

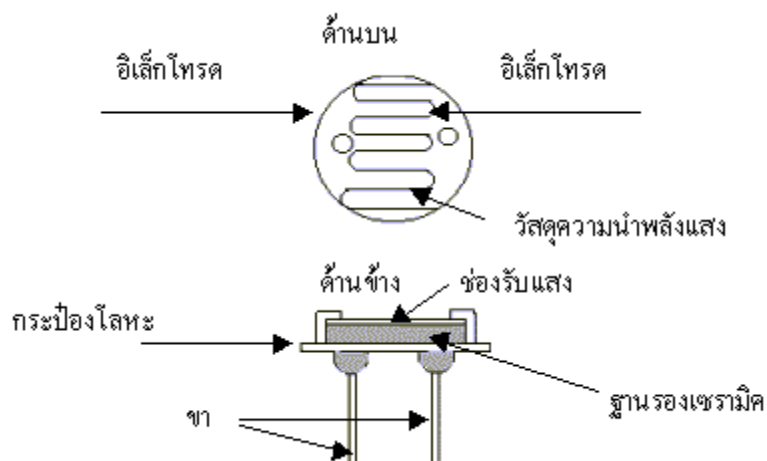
ใบความรู้ที่ 14.1

เรื่อง การใช้งาน LDR และเซอร์โวมอเตอร์ ใน Tinkercad

ในการใช้งานโปรแกรม Tinkercad เราสามารถประยุกต์ใช้กับเซนเซอร์และโมดูลค่อนข้างหลากหลาย เพื่อให้ประยุกต์ใช้งานกับระบบอัตโนมัติได้จริง จึงจำเป็นต้องมีการจำลองการทำงานของอุปกรณ์ผ่าน Simulator ของ Tinkercad ก่อน ในเนื้อหาวันนี้ นักเรียนจะประยุกต์การใช้อุปกรณ์ 2 อย่าง คือ ตัวต้านทานปรับค่าตามแสง และเซอร์โวมอเตอร์ ดังนี้

1. ตัวต้านทานปรับค่าตามแสง

LDR (Light Dependent Resistor) คือ ตัวต้านทานชนิดนี้สามารถเปลี่ยนความนำไฟฟ้าได้เมื่อมีแสงมาตกกระทบ โฟโตริซิสเตอร์ (Photo Resistor) หรือ โฟโตคอนดักเตอร์ (Photo Conductor) เป็นตัวต้านทานที่ทำมาจากสารกึ่งตัวนำ (Semiconductor) ประเภทแคดเมียมซัลไฟด์ (Cds : Cadmium Sulfide) หรือแคดเมียมซีลีไนด์ (CdSe : Cadmium Selenide) ซึ่งทั้งสองตัวนี้ก็เป็นสารประเภทกึ่งตัวนำ เอามาฉาบลงบนแผ่นเซรามิกที่ใช้เป็นฐานรองแล้วต่อขาจากสารที่ฉาบ ไข่ออกมา โครงสร้างของ LDR



รูปที่ 1 ภาพแสดงโครงสร้างของหลอด LDR

สำหรับการใช้ Tinkercad รับค่าจากหลอด LDR สามารถสร้างเขียน Code ได้ตามตัวอย่างที่ 1

ตัวอย่างที่ 1

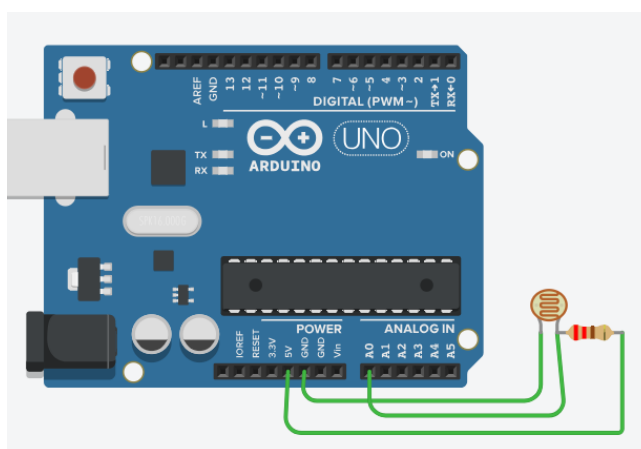
```
void setup()
{
  Serial.begin(9600); // เปิดการอ่านผ่าน Serial monitor
}
void loop()
{
  Serial.println(analogRead(A0)); // อ่านค่า LDR จากพอร์ต A0 ผ่าน Serial monitor
}
```

สำหรับการใช้งานคำสั่งรับค่าตัวต้านทานปรับค่าตามแสง LDR ในโปรแกรม Tinkercad สามารถทำได้ตามขั้นตอนดังต่อไปนี้

1. ข้อมูลอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการต่อวงจร

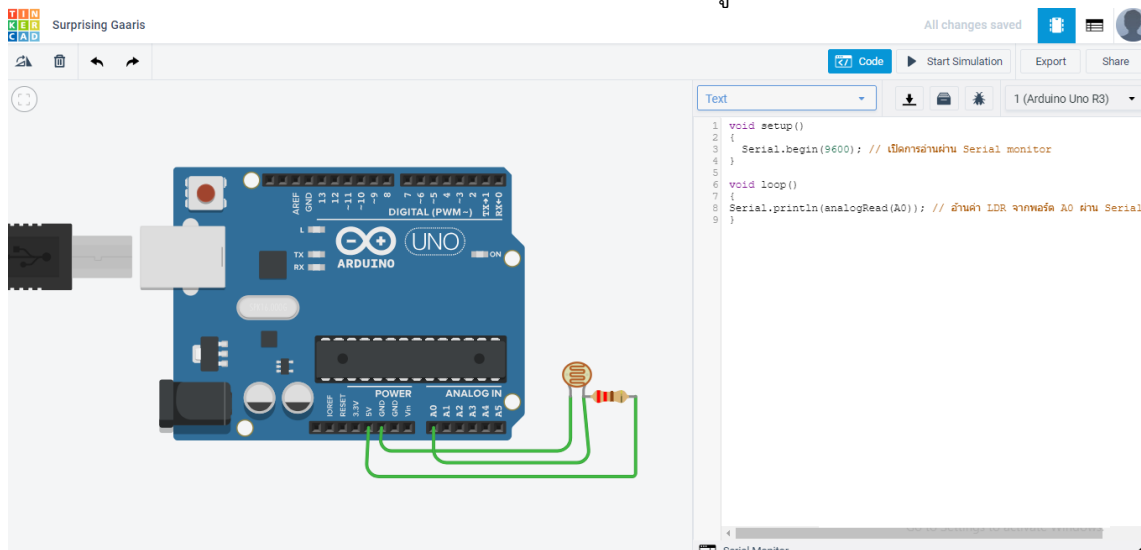
รูปอุปกรณ์	ชื่ออุปกรณ์	ขาสัญญาณที่ต่อ Arduino Uno/ชนิดสัญญาณ
	Photoresistor	A0
	Resistor	ตามภาพ

2. ต่ออุปกรณ์ดังรูปที่ 1



รูปที่ 1 ภาพแสดงการต่อวงจรประกอบตัวอย่างที่ 1 ใน tinkercad.com

3. นำตัวอย่าง code ตัวอย่างที่ 1 ไปวางใน Code ---> Text ดังรูปที่ 2



```

1 void setup()
2 {
3   Serial.begin(9600); // เปิดการอ่านผ่าน Serial monitor
4 }
5
6 void loop()
7 {
8   Serial.println(analogRead(A0)); // อ่านค่า LDR จากพิน A0 ผ่าน Serial
9 }

```

รูปที่ 2 ภาพแสดงการนำ Code ในตัวอย่างที่ 1 มาใช้ใน tinkercad.com

ผลการทำงานของโปรแกรม : เมื่อโปรแกรมทำงานเมื่อเพิ่มค่าแสดง ค่าจาก Serial Monitor ก็จะเพิ่มตาม และเมื่อลดค่าแสงค่าสัญญาณก็จะลดลงเช่นกัน

2. เซอร์โวมอเตอร์

เซอร์โวมอเตอร์ (Servo Motor) เป็นมอเตอร์ที่มีการควบคุมการเคลื่อนที่ของมัน (State) ไม่ว่าจะเป็น ระยะ ความเร็ว มุมการหมุน โดยใช้การควบคุมแบบป้อนกลับ (Feedback control) เป็นอุปกรณ์ที่สามารถควบคุมเครื่องจักรกล หรือระบบการทำงานอื่นๆ ให้เป็นไปตามความต้องการ เช่น ควบคุมความเร็ว (Speed), ควบคุมแรงบิด (Torque), ควบคุมแรงตำแหน่ง (Position), ระยะทางในการเคลื่อนที่(หมุน) (Position Control) ของตัวมอเตอร์ได้ ซึ่งมอเตอร์ทั่วไปไม่สามารถควบคุมในลักษณะงานเบื้องต้นได้ โดยให้ผลลัพธ์ตามความต้องการที่มีความแม่นยำสูง



รูปที่ 3 ภาพแสดงเซอร์โวมอเตอร์ที่มีอยู่ทั่วไปในท้องตลาด

โดยปกติทั่วไปเซอร์โวมอเตอร์จะมีสายสัญญาณกันอยู่แค่ 3 เส้น ดังตารางข้างล่าง

สี	ความหมาย
น้ำตาล หรือ ดำ	เป็นสาย Ground
แดง	เป็นไฟเลี้ยงอยู่ระหว่าง 4.8 – 6.6 V
เหลือง หรือขาว	เป็นสายสัญญาณ หรือ Signal

สำหรับการใช้งาน Tinkercad ในการควบคุมเซอร์โวมอเตอร์ สามารถเขียน Code ควบคุมองศา มีรูปแบบการเขียนหลักๆดังต่อไปนี้

```
sv1.write(องศาของมอเตอร์);
```

ตัวอย่างที่ 2 การสั่งงานเซอร์โวมอเตอร์

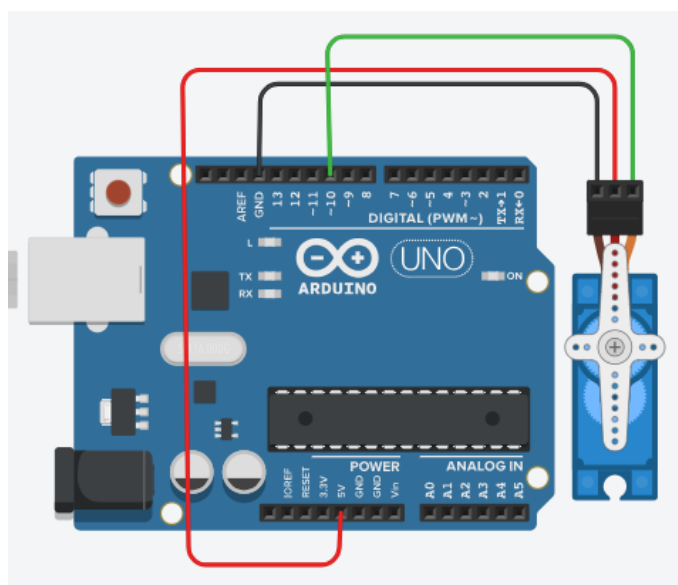
```
#include <Servo.h> // ผนวกไลบรารี
int s1 = 10; // เปิดใช้งาน s1 ที่พอร์ต 10
Servo sv1; // ประกาศการใช้งาน sv1
void setup() {
  sv1.attach(s1); // เรียกใช้งานเตรียมตอบสนอง sv1
}
void loop(){
  sv1.write(0); // sv1 ทำงานที่ 0 องศา
  delay(1000); // หน่วงเวลา 1 วินาที
  sv1.write(90); // sv1 ทำงานที่ 90 องศา
  delay(1000); // หน่วงเวลา 1 วินาที
  sv1.write(180); // sv1 ทำงานที่ 180 องศา
  delay(1000); // หน่วงเวลา 1 วินาที
}
```

สำหรับการใช้งานคำสั่งปรับองศาของเซอร์โวมอเตอร์ ในโปรแกรม Tinkercad สามารถทำได้ตามขั้นตอนดังต่อไปนี้

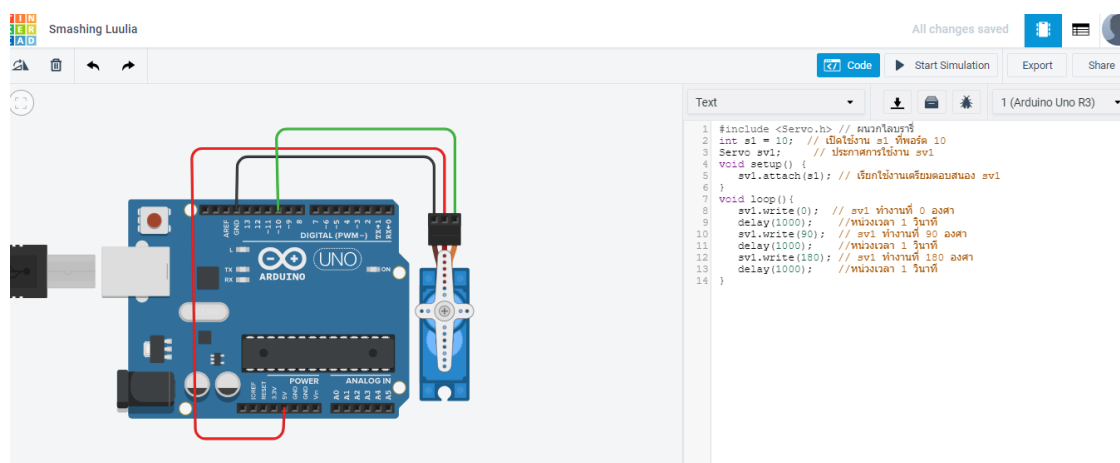
1. ข้อมูลอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการต่อวงจร

รูปอุปกรณ์	ชื่ออุปกรณ์	ขาสัญญาณที่ต่อ Arduino Uno/ชนิดสัญญาณ
	Micro Servo	สีส้มต่อที่พอร์ต 10 สีแดงต่อที่ 5V สีน้ำตลต่อที่ GND

2. ต่ออุปกรณ์ดังรูปที่ 4



รูปที่ 4 ภาพแสดงการต่อเซอร์โวมอเตอร์เข้ากับบอร์ด Arduino Uno



```

1 #include <Servo.h> // ตามกไลบรารี
2 int s1 = 10; // เปิดใช้งาน s1 ที่พอร์ต 10
3 Servo sw1; // ประกาศการใช้ชาน sw1
4 void setup() {
5   sw1.attach(s1); // เรียกใช้งานเตรียมคอมสแกน sw1
6 }
7 void loop(){
8   sw1.write(0); // sw1 ทำงานที่ 0 องศา
9   delay(1000); // ท่วงเวลา 1 วินาที
10  sw1.write(90); // sw1 ทำงานที่ 90 องศา
11  delay(1000); // ท่วงเวลา 1 วินาที
12  sw1.write(180); // sw1 ทำงานที่ 180 องศา
13  delay(1000); // ท่วงเวลา 1 วินาที
14 }

```

รูปที่ 5 ภาพแสดงการนำ Code ในตัวอย่างที่ 2 มาใช้ใน tinkercad.com

ผลการทำงานของโปรแกรม : เมื่อโปรแกรมทำงานเซอร์โวมอเตอร์จะทำงานที่ 0 องศา เป็นเวลา 1 วินาที ถัดไปจะทำงานที่ 90 องศา เป็นเวลา 1 วินาที และถัดไปก็จะทำงานที่ 180 องศา เป็นเวลา 1 วินาที จะทำงานอยู่ต่อเนื่องเพราะเขียนฟังก์ชันอยู่ในคำสั่ง loop (การวนซ้ำ) นั่นเอง

3. การประยุกต์ใช้งานตัวต้านทานปรับค่าตามแสง กับการใช้งานเปิดปิดไฟอัตโนมัติ จะมี Code ตัวอย่างในการศึกษาดังต่อไปนี้

ตัวอย่างที่ 3 การเขียน Code ในการควบคุมอุปกรณ์เปิดปิดไฟอัตโนมัติตามค่าแสง

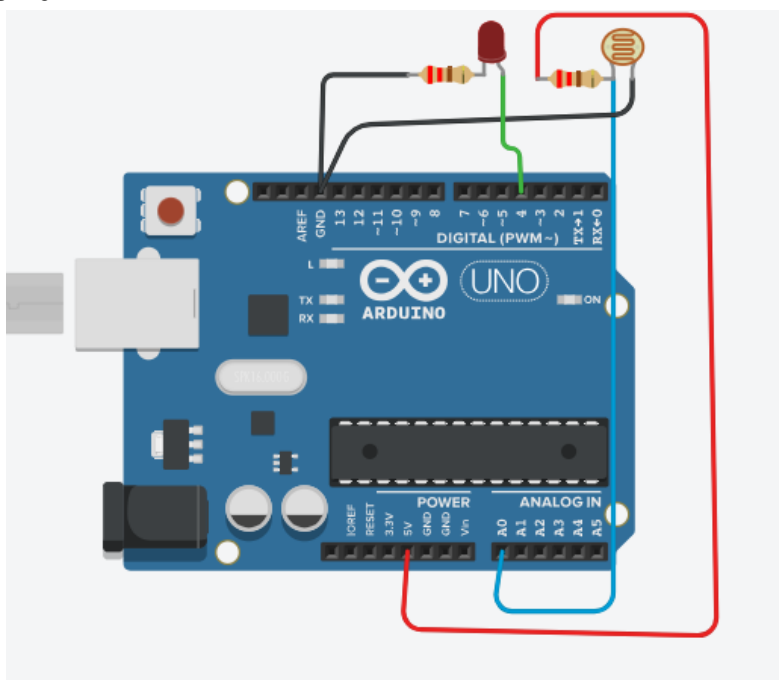
```
void setup()
{
  pinMode(4, OUTPUT); // กำหนดค่าพอร์ต 4 เป็น OUTPUT
  Serial.begin(9600); // เปิดการอ่านผ่าน Serial monitor
}
void loop()
{
  int x = analogRead(A0); // กำหนดให้ x เป็นค่าจำนวนเต็มซึ่งรับค่ามาจาก A0
  Serial.println(x); // อ่านค่า LDR จากพอร์ต A0 ผ่าน Serial monitor
  if (x < 500) { // ถ้าความเข้มแสงน้อยกว่า 500
    digitalWrite(4, HIGH); // หลอด LED ที่พอร์ต 4 จะติด
  }
  else { // นอกเหนือจากนั้น
    digitalWrite(4, LOW); // หลอด LED ที่พอร์ต 4 จะดับ
  }
}
```

โปรแกรม Tinkercad สามารถหาได้ตามขั้นตอนดังต่อไปนี้

1. ข้อมูลอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการต่อวงจร

รูปอุปกรณ์	ชื่ออุปกรณ์	ขาสัญญาณที่ต่อ Arduino Uno/ชนิดสัญญาณ
	Photoresistor	A0
	Resistor	ตามภาพ
	Resistor	ตามภาพ
	LED	ตามภาพ

2. ต่ออุปกรณ์ดังรูปที่ 6



รูปที่ 6 ภาพแสดงการต่อวงจรประกอบตัวอย่างที่ 3 ใน tinkercad.com

3. นำตัวอย่าง code ตัวอย่างที่ 3 ไปวางใน Code ----> Text ดังรูปที่ 7

```

1 void setup()
2 {
3   pinMode(4, OUTPUT); // กำหนดค่าพอร์ต 4 เป็น OUTPUT
4   Serial.begin(9600); // เปิดการอ่านผ่าน Serial monitor
5 }
6 void loop()
7 {
8   int x = analogRead(A0); // กำหนดให้ x เป็นค่าจำนวนเต็มซึ่งรับค่ามาจาก A0
9   Serial.println(x); // ส่งค่า LDR จากพอร์ต A0 ผ่าน Serial monitor
10  if (x < 500) { // ถ้าความเข้มแสงน้อยกว่า 500
11    digitalWrite(4, HIGH); // ทลอด LED ที่พอร์ต 4 จะติด
12  }
13  else {
14    digitalWrite(4, LOW); // นอกเหนือจากนั้น
15    // ทลอด LED ที่พอร์ต 4 จะดับ
16  }
17 }

```

รูปที่ 7 ภาพแสดงการนำ Code ในตัวอย่างที่ 3 มาใช้ใน tinkercad.com

ผลการทำงานของโปรแกรม : เมื่อโปรแกรมทำงานเมื่อเพิ่มค่าแสง Serial Monitor ก็จะเพิ่มตาม และเมื่อลดค่าแสงค่าสัญญาณก็จะลดลงเช่นกัน และจะทำงานเปิดหลอด LED แบบอัตโนมัติเมื่อที่ต่อเมื่อค่าแสงมีค่าน้อยกว่า 500 หากมากกว่านี้หลอด LED จะดับ