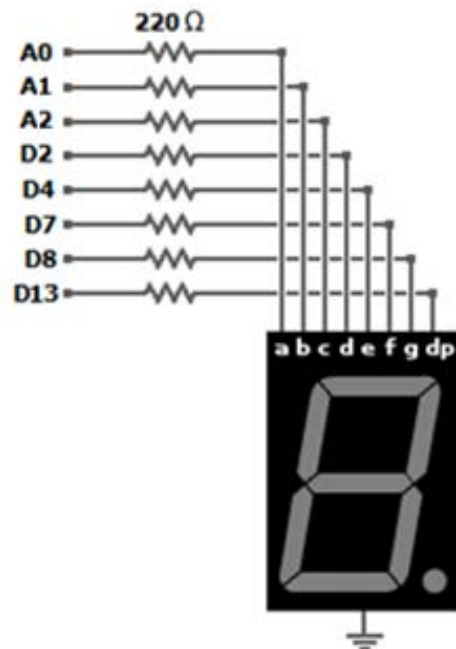
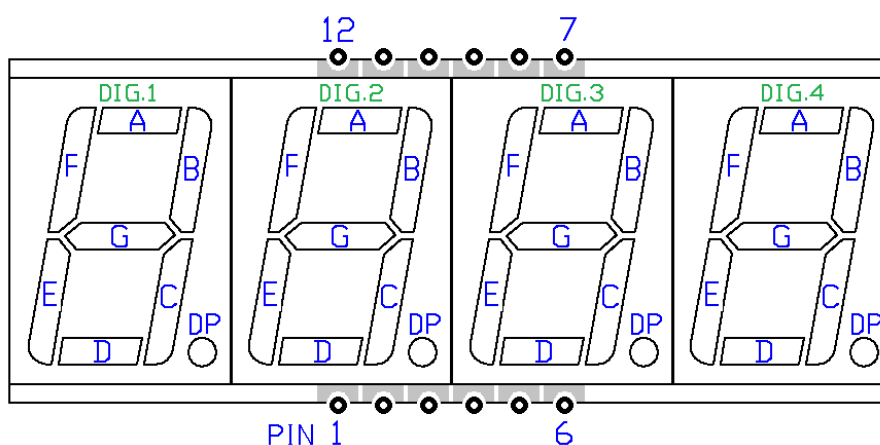


7-Segment การต่อวงจรเพื่อสั่งงานให้ไมโครคอนโทรลเลอร์แสดงผลของข้อมูลต่างๆ เป็นตัวเลขออก LED ที่เป็น 7-Segment แบบ 4 หลัก โดยใช้ LED ที่เป็นไดโอดเปล่งแสง จะต้องต่อขา Cathode คือขา Digit1 เข้ากับขั้วไฟลบหรือลงกราวด์ และจ่ายไฟเข้าขั้ว Anode ในแต่ละขาเป็นไฟบวกคือสัญญาณ HIGH เพื่อให้ LED สว่าง และสัญญาณ LOW เพื่อให้ LED ดับ โดยไฟที่จะสั่งให้ค่าในแต่ละบิต จำนวน 8 บิตไปแสดงผลเป็นค่าตัวเลขต่าง ๆ นั้นได้จากการต่อกับขั้วของบอร์ด Arduino ตามวงจรด้านล่าง ซึ่งจะมีรีซิสเตอร์ขนาด 220 โอห์ม มาใช้ในการควบคุมกระแสที่ไหลผ่านในแต่ละ Segment



ตัวอย่างของ 7-Segment เบอร์ 5641 จะมีตำแหน่งในแต่ละ Segment และขาต่างๆ ที่จะนำมาใช้ในการเชื่อมต่อแสดงได้ดังรูป รายละเอียดต่างๆเพิ่มเติมให้เปิดดูได้จากเอกสาร 7SEGMENT5641_DataSheet



1. ให้เชื่อมต่อสายวงจร 7-Segment กับบอร์ด Arduino ทำการเขียนโปรแกรมที่ทำหน้าที่สั่งงานให้ LED บน 7-Segment แสดงผลเป็นตัวเลขตั้งแต่ 1 ถึง 3 จากนั้นให้คอมไพล์แล้ว Upload โปรแกรมที่ได้ลงบนบอร์ด

```
// 4 digit 7 segment display
int segmentA = A0;
int segmentB = A1;
int segmentC = A2;
int segmentD = 2;
int segmentE = 4;
int segmentF = 7;
int segmentG = 8;
int segmentDP = 13;

void setup()
{
  pinMode(segmentA, OUTPUT);
  pinMode(segmentB, OUTPUT);
  pinMode(segmentC, OUTPUT);
  pinMode(segmentD, OUTPUT);
  pinMode(segmentE, OUTPUT);
  pinMode(segmentF, OUTPUT);
  pinMode(segmentG, OUTPUT);
  pinMode(segmentDP, OUTPUT);
}

void loop()
{
  displayNumber();
}

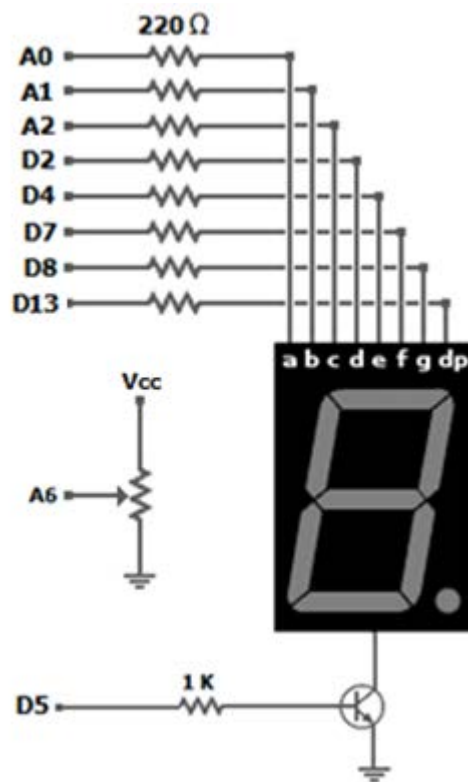
void displayNumber()
{
  for(int digit = 1 ; digit <= 3 ; digit++)
  {
    displaySegment(digit);
    delay(500);
  }
}

void displaySegment(int numberToDisplay)
{
  switch (numberToDisplay)
  {
    case 1:
      digitalWrite(segmentA, LOW);
      digitalWrite(segmentB, HIGH);
      digitalWrite(segmentC, HIGH);
      digitalWrite(segmentD, LOW);
      digitalWrite(segmentE, LOW);
      digitalWrite(segmentF, LOW);
      digitalWrite(segmentG, LOW);
      break;
      ; แสดงผลเลข 1

    case 2:
      digitalWrite(segmentA, HIGH);
      digitalWrite(segmentB, HIGH);
      digitalWrite(segmentC, LOW);
      digitalWrite(segmentD, HIGH);
      digitalWrite(segmentE, HIGH);
      digitalWrite(segmentF, LOW);
      digitalWrite(segmentG, HIGH);
      break;
      ; แสดงผลเลข 2

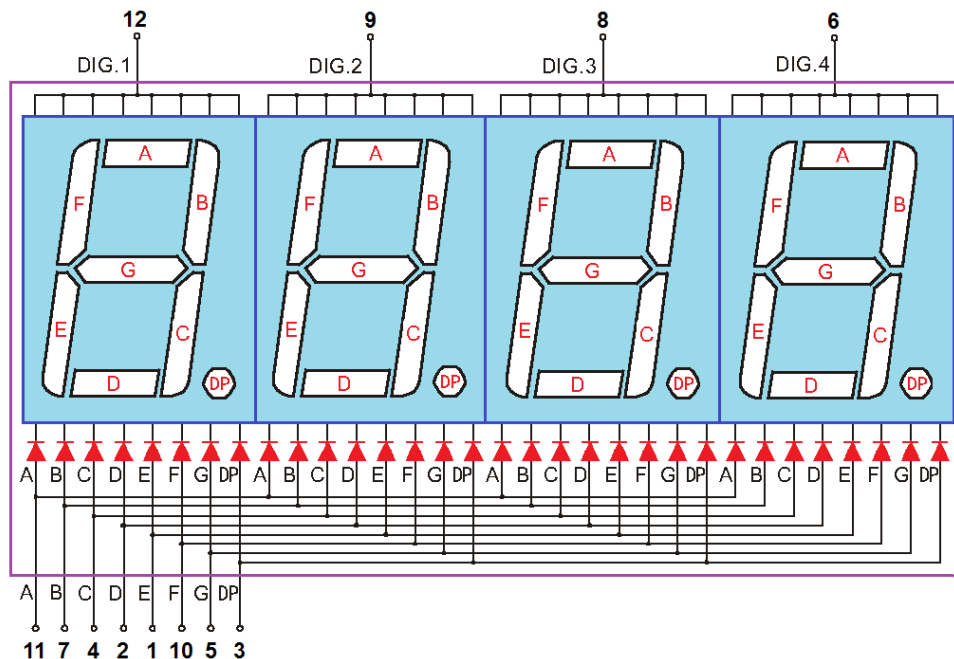
    case 3:
      digitalWrite(segmentA, HIGH);
      digitalWrite(segmentB, HIGH);
      digitalWrite(segmentC, HIGH);
      digitalWrite(segmentD, HIGH);
      digitalWrite(segmentE, LOW);
      digitalWrite(segmentF, LOW);
      digitalWrite(segmentG, HIGH);
      break;
      ; แสดงผลเลข 3
  }
}
```

2. ให้แก้ไขโปรแกรมเพื่อให้ 7 Segment แสดงผลเป็นการนับตัวเลขตั้งแต่ 0 ถึง 9 เรียงลำดับกันไป โดยใช้คำสั่ง case
3. จากข้อ 2 ให้เพิ่ม Switch ที่ต่อระหว่างขา D12 และขากราวด์ เสร็จแล้วให้เขียนโปรแกรมกำหนดขา D12 เป็น Input ที่มีการต่อ Internal Resistor แบบ pull-up โดยใช้คำสั่ง `pinMode(12, INPUT_PULLUP)` และให้ทำการตรวจสอบสถานะของขา D12 โดยใช้คำสั่ง `if(!digitalRead(12))` การทำงานกำหนดเงื่อนไขไว้ว่า เมื่อมีการกด Switch ให้ 7 Segment แสดงผลเป็นการนับตัวเลขถอยหลังตั้งแต่ 9 ถึง 0 เรียงลำดับกันไปครั้งละ 1 หลัก แต่ถ้าไม่ใช่คือไม่ได้กดสวิทช์ให้แสดงผลตามข้อ 2 เหมือนเดิม
4. ให้ต่อวงจรโดยเพิ่มตัว Transistor เข้าที่ขา Common Cathode ของ 7-Segment และมีตัวความต้านทานขนาด 1 K ต่อเข้าที่ขา B ของ Transistor ทำหน้าที่ป้อนไฟจากขา D5 เพื่อสั่งให้ Transistor เปิด-ปิดการทำงาน เสมือนทำหน้าที่เป็นสวิทช์เปิด-ปิดการแสดงผลของ 7-Segment ตามวงจรในรูป

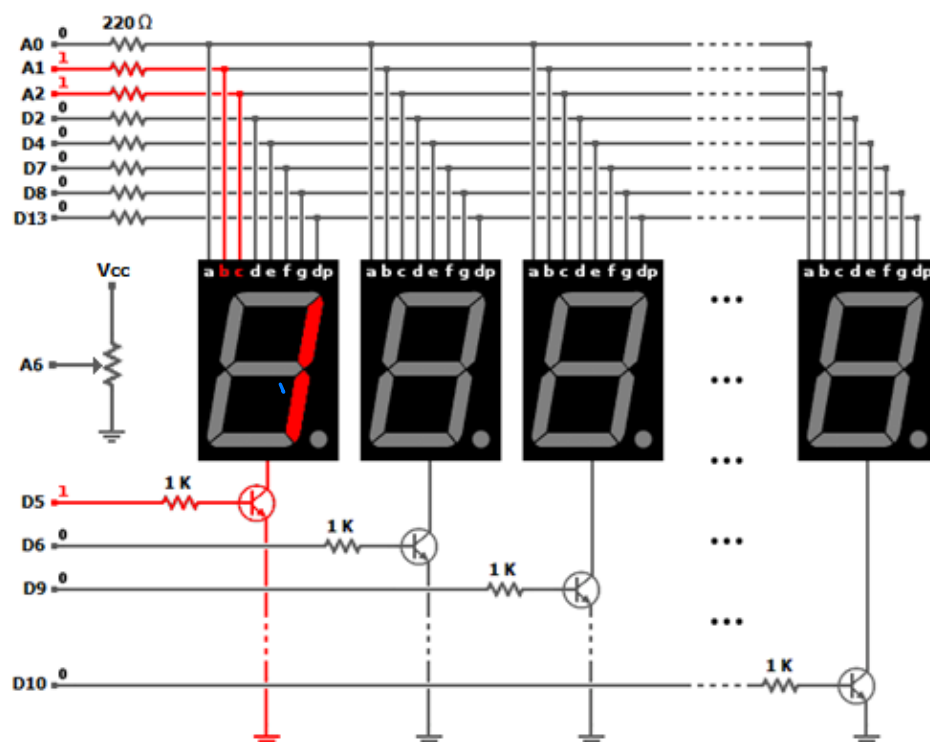


5. ให้ต่อวงจรโดยเพิ่มตัวความต้านทานปรับค่าได้ (VR) ทำหน้าที่แบ่งแรงดันไฟจาก Vcc ไปป้อนเข้าขาอนาล็อก อินพุต A6 ของบอร์ด Arduino Nano และให้แก้ไขโปรแกรมเพิ่มการอ่านค่าแรงดันไฟฟ้าที่ขา A6 ด้วยคำสั่ง `analogRead` ซึ่งค่าที่อ่านเข้ามาจะผ่านวงจร ADC ที่ทำหน้าที่แปลงแรงดันไฟฟ้าจากอนาล็อกไปเป็นสัญญาณดิจิทัลขนาด 10 บิต ในช่วงระหว่าง 0 ถึง 1023 โดยกำหนดว่าถ้าค่าเป็น 0 ให้ 7-Segment สว่างน้อยที่สุดและความสว่างจะเพิ่มขึ้นจนสูงสุดที่ค่า 1023 วิธีการปรับค่าความสว่างหรือหรี่หลอดไฟ ของ 7-Segment จะทำโดยการส่งสัญญาณที่เป็น PWM ไปที่ขา D5 โดยขา PWM นี้จะมีขนาด 8 บิต มีค่าระหว่าง 0 ถึง 255 เมื่อขา A6 มีอินพุตเข้ามาตามการปรับค่า VR จะต้องให้ขา D5 ส่งเอาท์พุทออกไปด้วยคำสั่ง `analogWrite` ควบคุมปรับค่าความถี่ไซเคิลตามการควบคุมคาบเวลาของสัญญาณที่เป็นลอจิก 1 เทียบกับคาบเวลาที่เป็นลอจิก 0 ซึ่งค่าที่ได้จะเป็นเปอร์เซ็นต์ตามค่าของอินพุตที่เข้ามาจากขา A6

7-Segment ที่ใช้ในการทดลองจะมี 4 หลัก โดยปกติแล้วจะใช้ขา Segment ร่วมกัน เพื่อประหยัดขาที่จะต่อออกมาภายนอก ถ้าวงจรเป็นแบบ Common Cathode จะต้องมีการ Common Cathode ของแต่ละ Digit แยกออกจากกันเป็น 4 ขาตามจำนวน Digit ที่ใช้ที่มีอยู่ 4 หลัก เมื่อต้องการแสดงตัวเลขบน Digit ใด ให้ส่งสัญญาณ HIGH และ LOW ไปที่ Segment Pin และต่อจุดร่วมที่เป็น Common Cathode ของ Digit นั้นลงกราวด์ ตัวอย่างของ 7-Segment เบอร์ FYLQ-5641A วงจรรวมภายในที่เป็นแบบ Common Cathode จะมีลักษณะดังนี้



6. ให้ต่อวงจรในส่วนของขา Common Cathode ในแต่ละ Digit เพิ่มตั้งแต่ Digit1 ถึง Digit4 โดยใช้ Transistor และตัวต้านทานขนาด 1 K เพื่อจะทำหน้าที่เป็นการ Scan ค่าของหลักที่ต้องการแสดง โดยใช้วงจรตามในรูป



การเขียนโปรแกรมจะต้องให้ Scan การแสดงผลตัวเลขทีละ Digit เรียงไปจนครบ 4 Digit โดยจะต้องส่งข้อมูลของ 7-Segment มา Latch ไว้ใน 7-Segment ของแต่ละ Digit ก่อน แล้วจึงทำการ Scan เพื่อให้ Digit นั้นทำงาน การทำงานของโปรแกรมจะต้องเร็วจนดูเหมือนว่า LED สว่างทั้ง 4 Digit พร้อมๆกัน เป็นการ Multiplex LED แบบหลาย Digit การทำงานของวงจรในลักษณะนี้ จะมีกระแสไหลผ่านในแต่ละ Digit ของขา Common Cathode เป็นจำนวนมากเกินกว่าที่ขาของ Arduino จะรับได้ จึงต้องใช้วิธีสั่งเปิด-ปิดในแต่ละ Digit ผ่านทางทรานซิสเตอร์เพื่อทำหน้าที่เป็นสวิตช์เปิด-ปิดแทน สำหรับการเขียนโปรแกรม Multiplex LED แบบ 4 หลัก ต้องสั่งให้ควบคุมทั้ง Segment และ Digit ให้ทำงานตรงกัน โดยการวนลูปให้แสดงทีละ Digit สลับไปเรื่อยๆ ซึ่งจะต้องลูปให้เร็วจนสายตามนุษย์มองไม่เห็นการกระพริบหรือดูไม่ทันการ Multiplex ในแต่ละ Digit จะใช้ช่วงเวลา 1 ใน 4 ของเวลาทั้งหมด ทำให้ความสว่างของ LED ในแต่ละ Digit จะลดลงเหลือเพียง 1/4 ของค่าความสว่างปกติ โดยทั่วไป LED จะยอมให้ Burst กระแสมากกว่าปกติในช่วงเวลาอันสั้น โดยดูได้จาก Parameter ค่า Peak Forward Current ของ Datasheet ดังนั้นหากต้องการให้ความสว่างมากขึ้นต้องคำนวณหาความต้านทานที่ใช้ในการควบคุมกระแสให้เหมาะสมในแต่ละ Segment

7. ให้คำนวณหากระแสที่ใช้ใน LED แต่ละหลอดใน 7 Segment แบบ 4 Digit เมื่อกำหนดให้ตัวความต้านทานเท่ากับ 220 โอห์ม และให้คำนวณหาค่าขนาดของความต้านทานค่าต่ำสุดที่จะใช้เพื่อต้องการให้ได้ความสว่างสูงสุด โดยค่าต่างๆที่นำมาใช้คำนวณให้อ้างอิงกับค่าใน Datasheet

$I_{LED} = \frac{V}{R}$ $= \frac{5-2}{220}$ $= \frac{3}{220}$ $= 0.013 \text{ A}$ $= 13 \text{ mA.}$	$R \propto \frac{1}{\text{Light}}$ <p>\therefore ถ้าแสงไฟส่องมาก R น้อยลง</p> $R = \frac{V}{I} = \frac{3}{0.03}$ $= 100 \text{ } \Omega$
--	---

8. ให้เพิ่มโปรแกรมเพื่อให้ 7 Segment แสดงผลเป็นแบบ 4 Digit โดยกำหนดตัวแปรเพิ่มจาก Hardware ที่ต่อไว้ดังนี้

```
int digit1 = 5;
int digit2 = 6;
int digit3 = 9;
int digit4 = 10;
```

การกำหนดให้ตำแหน่งขาที่ต่อเป็นเอาต์พุตลงใน void setup() เพิ่มเป็นดังนี้

```
pinMode(digit1, OUTPUT);
pinMode(digit2, OUTPUT);
pinMode(digit3, OUTPUT);
pinMode(digit4, OUTPUT);
```

9. ให้แก้ไขโปรแกรมเพื่อให้ 7 Segment แสดงผลเป็นเลข 1234 ในแต่ละ Digit ตามลำดับ โดยในส่วนของ displayNumber() ให้เพิ่มโปรแกรมการเปิดการทำงานของ Digit ต่างๆดังนี้

```
//Turn on a digit
```

```

switch(digit)
{
    case 1:
        digitalWrite(digit1, HIGH);
        break;
    case 2:
        digitalWrite(digit2, HIGH);
        break;
    case 3:
        digitalWrite(digit3, HIGH);
        break;
    case 4:
        digitalWrite(digit4, HIGH);
        break;
}

```

และเมื่อแสดงผลเรียบร้อยแล้วให้สั่งปิดการทำงานของ Digit ทั้งหมดดังนี้

```

//Turn off all digits
digitalWrite(digit1, LOW);
digitalWrite(digit2, LOW);
digitalWrite(digit3, LOW);
digitalWrite(digit4, LOW);

```

10. ถ้าจะแก้ไขโปรแกรมเพื่อให้ 7 Segment แบบ 4 Digit แสดงผลเป็นเลข 4321 พร้อมกันทั้ง 4 Digit โดยไม่มีการกระพริบต้องแก้ไขที่ส่วนไหนบ้าง

- เปลี่ยน delay ทุก 500 → 1 วินาทีให้หมดทั้ง 4 digit โดยไม่กระพริบได้
 - แก้ไขในส่วนของ case ให้เป็นไปเลขตาม โดยให้ displaySegment(0)
 ซึ่ง = เลขอะไรก็ได้ตาม case เช่น case 1; displaySegment(1)

11. ให้เขียนโปรแกรมเพื่อให้ 7 Segment แบบ 4 Digit แสดงผลเป็นรหัสนักศึกษาทั้ง 8 หลัก โดยเลื่อนข้อมูลจากขวาไปซ้าย

12. ให้ต่อวงจรเพื่อให้ทำหน้าที่เป็นโวลต์มิเตอร์ โดยใช้ตัวความต้านทานที่ปรับค่าได้ High Precision Trimmer Potentiometer Variable Resistor เบอร์ 3296W-102 ขนาด 1 K ohm ซึ่งสามารถปรับค่าความต้านทานได้โดยการหมุนปรับค่า จากขากลางของ VR ต่อเข้ากับขา A0 ของ Arduino และขาต้านข้างอีกสองขาให้ต่อกับกราวด์และขั้วของอุปกรณ์ที่ต้องการวัดค่าแรงดันไฟฟ้า เขียนโปรแกรมใช้คำสั่ง analogRead อ่านค่าจากขา A0 ซึ่งเป็นสัญญาณดิจิตอลขนาด 10 บิต แล้วแปลงค่าที่ได้เป็นระดับแรงดันไฟฟ้าออกมาแสดงค่าบนจอภาพ โดยใช้ serial monitor ซึ่งเป็นการสื่อสารแบบอนุกรมส่งค่าจากไมโครคอนโทรลเลอร์มายังคอมพิวเตอร์ ซึ่งมีประโยชน์ใช้ในการตรวจสอบการทำงานของไมโครคอนโทรลเลอร์ได้

```

void setup()
{
    Serial.begin(9600); // initialize serial communication at 9600 bits per second
}

void loop()
{
    int sensorValue = analogRead(A0); // read the input on analog pin 0
    float voltage = sensorValue * (5.0 / 1023.0); // Convert the analog reading (0 - 1023 to 0 - 5V)

    Serial.println(voltage); // print out the value
}

```

13. ถ้าต้องการทำเป็นโวลต์มิเตอร์ที่วัดค่าได้ตั้งแต่ 0 ถึง 10.00 V โดยมีความถูกต้องในแต่ละ step เท่ากับ 0.01 V จะต้องแก้ไขในส่วนใดบ้าง ให้แสดงวิธีคำนวณหาค่าพร้อมทั้งยกตัวอย่าง

$$\text{float voltage} = \text{Sensor Value} \times \frac{(5.00/1023)}{(10.00/1023)} \text{ โวลต์}$$

ex. $A_1 = 800$ | $V = \frac{10}{1023} \times 800$
 $= 7.82 \text{ V.}$

14. จากข้อ 12 ให้แก้ไขโปรแกรมสร้างเป็นโวลต์มิเตอร์ที่วัดค่าได้ตั้งแต่ 0 ถึง 10.00 V โดยเขียนโปรแกรมอ่านค่าที่ได้จากการวัด ซึ่งเป็นสัญญาณดิจิตอลขนาด 10 บิต แล้วแปลงค่าที่ได้ไปเป็นระดับแรงดันไฟฟ้าออกมาแสดงผลบน 7 Segment แบบ 4 Digit ให้มีความถูกต้องในแต่ละ step เท่ากับ 0.01 V
15. จากการทดลองเรื่องการวัดอุณหภูมิ โดยใช้ตัวเทอร์มิสเตอร์มาใช้เป็นเซ็นเซอร์ ที่แสดงผลออกมาทาง Serial Monitor ให้แก้ไขโปรแกรมเพิ่มการแสดงผลให้ไปแสดงผลที่ 7-segment ขนาด 4 หลัก โดยให้แสดงผลการวัดอุณหภูมิเป็นทศนิยม 2 ตำแหน่ง
16. จากข้อ 14 ให้แก้ไขวงจรและโปรแกรม เพื่อให้ใช้เป็นเครื่องวัดค่าตัวความต้านทาน หรือโอห์มมิเตอร์ ที่มีช่วงการวัดได้ 200 K Ω โดยให้แสดงผลเป็น 7-segment ขนาด 4 หลัก
17. จากข้อ 14 ให้แก้ไขวงจรและโปรแกรม เพื่อให้เป็นเครื่องวัดค่า Diode หรือ LED โดยให้แสดงค่า Forward Voltage ของอุปกรณ์ที่วัดได้เป็นทศนิยม 1 ตำแหน่ง
18. จากข้อ 17 ให้นำเอาเครื่องวัด LED ไปใช้ในการวัด LED แบบต่างๆและบันทึกผลที่ได้ลงในตาราง

LED	Wavelength (nm)	ค่าที่วัดได้ Forward Voltage (V)
Infrared		
สีแดง		
สีเขียว		
สีน้ำเงิน		
Ultra Violet		

