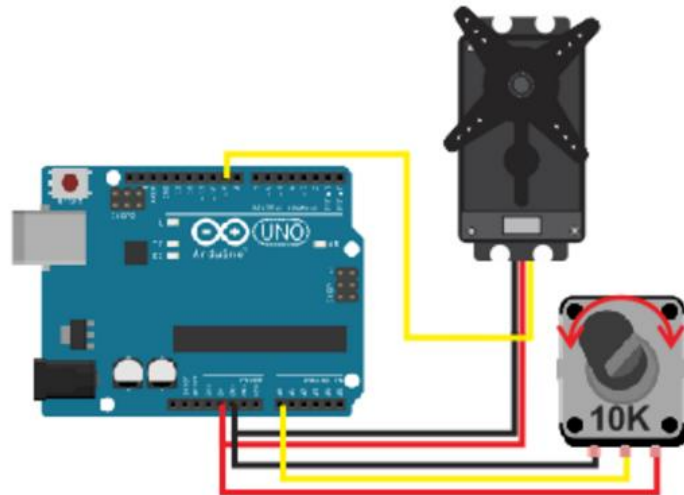
```
for(pos = 180; pos>=1; pos-=1) {
myservo.write(pos);
delay(15); }
```

ใน loop for ที่สองนี้จะทำงานเช่นเดียวกับใน loop for แรกเพียงแต่เปลี่ยนค่าเริ่มต้นจาก 180 เป็น 0 และลดค่าลง 1 ค่าทุกๆ การทำงาน 1 รอบ ส่งผลให้ Servo Motor จะหมุนจากตำแหน่งมุม 180 องศา ไปยังมุม 0 องศาอย่างช้าๆ

เมื่อเพิ่มค่าในคำสั่ง delay() ให้มากขึ้นจะพบว่า Servo Motor จะหมุนช้าลงและในทางกลับกันหากลดค่าใน delay() ลงจะพบว่า Servo Motor จะหมุนเร็วขึ้น



## 9.7 การควบคุมตำแหน่ง RC Servo Motor โดยใช้ Potentiometer



รูปที่ 9.13 การควบคุมตำแหน่ง RC Servo Motor โดยใช้ Potentiometer  
(ที่มา [www.Sparkfun.com/tutorials/283](http://www.Sparkfun.com/tutorials/283))

### โปรแกรมที่ 9.3 การควบคุมตำแหน่ง RC Servo Motor โดยใช้ Potentiometer

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup(){
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop(){
  val = analogRead(potpin); // reads the value of the potentiometer value
  // between 0 and 1023)
  val = map(val, 0, 1023, 0, 179); // scale it to use it with the servo (value
  // between 0 and 180)
  myservo.write(val); // sets the servo position according to the
  // scaled value
  delay(15); // waits for the servo to get there}
```

ผลการทำงานของโค้ดการควบคุมตำแหน่ง RC Servo Motor โดยใช้ Potentiometer

```
val = analogRead(potpin);
```

อ่านค่า Analog จาก Potentiometer ที่ต่ออยู่ที่ขา A0 เก็บไว้ในตัวแปร val

```
val = map(val, 0, 1023, 0, 179);
```

เนื่องจาก ADC ภายใน Arduino เป็น ADC ขนาด 10-bit จึงอ่านค่า Analog ได้ตั้งแต่ 0 – 1023 แต่ RC Servo Motor สามารถหมุนได้เพียงแค่ 1-180 องศา จึงต้องใช้ Function map เพื่อทำการสเกลค่าลงจาก 0-1023 เป็น 0-179 แล้วนำไปเก็บไว้ในตัวแปร val

```
myservo.write(val);
```

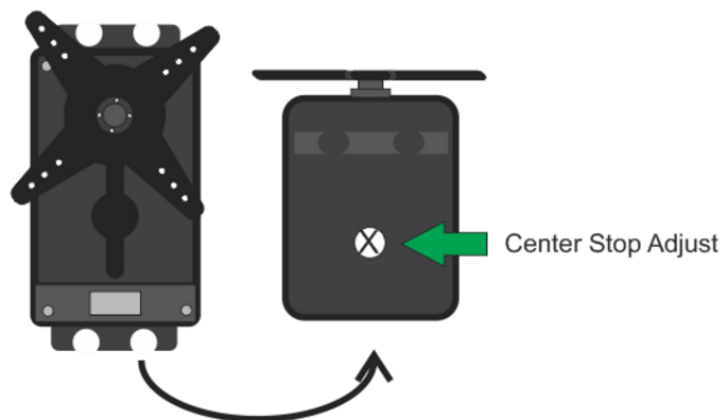
เมื่อสเกลค่า จาก 0-1023 ลงเหลือ 0-179 แล้วก็นำมาสั่งให้ Servo Motor หมุนไปยังตำแหน่งในค่าตัวแปร val

```
delay(15);
```

หน่วงเวลา 15 ms

ผลของการทำงานทำให้สามารถปรับตำแหน่งองศาของ Servo Motor ได้โดยการหมุนปรับค่า Potentiometer

### 9.7.1 Continuous Rotation Servo

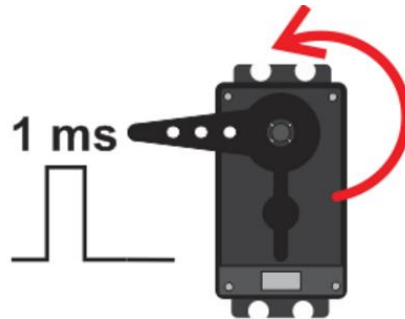


รูปที่ 9.14 RC Servo Motor แบบที่สามารถหมุนได้ 360 องศา  
(ที่มาจาก [www.Pololu.com/blog/13/gettin-all-up-in-your-servos](http://www.Pololu.com/blog/13/gettin-all-up-in-your-servos))

Continuous Rotation Servo คือ RC Servo Motor แบบที่สามารถหมุนได้ 360 องศา ส่วนประกอบภายนอกนั้นจะมีหน้าตาคล้ายกับ RC Servo Motor แบบที่หมุนได้ 180 องศา เพียงแต่จะมี Potentiometer เพื่อใช้สำหรับปรับ ตำแหน่ง Center Stop Adjust ของตัว Servo

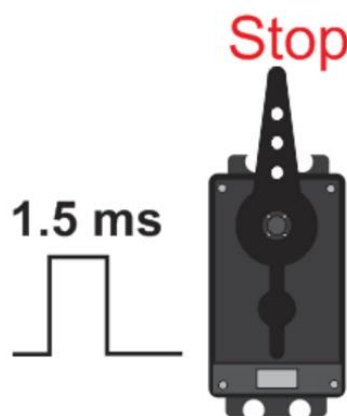
ลักษณะการใช้งาน RC Servo Motor ชนิดนี้จะแตกต่างจากการใช้งาน RC Servo Motor แบบ 180 องศาตรงที่ Servo ชนิดนี้จะใช้ความกว้างของสัญญาณพัลส์ในการกำหนดความเร็วและทิศทางในการหมุน ไม่ได้ใช้เพื่อกำหนดมุมจึงไม่สามารถกำหนดให้ Motor หมุนไปยังตำแหน่งมุมต่างๆ ตามความต้องการได้ สัญญาณความกว้างของพัลส์ที่ใช้ควบคุมจะอยู่ในช่วง 1000-2000 us แต่จะมีความแตกต่างในความหมายของแต่ละความกว้างของพัลส์ดังนี้

ความกว้าง 1000 us หมายถึงการหมุนไปทางซ้ายด้วยความเร็วสูงสุดที่ Servo Motor จะหมุนได้



รูปที่ 9.15 การหมุนไปทางซ้ายด้วยความเร็วสูงสุดของ Servo Motor  
(ที่มาจาก [www.Pololu.com/blog/13/gettin-all-up-in-your-servos](http://www.Pololu.com/blog/13/gettin-all-up-in-your-servos))

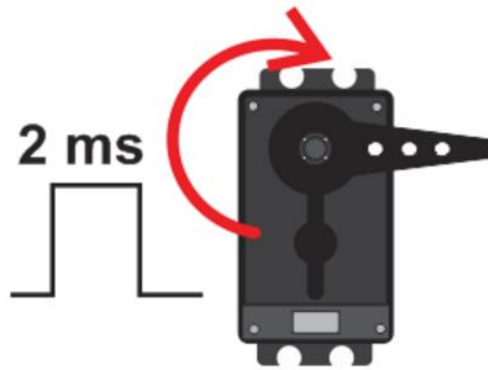
ความกว้าง 1500 us หมายถึงการสั่งให้ Servo Motor หยุดหมุน



รูปที่ 9.16 การสั่งให้ Servo Motor หยุดหมุน

(ที่มาจาก [www.Pololu.com/blog/13/gettin-all-up-in-your-servos](http://www.Pololu.com/blog/13/gettin-all-up-in-your-servos))

ความกว้าง 2000 us หมายถึงการหมุนไปทางขวาด้วยความเร็วสูงสุดที่ Servo Motor จะหมุนได้



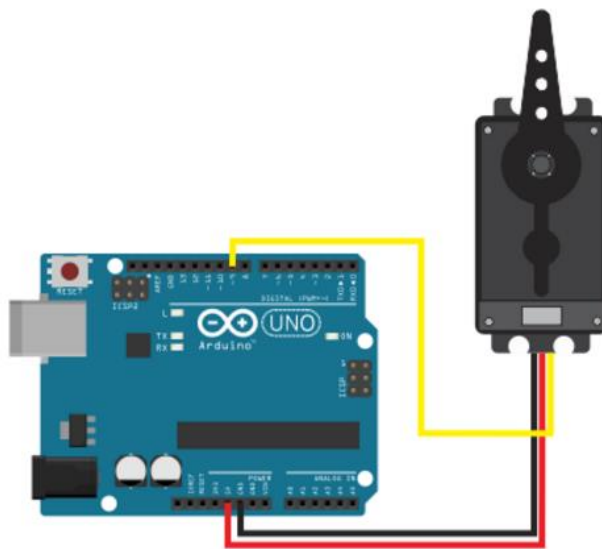
รูปที่ 9.17 การหมุนไปทางขวาด้วยความเร็วสูงสุด

(ที่มาจาก [www.Pololu.com/blog/13/gettin-all-up-in-your-servos](http://www.Pololu.com/blog/13/gettin-all-up-in-your-servos))

### 9.7.2 การ Calibrate Center Stop

ในการใช้งาน Continuous Rotation Servo เมื่อซื้อใหม่หรือใช้งานไปสักระยะหนึ่งจุด Center Stop อาจมีการคลาดเคลื่อนได้ ซึ่งแม้เราสั่งให้สัญญาณพัลส์มีความกว้างเท่ากับ 1500 us ไป Continuous rotation servo ก็จะไม่หยุดหมุน เราจึงต้องปรับตั้งค่า Center Stop ดังนี้

- ต่อ Continuous rotation servo เข้ากับ Arduino



รูปที่ 9.18 การ Calibrate Center Stop

(ที่มาจาก [www.Pololu.com/blog/13/gettin-all-up-in-your-servos](http://www.Pololu.com/blog/13/gettin-all-up-in-your-servos))

- เขียนโปรแกรมจ่ายความกว้างพัลส์ 1500 us ให้กับ Servo Motor
 

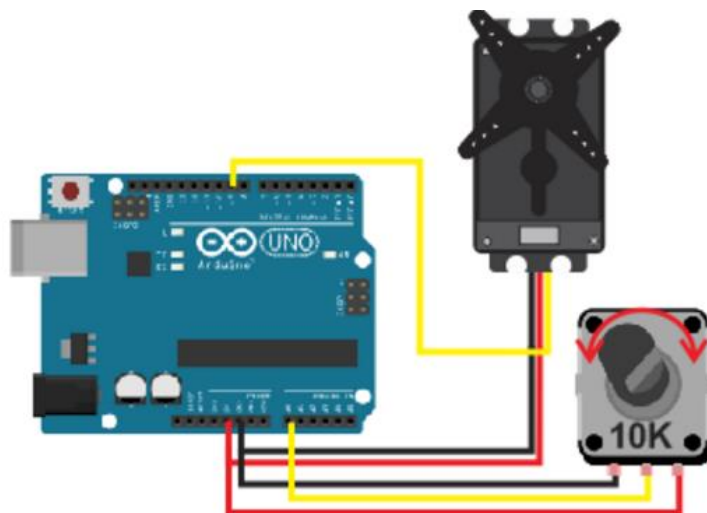
```
#include <Servo.h>
Servo myServo;
void setup() {
  myServo.attach(9);
  myServo.writeMicroseconds(1500); // Stop }
```
- เมื่อรันโปรแกรมจ่ายความกว้างพัลส์ 1500 us แล้ว Servo Motor ไม่หยุดหมุน ให้ใช้ไขควงขนาดเล็กหมุนปรับ Center Stop Adjust จน Servo Motor หยุดหมุน



รูปที่ 9.19 การปรับ Center Stop Adjust

(ที่มาจาก [www.Pololu.com/blog/13/gettin-all-up-in-your-servos](http://www.Pololu.com/blog/13/gettin-all-up-in-your-servos))

การควบคุม Continuous Rotation Servo โดยใช้ Potentiometer ปรับความเร็วและทิศทางการหมุน



รูปที่ 9.20 ควบคุม Continuous Rotation Servo โดยใช้ Potentiometer

(ที่มาจาก [www.Pololu.com/blog/13/gettin-all-up-in-your-servos](http://www.Pololu.com/blog/13/gettin-all-up-in-your-servos))

### โปรแกรมที่ 9.4 การควบคุม Continuous Rotation Servo โดยใช้ Potentiometer

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup(){
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop(){
  val = analogRead(potpin); // reads the value of the potentiometer value
  // between 0 and 1023)
  val = map(val, 0, 1023, 0, 179); // scale it to use it with the servo value
  // between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); // waits for the servo to get there
}
```

โค้ดเหมือนกับตัวอย่าง RC Servo Motor โดยใช้ Potentiometer แต่ผลการทำงานจะแตกต่างกัน ผลการทำงานของ Code การควบคุม Continuous Rotation Servo โดยใช้ Potentiometer

```
val = analogRead(potpin);
```

อ่านค่า Analog จาก Potentiometer ที่ต่ออยู่ที่ขา A0 เก็บไว้ในตัวแปร val

```
val = map(val, 0, 1023, 0, 179);
```

เนื่องจาก ADC ภายใน Arduino เป็น ADC ขนาด 10-bit จึงอ่านค่า Analog ได้ตั้งแต่ 0-1023 แต่ Continuous Rotation Servo รับค่าได้ในช่วง 1-180 จึงต้องใช้ Function map เพื่อสเกลค่าจาก 0-1023 เป็น 0-179 แล้วนำไปเก็บไว้ในตัวแปร val

```
myservo.write(val);
```

เมื่อสเกลค่า จาก 0-1023 ลงเหลือ 0-179 แล้วก็นำมาสั่งให้ Servo Motor หมุนในความเร็วและทิศทางตามค่าความกว้างของพัลส์ที่จ่ายออกไป

```
delay(15);
```

หน่วงเวลา 15 ms

ผลของการทำงานในโค้ดนี้จะเห็นได้ว่า หากปรับ Potentiometer ไปทางด้านใดด้านหนึ่งจนสุด Continuous Rotation Servo จะหมุนในทิศทางหนึ่งด้วยความเร็วสูงสุดและเมื่อเราค่อยๆ ปรับ Potentiometer กลับมาให้มาอยู่ในตำแหน่งกึ่งกลาง Servo Motor จะค่อยๆ หมุนช้าลงแต่ยังหมุนในทิศทางเดิม และจะหยุดสนิทหากเราปรับ Potentiometer มาที่ตำแหน่งกึ่งกลางพอดี เมื่อปรับ Potentiometer เกินกว่า ครึ่งหนึ่งในทิศตรงกันข้าม Servo Motor จะเปลี่ยนทิศการหมุน แต่จะหมุนอย่างช้าๆ และจะค่อยๆ หมุนเร็วขึ้นเมื่อเราปรับ Potentiometer มากขึ้น

## สรุปเนื้อหาสาระสำคัญ

Arduino มีไลบรารีสำหรับสั่งงาน RC Servo Motor มาให้ใช้งานอยู่แล้ว เป็นฟังก์ชันสำเร็จรูปและใช้งานได้ง่าย ในหน้าเว็บไซต์ <http://arduino.cc/en/reference/servo> ได้ให้ข้อมูลไว้ว่า Servo Library ของ Arduino สามารถสั่งงาน RC Servo Motor ได้ทั้งแบบหมุนไป-กลับได้ 0-180 องศา และแบบต่อเนื่องที่หมุนครบรอบได้ เรียกว่าเป็น Continuous Rotation Servo โดยสามารถรองรับการเชื่อมต่อ RC Servo Motor ได้ถึง 12 ตัวกับ หากปรับ Potentiometer ไปทางด้านใดด้านหนึ่งจนสุด Continuous Rotation Servo จะหมุนในทิศทางหนึ่ง ด้วยความเร็วสูงสุดและเมื่อเราค่อยๆ ปรับ Potentiometer กลับมาให้มาอยู่ในตำแหน่งกึ่งกลาง Servo Motor จะค่อยๆ หมุนช้าลงแต่ยังหมุนในทิศทางเดิม และจะหยุดสนิทหากเราปรับ Potentiometer มาที่ตำแหน่งกึ่งกลางพอดี เมื่อปรับ Potentiometer เกินกว่าครึ่งหนึ่งในทิศตรงกันข้าม Servo Motor จะเปลี่ยนทิศการหมุน แต่จะหมุนอย่างช้าๆ และจะค่อยๆ หมุนเร็วขึ้นเมื่อเราปรับ Potentiometer มากขึ้น





## แบบฝึกหัดหน่วยที่ 9

เรื่อง การคุมเซอร์โวมอเตอร์ด้วย Arduino

ใช้เวลา 20 นาที

\*\*\*\*\*

**คำชี้แจง** แบบฝึกหัดมีทั้งหมด 2 ตอน ประกอบด้วยตอนที่ 1 และตอนที่ 2 (20 คะแนน)

1. แบบฝึกหัดตอนที่ 1 เป็นคำถามแบบถูก-ผิด มีทั้งหมด 10 ข้อ (10 คะแนน)
2. แบบฝึกหัดตอนที่ 2 เป็นคำถามแบบปรนัย มีทั้งหมด 10 ข้อ (10 คะแนน)



### แบบฝึกหัดตอนที่ 1

**คำชี้แจง** ให้ผู้เรียนกาเครื่องหมายถูก ✓ ในข้อที่คิดว่าถูก และกาเครื่องหมายผิด ✗ ในข้อที่คิดว่าผิด

- 1. Case ตัวถัง หรือ ส่วนจับยึดตัว Servo กับชิ้นงาน
- 2. Mounting Tab กรอบของตัว Servo Motor
- 3. Output Shaft เพลาส่งกำลัง
- 4. Servo Horns ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไก
- 5. Cable สายเชื่อมต่อเพื่อ จ่ายไฟฟ้าอย่างเดียว
- 6. สายเชื่อมต่อจะประกอบด้วยสายไฟ 3 เส้น มีสีของสายแตกต่างกันไป
- 7. สายสีแดง คือ ไฟเลี้ยง กราวด์
- 8. สายสีดำ หรือ น้ำตาล คือ (4.8-6V)
- 9. สายสีเหลือง (ส้อม ขาว หรือฟ้า) คือ สายส่งสัญญาณพัลส์ควบคุม (3-5V)
- 10. Connector จุดเชื่อมต่อสายไฟ

 **แบบฝึกหัดตอนที่ 2**

**คำชี้แจง** ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ให้ครบทุกข้อ

1. ในเชิงความหมายของ Servo Motor คือ
  - ก. ระบบควบคุมอัตโนมัติ
  - ข. มอเตอร์ที่สั่งให้หมุนไปยังตำแหน่งองศาได้เองอย่างถูกต้อง
  - ค. มอเตอร์
  - ง. การควบคุมแบบป้อนกลับ
2. Feedback Control คือ
  - ก. ระบบควบคุมที่มีการนำค่าอินพุตมาเปรียบเทียบกับค่าเอาต์พุต
  - ข. ระบบควบคุมที่มีการนำค่าเอาต์พุตมาเปรียบเทียบกับค่าอินพุต
  - ค. ระบบควบคุมที่มีการนำค่าแรงดันมาเปรียบเทียบกับค่ากระแส
  - ง. ระบบควบคุมที่มีการนำค่ากระแสมาเปรียบเทียบกับค่าแรงดัน
3. Position Sensor คือ
  - ก. เป็นส่วนของตัวมอเตอร์
  - ข. เป็นส่วนที่ควบคุมและประมวลผล
  - ค. เซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
  - ง. จุดเชื่อมต่อสายไฟ
4. การอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อ
  - ก. แปลค่าเป็นตำแหน่งองศาที่ต้องการ
  - ข. จ่ายสัญญาณพัลส์
  - ค. เปรียบเทียบกับค่าอินพุต
  - ง. เปรียบเทียบกับค่าเอาต์พุต
5. สัญญาณควบคุม RC Servo Motor อยู่ในรูปแบบใด
  - ก. PAM
  - ข. PCM
  - ค. PWM
  - ง. SAM
6. มุมหรือองศาที่หมุนจะขึ้นอยู่กับ
  - ก. ความถี่ของสัญญาณพัลส์
  - ข. ความกว้างของสัญญาณพัลส์
  - ค. คาบของสัญญาณพัลส์
  - ง. ความสูงของสัญญาณพัลส์

7. กำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms ตัว Servo Motor หมุนแบบใด
- ไปทางซ้ายจนสุด
  - ไปทางซ้ายจนสุด
  - เดินหน้าจนสุด
  - ถอยหลังจนสุด
8. กำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor หมุนแบบใด
- ไปทางซ้ายจนสุด
  - ไปทางซ้ายจนสุด
  - เดินหน้าจนสุด
  - อยู่ที่ตำแหน่งตรงกลางพอดี
9. ที่ 0 องศา ใช้ความกว้างพัลส์เท่ากับ 1000 us ที่ 180 องศา ความกว้างพัลส์เท่ากับ 2000 us ที่ 1 องศา จะใช้ความกว้างพัลส์เท่าใด
- 4.44 us
  - 5.55 us
  - 6.66 us
  - 7.77 us
10. คำมูมเท่ากับ 90 คือคำสั่งให้ Servo Motor หมุนอย่างไร
- หมุนด้วยความเร็วสูงสุดในทิศทางหนึ่ง
  - หมุนด้วยความเร็วสูงสุดในทิศทางตรงกันข้าม
  - หยุดหมุน
  - หมุนด้วยความเร็วสูงสุด

## ปฏิบัติการทดลองหน่วยที่ 9

### เรื่อง การคุมเซอร์โวมอเตอร์ด้วย Arduino

\*\*\*\*\*

**คำชี้แจง** ให้ผู้เรียนทุกคนทำการทดลองตามปฏิบัติการทดลองหน่วยที่ 9 เรื่อง การคุมเซอร์โวมอเตอร์ด้วย Arduino ใช้เวลา 180 นาที ( 20 คะแนน)

#### จุดประสงค์เชิงพฤติกรรม

1. สามารถใช้ฟังก์ชันได้ถูกต้อง
2. สามารถแก้ปัญหาในการทำงานของบอร์ด Arduino Uno R3 ได้
3. สามารถต่อใช้งานและอัปโหลดโปรแกรมให้กับบอร์ด Arduino Uno R3 ได้

#### อุปกรณ์การทดลอง

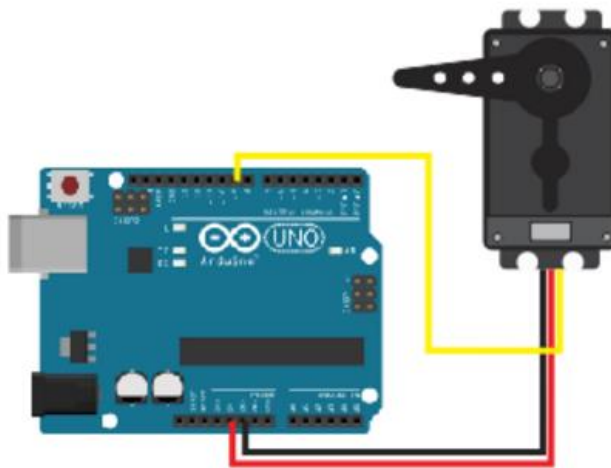
- |   |    |       |
|---|----|-------|
| 1. เครื่องคอมพิวเตอร์และโปรแกรม Arduino IDE 1.6.9 | 1  | ชุด   |
| 2. USB Cable Arduino Uno R3                       | 1  | เส้น  |
| 3. Arduino Uno R3 Board                           | 1  | บอร์ด |
| 4. Hook-up Wires                                  | 10 | เส้น  |
| 5. Breadboard                                     | 1  | แผง   |
| 6. RC Servo Motor                                 | 1  | ตัว   |
| 7. 10k Ohm Potentiometer                          | 1  | ตัว   |

#### ข้อควรระวัง

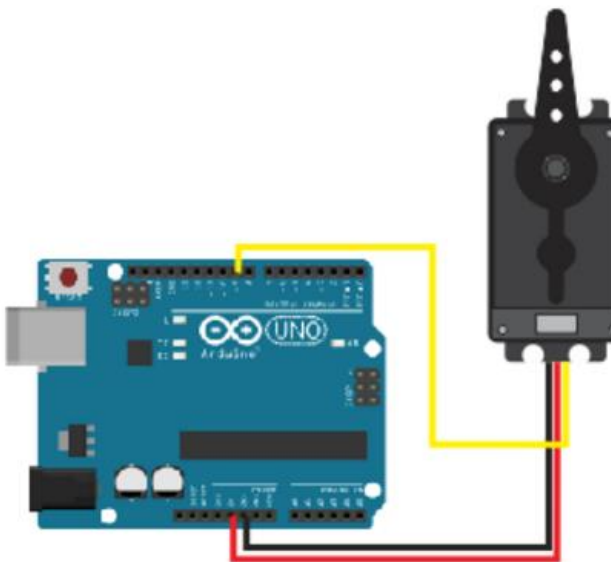
1. ควรระวังไม่วางบอร์ด Arduino Uno R3 หรือซิลต่างๆ บนโต๊ะโลหะหรือที่วางที่เป็นโลหะเพราะอาจเกิดการลัดวงจรของภาคจ่ายไฟได้
2. ไม่ควรต่อสายต่อวงจรในบอร์ด Arduino Uno R3 ทิ้งไว้ ควรถอดสายต่อวงจรออกให้หมด เพราะผลการทดลองอาจเกิดการผิดพลาดไม่เป็นไปตามทฤษฎีได้
3. ไม่ควรถอดสายสายโหลด USB เข้าออกตลอดเวลา เพราะอาจทำให้ภาคจ่ายไฟของบอร์ด Arduino Uno R3 เสียหายได้

## การทดลองที่ 9.1 การควบคุมตำแหน่ง RC Servo Motor

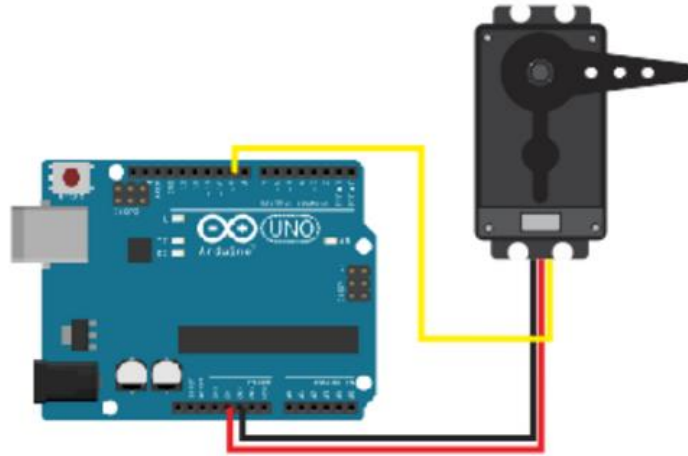
### Hardware Required/Circuit



รูปที่ 9.21 ควบคุม `myservo.write(0);delay(1000);`



รูปที่ 9.22 ควบคุม `myservo.write(90);delay(1000);`



รูปที่ 9.23 ควบคุม myservo.write(180);delay(1000);

## Code

```

#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(9);
}
void loop()
{
  myservo.write(0);
  delay(1000);
  myservo.write(90);
  delay(1000);
  myservo.write(180);
  delay(1000);      }

```

## ผลการทดลอง

.....

.....

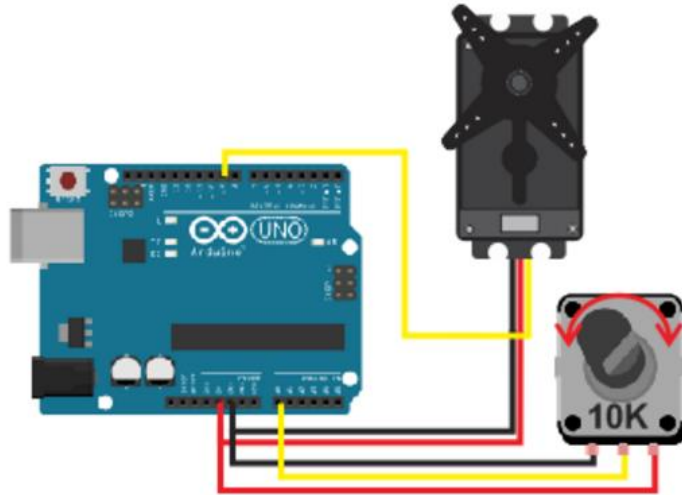
.....

.....

.....

## การทดลองที่ 9.2 การควบคุมตำแหน่ง RC Servo Motor โดยใช้ Potentiometer

### Hardware Required/Circuit



รูปที่ 9.24 การควบคุมตำแหน่ง RC Servo Motor โดยใช้ Potentiometer

### Code

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup(){myservo.attach(9); } // attaches the servo on pin 9 to the servo object
void loop(){
    val = analogRead(potpin); // reads the value of the potentiometer value
    // between 0 and 1023
    val = map(val, 0, 1023, 0, 179); // scale it to use it with the servo value
    // between 0 and 180
    myservo.write(val); // sets the servo position according to the
    // scaled value
    delay(15); // waits for the servo to get there}
```

### ผลการทดลอง

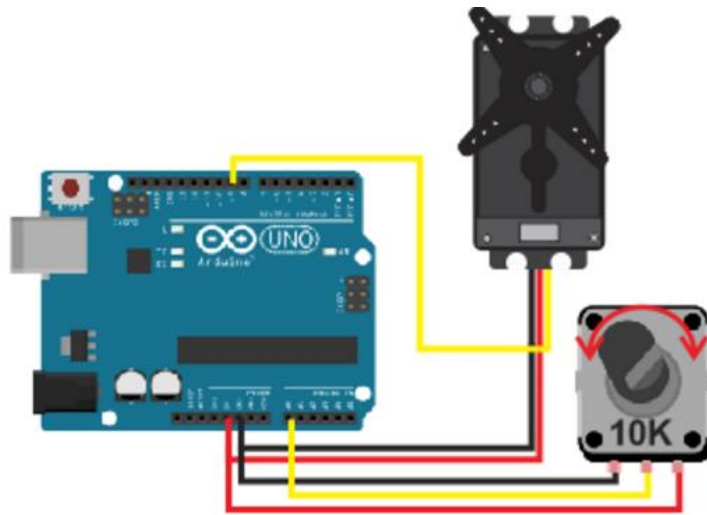
.....

.....

.....

### การทดลองที่ 9.3 การควบคุม Continuous Rotation Servo โดยใช้ Potentiometer ปรับความเร็วและทิศทางการหมุน

#### Hardware Required/Circuit



รูปที่ 9.25 ควบคุมการควบคุม Continuous Rotation Servo โดยใช้ Potentiometer

#### Code

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup(){
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop(){
  val = analogRead(potpin); // reads the value of the potentiometer value
  val = map(val, 0, 1023, 0, 179); // scale it to use it with the servo value
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); }
```

#### ผลการทดลอง

.....

.....

.....





## แบบทดสอบหลังเรียน หน่วยที่ 9

### เรื่อง การคุมเซอร์โวมอเตอร์ด้วย Arduino

เรื่อง การคุมเซอร์โวมอเตอร์ด้วย Arduino

ใช้เวลา 20 นาที

วิชา ไมโครคอนโทรลเลอร์เบื้องต้น

รหัสวิชา (2127-2107)

ระดับชั้น ประกาศนียบัตรวิชาชีพ (ปวช.)

สาขาวิชา เมคคาทรอนิกส์

\*\*\*\*\*

**คำชี้แจง** 1. แบบทดสอบมีทั้งหมด 10 ข้อ (10 คะแนน)

2. ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ลงในกระดาษคำตอบ

2.Servo มาจากภาษาละตินจากคำว่า Sevus หมายถึงอะไร

- ก. มอเตอร์
- ข. อติโนมิติ
- ค. ทาส
- ง. เจ้านาย

2. RC Servo Motor ซึ่งนิยมนำมาใช้กับอะไร

- ก. เครื่องเล่นที่บังคับด้วยคลื่นโทรทัศน์
- ข. เครื่องเล่นที่บังคับด้วยคลื่นโทรศัพท์
- ค. เครื่องเล่นที่บังคับด้วยคลื่นไมโครเวฟ
- ง. เครื่องเล่นที่บังคับด้วยคลื่นวิทยุ

3. Gear Train หรือ Gearbox คือ

- ก. เป็นชุดเกียร์ทดแรง
- ข. เป็นชุดเกียร์เพิ่มแรง
- ค. เป็นส่วนของตัวมอเตอร์
- ง. เป็นส่วนที่ควบคุมและประมวลผล

4. Electronic Control System คือ

- ก. เป็นส่วนของตัวมอเตอร์
- ข. เป็นส่วนที่ควบคุมและประมวลผล
- ค. เซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
- ง. จุดเชื่อมต่อสายไฟ

5. Position Sensor เป็นตัวเซ็นเซอร์ที่วัดค่ามุมที่กำลังหมุนส่งกลับมา เพื่ออะไร
  - ก. ขับ Gear Train
  - ข. ประมวลผลค่าความกว้างของสัญญาณพัลส์
  - ค. ให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการ
  - ง. ขับ Connector
6. กำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor หมุนแบบใด
  - ก. ไปทางซ้ายจนสุด
  - ข. ไปทางขวาจนสุด
  - ค. เดินหน้าจนสุด
  - ง. อยู่ที่ตำแหน่งตรงกลาง
7. ที่ 1 องศาความกว้างพัลส์เท่ากับ 5.55 us ที่มุม 45 องศา หาค่าความกว้างพัลส์ที่ต้องการได้เท่าใด
  - ก. 107.75 us
  - ข. 222.75 us
  - ค. 249.75 us
  - ง. 274.75 us
8. ฟังก์ชันของ Arduino ภายใน Servo Library ชื่อ attach() คือ
  - ก. ฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ให้ Servo Motor หมุนไปยังตำแหน่งองศาที่กำหนด
  - ข. ฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ต้องการ
  - ค. ฟังก์ชันที่ใช้ในการกำหนดขาสัญญาณ
  - ง. ฟังก์ชันอ่านค่าองศาที่ส่งเข้าไปด้วยฟังก์ชัน write()
9. ฟังก์ชันของ Arduino ภายใน Servo Library ชื่อ read() คือ
  - ก. ฟังก์ชันชะลอการทำงานในหน่วยไมโครวินาที
  - ข. ฟังก์ชันชะลอการทำงานในหน่วยวินาที
  - ค. ฟังก์ชันชะลอการทำงานในหน่วยนาฬิกา
  - ง. ฟังก์ชันอ่านค่าองศาที่ส่งเข้าไปด้วยฟังก์ชัน write()
10. Servo Library ของ Arduino สามารถสั่งงาน RC Servo Motor ได้แบบใดบ้าง
  - ก. 0-45 องศา และ หมุนครบรอบ
  - ข. 0-90 องศา และ หมุนครบรอบ
  - ค. 0-120 องศา และ หมุนครบรอบ
  - ง. 0-180 องศา และ หมุนครบรอบ