



คู่มือแนะนำการเรียนรู้ ARDUINO
สำหรับผู้เริ่มใช้

MAKER UNO



สวัสดี

ผม Eng Tong จาก Cytron Technologies ผมได้เขียนคู่มือนี้กับคุณด้วยใจ ไม่ว่าจะ เป็น นักเรียนหรือครู คู่มือนี้จะเป็นคู่มือเพื่อใช้สำหรับการเรียนรู้ Arduino ครั้งแรกของคุณ

เมื่อผมได้เริ่มเรียนรู้ Arduino เป็นครั้งแรก ผมก็ได้พบว่า มีเอกสารมากมาย ทั้งคู่มือการเรียนรู้และการฝึกสอนการใช้งานอย่างมากมาย ค่อนข้างกลืนกลาดบนอินเทอร์เน็ต เป็น เรื่องที่ตลกดีเหมือนกัน มันกลายเป็นปัญหาที่ ยิ่งใหญ่ของผม! มันมีเยอะเกินไปจนผมไม่รู้ว่า จะเริ่มจากตรงไหนแล้วพูดถึงเรื่องอะไรบ้าง

ทั้งนี้คู่มือการเรียนรู้หลายๆ ฉบับอ้างว่า ถูกสร้างขึ้นสำหรับผู้เริ่มต้น ซึ่งค่อนข้าง ผิดเพี้ยนเกินไป ทำให้เสียเวลาไปเยอะมาก ส่วนใหญ่จะแสดงให้คุณเห็นผังวงจร แล้วมี โค้ดง่ายๆให้คุณ และบอกให้คุณรู้ถึงสิ่งที่คาดหวัง แต่พวกเขาก็ไม่ได้อธิบายว่ามันทำงาน อย่างไร

มันน่าผิดหวังมากสำหรับคนอย่างผมที่ไม่มีพื้นฐานการเขียนโปรแกรมภาษา C แม้ว่าผม จะสามารถบรรลุผลลัพธ์ตามที่คาดหวังได้ โดยทำตามผังวงจรและคัดลอกโค้ดที่เขาให้มา ผมค่อนข้างแน่ใจว่า ผมไม่น่าจะสร้างโครงการของตนเองได้สำเร็จ หลังจากที่ผมจบ บทเรียน ทั้งหมดแล้วก็ตาม

นอกจากนี้ ผมสังเกตว่าคู่มือการเรียนรู้ส่วนใหญ่เขียนโดยนักเทคโนโลยีที่มีแนวโน้มว่า จะเข้าใจบางอย่างแบบเจาะลึก กล่าวอีกนัยหนึ่งว่า ความรู้พื้นฐานไม่ครอบคลุม และสิ่งที่นำเสนอเป็นแบบเชิง เทคนิคมากเกินไปสำหรับผู้เริ่มต้นที่จะเข้าใจ

สำหรับโครงการในคู่มือการเรียนรู้นี้ ไม่มีอะไรพิเศษพิสดาร โดยพื้นฐานแล้วผมเพิ่มคำ อธิบายในบางแง่มุม ที่ผมได้พยายามในตอนแรก และยังทำให้ส่วนเทคนิคดูง่ายขึ้นเพื่อ ความเข้าใจที่ดีขึ้น ผมหวังเป็นอย่างยิ่งว่าผู้เริ่มต้นจะได้รับประโยชน์จากคู่มือการเรียนรู้นี้ และสนุกกับการเรียนรู้ Arduino ครับ

ขอบคุณ



Tan Eng Tong
ผู้ก่อตั้งและ CEO
Cytron Technologies



ผจญไปกับ STEM และ Coding ด้วย Maker UNO X

เขียนโดย ET Tan

เรียบเรียงโดย Cheryl Ng

ภาพประกอบโดย Suhana Oazmi

พิสูจน์อักษรโดย รวินทร์ ไชยสิทธิพร

แปลเอกสารโดย

दनัย สุขเจริญ

วิทวัส โต๊ะเส้น

วรากร เทียนไทย

ภูวนาท เพื่อกทอง

พงศธรณ์ สระอุทัย

รวิน อัสนะบำรุงรัตน์

มงคล เพื่องฟูตระกูล

Johannes H Redelinghuys

v1.4.1 June 2020

Published by Cytron Technologies

ภาพรวม

บทที่ 0 การติดตั้ง

| | |
|---|---|
| ฮาร์ดแวร์ | 6 |
| ซอฟต์แวร์ - Arduino IDE และไดรฟ์เวอร์ | 7 |

LESSON 1 การส่งข้อมูลออกแบบดิจิทัล (DIGITAL OUTPUT)

| | |
|--|----|
| Project 1: เปิดไฟ LED..... | 14 |
| Project 2: สร้างไฟ LED ให้กะพริบ..... | 18 |
| Project 3: ให้ไฟ LEDs กะพริบเป็นลำดับ..... | 21 |

LESSON 2 การนำข้อมูลเข้าแบบดิจิทัล (DIGITAL INPUT)

| | |
|-------------------------------------|----|
| Project 4: สวิตช์ปุ่มกดบนบอร์ด..... | 27 |
| Project 5: สวิตช์ปุ่มกดภายนอก..... | 30 |

LESSON 3 การอ่านข้อมูลออกแบบอนาล็อก (ANALOG OUTPUT)

| | |
|-------------------------------|----|
| Project 6: สร้างวงจร LED..... | 36 |
| Project 7: จางแสง LED..... | 40 |

LESSON 4 เสียงเพลง (MELODY TONE)

| | |
|---|----|
| Project 8: การแต่งเพลงแบบง่ายๆ..... | 47 |
| Project 9: การแต่งทำนองเพลง "Happy Birthday"..... | 51 |
| Project 10: ปรับแต่งโค้ดของคุณ..... | 54 |

LESSON 5 การอ่านข้อมูลเข้าแบบอนาล็อก (ANALOG INPUT)

| | |
|--|----|
| Project 11: แสดงค่าอนาล็อกบน Serial Monitor..... | 61 |
| Project 12: อ่านค่าอนาล็อกจากอินฟราเรดเซ็นเซอร์..... | 65 |
| Project 13: การใช้อินฟราเรดเซ็นเซอร์ตรวจสอบเส้นสีดำ..... | 68 |

LESSON 6 การควบคุม DC MOTOR

| | |
|---|----|
| Project 14: การควบคุม DC Motor..... | 76 |
| Project 15: การควบคุมความเร็วมอเตอร์โดยใช้สวิตช์..... | 82 |

LESSON 7 เซ็นเซอร์อัลตราโซนิก

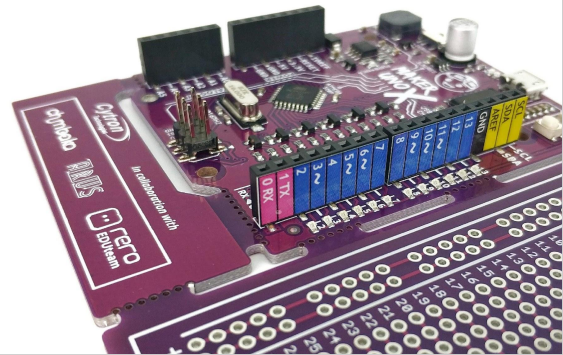
| | |
|--|----|
| Project 16: การเตรียมอัลตราโซนิกเซ็นเซอร์..... | 89 |
| Project 17: สร้างกันชนท้ายสำหรับรถยนต์..... | 97 |

บทเรียนที่ 0

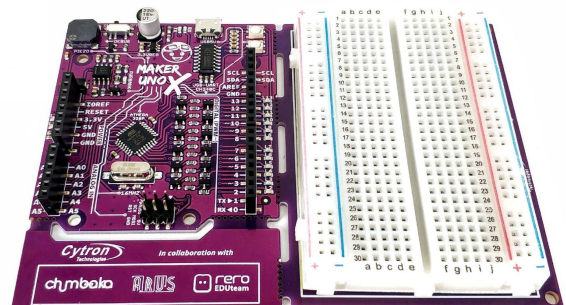
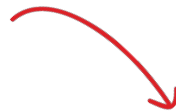
การติดตั้งฮาร์ดแวร์และ
ซอฟต์แวร์



-



UNO X

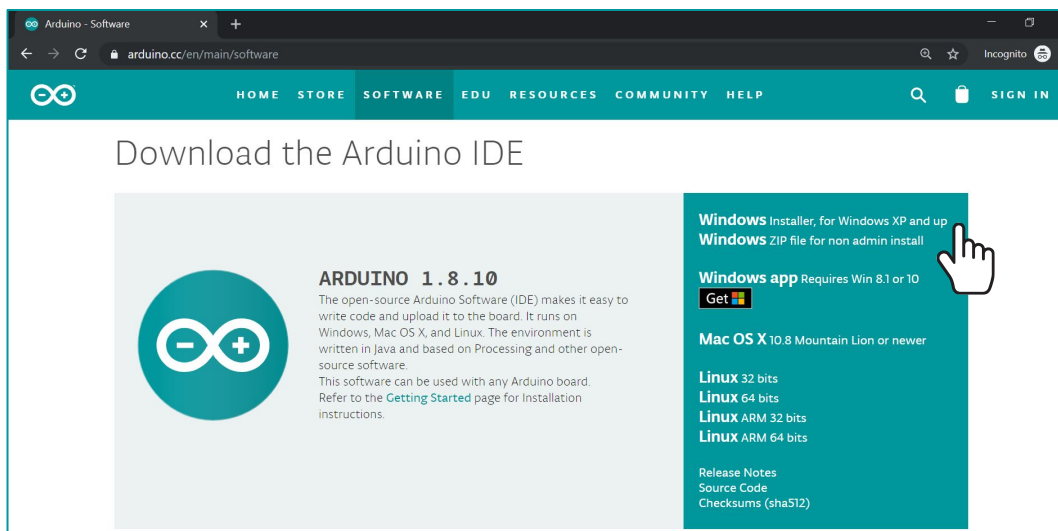


3. เชื่อมต่อบอร์ด Maker UNO X กับคอมพิวเตอร์ของคุณด้วยสาย micro USB



ส่วนที่ 2: ดาวน์โหลดและติดตั้ง Arduino IDE

1. เข้าไปที่เว็บ <https://www.arduino.cc/en/main/software>.
2. เลือกระบบปฏิบัติการ (OS) ของคุณเพื่อดำเนินการต่อ



สำหรับผู้ใช้ Windows ขอแนะนำให้เลือก **Windows Installer, for Windows XP** หรือเวอร์ชันสูงกว่า.



3. Arduino IDE เป็นซอฟต์แวร์แบบ open source ให้คลิกที่ปุ่ม “**JUST DOWNLOAD**” เพื่อดาวน์โหลดและใช้งานได้ฟรี



หมายเหตุ: ผู้ใช้ควรสนับสนุน ด้วยการบริจาคเงินเพื่อช่วยในการพัฒนา ผลงานที่ดีต่อไป



4. ดับเบิลคลิกที่ไฟล์ที่ดาวน์โหลดมาเพื่อดำเนินการต่อ



arduino-1.8.6-win
dows

สำหรับ Windows



arduino-1.8.6-macosx.zip
185.7 MB

สำหรับ Mac

5. เมื่อการติดตั้งเสร็จเรียบร้อยแล้ว ไอคอนของ Arduino จะปรากฏขึ้น ให้ดับเบิลคลิกที่ไอคอน Arduino เพื่อเปิดเข้าสู่โปรแกรม Arduino IDE



Arduino



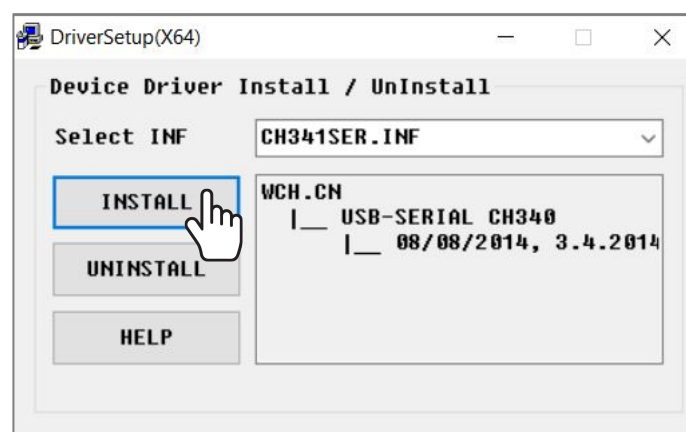
ส่วนที่ 3: การดาวน์โหลดและติดตั้ง Driver

สำหรับผู้ใช้งาน Windows:

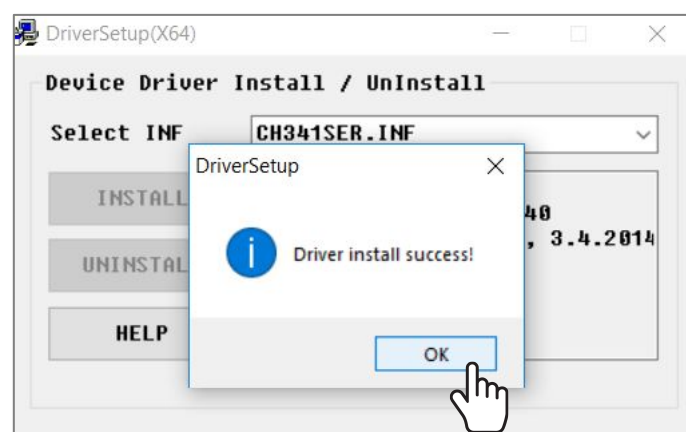
1. ดาวน์โหลด Driver จากที่นี่:
<https://cdn.cytron.io/makeruno/CH341SER.EXE>
2. ดับเบิลคลิกที่ไอคอน “CH341SER” เพื่อเริ่มติดตั้ง



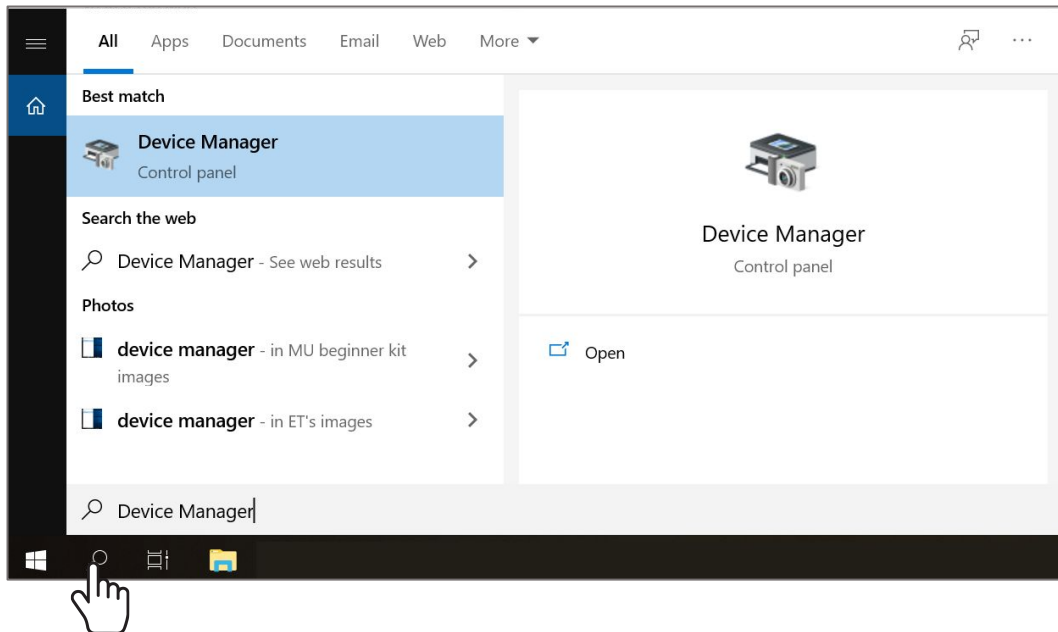
3. คลิกที่ “INSTALL”



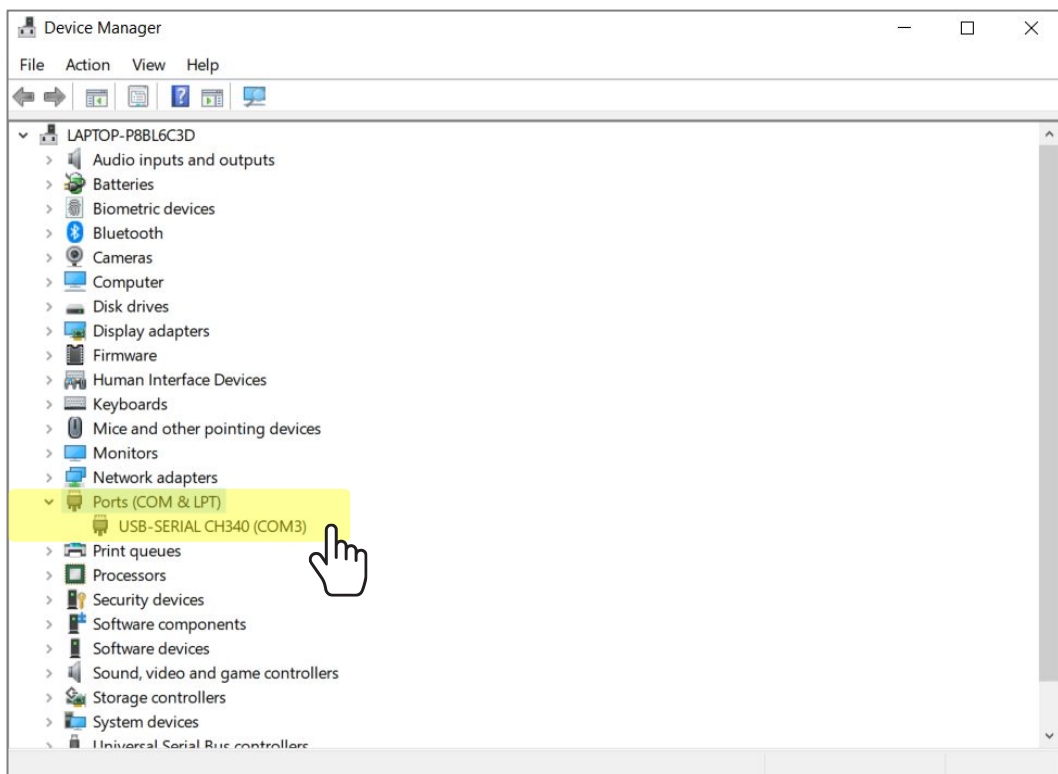
4. คลิกที่ “OK”.



5. คลิกที่ไอคอน Windows Search  แล้วพิมพ์ข้อความ “**Device Manager**” หรือ “ตัวจัดการอุปกรณ์” ในช่องค้นหา แล้วคลิก Open (เปิด)



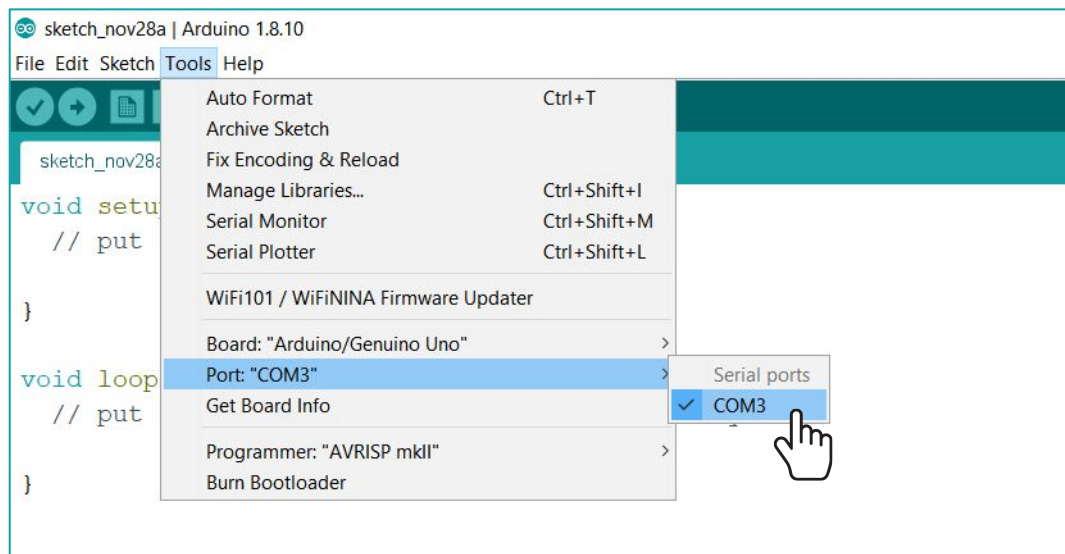
6. คลิกที่ “**Ports (COM & LPT)**”. ให้ขยายออกมา ตรวจสอบว่าพอร์ตใดที่ Driver ชื่อ CH340 ใช้งานอยู่ ให้จดหมายเลขพอร์ต COM นั้นไว้ (ตัวอย่าง: Driver ใช้พอร์ต COM3)



Maker UNO X ของคุณจำเป็นต้องเชื่อมต่อกับคอมพิวเตอร์



7. เข้าโปรแกรม Arduino IDE ไปที่ **Tools > Ports > COM_** แล้วเลือกพอร์ต COM ที่ถูกต้อง (ตามที่จดไว้).



สำหรับผู้ Mac:

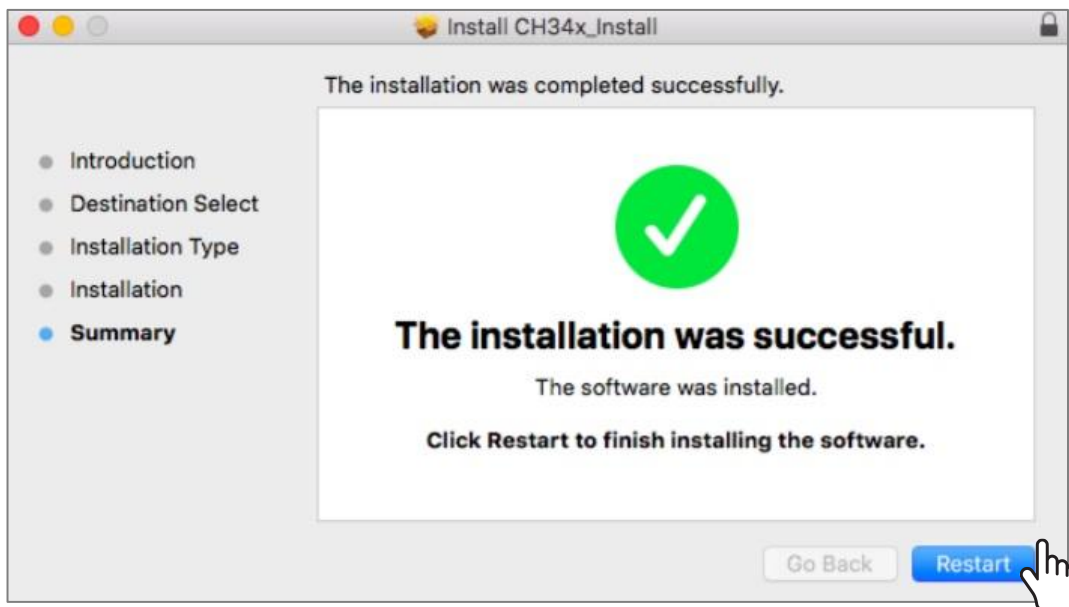
1. ดาวน์โหลดไดรฟ์เวอร์จากที่นี่:
https://cdn.cytron.io/makeruno/CH341SER_MAC.ZIP
2. ดับเบิลคลิกที่ไฟล์ zip เปิดโฟลเดอร์ที่แตกซิปออกมา แล้วดับเบิลคลิกที่ไฟล์ pkg



3. คลิก **“CONTINUE”** เพื่อเริ่มติดตั้ง

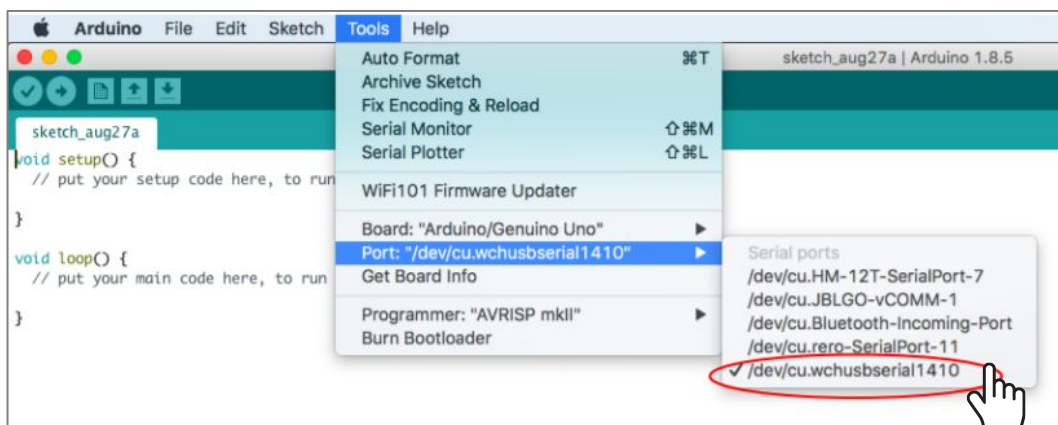


4. เมื่อเรียบร้อยแล้วคลิก **“Restart”**



5. หลังจากคุณรีสตาร์ทแมคของคุณแล้ว ให้เข้าโปรแกรม Arduino IDE แล้วเข้าไปที่

Tools > Port > /dev/cu.wchusbserial1410



หมายเหตุ: ถ้าคุณพบปัญหาในระหว่างติดตั้ง Driver ให้คุณทำตามวิธีแก้ปัญหานี้:

https://cdn.cytron.io/makeruno/Troubleshooting_CH431_Driver_For_Mac.pdf



บทเรียนที่ 1

การส่งข้อมูลออกแบบดิจิทัล
(DIGITAL OUTPUT)

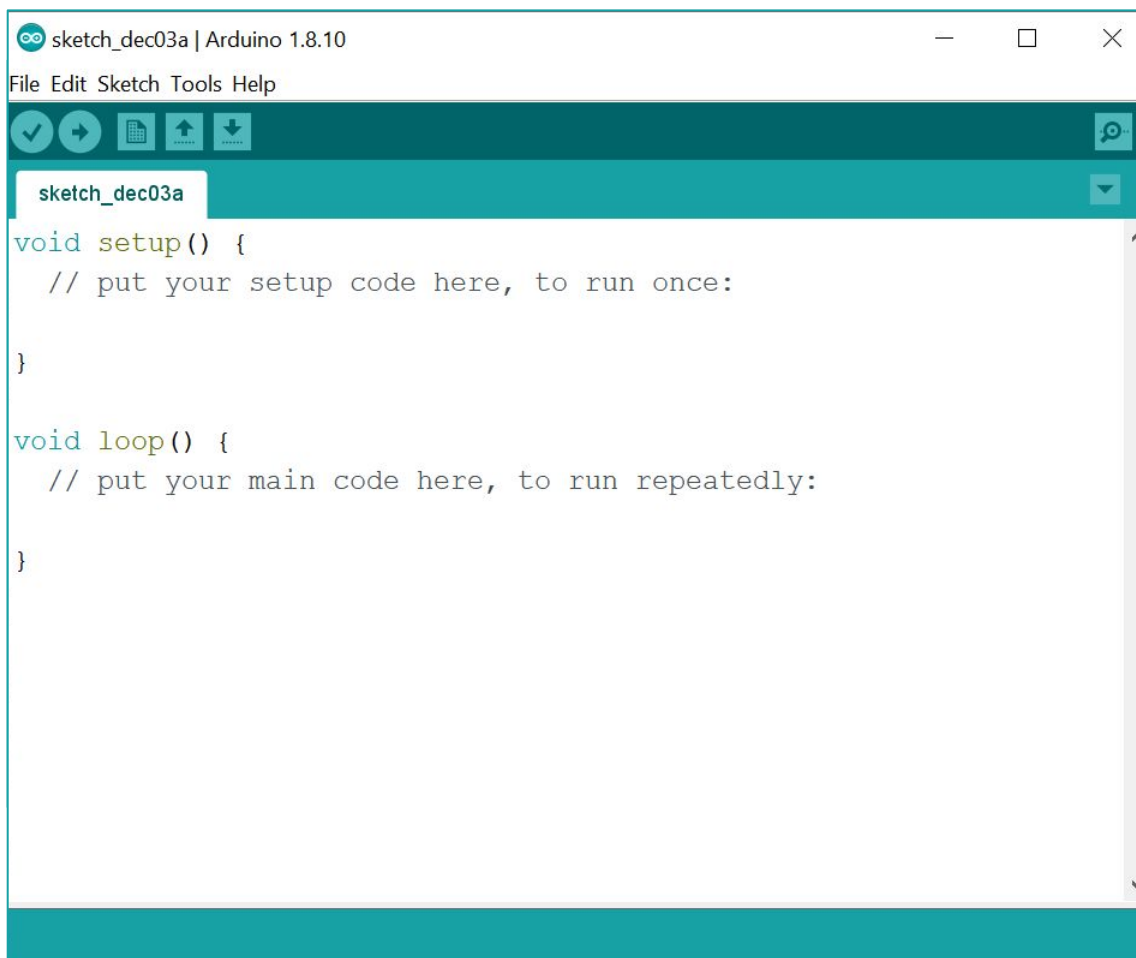


โปรเจกต์ 1: เปิดไฟ LED

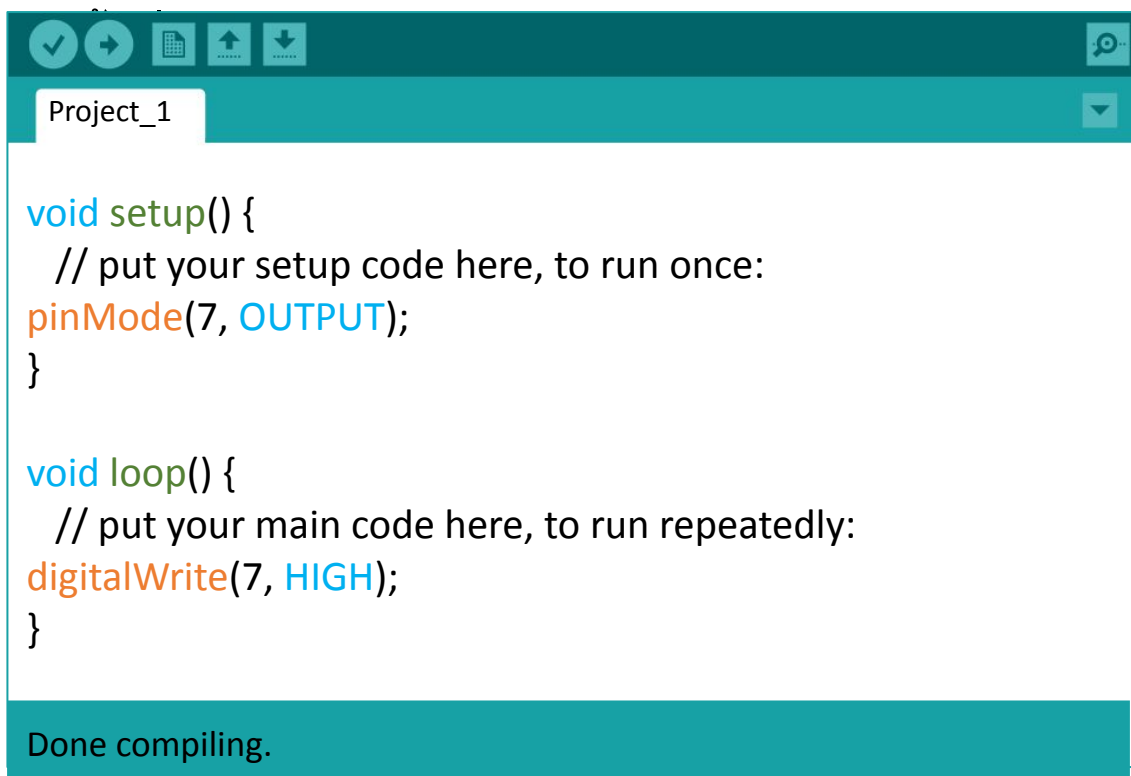
1. เชื่อมต่อ Maker UNO X กับคอมพิวเตอร์ด้วยสายเคเบิล Micro USB



2. เปิดโปรแกรม Arduino IDE จะขึ้นหน้าจอดังรูป



3. เขียนโค้ดตามรูป และกดปุ่ม  เพื่อคอมไพล์โค้ดดังกล่าว รอสักพักจะเห็นคำว่า “Done Compiling” หมายความว่า โค้ดของเราคอมไพล์ได้เสร็จสมบูรณ์ตรงด้านล่าง



```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(7, OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(7, HIGH);  
}
```

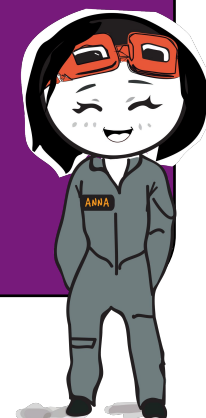
Done compiling.



การแก้ไขกรณีเกิดปัญหา

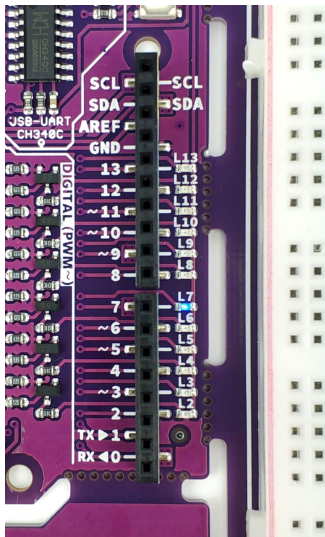
ถ้าเกิดปัญหาขึ้น ให้ลองตรวจโค้ดอีกครั้งว่าเขียนถูกหรือไม่ ที่
ละบรรทัด แล้วกดปุ่มอีกครั้งเพื่อคอมไพล์ใหม่

- ✓ ตัวอักษรพิมพ์เล็กและตัวอักษรพิมพ์ใหญ่มีความสำคัญในโค้ด Arduino (case sensitivity) ดังนั้นให้ตรวจสอบทุกครั้งเสมอว่าพิมพ์ตัวอักษรถูกหรือไม่ เช่นคำว่า OUTPUT ที่ต้องพิมพ์เป็นตัวอักษรพิมพ์ใหญ่ทุกตัว และคำว่า pinMode หรือ digitalWrite ที่มีตัวพิมพ์ใหญ่ระหว่างคำด้วย เรียกว่า Camel case
- ✓ ตรวจสอบว่าไม่ได้ลืมเครื่องหมายเหล่านี้ ; , () { }
- ✓ อย่าเผลอใช้เครื่องหมาย () และ { } ด้วยกัน เราต้องระลึกไว้เสมอว่ามันมีความหมายต่างกันนะ



4. ขั้นตอนต่อไปให้คลิกปุ่ม ➡ เพื่ออัปโหลดโค้ดลงในบอร์ด Maker UNO X เมื่อเสร็จแล้วจะเห็นคำว่า “Done Uploading” ขึ้นในตำแหน่งด้านล่างของโค้ด
5. ให้เลื่อนสวิตช์ของ Maker UNO X เพื่อเปิดโหมดดีบั๊ก แล้วดูผลการทดลอง

TIPS คำว่าดีบั๊ก (debug) หมายถึงการแก้ไขจุดบกพร่อง

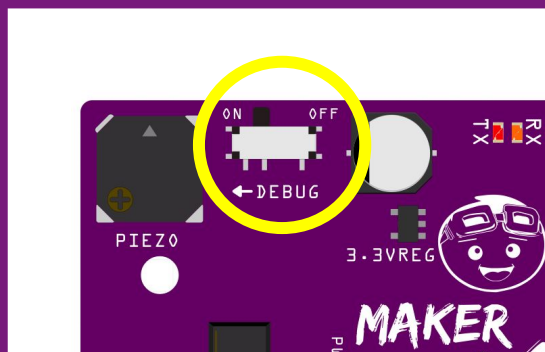


ไฟ LED ที่หมายเลข 7 จะเปิดขึ้นมา



การแก้ไขกรณีเกิดปัญหา

1. ถ้าเกิดข้อผิดพลาดขณะอัปโหลดโค้ดเข้าบอร์ด ให้ตรวจสอบว่าเชื่อมต่อบอร์ดกับคอมพิวเตอร์หรือยัง และตรวจสอบว่าได้เลือกพอร์ตถูกต้องแล้วหรือไม่ (ไปที่ Tools > Port > COM_).
2. จำไว้ว่าให้เปิดโหมดดีบั๊ก (Debug) เพื่อให้ LEDs ใน Maker UNO X เริ่มทำงานเสมอ





มันทำงานได้อย่างไร

```
Project_1

void setup() {
  // put your setup code here, to run once:
  pinMode(7, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(7, HIGH);
}
```

กำหนดให้พินที่ 7 เป็นพินสำหรับส่งออกข้อมูล (output)

กำหนดสถานะให้พินที่ 7 เป็น HIGH ซึ่งก็คือเปิดไฟนั่นเอง



เรื่องที่เราควรรู้

ฟังก์ชันใน Arduino

1. การกำหนดสถานะให้พิน เช่นเป็นพินนำเข้าข้อมูล (input) หรือเป็นพินส่งออกข้อมูล (output)

pinMode (หมายเลขพิน, สถานะ)

โดยเราสามารถกำหนดสถานะได้เป็น
INPUT, OUTPUT, หรือ INPUT_PULLUP

2. การกำหนดค่าให้พินเป็น HIGH หรือ LOW

digitalWrite(หมายเลขพิน, ค่า)

หมายเลขพินคือหมายเลขของพินที่ต้องการ
กำหนดค่าให้

ค่าก็หมายถึง HIGH (เปิด) หรือ LOW (ปิด)





โปรเจกต์ 2: สร้างไฟ LED ให้กระพริบ

1. เพิ่มโค้ดจากโปรเจกต์ 1 ให้เป็นตามรูป

```
Project_2

void setup() {
  // put your setup code here, to run once:
  pinMode(7, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(7, HIGH);
  delay (100);
  digitalWrite(7, LOW);
  delay (100);
}
```

2. คอมีไฟล์โค้ด และอัปโหลดโค้ดเข้า Maker UNO X
3. ดูผลการทดลอง

คลิกที่นี่หรือสแกนคิวอาร์โค้ดเพื่อชมวิดีโอ

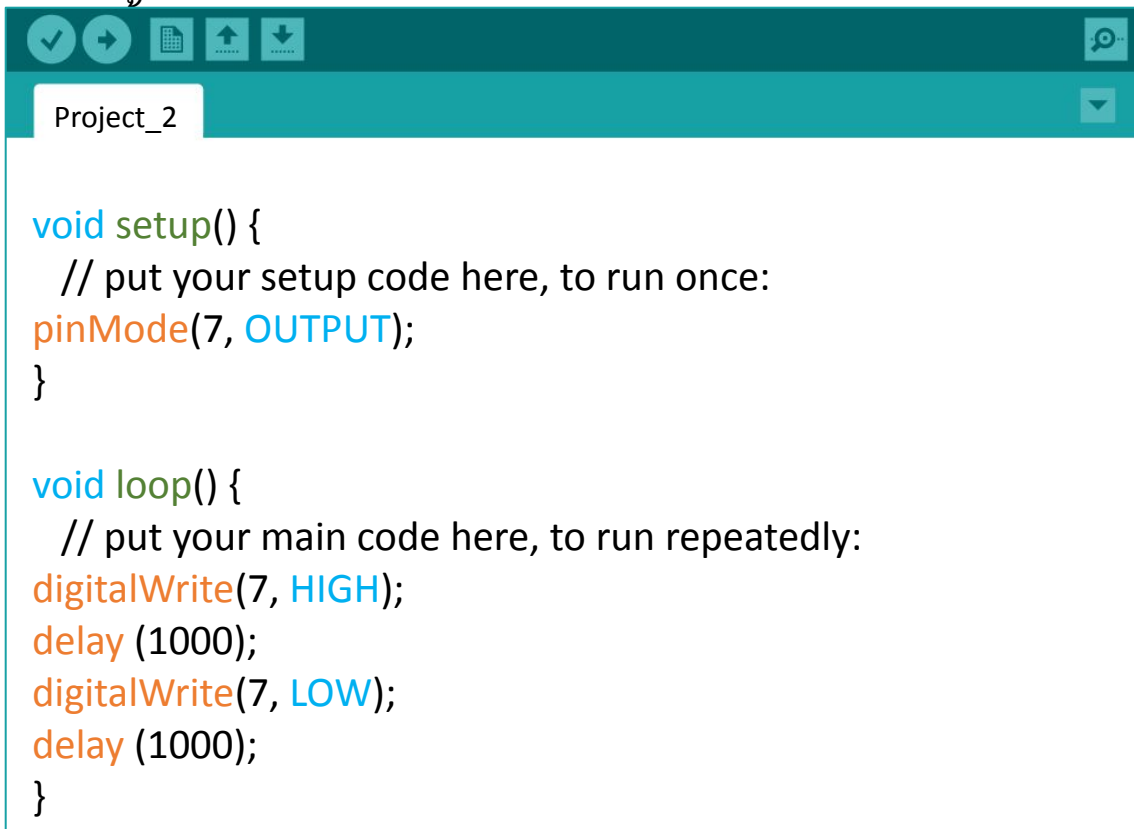
สาธิต



LED 7 กระพริบในขณะที่ LEDs อื่น ๆ ดับ



4. เปลี่ยนเลขในวงเล็บของ delay จาก 100 ให้เป็น 1000 และลอง



```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(7, OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(7, HIGH);  
  delay (1000);  
  digitalWrite(7, LOW);  
  delay (1000);  
}
```

5. เปรียบเทียบผลการทดลองครั้งนี้กับรอบที่แล้ว หากความแตกต่างดู



มันทำงานได้อย่างไร

```
Project_2

void setup() {
  // put your setup code here, to run once:
  pinMode(7, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(7, HIGH);
  delay (1000);

  digitalWrite(7, LOW);
  delay (1000);
}
```

กำหนดให้พินที่ 7 เป็นพินสำหรับส่งออก

กดเปิดไฟ LED ที่ 7
และรอ 1000 มิลลิวินาที

กดปิดไฟ LED ที่ 7
และรอ 1000 มิลลิวินาที



และโปรแกรมจะ
ทำวนไปเรื่อย ๆ



เรื่องที่คุณควรรู้

ฟังก์ชันใน Arduino

1. การสั่งให้โปรแกรมหยุดรอ (หน่วงเวลา)
delay(เวลาในหน่วยมิลลิวินาที)

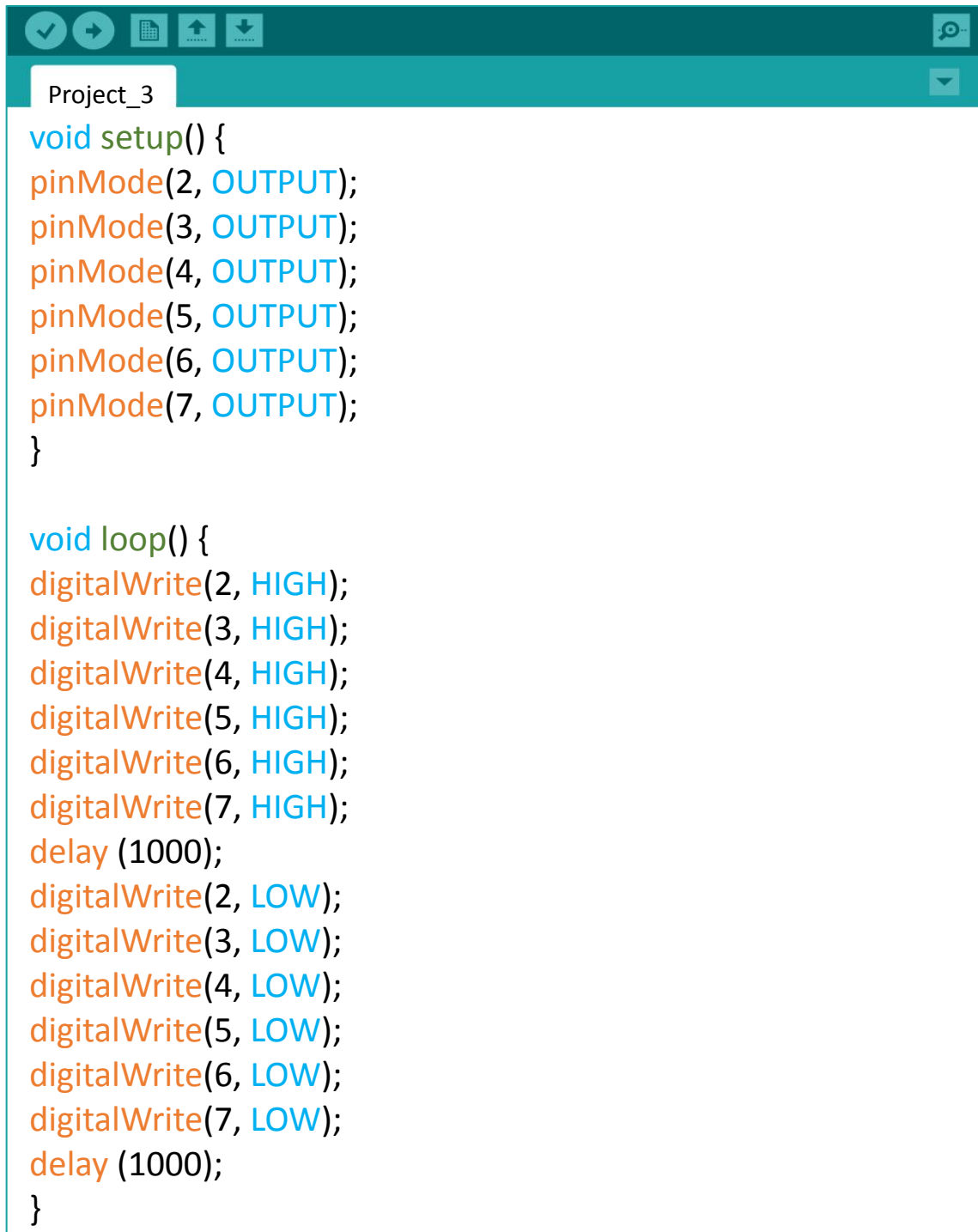


โปรเจกต์ 3: ให้ไฟ LEDs กระพริบเป็นลำดับ

LEDs หลายตัวจะกระพริบเป็นลำดับกัน และจะใช้ LEDs ทั้งหมด 6 ตัว ประกอบด้วย

LED 2, LED 3, LED 4, LED 5, LED 6, และ LED 7

1. แก้โค้ดให้เป็นตามรูป



```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(2, HIGH);  
  digitalWrite(3, HIGH);  
  digitalWrite(4, HIGH);  
  digitalWrite(5, HIGH);  
  digitalWrite(6, HIGH);  
  digitalWrite(7, HIGH);  
  delay (1000);  
  digitalWrite(2, LOW);  
  digitalWrite(3, LOW);  
  digitalWrite(4, LOW);  
  digitalWrite(5, LOW);  
  digitalWrite(6, LOW);  
  digitalWrite(7, LOW);  
  delay (1000);  
}
```

2. คอมีไฟล์โค้ด และอัปโหลดโค้ดเข้า Maker UNO X

3. ดูผลการทดลอง

คลิกที่นี่หรือสแกนคิวอาร์โค้ดเพื่อชมวิดีโอสาธิต



LED2 ไปจนถึง LED 7 กระพริบไปพร้อม ๆ กันใน ขณะที่ LEDs อื่น ๆ ดับ แต่ว่าทำไมมันไม่กระพริบ เป็นลำดับเรียงกันไปล่ะ?



มันทำงานได้อย่างไร

```
Project_3

void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  delay (1000);

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  delay (1000);
}
```

กำหนดให้พินที่ 2 ถึงพินที่ 7 เป็นพินสำหรับส่งออกข้อมูล (output)

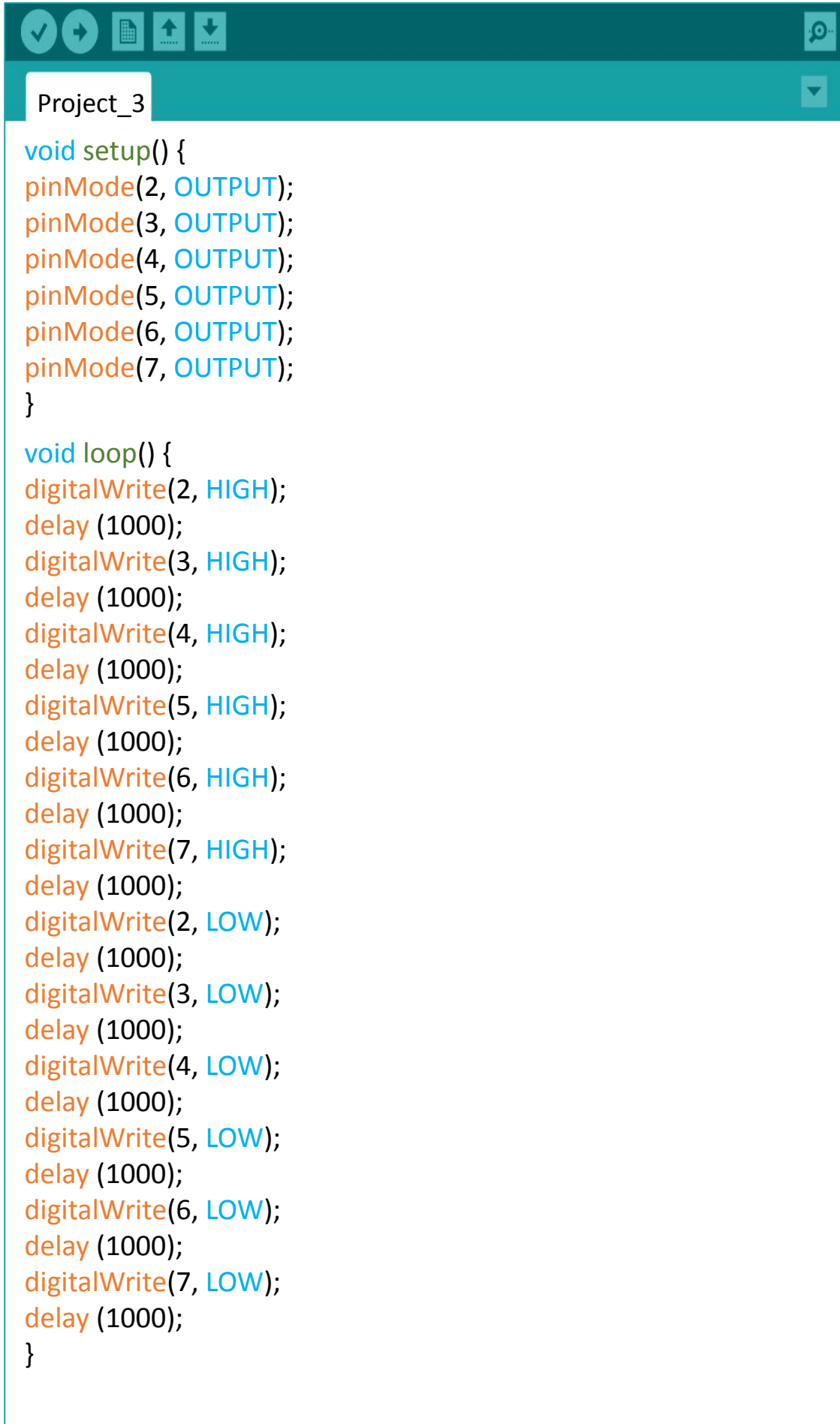
กดเปิดไฟ LED ที่ 2 ไปจนถึง LED ที่ 7

และรอ 1000 มิลลิวินาที

กดปิดไฟ LED ที่ 2 ไปจนถึง LED ที่ 7

และรอ 1000 มิลลิวินาที

4. แก๊วโค้ดอีกรอบให้เป็นตามรูป



```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(2, HIGH);  
  delay (1000);  
  digitalWrite(3, HIGH);  
  delay (1000);  
  digitalWrite(4, HIGH);  
  delay (1000);  
  digitalWrite(5, HIGH);  
  delay (1000);  
  digitalWrite(6, HIGH);  
  delay (1000);  
  digitalWrite(7, HIGH);  
  delay (1000);  
  digitalWrite(2, LOW);  
  delay (1000);  
  digitalWrite(3, LOW);  
  delay (1000);  
  digitalWrite(4, LOW);  
  delay (1000);  
  digitalWrite(5, LOW);  
  delay (1000);  
  digitalWrite(6, LOW);  
  delay (1000);  
  digitalWrite(7, LOW);  
  delay (1000);  
}
```

5. ดูผลการทดลอง

คลิกที่นี่หรือสแกนคิวอาร์โค้ดเพื่อชมวิดีโอสาธิต



LEDs ทั้ง 6 หลอด กระพริบเปิดที่ละดวง และปิดที่ละดวง



เรื่องที่คุณควรรู้

```
void loop() {  
  digitalWrite(2, HIGH);  
  digitalWrite(3, HIGH);  
  digitalWrite(4, HIGH);  
  digitalWrite(5, HIGH);  
  digitalWrite(6, HIGH);  
  digitalWrite(7, HIGH);  
  delay (1000);  
}
```

เวอร์ชันแรก

```
void loop() {  
  digitalWrite(2, HIGH);  
  delay (1000);  
  digitalWrite(3, HIGH);  
  delay (1000);  
  digitalWrite(4, HIGH);  
  delay (1000);  
  digitalWrite(5, HIGH);  
  delay (1000);  
  digitalWrite(6, HIGH);  
  delay (1000);  
  digitalWrite(7, HIGH);  
  delay (1000);  
}
```

เวอร์ชันที่สอง

แม้ว่าทั้งสองเวอร์ชันมีการเปิดไฟ LED และปิดไฟ LED เหมือนกัน แต่ผลการทดลองที่ได้นั้นต่างกัน เป็นเพราะว่าในเวอร์ชันแรก LEDs ถูกสั่งให้เปิดโดยไม่มีดีเลย์ และไฟในบอร์ดทำตามคำสั่งที่ละคำสั่งเร็วมาก (ใช้เวลาน้อยกว่าหนึ่งไมโครวินาทีหรือหนึ่งในล้านส่วนของวินาที)

ดังนั้นเพื่อให้ไฟ LEDs เปิดหรือปิดทีละดวงให้เห็นได้ด้วยตา เราจึงต้องใส่คำสั่ง delay หรือก็คือคำสั่งหยุดหลังจากจบทุก ๆ บรรทัดของการสั่งเปิดปิดไฟ ดังโค้ดของเวอร์ชันที่สอง





คำสั่ง: เขียนคำสั่งให้ LED ที่ 2 ไปจนถึง LED ที่ 13 กระพริบตามวิดีโอในตัวอย่าง

คลิกหรือสแกนคิวอาร์โค้ดเพื่อรับวิดีโอ



ยินดีด้วย! คุณได้ผ่านการเรียนรู้ในบทเรียนที่ 1 แล้ว ตอนนี้คุณสามารถทำงานดังต่อไปนี้ได้แล้ว

1. กำหนดพินให้เป็นพินส่งข้อมูลขาออกแบบดิจิทัล(เปิด-ปิด)
2. กำหนดค่าให้พินดิจิทัลเป็น HIGH หรือ LOW
3. ใช้ฟังก์ชัน delay เพื่อให้บอร์ดหยุดรอ

ขอให้โชคดีในบทต่อ ๆ ไปนะครับ (จากผู้แปล)



บทเรียนที่ 2

การนำข้อมูลเข้าแบบดิจิทัล
(DIGITAL INPUT)



โปรเจก 4: สวิตช์ปุ่มกดบนบอร์ด

ในโปรเจกนี้, เราต้องการควบคุมไฟ LED โดยใช้สวิตช์ปุ่มกดบนบอร์ด

1. เปิดแบบร่างใหม่และจากนั้นเขียนโค้ดดังต่อไปนี้

```
void setup() {  
  pinMode(4, OUTPUT);  
  pinMode(2, INPUT_PULLUP);  
}  
  
void loop() {  
  if (digitalRead(2) == LOW) {  
    digitalWrite(4, HIGH);  
  }  
  else {  
    digitalWrite(4, LOW);  
  }  
}
```

2. คอมไพล์และอัปโหลดโปรแกรม ตรวจสอบผลลัพธ์ของคุณ

คลิกที่นี่ หรือ สแกน
QR code เพื่อรับชม
วิดีโอตัวอย่าง



ไฟ LED ดวงที่ 4 จะสว่างขึ้น
เมื่อปุ่มถูกกด





มันทำงานอย่างไร

```
Project_4
void setup() {
  pinMode(4, OUTPUT);
  pinMode(2, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(2) == LOW) {
    digitalWrite(4, HIGH);
  }
  else {
    digitalWrite(4, LOW);
  }
}
```

กำหนดพิน 4 เป็นเอาต์พุต
กำหนดพิน 2 เป็นอินพุตสวิตช์บนบอร์ด

อ่านค่าจากพิน 2 ถ้ามีค่าเป็น LOW (สวิตช์ถูกกด)
จะกำหนดพิน 4 เป็น HIGH (เปิดไฟ LED ดวงที่ 4)

ถ้าเป็นอย่างอื่น (สวิตช์ไม่ถูกกด) จะกำหนดพิน
4 เป็น LOW (ปิดไฟ LED ดวงที่ 4)



สิ่งดีที่ควรรู้

ชุดคำสั่ง ARDUINO

1. ใช้สวิตช์ปุ่มกดบนบอร์ด, เราต้องกำหนดมันเป็นอินพุตแบบพูลอัพภายใน `pinMode(2, INPUT_PULLUP);`
2. ไฟ LED บนบอร์ดที่พิน 2 จะทำหน้าที่เป็นตัวชี้วัดของอินพุตเมื่อคุณเปิดโหมด debug, ไฟ LED ดวงที่ 2 จะไม่สว่างเมื่อสวิตช์บนบอร์ดถูกกด
3. สวิตช์บนบอร์ดถูกเชื่อมต่อกันภายในไปยังพิน 2 หมายความว่ามันถูกใช้งานอยู่ จะไม่สามารถเชื่อมต่อไปยังอุปกรณ์ภายนอกอื่นๆ ได้เมื่อมันถูกใช้งาน.

รูปแบบการเขียน

1. ใช้คำสั่ง “if-else”

```
if (เงื่อนไขที่ 1) {  
    // ทำ A  
}  
else if (เงื่อนไขที่ 2) {  
    // ทำ B  
}  
else {  
    // ทำ C  
}
```

2. เราสามารถใช้ // (เครื่องหมาย / ติดกันสองครั้ง) เพื่อละไว้ ใช้สำหรับช่วยเตือนหรืออธิบายระหว่างเขียนโปรแกรม การเขียนสิ่งใดๆ หลังเครื่องหมายนี้จะถูกละเว้น โดยคำสั่งข้างหลังจะไม่ถูกประมวลผล

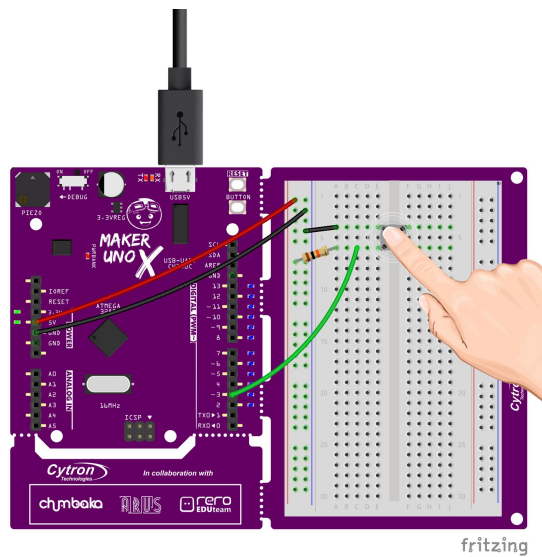
```
// เขียนคำอธิบายของคุณที่นี่
```



1. นำส่วนประกอบเหล่านี้



3. กดสวิตช์ปุ่มภายนอก สังกัดไฟ LED ดวงที่ 3 ว่ามันจะสว่างก่อนที่คุณกดสวิตช์ และดับเมื่อสวิตช์ถูกกด ถ้ามันไม่ทำงาน, โปรดแก้ไขวงจรไฟฟ้าของคุณ ก่อนที่คุณดำเนินการไปยังขั้นตอนต่อไป



4. แก้ไขโค้ดโปรเจก 4 ของคุณตามดังนี้ และอัปโหลดไปยัง Maker UNO X ของคุณ

```
Project_5

void setup() {
  pinMode(4, OUTPUT);
  pinMode(3, INPUT);
}

void loop() {
  if (digitalRead(3) == LOW) {
    digitalWrite(4, HIGH);
  }
  else {
    digitalWrite(4, LOW);
  }
}
```

5. ตรวจสอบผลลัพธ์ของคุณ

คลิกที่นี่ หรือ สแกน
QR code เพื่อรับชม
วิดีโอตัวอย่าง



เมื่อสวิตช์ปุ่มกดภายนอกถูกกด, ไฟ LED ดวงที่ 4 จะสว่าง



มันทำงานอย่างไร

```
Project_5

void setup() {
  pinMode(4, OUTPUT);
  pinMode(3, INPUT);
}

void loop() {
  if (digitalRead(3) == LOW) {
    digitalWrite(4, HIGH);
  }
  else {
    digitalWrite(4, LOW);
  }
}
```

กำหนดพิน 4 เป็นเอาต์พุต
กำหนดพิน 3 เป็นอินพุต

อ่านค่าจากพิน 3 ถ้ามันเป็น LOW (สวิตช์ถูกกด),
กำหนดพิน 4 เป็น HIGH (เปิดไฟ LED ดวงที่ 4)

ถ้าเป็นอย่างอื่น (สวิตช์ไม่ถูกกด),
กำหนดพิน 4 เป็น LOW (ปิดไฟ LED ดวงที่ 4)



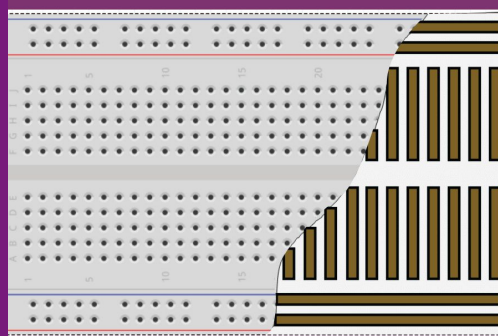
สิ่งดีที่ควรรู้

ชุดคำสั่ง ARDUINO

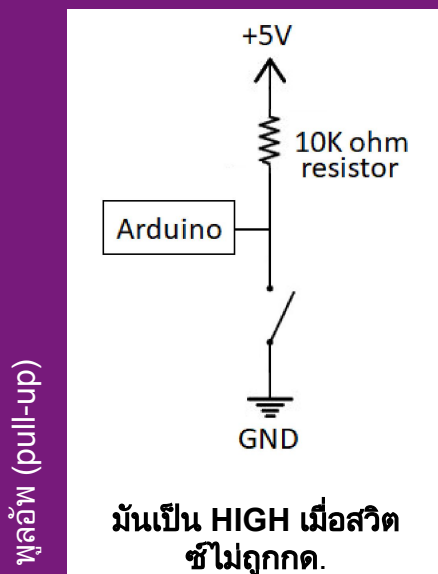
1. สวิตช์คืออุปกรณ์อินพุตแบบหนึ่ง คุณต้องกำหนดพินนั้นเป็นอินพุตก่อนที่คุณจะใช้มัน
`pinMode(pin, INPUT);`

อิเล็กทรอนิกส์พื้นฐาน

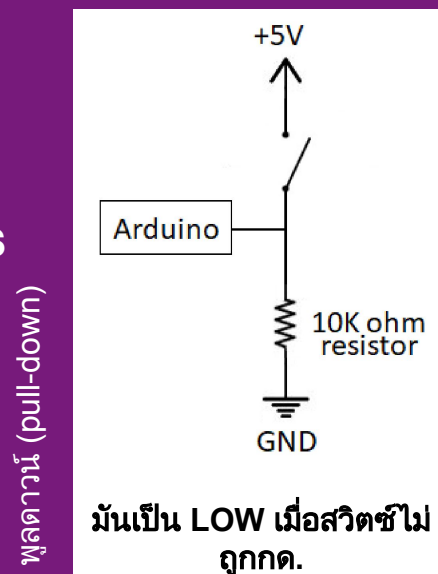
1. การเชื่อมต่อภายในของเบรด์บอร์ด(Breadboard)



2. สำหรับอินพุตดิจิทัลใดๆ, คุณสามารถทำมันเป็นวงจรไฟฟ้าแบบพูลอัพ (pull-up) หรือแบบพูลดาวน์ (pull-down) ได้



VS



ในโปรเจกต์นี้, เราใช้วงจรไฟฟ้าแบบพูลอัพ (pull-up) คุณสามารถเลือกใช้วงจรไฟฟ้าแต่ละแบบสำหรับโปรเจกต์ในอนาคตของคุณ แต่อย่าลืมเปลี่ยนโค้ดให้สอดคล้องตาม



ขอทำ

งาน: ใช้ทั้งสวิตช์บนบอร์ดและภายนอก
เมื่อสวิตช์ทั้งคู่ไม่ถูกกด, ไฟ LED ดวงที่ 4 และ 5 ทั้งคู่จะสว่าง
ถ้าสวิตช์บนบอร์ดถูกกด, ไฟ LED ดวงที่ 4 ดับ
ถ้าสวิตช์ภายนอกถูกกด, ไฟ LED ดวงที่ 5 ดับ

คลิกที่นี่ หรือ สแกน
QR code เพื่อรับชม
วิดีโอตัวอย่าง



ขอแสดงความยินดี! คุณได้จบบทเรียนที่ 2 และได้เรียนรู้ดังนี้:

1. วิธีอ่านสัญญาณอินพุตดิจิทัล
2. วิธีควบคุมไฟ LED โดยใช้สวิตช์
3. วิธีสร้างวงจรไฟฟ้าแบบพูลอัพ(pull-up) และแบบพูลดาวน์(pull-down) แบบง่าย
4. วิธีใช้คำสั่ง if-else



บทเรียนที่ 3

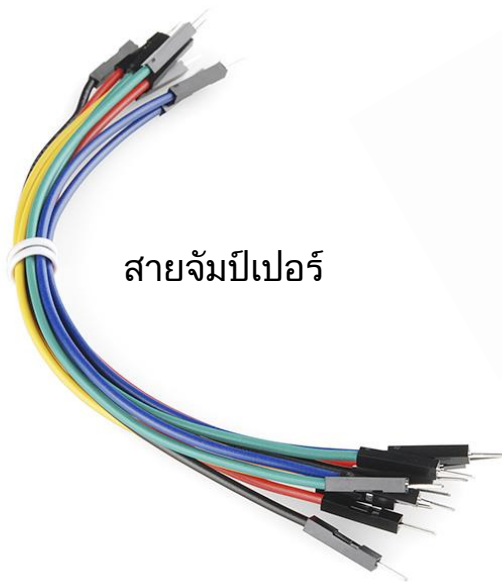
การส่งข้อมูลออกแบบอนาล็อก
(ANALOG OUTPUT)



โครงการ 6: สร้างวงจร LED

โครงการนี้ เราจะทำการสร้างวงจรอย่างง่ายด้วย LED หนึ่งดวง

1. เตรียมอุปกรณ์ดังนี้



สายจัมป์เปอร์

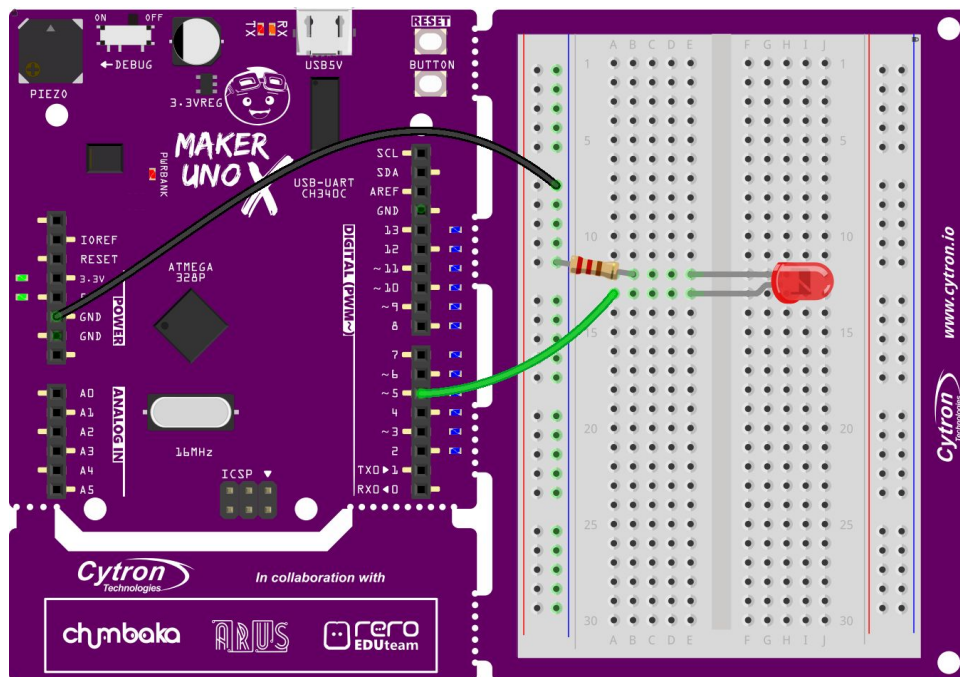


LED



ตัวต้านทาน 220 โอห์ม

1. สร้างวงจรตามรูปด้านล่าง



fritzing

3. เขียนและอัปโหลดโค้ดตามด้านล่างไปที่บอร์ดของคุณ

```
Project_6

void setup() {
  pinMode(2, INPUT_PULLUP);
  pinMode(5, OUTPUT);
}

void loop() {
  if (digitalRead(2) == LOW)
    while (1) {
      digitalWrite(5, HIGH);
      delay(200);
      digitalWrite(5, LOW);
      delay(200);
    }
  else digitalWrite(5, LOW);
}
```

3. ตรวจสอบผลลัพธ์

คลิกที่นี่หรือสแกน
QR code เพื่อดูวิดีโอสาธิต



ทั้ง LED บนบอร์ด และ LED ภายนอก ที่ขา Pin 5
จะกระพริบหลังจากทำการกดสวิตช์บนบอร์ด
LED 5 จะยังไม่หยุดกระพริบจนกว่าจะกดปุ่มรีเซ็ต





ทำงานอย่างไร

```
Project_6

void setup() {
  pinMode(2, INPUT_PULLUP);
  pinMode(5, OUTPUT);
}

void loop() {
  if (digitalRead(2) == LOW)
    while (1){
      digitalWrite(5, HIGH);
      delay(200);
      digitalWrite(5, LOW);
      delay(200);
    }
  else digitalWrite(5, LOW);
}
```

ตั้งค่า Pin 2 เป็นพูลอัปอินพุตสวิตช์
ตั้งค่า Pin 5 (ต่อกับ LED ภายนอก) เป็น
เอาต์พุต

อ่าน Pin 2 ถ้าค่าเป็น LOW (สวิตช์ถูกกด)
ตั้งค่า Pin 5 เป็น HIGH นาน 200ms จาก
นั้นตั้งค่าเป็น LOW นาน 200ms

*เมื่อ Pin 2 ถูกกด LED จะกระพริบต่อเนื่อง
และจะไม่หยุดจนกว่าจะกดปุ่มรีเซ็ต.

หรือถ้าสวิตช์ไม่ถูกกด ทำการตั้งค่า Pin 5
เป็น LOW (LED ที่ต่อกับ Pin 5 จะไม่เปล่ง
แสง)

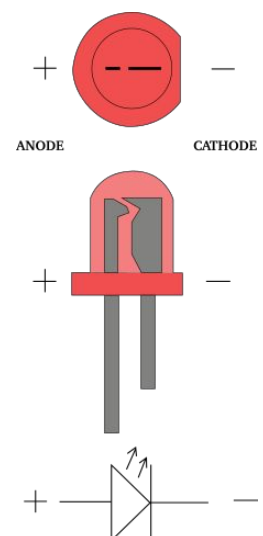


เรื่องดีน่ารู้

อิเล็กทรอนิกส์พื้นฐาน

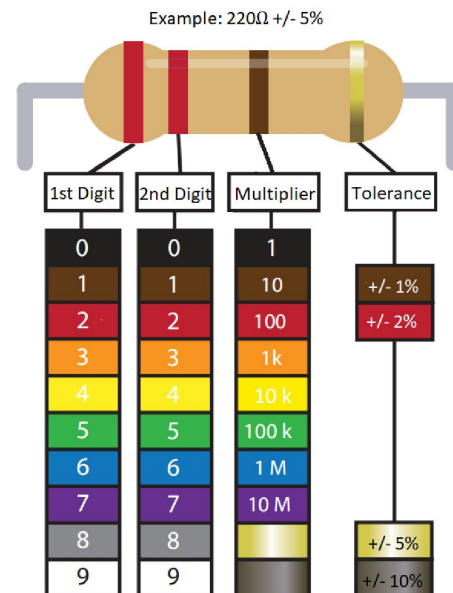
1. ไดโอดเปล่งแสง (LED) มี 2 ขา คุณต้องต่อขาด้านที่ยาวกว่าเข้ากับขั้วบวก + และขาด้านที่สั้นกว่า (ขอบเรียบบนปลอก) เข้ากับขั้วลบ -

LED มีขั้วสองด้าน ต้องทำการต่อขั้วให้ถูกต้องจึงจะเปล่งแสงได้



2. ตัวต้านทานไม่มีขั้ว ดังนั้นคุณสามารถต่อขาпинทิศทางไหนก็ได้

3. แถบสีบนตัวต้านทานเป็นตัวบอกค่าความต้านทาน คุณสามารถอ้างอิงจากแผงผังเพื่อดูวิธีอ่านค่าความต้านทานได้



โครงสร้างการเขียนโค้ด

คำสั่ง while loop จะวนทำงานต่อเนื่องไม่สิ้นสุดจนกว่าตัวแปรในวงเล็บ() จะเปลี่ยนค่าเป็น false.

```
while (เงื่อนไข)
{
    // สั่งให้ทำบางอย่าง
}
```

ตัวอย่าง

1. สั่งให้ทำ 200 รอบ.

```
var = 0;
while (var < 200)
{
    // สั่งให้ทำบางอย่าง
    var++;
}
```

2. สั่งให้ทำแบบไม่สิ้นสุด.

```
while (1)
{
    // สั่งให้ทำบางอย่าง
}
```





โครงการ 7: จางแสง LED

1. ให้ใช้วงจรเหมือนโครงการ 6 ทำการเขียนและอัปโหลดโค้ดดังต่อไปนี้ไปที่บอร์ดของคุณ.

```
Project_7

void setup() {
  pinMode(5, OUTPUT);
}

void loop() {
  analogWrite(5, 60);
  delay (200);
  analogWrite(5, 50);
  delay (200);
  analogWrite(5, 40);
  delay (200);
  analogWrite(5, 30);
  delay (200);
  analogWrite(5, 20);
  delay (200);
  analogWrite(5, 10);
  delay (200);
  analogWrite(5, 0);
  delay (1000);
}
```


2. ตรวจสอบผลลัพธ์

คลิกที่นี่หรือสแกน
QR code เพื่อดูวิดีโอสาธิต



คุณ将会เห็นว่าแสง LED สว่างขึ้นและจะค่อยๆ
หรือลงจนกระทั่งดับไป วัฏจักรแบบนี้จะวนซ้ำไป
เรื่อยๆ จนกว่าคุณจะหยุดจ่ายไฟเลี้ยง



ทำงานอย่างไร

| Project_7 | |
|--|--|
| <pre>void setup() { pinMode(5, OUTPUT); }</pre> | กำหนด Pin 5 (ต่อกับ LED ภายนอก) ให้เป็น สัญญาณขาออก |
| <pre>void loop() { analogWrite(5, 60); delay(200);</pre> | เปล่งแสง LED ที่ความสว่าง 60 นาน 200ms |
| <pre> analogWrite(5, 50); delay(200);</pre> | เปล่งแสง LED ที่ความสว่าง 50 นาน 200ms |
| <pre> analogWrite(5, 40); delay(200);</pre> | เปล่งแสง LED ที่ความสว่าง 40 นาน 200ms |
| <pre> analogWrite(5, 30); delay(200);</pre> | เปล่งแสง LED ที่ความสว่าง 30 นาน 200ms |
| <pre> analogWrite(5, 20); delay(200);</pre> | เปล่งแสง LED ที่ความสว่าง 20 นาน 200ms |
| <pre> analogWrite(5, 10); delay(200);</pre> | เปล่งแสง LED ที่ความสว่าง 10 นาน 200ms |
| <pre> analogWrite(5, 0); delay(1000); }</pre> | หยุดเปล่งแสง LED (ความสว่างเป็นศูนย์) นาน 1s |

3. มีวิธีที่ง่ายขึ้นที่ให้ได้ผลลัพธ์เดียวกัน ให้ทำการเปลี่ยนโค้ดตามตัวอย่างและอัปโหลดไปที่บอร์ดของคุณ

```
Project_7a

int brightness = 60
void setup() {
  pinMode(5, OUTPUT);
}

void loop() {
  analogWrite(5, brightness);
  delay (200);
  if (brightness == 0) {
    delay (1000);
    brightness = 60;
  }
  else {
    brightness = brightness-10;
  }
}
```

3. ตรวจสอบผลลัพธ์

คลิกที่นี่หรือสแกน
QR code เพื่อดูวิดีโอ
สาธิต.



คุณได้ผลลัพธ์เหมือนโครงงานก่อนหน้านี้ไหม?





ทำงานอย่างไร

| | |
|---|--|
| Project_7a | |
| <code>int brightness = 60</code> | กำหนดตัวแปร (brightness) และกำหนดค่าเริ่มต้นเป็นค่า 60 |
| <code>void setup() { pinMode(5, OUTPUT); }</code> | ตั้งค่า Pin 5 (LED ภายนอก) ให้เป็นสัญญาณขาออก |
| <code>void loop() { analogWrite(5, brightness); delay (200); }</code> | เปล่งแสง LED ด้วยค่าปัจจุบันที่กำหนดให้กับตัวแปร 'brightness' และหน่วงเวลา 200ms |
| <code>if (brightness == 0) { delay (1000); brightness = 60; }</code> | ตรวจสอบว่าถ้าตัวแปร 'brightness' มีค่าเท่ากับ 0 (LED หยุดทำงาน) ให้หน่วงเวลา 1000ms จากนั้นกำหนดค่าใหม่เป็น 60 ให้กับตัวแปร 'brightness' |
| <code>else { brightness = brightness-10; }</code> | หรือถ้าค่า brightness ไม่เท่ากับ 0 ให้อัปเดตตัวแปร 'brightness' ด้วยค่าปัจจุบันลบด้วย 10 (หรือความสว่างของ LED ลง) |



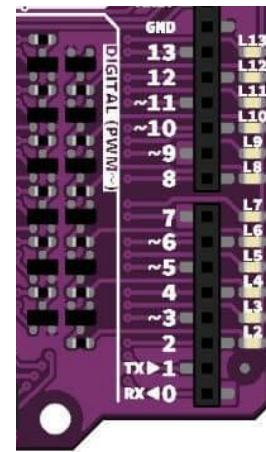
เรื่องที่ต้องรู้

ฟังก์ชัน ARDUINO

- ข้อมูลส่งออกมี 2 แบบ คือ ดิจิทัลและอนาล็อก ข้อมูลส่งออกแบบดิจิทัลมีค่าเพียงสองค่าคือ **0 (LOW)** หรือ **1 (HIGH)** ส่วนค่าข้อมูลส่งออกแบบอนาล็อกจะมีค่าเป็นช่วงตั้งแต่ 0 ถึง 255 คุณสามารถใช้ฟังก์ชันต่อไปนี้ในการควบคุมข้อมูลส่งออกแบบอนาล็อก

`analogWrite (pin, value);`

2. อย่างไรก็ตามไม่ใช่ทุกขาพิน บนบอร์ดที่สามารถใช้เป็นข้อมูลส่งออกแบบอนาล็อกได้ มีเพียงแค่ขา 3, 5, 6, 9, 10 และ 11 เท่านั้น (ระบุด้วยเครื่องหมาย ~) ที่นำไปใช้งานเป็นข้อมูลส่งออกแบบอนาล็อกได้



โครงสร้างการเขียนโค้ด

1. ความสวยงามของการเขียนโค้ดคือการที่โปรแกรมเมอร์ทำโค้ดให้ง่ายจากชุดคำสั่งที่ยาวมากๆ ด้วยโค้ดเพียงไม่กี่บรรทัดที่สามารถสั่งให้ทำงานได้เหมือนกัน
1. การลดจำนวนบรรทัดลงยังช่วยลดระยะเวลาประมวลผล ดังนั้นเป็นเรื่องสำคัญมากที่ต้องคอยเพิ่มประสิทธิภาพของโค้ดของคุณเสมอ
1. ในโครงงานนี้ เราต้องการลดความสว่างของ LED ลงทุกๆ 200ms ดังนั้นเราจึงกำหนดตัวแปร brightness เป็นจำนวนเต็มในตอนเริ่มต้นของโปรแกรม "brightness" เป็นเพียงแค่ชื่อตัวแปรดังนั้นคุณสามารถเลือกใช้ชื่ออื่นได้ตามต้องการ
1. คำสั่งต่อไปนี้เป็นคำสั่งเปรียบเทียบที่คุณสามารถใช้ในโค้ดของคุณได้

$x == y$ (x เท่ากับ y)
 $x != y$ (x ไม่เท่ากับ y)
 $x < y$ (x น้อยกว่า y)
 $x > y$ (x มากกว่า y)
 $x <= y$ (x น้อยกว่าหรือเท่ากับ y)
 $x >= y$ (x มากกว่าหรือเท่ากับ y)





ภารกิจท้าทาย

ภารกิจ: ให้ลดความสว่างของแสง LED 5 เมื่อสวิตช์บนบอร์ดถูกกดเท่านั้น

คลิกที่นี่หรือสแกน
QR code เพื่อดูวิดีโอ
สาธิต



ขอแสดงความยินดี! คุณได้จบบทเรียนที่ 3 และได้เรียนรู้สิ่งต่อไปนี้:

1. การสร้างวงจร LED พื้นฐาน
2. การใช้คำสั่ง while loop
3. ความแตกต่างของข้อมูลส่งออกแบบดิจิทัลและข้อมูลส่งออกแบบอนาล็อก
4. ความสำคัญของการลดจำนวนบรรทัดของโค้ด



บทเรียนที่ 4

เสียงเพลง
(MELODY TONE)



Project 8: แต่งเพลงพื้นฐาน

1. เปิดโปรแกรม Arduino แล้วเขียนโปรแกรมด้านล่างนี้พร้อมอัปโหลดไปยังบอร์ด Maker Uno X

```
void setup() {  
  pinMode(8, OUTPUT);  
}  
  
void loop() {  
  tone (8, 262, 250);  
  delay (325);  
  tone (8, 294, 250);  
  delay (325);  
  tone (8, 330, 250);  
  delay (325);  
  tone (8, 349, 250);  
  delay (325);  
  tone (8, 392, 250);  
  delay (325);  
  tone (8, 440, 250);  
  delay (325);  
  tone (8, 494, 250);  
  delay (325);  
  tone (8, 523, 250);  
  delay (1000);  
}
```

2. ตรวจสอบผลลัพธ์

คลิกที่นี่ หรือ สแกน QR code เพื่อดูวิดีโอตัวอย่าง



ลำโพงบัสเซอร์ จะเล่นเสียง “โด เร มี ฟา ซอล ลา ที โด” วนซ้ำๆ.



How it works

```
Project_8

void setup() {
  pinMode(8, OUTPUT);
}

void loop() {
  tone (8, 262, 250);
  delay (325);
  tone (8, 294, 250);
  delay (325);
  tone (8, 330, 250);
  delay (325);
  tone (8, 349, 250);
  delay (325);
  tone (8, 392, 250);
  delay (325);
  tone (8, 440, 250);
  delay (325);
  tone (8, 494, 250);
  delay (325);
  tone (8, 523, 250);
  delay (1000);
}
```

ตั้งค่าขา 8 (เชื่อมต่ออยู่กับลำโพงบัสเซอร์บนบอร์ด) เป็นเอาต์พุต (output)

เล่นเสียง “โด” ความถี่ 262 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)

เล่นเสียง “เร” ความถี่ 294 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)

เล่นเสียง “มี” ความถี่ 330 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)

เล่นเสียง “ฟา” ความถี่ 392 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)

เล่นเสียง “ซอล” ความถี่ 392 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)

เล่นเสียง “ลา” ความถี่ 440 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)

เล่นเสียง “ที” ความถี่ 494 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)

เล่นเสียง “โดสูง” ความถี่ 523 เฮิรต(Hz) เป็นเวลา 250 มิลลิวินาที(ms)



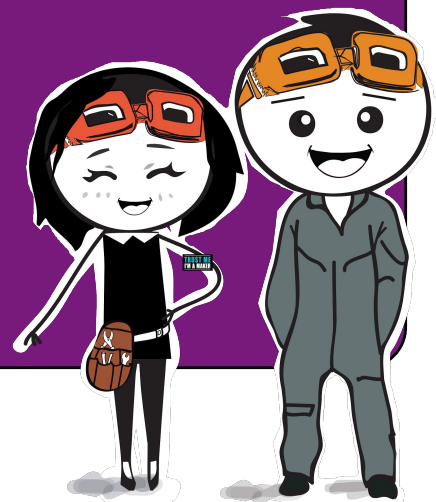
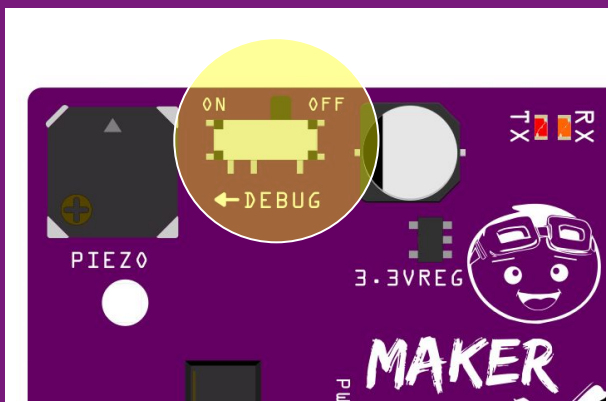
Good To Know

ชุดคำสั่ง ARDUINO

1. การเล่นเสียงของลำโพงบัซเซอร์
tone (pin, frequency, duration);
pin คือขาสัญญาณบนบอร์ดที่เชื่อมต่อกับลำโพงบัซเซอร์
frequency คือความถี่ของโน้ตมีหน่วยเป็นเฮิรต (Hz)
duration คือระยะเวลาของการเล่นเสียงมีหน่วยเป็นมิลลิวินาที (ms)
2. เพื่อให้สามารถแยกความแตกต่างของโน้ตต่างๆในทำนองเพลง เราต้องทำการเพิ่มการหน่วงเวลาระหว่างโน้ต การหน่วงเวลาต้องลากยาวกว่าระยะเวลาที่เล่นโน้ต 30% เพื่อให้โน้ตเล่นออกไปอย่างเต็มที่ ตัวอย่าง เช่น ถ้าโน้ตมีความยาว 250 มิลลิวินาที(ms) เราจะเพิ่มการหน่วงเวลา 325 มิลลิวินาที(ms) (250ms + 75ms)

คุณสมบัติของ MAKER UNO X'S

1. มีลำโพงบัซเซอร์บนบอร์ดเชื่อมต่ออยู่ที่ขา 8 คุณต้องทำการเปิดการใช้งานโดยดันสวิตช์ DEBUG ไปที่ ON
2. ถ้าคุณต้องการเชื่อมต่อขา 8 กับอุปกรณ์อินพุต/เอาต์พุต (I/O) อื่นๆ อย่าลืมว่าจะต้องปิดการใช้งานลำโพงบัซเซอร์บนบอร์ด โดยดันสวิตช์ DEBUG ไปที่ OFF

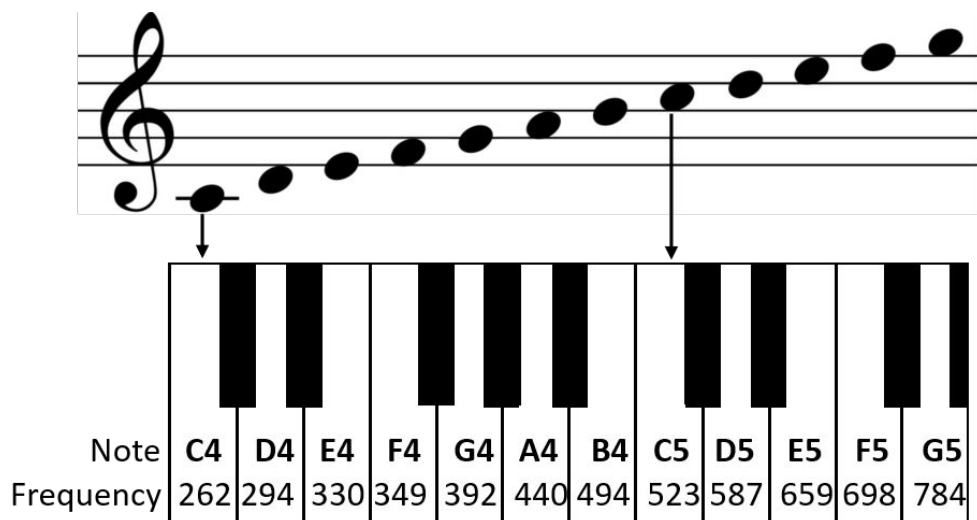




Good To Know

การอ่านโน้ตเพลง

- ตำแหน่งของโน้ตเพลงบนบรรทัด 5 เส้น (the five horizontal lines) คือ การกำหนดเสียง โน้ตที่อยู่ตำแหน่งที่สูงกว่า ก็จะมีค่าที่สูงกว่าโน้ตที่อยู่ต่ำกว่า คุณสามารถดูที่ลิงค์ต่อไปเพื่อดูความถี่ทั้งหมด 88 เสียงที่มีอยู่บนเปียโน <https://www.arduino.cc/en/Tutorial/ToneMelody>



- มีการใช้สัญลักษณ์ทางดนตรีที่แตกต่างกันเพื่อบอกระยะเวลา (ความยาวของเสียง) ในไลบรารี tone ของ Arduino ระยะเวลาในการเล่น 1/4 โน้ต (1 จังหวะ) คือ 250มิลลิวินาที(ms) ในกรณีที่กำหนด 1 โน้ตเป็นเวลา 1 วินาที

| ตัวโน้ต | | | | | |
|---------------|------------|------------|-----------|--------------|--------------|
| ตัวหยุด | | | | | |
| ความยาว | โน้ตตัวกลม | โน้ตตัวขาว | โน้ตตัวดำ | เขบ็ต 1 ชั้น | เขบ็ต 2 ชั้น |
| จังหวะ | 4 | 2 | 1 | 1/2 | 1/4 |
| ระยะเวลา (ms) | 1000 | 500 | 250 | 125 | 63 |



Project 9: แต่งเพลง "Happy Birthday"

ลองเขียนท่อนแรกของเพลง Happy Birthday กันเลย



| โน้ต | Hap- ซอล | -py ซอล | Birth- ลา | -day ซอล | To โดสูง | You ที |
|---------------|-------------|------------|--------------|-------------|-------------|-----------|
| ความถี่ (Hz) | 392 | 392 | 440 | 392 | 523 | 494 |
| ระยะเวลา (ms) | 125 | 125 | 250 | 250 | 250 | 500 |

1. แก้วไขชอร์สโค้ดก่อนหน้าของคุณตามด้านล่างนี้และอัปโหลดไปที่บอร์ด

```
Project_9

void setup() {
  pinMode(8, OUTPUT);
  tone (8, 392, 125);
  delay (163);
  tone (8, 392, 125);
  delay (163);
  tone (8, 440, 250);
  delay (325);
  tone (8, 392, 250);
  delay (325);
  tone (8, 523, 250);
  delay (325);
  tone (8, 494, 500);
  delay (650);
}
```

2. ตรวจสอบผลลัพธ์

คลิกที่นี่ หรือ สแกน QR code เพื่อดูวิดีโอตัวอย่าง



ตอนนี้ลำโพงบัสเซอร์เล่นท่อนแรกของเพลง "Happy Birthday" หนึ่งรอบ.



How it works

Project_9

```
void setup() {  
  pinMode(8, OUTPUT);  
  
  tone (8, 392, 125);  
  delay (163);  
  
  tone (8, 392, 125);  
  delay (163);  
  
  tone (8, 440, 250);  
  delay (325);  
  
  tone (8, 392, 250);  
  delay (325);  
  
  tone (8, 523, 250);  
  delay (325);  
  
  tone (8, 494, 500);  
  delay (650);  
}
```

ตั้งค่าขา 8 (เชื่อมต่ออยู่กับลำโพงบนบอร์ด) เป็นเอาต์พุต (output)

เล่นเสียง 'ซอล' ความถี่ 392เฮิรต(Hz) เป็นเวลา 125มิลลิวินาที (ms)

เล่นเสียง 'ซอล' ความถี่ 392เฮิรต(Hz) เป็นเวลา 125มิลลิวินาที (ms)

เล่นเสียง 'ลา' ความถี่ 440เฮิรต(Hz) เป็นเวลา 250มิลลิวินาที (ms)

เล่นเสียง 'ซอล' ความถี่ 392เฮิรต(Hz) เป็นเวลา 250มิลลิวินาที (ms)

เล่นเสียง 'โด' ความถี่ 523เฮิรต(Hz) เป็นเวลา 250มิลลิวินาที(ms)

เล่นเสียง 'ที' ความถี่ 494เฮิรต(Hz) เป็นเวลา 500มิลลิวินาที(ms)

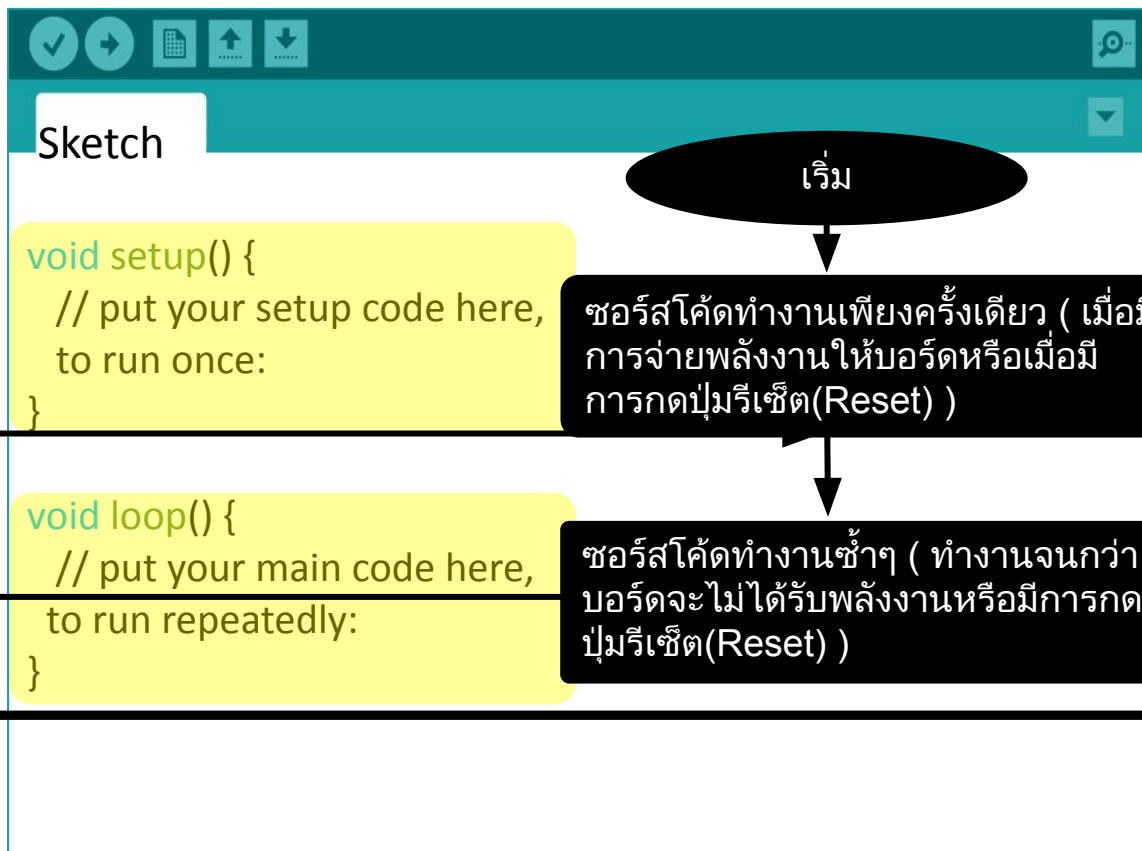


Good To Know

ไวยากรณ์ของซอร์สโค้ด

1. คุณสังเกตไหมว่าทำนองเพลงใน Project_9 เล่นเพียงครั้งเดียวและไม่ซ้ำเหมือนในซอร์สโค้ดก่อนหน้านี้

นี่เป็นเพราะซอร์สโค้ดได้ถูกเขียนอยู่ภายใน ฟังก์ชัน `setup()` แทนที่จะเขียนอยู่ในฟังก์ชัน `loop()` กล่าวอีกนัยหนึ่งซอร์สโค้ดจะเริ่มทำงานบรรทัดต่อบรรทัดเมื่อจ่ายพลังงานให้กับบอร์ด Maker UNO X จะไม่ทำงานซ้ำอีก เว้นแต่ว่าจะมีการกดปุ่มรีเซ็ต(Reset)

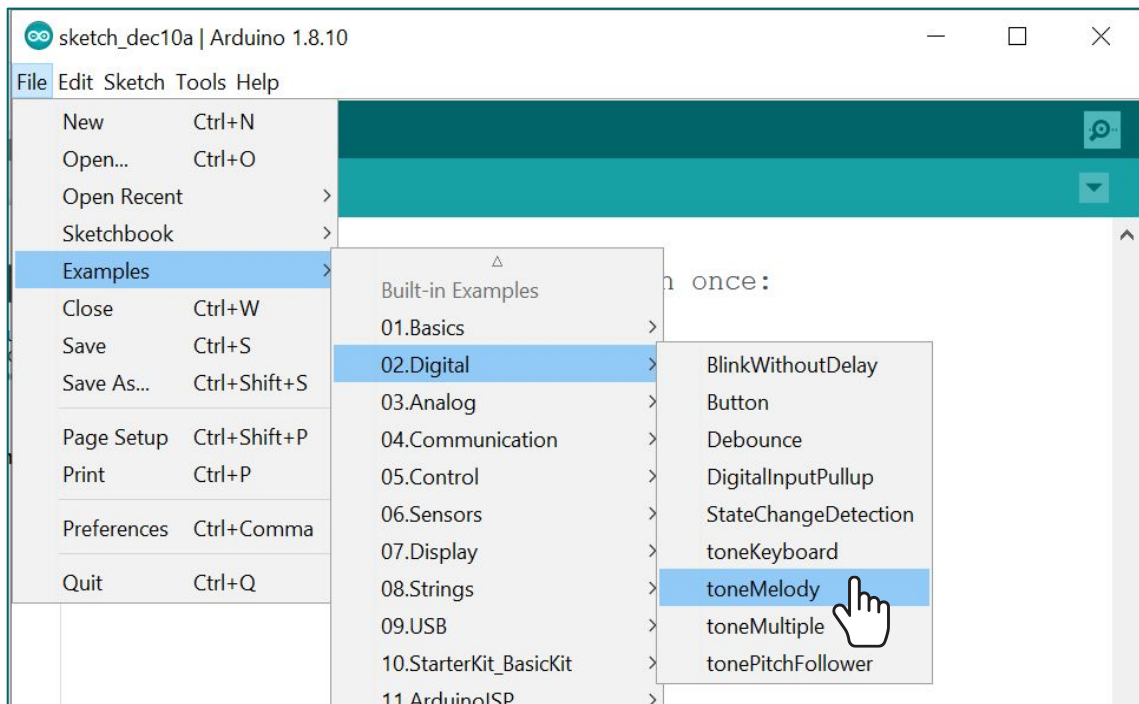




Project 10: ปรับแต่งซอร์สโค้ดของคุณ

แทนที่จะป้อนค่าสิ่งของโน้ต และ ระยะเวลาในการเล่นเสียงทีละบรรทัด เราสามารถเพิ่มประสิทธิภาพซอร์สโค้ดของเราได้โดยใช้ซอร์สโค้ดตัวอย่าง ซึ่งจะช่วยให้เราแต่งเพลงยาวๆ ได้ง่ายขึ้น

1. เปิดซอร์สโค้ดตัวอย่างไปที่เมนู
File > Examples > 02 Digital > toneMelody



ในตัวอย่างที่ได้ ทุกๆค่าสิ่งของโน้ตทั้งหมด 88 ตัวโน้ต จะถูกกำหนดไว้ล่วงหน้าเริ่มตั้งแต่ B0 ถึง DS8 คุณสามารถคลิกที่แท็บ **pitches.h** เพื่อดูค่าสิ่งของโน้ตที่กำหนดไว้



```
toneMelody pitches.h

#include "pitches.h"

// โน้ตในทำนองเพลง
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3,
  NOTE_C4
};

// ความยาวโน้ต 4 = โน้ตตัวดำ, 8 = เขบ็ตหนึ่งชั้น เป็นต้น
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

void setup() {
  // วนรอบเพื่อเล่นเสียง
  for (int thisNote = 0; thisNote < 8; thisNote++) {

    // คำนวณความยาวโน้ตที่จะเล่น โดยใช้ความยาว 1 ตัวโน้ตเท่ากับ
    1 วินาทีหารด้วยประเภทของความยาวตัวโน้ต
    // ตัวอย่าง โน้ตตัวดำเท่ากับ 1000 / 4, เขบ็ตหนึ่งชั้นเท่ากับ 1000/8,
    เป็นต้น
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);

    // เพื่อให้การเล่นเสียงชัดเจน เราจะหน่วงเวลาเล็กน้อยระหว่างโน้ต
    // หน่วงเวลาเพิ่ม 30% จากความยาวเสียงที่กำหนดจะทำให้การเล่น
    นเสียงออกมาดีที่สุด:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    // หยุดการเล่นเสียง
    noTone(8);
  }
}

void loop() {
  // no need to repeat the melody.
}
```

ซอร์สโค้ดตัวอย่างนี้สร้างโดยคุณ Tom Igoe
อัปโหลดซอร์สโค้ดไปยังบอร์ด Maker UNO X
แล้วคุณจะได้ยินเสียงเพลง "Shave and a
Haircut, Two Bits".



2. แก้ไขซอร์สโค้ดตัวอย่างเพื่อเล่นท่อนแรกของเพลง “Happy Birthday” ใส่โน้ตที่เล่นไว้ในตัวแปรอาร์เรย์ melody พร้อมทั้งใส่ความยาวเสียงตามจำนวนโน้ตที่จะเล่นทั้งหมดตามซอร์สโค้ดด้านล่าง ดังนี้

| โน้ต | Hap- ซอล | -py ซอล | Birch- ลา | -day ซอล | To โดสูง | You ที |
|----------|-------------|------------|--------------|-------------|-------------|-----------|
| ระยะเวลา | 8 | 8 | 4 | 4 | 4 | 2 |

```

Project_10 pitches.h

#include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_G4, NOTE_G4, NOTE_A4, NOTE_G4, NOTE_C5, NOTE_B4
};

// ความยาวโน้ต 4 = โน้ตตัวดำ, 8 = เข้บิตหนึ่งชั้น เป็นต้น
int noteDurations[] = {
  8, 8, 4, 4, 4, 4, 2
};

void setup() {
  // วงรอบเพื่อนเล่นเสียง
  for (int thisNote = 0; thisNote < 6; thisNote++) {

```

2. คอลไฟล์ อัปโหลดซอร์สโค้ด และตรวจสอบผลลัพธ์



เหมือนกับ Project 9 ถ้าโฟกัสเซอร์เล่นทำนองเพลง “Happy Birthday” หนึ่งรอบ





How it works

| toneMelody | pitches.h |
|---|---|
| <pre>#include "pitches.h"</pre> | นำเข้าค่าเสียงทั้งหมดตามที่กำหนดไว้ใน "pitches.h" |
| <pre>int melody[] = { NOTE_G4, NOTE_G4, NOTE_A4, NOTE_G4, NOTE_C5, NOTE_B4 };</pre> | กำหนดตัวแปรประเภทอาเรียชื่อ melody[] ภายในเก็บโน้ตที่จะเล่นเรียงตามลำดับ |
| <pre>int noteDurations[] = { 8, 8, 4, 4, 4, 4, 2 };</pre> | กำหนดตัวแปรประเภทอาเรียชื่อ noteDurations[] ภายในเก็บค่าประเภทของความยาวโน้ตที่จะเล่นเรียงลำดับตามโน้ตที่กำหนดไว้ข้างต้น |
| <pre>void setup() { for (int thisNote = 0; thisNote < 6; thisNote++) {</pre> | ใช้การวนรอบแบบ for เพื่อเล่นโน้ตตามลำดับสำหรับเลข 6 ในที่นี้หมายถึงจำนวนโน้ตที่จะเล่น |
| <pre> int noteDuration = 1000 / noteDurations[thisNote]; tone(8, melody[thisNote], noteDuration); int pauseBetweenNotes = noteDuration * 1.30; delay(pauseBetweenNotes);</pre> | เช่นเดียวกับซอร์สโค้ดใน Project_9 นี่เป็นการบอกให้โปรแกรมเล่นแต่ละเสียงแล้วค้างไว้นานขึ้น 30% ของระยะเวลาที่เล่นเสียง |
| <pre> noTone(8); } }</pre> | หยุดเล่นเสียงทั้งหมด |



ไวยากรณ์ของซอร์สโค้ด

1. ใช้คำสั่ง “for”
for (initialization; condition; increment)
{
 //คำสั่ง
}

ตัวอย่าง:

```
void loop() {  
  digitalWrite(2,HIGH);  
  delay (1000);  
  digitalWrite(3,HIGH);  
  delay (1000);  
  digitalWrite(4,HIGH);  
  delay (1000);  
  digitalWrite(5,HIGH);  
  delay (1000);  
  digitalWrite(6,HIGH);  
  delay (1000);  
  digitalWrite(7,HIGH);  
  delay (1000);  
}
```

ซอร์สโค้ดจาก Project_3 สามารถเขียนให้สั่งลงเหลือเพียง 3 บรรทัดโดยใช้คำสั่ง **for** ตามที่แสดงด้านล่างนี้

```
for (int i=2; i<8; i++)  
{  
  digitalWrite (i , HIGH);  
  delay (1000);  
}
```

2. เมื่อคุณมีตัวแปรหลายตัวที่เป็นชนิดเดียวกัน คุณสามารถใช้ อาเรย์ และกำหนดค่าให้ตัวแปรทั้งหมดได้ในบรรทัดเดียว
วิธีการใช้ “อาเรย์”

type arrayName [] = { ค่าในตัวแปร }
type: ชนิดของตัวแปร เช่น int, char ,
arrayName = ชื่อขอตัวแปรอาเรย์

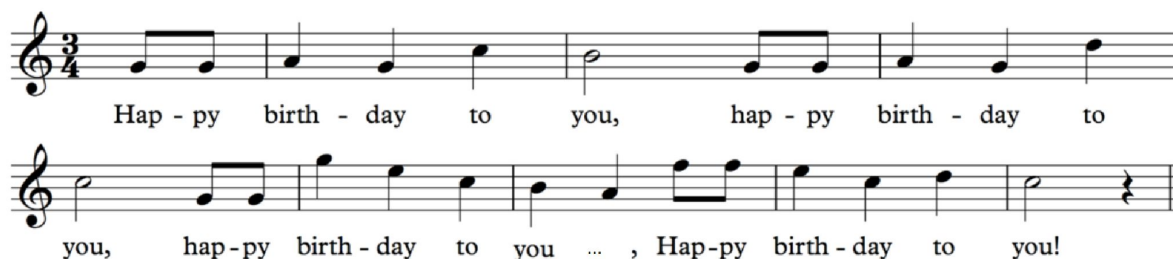
ตัวอย่าง

```
int melody[ ] = { NOTE_C4, NOTE_D4, NOTE_E4 };
```



Challenge

คำสั่ง: ป้อนคำสั่งให้บอร์ด Maker UNO X ของคุณเล่นเพลง “Happy Birthday”
เมื่อกดสวิตช์บนบอร์ด



คลิกที่นี่หรือสแกน
QR code เพื่อดูวิดีโอตัวอย่าง



ขอแสดงความยินดี! คุณเรียนจบบทที่ 4 และเรียนรู้สิ่งต่อไปนี้แล้ว

1. วิธีป้อนคำสั่งให้บอร์ด Maker UNO X เล่นเพลง
2. วิธีแต่งเพลงเบื้องต้นโดยใช้บอร์ด Maker UNO X
3. วิธีโหลดโปรแกรมตัวอย่างจากไลบรารี Arduino
4. วิธีใช้คำสั่ง for



บทที่ 5

การอ่านข้อมูลแบบอนาล็อก (ANALOG INPUT)



ชิ้นงานที่ 11 : การแสดงคลื่นสัญญาณบนหน้าจอแสดงผล

1. จัดเตรียมอุปกรณ์ต่างๆ ดังต่อไปนี้

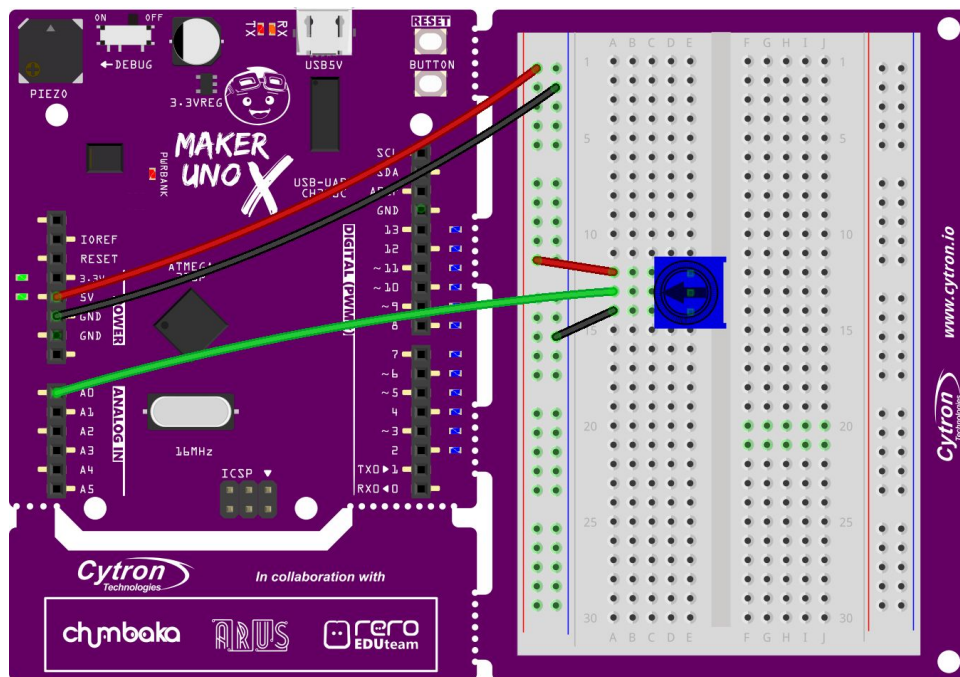


สายไฟจัมเปอร์ชนิดผู้-ผู้



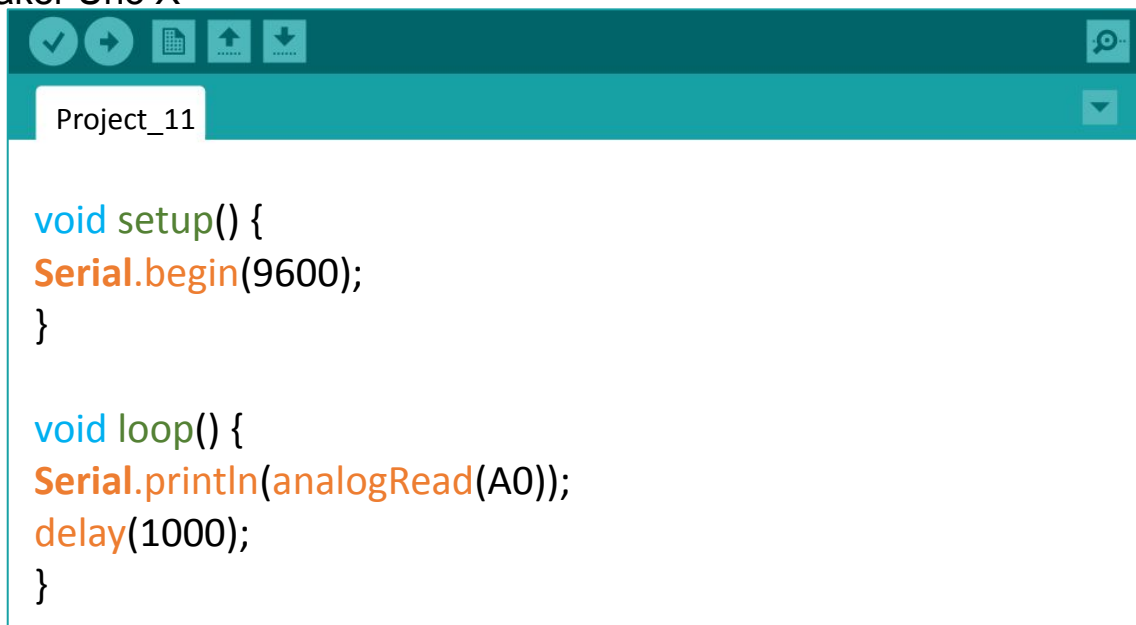
ตัวต้านทานปรับค่าได้

2. ติดตั้งวงจร Arduino ดังรูปด้านล่าง



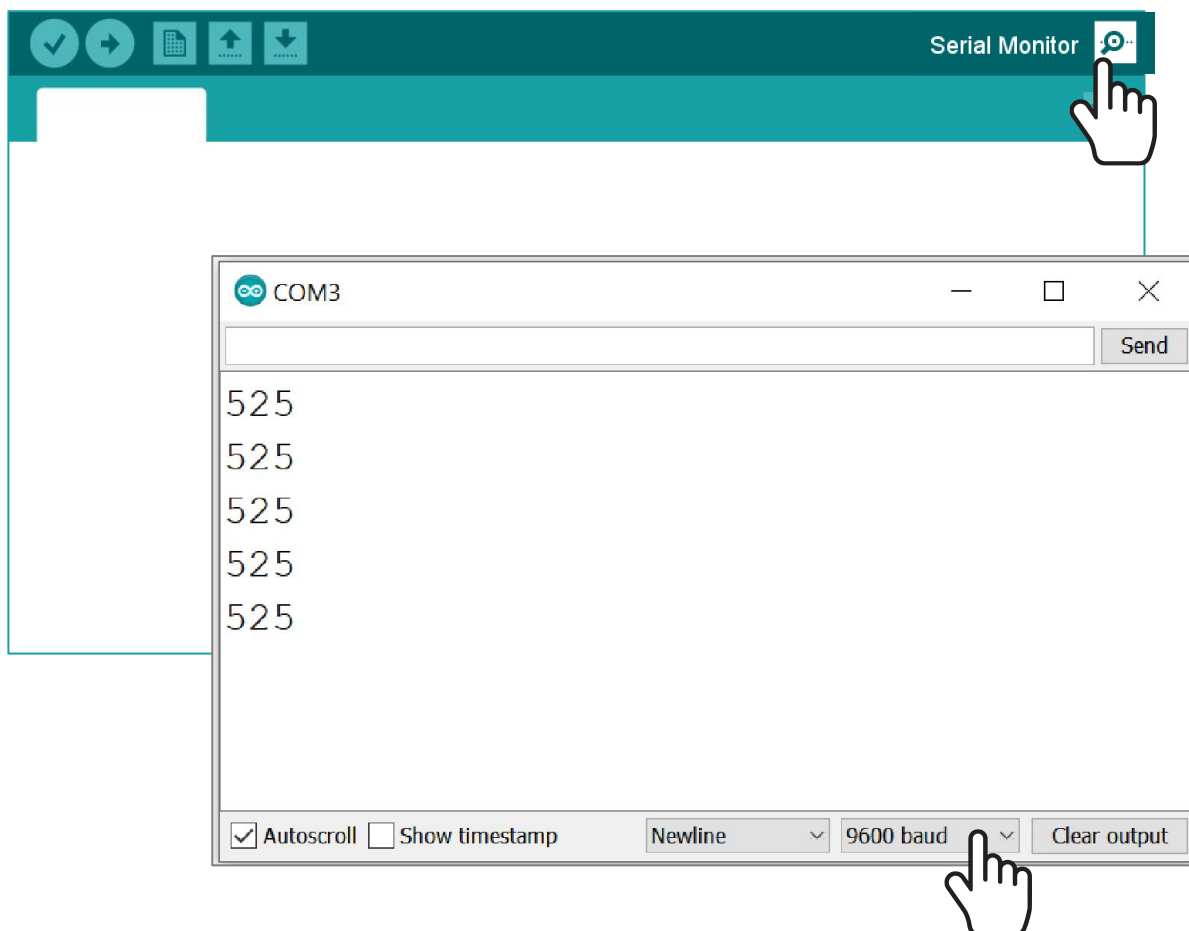
fritzing

3. เปิดโปรแกรม Arduino แล้วเขียนโปรแกรมด้านล่างนี้พร้อมอัปโหลดไปยังบอร์ด Maker Uno X

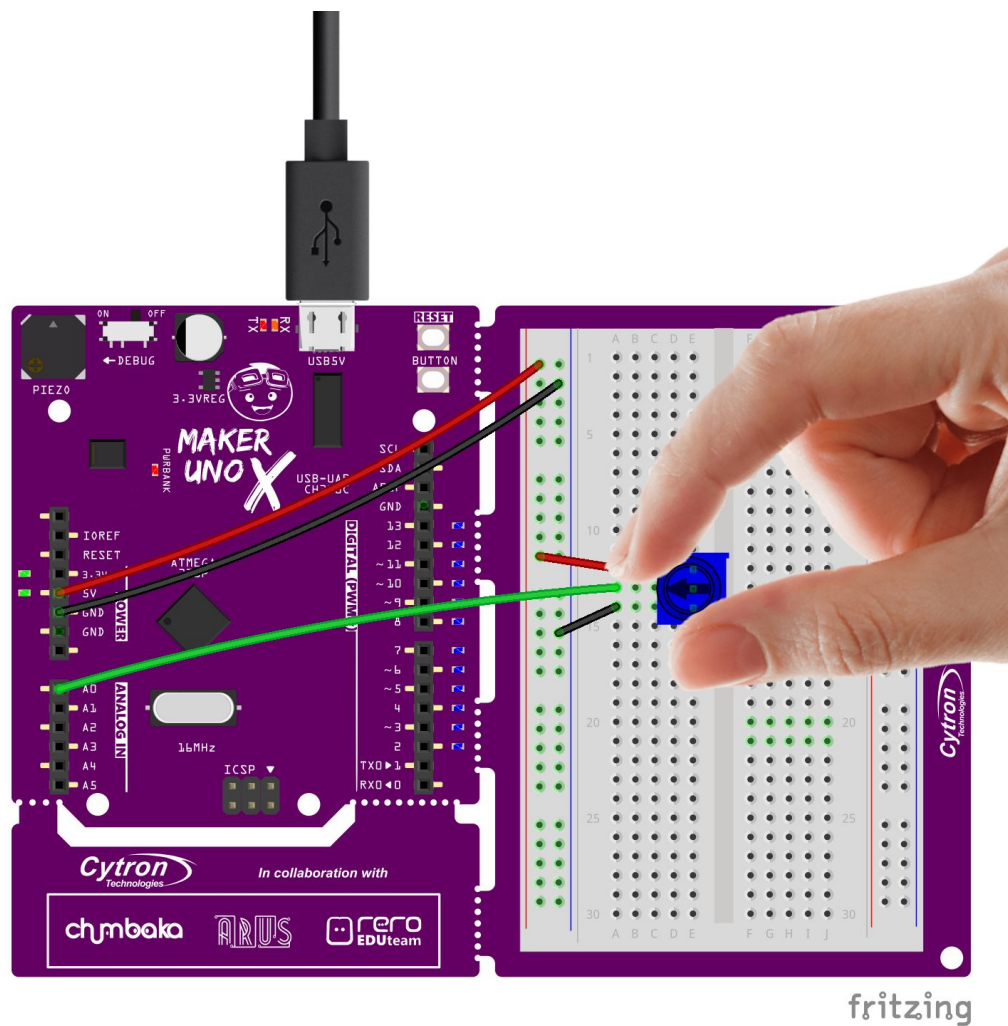


```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println(analogRead(A0));  
  delay(1000);  
}
```

4. เมื่ออัปโหลดเสร็จ ให้ทำการคลิกที่ปุ่ม Serial Monitor ที่ด้านบนของแถบเครื่องมือ หลังจากนั้นจะปรากฏหน้าต่างใหม่ขึ้นมาให้เลือกอัตราการรับส่งข้อมูลที่ 9600 baud



5. สังเกตการแสดงค่าบนจอแสดงผลในขณะที่หมุนโพเทนชิโอมิเตอร์หรือตัวต้านทานปรับค่าได้ (Potentiometer) ตามเข็มนาฬิกาและทวนเข็มนาฬิกา



6. ตรวจสอบผลลัพธ์



การแสดงค่าบนจอภาพแสดงผลจะปรากฏการอ่านค่าแบบอนาล็อกที่โพเทนชิโอมิเตอร์อ่านค่าได้





How it works

....อธิบายกระบวนการทำงาน....

| Project_11 | |
|--|--|
| <pre>void setup() { Serial.begin(9600); }</pre> | ตั้งค่าการเชื่อมต่อที่สัญญาณ 9600 |
| <pre>void loop() { Serial.println(analogRead(A0)); delay(1000); }</pre> | อ่านค่าในช่อง A0 และแสดงค่าออกมาในหน้าต่างแสดงผล รอประมาณ 1 วินาที. |



Good To Know

...เกร็ดน่ารู้...

ชุดคำสั่ง ARDUINO

1. เฉพาะช่อง A0,A1,A2,A3,A4 และ A5 จะมีฟังก์ชันข้อมูลขาเข้าแบบอนาล็อก (Analog input) ด้วยเหตุนี้เราจำเป็นต้องเชื่อมอนาล็อกเซ็นเซอร์ไปยังช่องต่างๆ ดังกล่าวหากเราต้องการรับค่าของข้อมูลขาเข้าที่อนาล็อกเซ็นเซอร์อ่านค่าได้
2. ข้อมูลขาเข้าแบบดิจิตอล (Digital input) มีเพียง 2 ค่า นั่นคือ 0 (LOW) หรือ 1 (HIGH) และค่าข้อมูลขาเข้าแบบดิจิตอล (Digital input) จะเริ่มนับจาก 0 ไปถึง 1023.

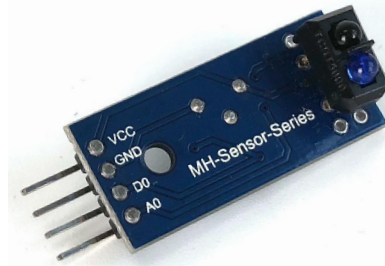


ชิ้นงานที่12: การอ่านค่าของเซ็นเซอร์ IR แบบอนาล็อก

1. จัดเตรียมอุปกรณ์ต่างๆ ต่อไปนี้.

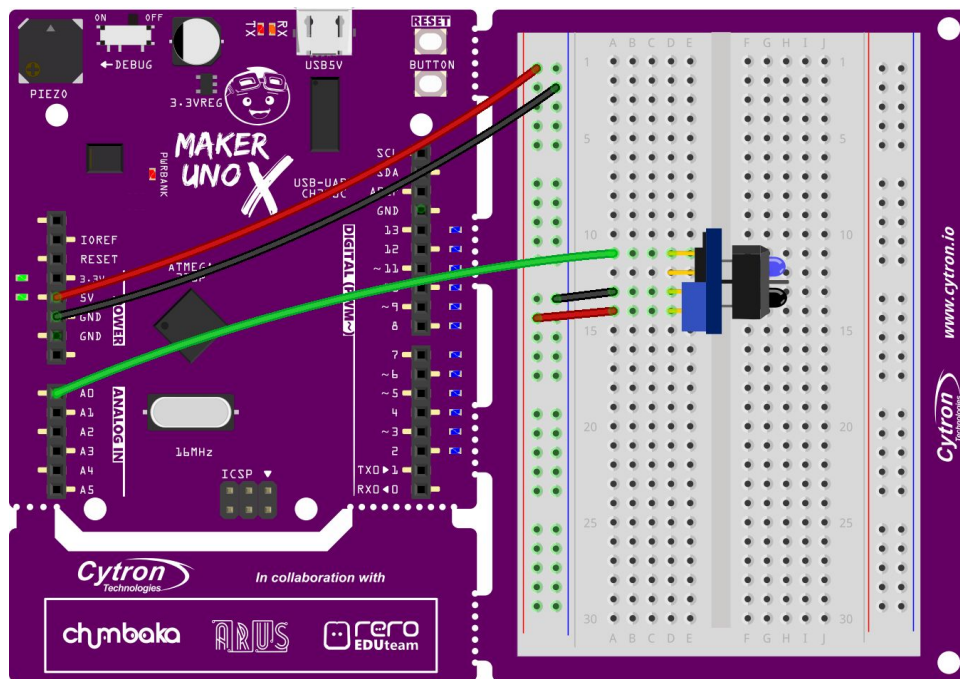


สายไฟจัมเปอร์ชนิดผู้-ผู้



IR Sensor

1. ติดตั้งวงจร Arduino ดังรูปด้านล่าง.



fritzing

| | | | | |
|-------------|-----|-----|----|----|
| IR Sensor | VCC | GND | D0 | A0 |
| Maker UNO X | 5V | GND | -- | A0 |

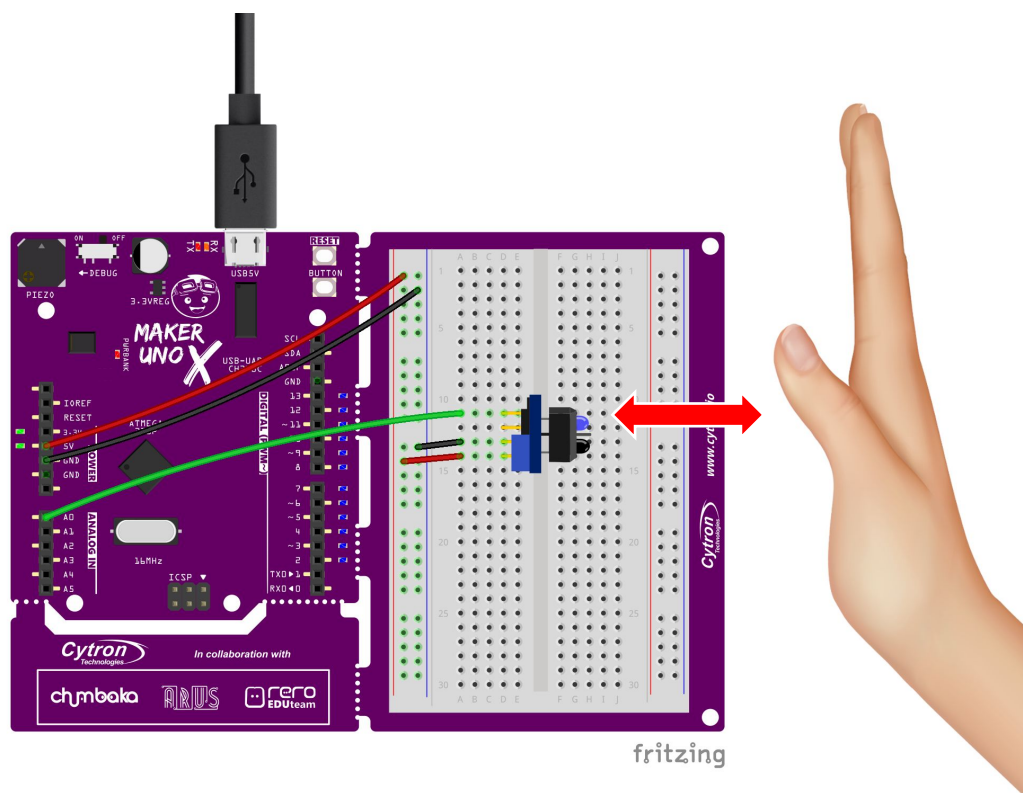
3. อัปโหลดโปรแกรมเดียวกันที่ใช้ในชิ้นงานที่ 11 บนบอร์ด Maker Uno X.

```
Project_12

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(A0));
  delay(1000);
}
```

4. เมื่ออัปโหลดเสร็จสิ้น, ให้คลิกที่ serial monitor. หลังจากนั้นวางมือหน้าเซ็นเซอร์. ให้สังเกตการแสดงผลค่าบนจอแสดงผลในขณะที่เคลื่อนฝ่ามือเข้าไปยัง IR เซ็นเซอร์. หลังจากนั้นค่อยๆ เคลื่อนฝ่ามือออกจาก IR เซ็นเซอร์



5. ตรวจสอบผลลัพธ์.

กดหรือสแกน QR code เพื่อดูวิดีโอผลลัพธ์.



คุณสังเกตเห็นอะไรบ้าง? เซ็นเซอร์จะอ่านค่าน้อยหรือมากเมื่อคุณเคลื่อนมือไปยังเซ็นเซอร์? แล้วค่าอะไรที่เซ็นเซอร์อ่านค่าได้เมื่อคุณนำมือห่างออกจากเซ็นเซอร์ (ไม่มีสิ่งกีดขวางหน้าเซ็นเซอร์)



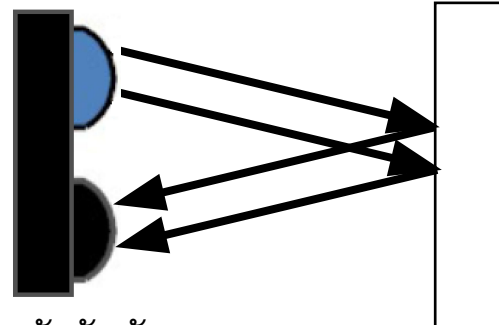
Good To Know

.....เกร็ดน่ารู้.....

เซ็นเซอร์ Infrared (IR)

1. เซ็นเซอร์อินฟราเรด (infrared sensor) หรือ IR มีองค์ประกอบสองอย่าง คือ ตัวส่งสัญญาณ (IR transmitter) หรือ (IR LED) และ ตัวรับสัญญาณ.
2. เมื่อเซ็นเซอร์ทำงาน ตัวปล่อยสัญญาณจะปล่อยแสงอินฟราเรด (IR) หากมีวัตถุวางไว้ด้านหน้าเซ็นเซอร์ แสงอินฟราเรด(IR)จะสะท้อนกลับไปยังตัวรับสัญญาณ.
3. ความแรงของแสงอินฟราเรด (IR) จะถูกตรวจพบโดยตัวรับสัญญาณจะถูกแปลงเป็นค่าอนาล็อกซึ่งคุณสามารถสังเกตได้ในหน้าต่างแสดงผล.
4. การเกิดการเปลี่ยนแปลงของค่าอินฟราเรด(IR) ที่อ่านได้ขึ้นอยู่กับสัดส่วนของระยะห่างระหว่างวัตถุกับเซ็นเซอร์อินฟราเรด (IR Sensor).

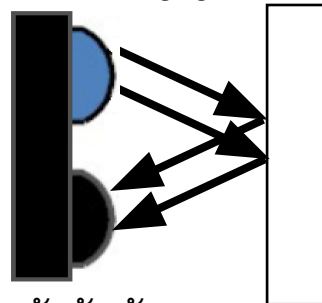
ตัวส่งสัญญาณ IR



ตัวรับสัญญาณ IR

วัตถุ: อยู่ห่าง - การอ่านค่า IR : ค่าที่ได้จะสูง

ตัวส่งสัญญาณ IR



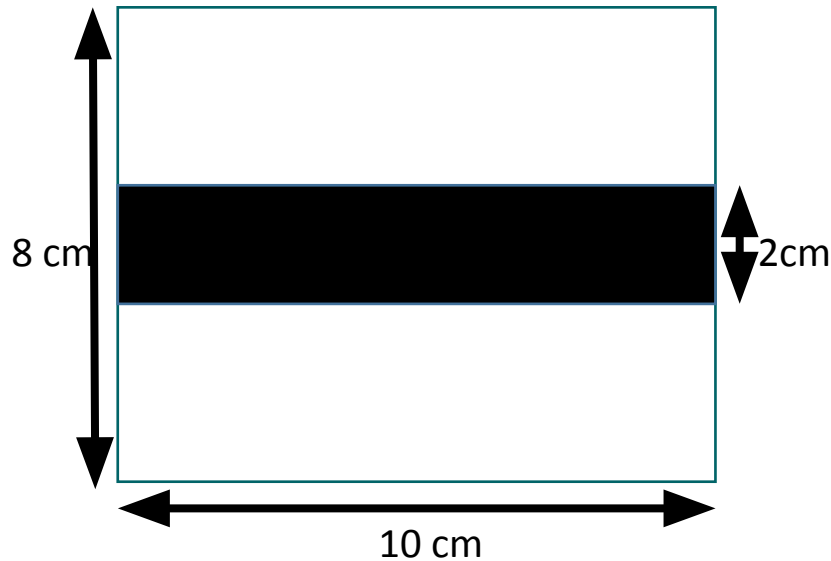
ตัวรับสัญญาณ IR

วัตถุ: อยู่ใกล้ - การอ่านค่า IR : ค่าที่ได้จะต่ำ

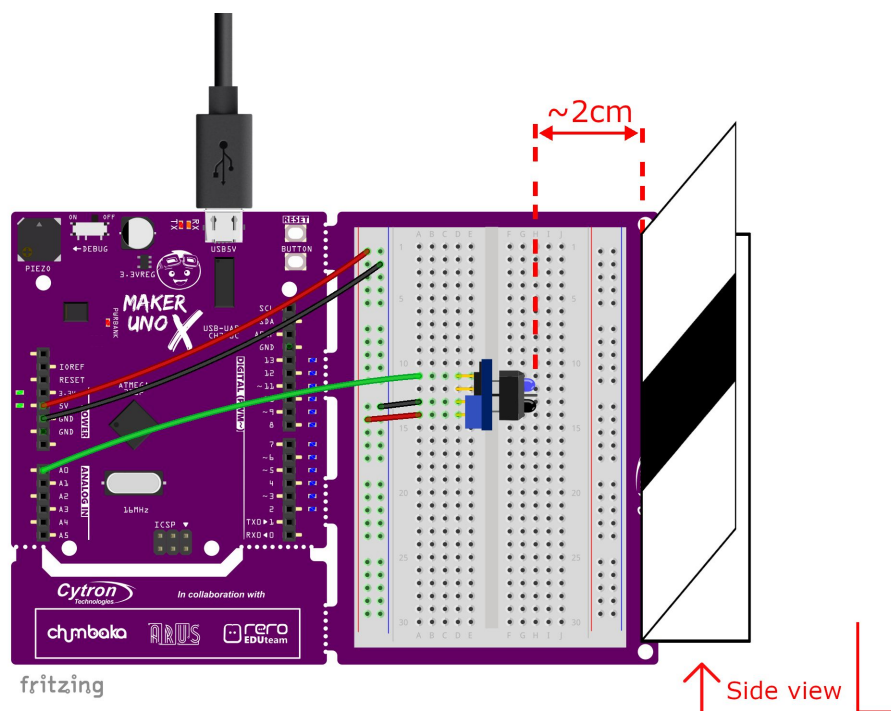


ชิ้นงานที่ 13 : การใช้เซ็นเซอร์อินฟราเรด (IR Sensor) ตรวจจับเส้นสีดำ

1. เตรียมกระดาษแข็ง 1 ชิ้น และปากกามาร์กเกอร์สีดำ วาดเส้นสีดำหนึ่งเส้น (หนาประมาณ 2 ซม.) ให้อยู่ในจุดศูนย์กลางของกระดาษแข็งตามตัวอย่างด้านล่าง.



2. พับกระดาษแข็งและวางไว้ด้านหลังของอินฟราเรดเซ็นเซอร์ (IR Sensor) ตามตัวอย่างที่แสดงด้านล่าง กำหนดระยะห่างของกระดาษแข็งและเซ็นเซอร์ประมาณ 2 ซม.



3. เลื่อนกระดาษตามแนวนอนจากปลายด้านหนึ่งไปอีกด้านหนึ่ง สังเกตและบันทึกการอ่านค่าของอินฟราเรด (IR) จากจอแสดงผล เมื่ออินฟราเรด(IR) อ่านค่าพื้นสีขาวหลังจากนั้นอ่านค่าพื้นสีดำ.

| Condition | IR Reading |
|---------------|------------|
| White Surface | |
| Black Line | |

4. เขียนโค้ดต่อไปนี้แล้วอัปโหลดไปยังบอร์ดของคุณ.

```
Project_13

int irValue;
void setup() {
  Serial.begin(9600);
  pinMode (3, OUTPUT);
}

void loop() {
  Serial.println(analogRead(A0));
  irValue = analogRead(A0);
  if (irValue > 500) {
    digitalWrite (3, HIGH);
    delay(200);
  }
  else {
    digitalWrite (3, LOW);
    delay(200);
  }
}
```

5. ตรวจสอบผลลัพธ์.

กด หรือ แสกน QR
code เพื่อดูวิดีโอ
ผลลัพธ์.



หลอดไฟ LED 3 จะสว่างขึ้นเมื่อเซ็นเซอร์ตรวจพบเส้นสีดำใช่หรือไม่? และจะดับลงเมื่อเซ็นเซอร์ตรวจพบพื้นสีขาวใช่หรือไม่?



Troubleshooting

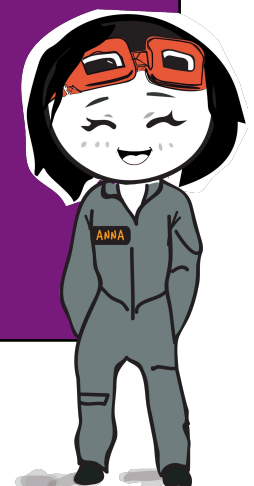
.....การแก้ไขหากเกิดปัญหา.....

ค่า "irValue > 500" คือค่าเริ่มต้นที่ฉันได้ทำการศึกษา. หากคุณต้องการเปลี่ยนแปลงค่าเริ่มต้นในขั้นตอนที่ 4 คุณก็สามารถใส่ค่าอื่นแทนค่าเริ่มต้นได้โดยให้มีเกณฑ์การเปรียบเทียบระหว่างสองเงื่อนไข แต่จะเป็นการดีหากคุณสามารถตั้งค่าที่จุดกึ่งกลาง.

คุณสามารถใช้สมการต่อไปนี้เพื่อตรวจสอบค่าเกณฑ์ที่เหมาะสม.

ค่าเกณฑ์ =


$$\left(\frac{\text{การอ่านผล IR เมื่อตรวจพบพื้นสีขาว} + \text{การอ่านผล IR เมื่อตรวจพบเส้นสีดำ}}{2} \right)$$





How it works

...กระบวนการทำงานของโปรแกรม...

| | |
|--|--|
|  | |
| Project_13 | |
| <code>int irValue;</code> | กำหนดค่าตัวแปร - "irValue" |
| <code>void setup() { Serial.begin(9600); pinMode (3, OUTPUT); }</code> | ตั้งค่าการเชื่อมต่อข้อมูลที่สัญญาณ 9600. ตั้งค่า Pin 3 เป็นการส่งข้อมูลขาออก. |
| <code>void loop() { Serial.println(analogRead(A0));</code> | อ่านค่าใน pin A0 และแสดงผลใน หน้าต่างแสดงผล. |
| <code> irValue = analogRead(A0);</code> | กำหนดให้ค่าของ pin A0 มีค่าเท่ากับ ตัวแปร irValue. |
| <code> if (irValue > 500) { digitalWrite (3, HIGH); delay(200); }</code> | ตรวจสอบตัวแปร irValue: หากค่า irValue > 500 (เซ็นเซอร์ตรวจ เจอเส้นสีดำ), หลอดไฟ LED 3 จะสว่าง ประมาณ 200 มิลลิวินาที. |
| <code> else { digitalWrite (3, LOW); delay(200); }</code> | หากไม่เช่นนั้น, หลอดไฟ LED 3 จะดับ. ประมาณ 200 มิลลิวินาที. |
| <code>}</code> | |



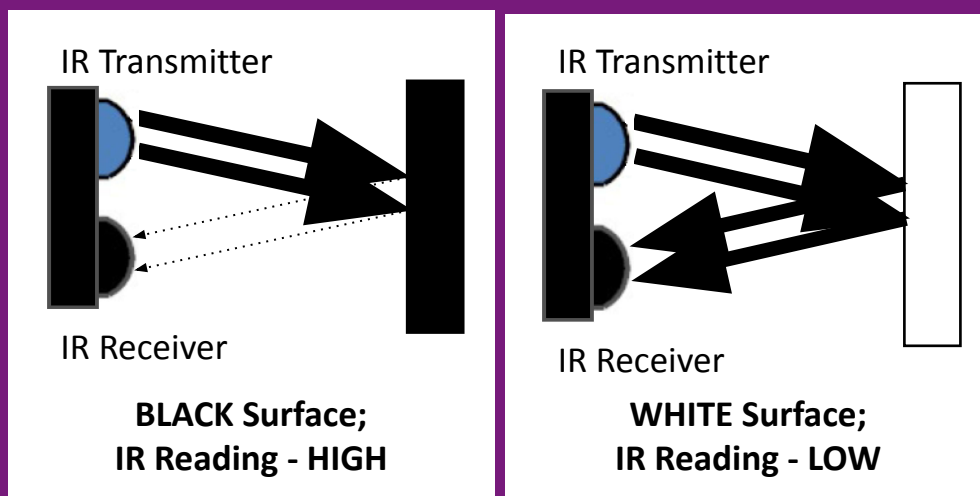
Good To Know

....เกร็ดความรู้....

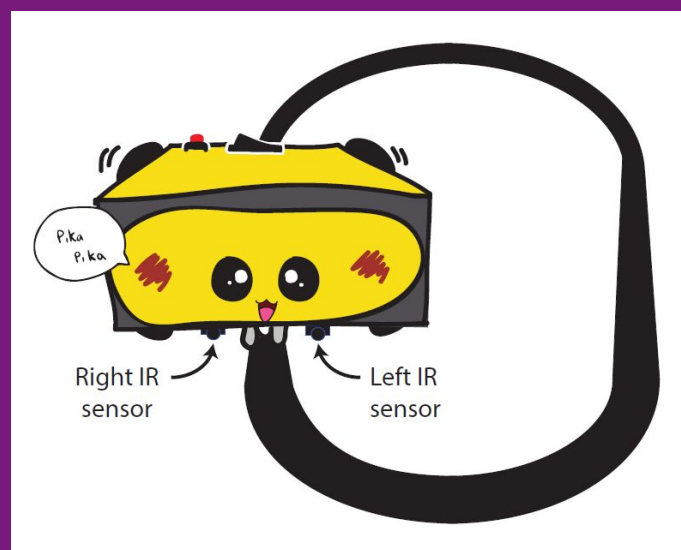
เซ็นเซอร์อินฟราเรด (IR Sensor)

เซ็นเซอร์อินฟราเรด (IR Sensor) คืออุปกรณ์ที่ทำหน้าที่ในการตรวจสอบและวัดระยะทาง นอกจากนั้นเรายังสามารถใช้เซ็นเซอร์อินฟราเรด (IR Sensor) ในการแยกความแตกต่างระหว่างพื้นผิวสีดำและสีขาวได้

นั่นเป็นเพราะว่าแสงสีขาวยังสามารถสะท้อนแสงส่วนใหญ่จากตัวส่งสัญญาณ เซ็นเซอร์อินฟราเรด (IR Sensor) ในขณะที่แสงสีดำนั้นจะดูดซับแสง ด้วยเหตุนี้ตัวรับสัญญาณเซ็นเซอร์อินฟราเรด (IR Sensor) จะตรวจจับแสงที่สะท้อนจากพื้นผิว สีขาวมากกว่าพื้นผิวสีดำ



สำหรับแนวคิดนี้เราสามารถใช้อินฟราเรด (IR Sensor) จัดเรียงเป็นเส้นตรงเพื่อสร้างหุ่นยนต์เดินตรงตามเส้นสีดำได้





Good To Know

.....เกร็ดน่ารู้.....

การแปลงรหัสสัญญาณ

ในโปรเจกต์นี้ ค่าของอุปกรณ์ส่งสัญญาณขาเข้าแบบอนาล็อกจากเซ็นเซอร์อินฟราเรด (IR Sensor) จะเปลี่ยนแปลงตลอดเวลาขึ้นอยู่กับสีของวัตถุที่อ่านค่าได้ เราจำเป็นต้องเก็บค่านั้นไว้ในตำแหน่ง (ตัวแปร) ที่แน่นอนและจะเรียกค่านั้นขึ้นมาใช้เมื่อต้องการ ซึ่งเราสามารถกำหนดตัวแปรได้ดังนี้

int variableName(ชื่อตัวแปร) = value; หรือ int variableName;

ชื่อตัวแปร(variableName) สามารถกำหนดเป็นค่าอะไรก็ได้ ยกเว้นค่าเฉพาะที่มีอยู่แล้วใน Arduino IDE ซึ่งได้รับการสนับสนุนเพื่อให้ใช้ชื่อที่สื่อถึงความหมายเช่น “irValue” หรือ “distance” ดังนั้นคุณสามารถเข้าใจได้ง่ายขึ้นว่าตัวแปรดังกล่าวหมายถึงอะไรหรือบุคคลอื่นก็สามารถเข้าใจในโค้ดที่คุณเขียนได้ง่ายยิ่งขึ้น



Challenge

.....โจทย์ท้าทาย.....

Task: ใช้ค่าอนาล็อกจากโพเทนชิโอมิเตอร์ (Potentiometer) เพื่อเปลี่ยนความเร็วในการกระพริบของหลอดไฟ LED 3

กดที่นี่หรือสแกน
QR code เพื่อรับชม
ผลลัพธ์.



ยินดีด้วย! คุณสำเร็จการเรียนรู้ในบทที่ 5 ในหัวข้อต่อไปนี้:

1. วิธีอ่านข้อมูลขาเข้าแบบอนาล็อก (analog input)
2. วิธีใช้จอแสดงผลข้อมูล
3. เซ็นเซอร์อินฟราเรด (IR sensor) และกระบวนการทำงาน
4. วิธีกำหนดค่าตัวแปร



บทเรียนที่ 6

การควบคุม DC มอเตอร์
(DC MOTOR)



กิจกรรมที่ 14 การควบคุมมอเตอร์ DC

1. เตรียมส่วนประกอบต่อไปนี้ให้พร้อม



สายเชื่อมต่อไฟฟ้า
(พร้อมหมุดหัวผู้-ผู้)



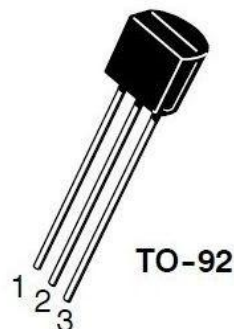
มอเตอร์ DC พร้อมใบพัด



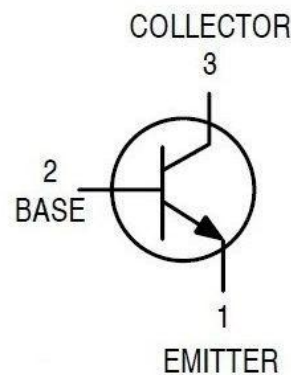
ไดโอด (Diode)
ขั้วต่อที่มีแถบสีขาวควรเชื่อม
ต่อกับสายบวก (สีแดง)



ตัวต้านทาน (Resistor)
220 โอห์ม

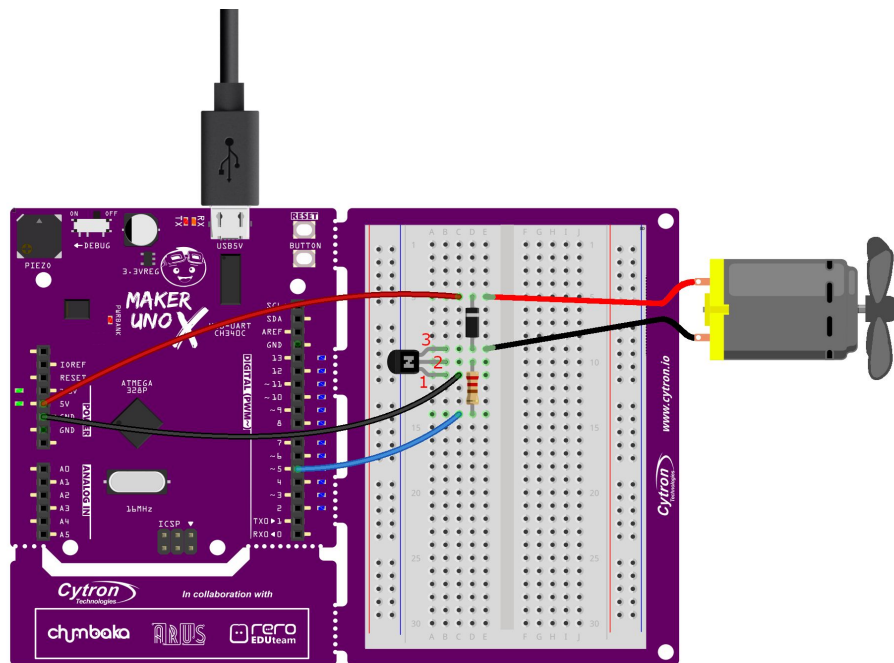


TO-92



2N2222 หรือ TO-92 NPN ทรานซิสเตอร์

2. สร้างวงจรตามที่แสดงในภาพด้านล่าง:



fritzing

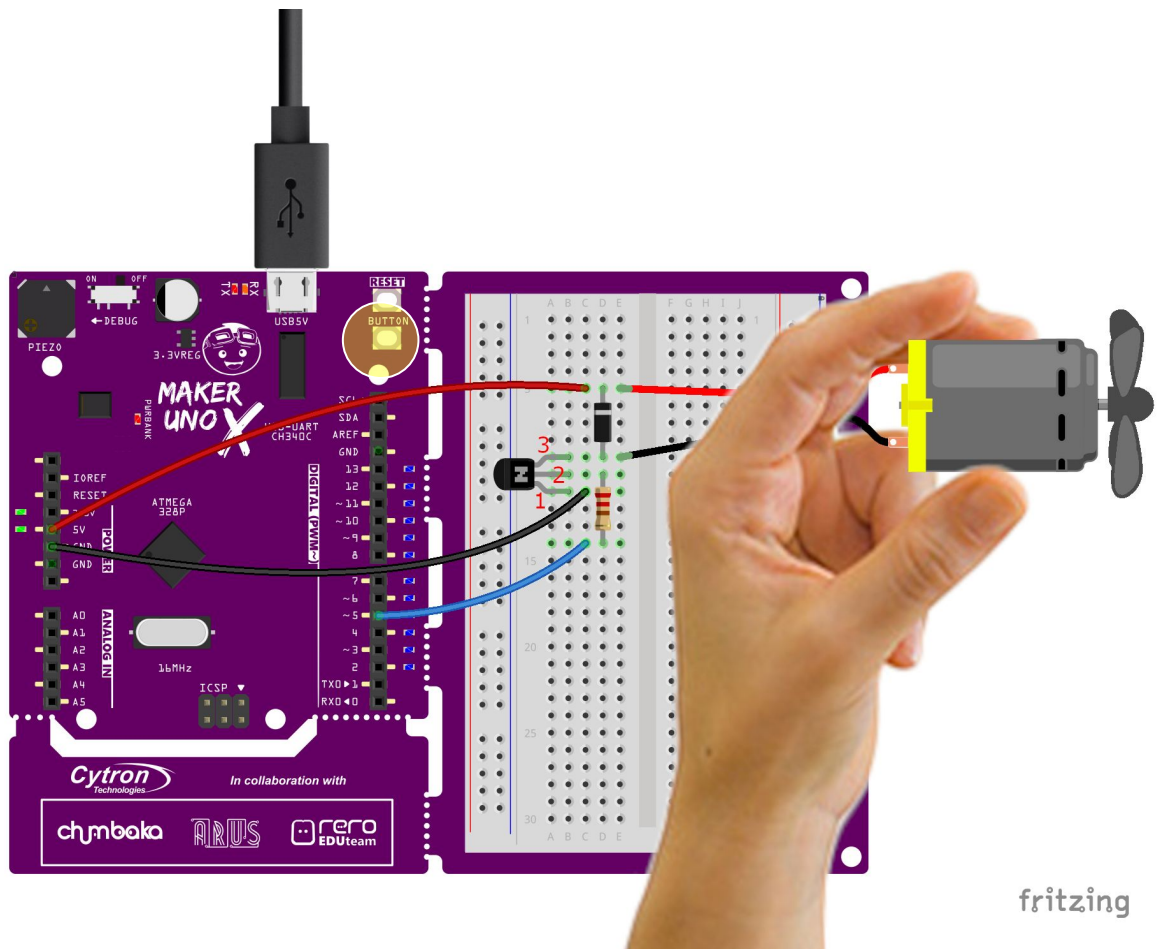
2. เขียนและอัปโหลดซอร์สโค้ดต่อไปนี้ลงในบอร์ด UNO X ของคุณ

```
กิจกรรมที่ 14

void setup() {
  pinMode(2, INPUT_PULLUP);
  pinMode(5, OUTPUT);
}

void loop() {
  if (digitalRead(2)==LOW) {
    while(1) {
      analogWrite(5, 255);
      delay(3000);
      analogWrite(5, 80);
      delay(3000);
    }
  }
  else digitalWrite(3, LOW);
}
```

4. เมื่ออัปโหลดแล้วกดสวิทช์บอร์ดและสังเกตผล...



fritzing

คลิกที่นี่หรือสแกน
รหัส QR เพื่อดูวิดีโอ
สาธิต



เมื่อคุณกดสวิทช์คุณเห็นมอเตอร์เริ่มหมุนหรือไม่?
มันจะหมุนอย่างต่อเนื่อง ขณะที่เปลี่ยนความเร็ว
ในช่วงระหว่างเต็มที่และขนาดกลาง, จนกว่าคุณ
จะปล่อย สวิทช์หรือกดปุ่มรีเซ็ต





มันทำงานยังไ

✓ ↻ 📄 ⬆ ⬇
🔍

กิจกรรมที่ 14

```

void setup() {
  pinMode(2, INPUT_PULLUP);
  pinMode(5, OUTPUT);
}

void loop() {
  if (digitalRead(2) == LOW) {
    while(1) {
      analogWrite(5, 255);
      delay(3000);
      analogWrite(5, 80);
      delay(3000);
    }
  }
  else digitalWrite(5, LOW);
}
```

ตั้งค่าขา 2 (DIO2) เป็น input_pullup (สวิตช์)
ตั้งค่าขา 5 (DIO5) เป็น เอาท์พุต (output)

ตรวจสอบค่าขา 2 ถ้าค่าของขา 2 = LOW (สวิตช์ถูกกด), กำหนดค่าของขาที่ 5 ถึง 255 (มอเตอร์หมุนด้วยความเร็วสูงสุด) เป็นเวลา 3 วินาที, หลังจากนั้นกำหนดค่าของขาที่ 5 ถึง 80 (มอเตอร์หมุนด้วยความเร็วปานกลาง) เป็นเวลา 3 วินาที ทำซ้ำจนกว่าจะปล่อยสวิตช์

หากเป็นเงื่อนไขอื่นอื่นตั้งขา 5 เป็น LOW [ไม่ได้กดสวิตช์] (มอเตอร์ถูกปิดมันไม่หมุน)



การแก้ไขปัญหา

1. ตรวจสอบให้แน่ใจว่าวางมอเตอร์ DC ไว้บนพื้นผิวที่ใบพัดหมุนได้อย่างอิสระเช่นบนขอบโต๊ะของคุณมอเตอร์อาจมีแรงบิดไม่เพียงพอที่จะหมุนถ้าใบพัดอยู่บนพื้นผิว
1. เปิดโหมด Debug และสังเกตหลอดไฟ LED ในตัวที่ขา 5 หลังจากทีสวิตช์กดหลอดไฟ LED ควรเปลี่ยนในที่สว่างเต็มเวลา 3 วินาทีแล้วจางสว่างอีก 3 วินาที.

ถ้าขั้นตอนที่ 2 ด้านบนทำงานได้ แต่มอเตอร์ยังไม่หมุน คุณต้องตรวจสอบวงจรอิเล็กทรอนิกส์ของคุณและตรวจสอบให้แน่ใจว่าคุณได้เชื่อมต่อส่วนประกอบอิเล็กทรอนิกส์ทั้งหมดอย่างถูกต้อง



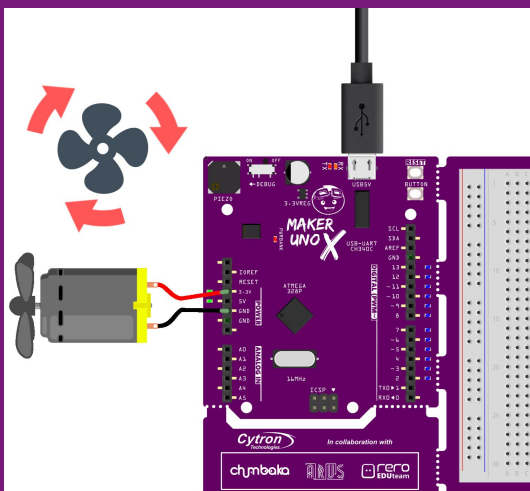


ดีแล้วที่รู้

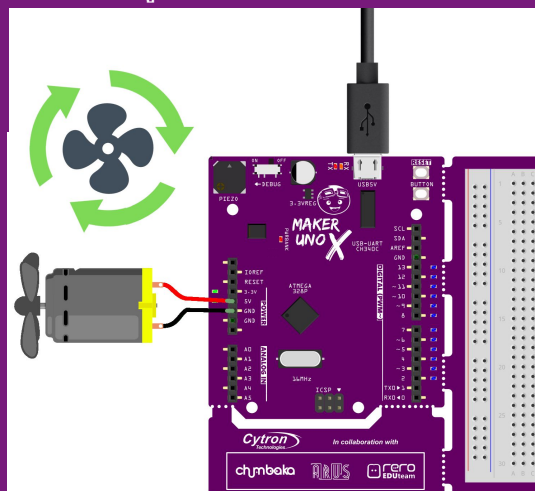
วิธีการควบคุมความเร็วของมอเตอร์ DC

ในการหมุนมอเตอร์กระแสตรงคุณจำเป็นต้องจ่ายแรงดันไฟที่เหมาะสม แรงดันไฟฟ้าอินพุตที่เหมาะสมโดยทั่วไปสำหรับมอเตอร์ DC คือ 3V, 6V และ 12V ผู้ผลิตมอเตอร์มักจะระบุแรงดันไฟฟ้าที่แนะนำในข้อกำหนดของผลิตภัณฑ์

แต่มอเตอร์ยังสามารถหมุนได้แม้ในขณะที่คุณใช้แรงดันไฟฟ้ามากหรือน้อยกว่าที่แนะนำ สำหรับมอเตอร์ DC ที่เราจะใช้ในกิจกรรมนี้แรงดันไฟฟ้าที่แนะนำคือ 3-6V ลองนำไปใช้ 3.3V และ 5V และดูสิ่งที่เกิดขึ้น



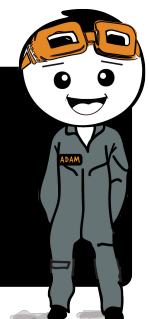
สายไฟสีดำเชื่อมต่อกับขา GND
และสายสีแดงกับขา 3.3V



สายไฟสีดำเชื่อมต่อกับขา GND
และสายสีแดงกับขา 5V

เราสามารถสังเกตได้ว่าความเร็วของมอเตอร์ DC เปลี่ยนไปตามแรงดันไฟฟ้าขาเข้า มอเตอร์จะหมุนเร็วขึ้นด้วยแรงดันไฟฟ้าสูงและหมุนช้าลงด้วยแรงดันไฟฟ้าต่ำ ดังนั้นในกิจกรรมที่ 14 เราเห็นว่าเมื่อเราตั้งขา 5 ให้มีค่าถึง 255 มอเตอร์จะหมุนที่ความเร็วเต็มที่และมันหมุนด้วยความเร็วที่ช้าลงเมื่อขา 5 คุ่มค่าถูกตั้งไว้ที่ 80

ข้อควรระวัง! การใช้ไฟฟ้าแรงสูงกับมอเตอร์ (นอกเหนือจากแรงดันไฟฟ้าที่แนะนำ) จะทำให้มอเตอร์เสียหายในระยะยาว

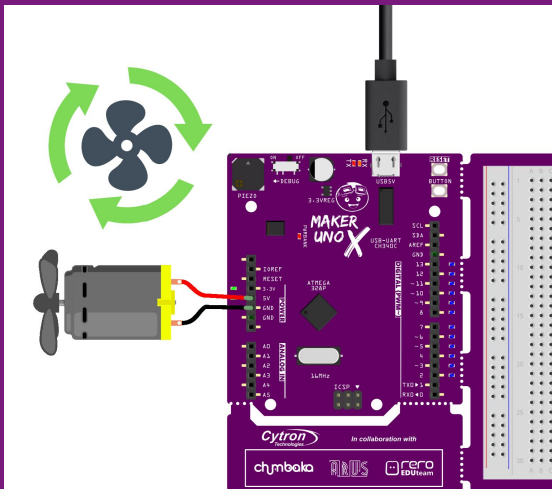




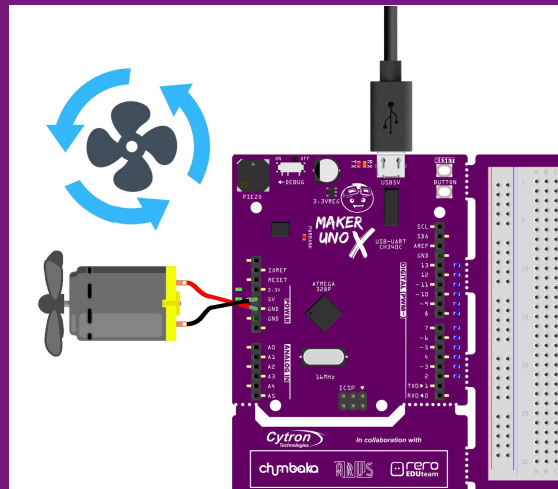
ดีแล้วที่รู้

วิธีควบคุมทิศทางการหมุนของมอเตอร์ DC

มันเป็นเรื่องง่ายมากที่จะเปลี่ยนทิศทางที่มอเตอร์หมุน - เพียงแค่กลับขั้วของมอเตอร์โดยการเปลี่ยนการเชื่อมต่อของสายไฟสีแดงและสีดำ



สายสีดำเชื่อมต่อกับขา GND
และสายสีแดงเชื่อมต่อกับขา 5V

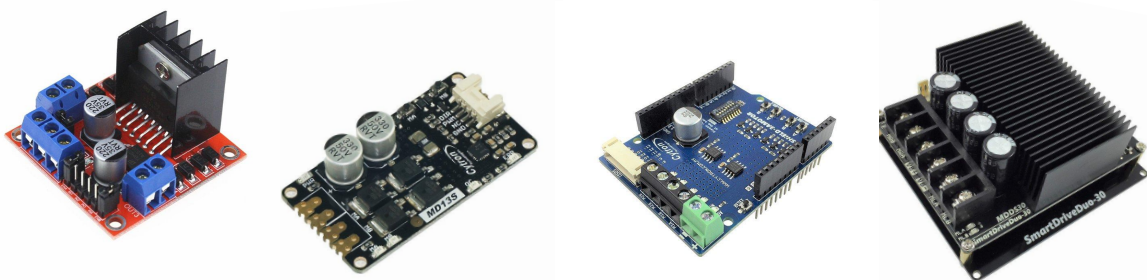


สายสีดำเชื่อมต่อกับขา 5V และสาย
สีแดงเชื่อมต่อกับขา GND

คุณสังเกตเห็นว่ามอเตอร์หมุนตรงข้ามทิศทางเมื่อเรา
สลับการเชื่อมต่อสายไฟ?



เป็นเรื่องยากที่คุณจะเปลี่ยนการเชื่อมต่อสายไฟทุกครั้งที่คุณต้องการที่จะ
เปลี่ยนทิศทางที่มอเตอร์หมุนเพื่อที่จะควบคุมได้อย่างง่ายดายทั้งทิศทาง
หมุนและความเร็วของมอเตอร์ DC (โดยไม่ต้องแก้ไขวงจร) เราสามารถใช้
วงจรควบคุมมอเตอร์เช่นต่อไปนี้





กิจกรรมที่ 15: ควบคุมความเร็วมอเตอร์โดยใช้สวิตช์

1. เขียนและอัปโหลดซอร์สโค้ดต่อไปนี้ลงในบอร์ด UNO X ของคุณ

```
Project_15

int mode = 0;
void setup() {
  pinMode(2, INPUT_PULLUP);
  pinMode(5, OUTPUT);
}
void loop() {
  switch(mode) {
    case 0:
      analogWrite (5, 0);
      break;
    case 1:
      analogWrite (5, 100);
      break;
    case 2:
      analogWrite (5, 255);
      break;
    default:
      mode = 0;
      break;
  }
  if (digitalRead(2)==LOW) {
    while(digitalRead(2)==LOW);
    mode++;
    if (mode==3)
      mode=0;
  }
}
```

3. เมื่ออัปโหลดแล้วกดสวิทช์ออนบอร์ดและสังเกตผล...
ทำซ้ำหลายครั้งและสังเกตการเปลี่ยนแปลงความเร็วมอเตอร์ในแต่ละครั้งที่กดสวิทช์

คลิกที่นี่หรือสแกน
รหัส QR เพื่อดูวิดีโอ
สาธิต



เมื่อคุณกดสวิทช์ครั้งแรกมอเตอร์จะเริ่มหมุนและ
เมื่อคุณกดสวิทช์ครั้งที่สองมอเตอร์จะหมุนเร็วขึ้น
(ด้วยความเร็วเต็ม) ครั้งที่สามที่คุณกดสวิ
ทช์มอเตอร์จะหยุดหมุน





มันทำงานยังไง

| | |
|--|--|
| Project_15 | |
| <code>int mode = 0;</code> | กำหนดตัวแปรชื่อ "โหมด" และตั้งค่าเริ่มต้นเป็น 0 |
| <code>void setup() { pinMode(2, INPUT_PULLUP); pinMode(5, OUTPUT); }</code> | ตั้งค่าขา 2 (DIO2) เป็น input_pullup (สวิตช์) ตั้งค่าขา 5 (DIO5) เป็น เอาท์พุต (output) |
| <code>void loop() { switch(mode) { case 0; analogWrite (5, 0); break; case 1: analogWrite (5, 100); break; case 2: analogWrite (5, 255); break; default: mode = 0; break; } if (digitalRead(2)==LOW) { while(digitalRead(2)==LOW); mode++; } if (mode==3) mode=0; }</code> | ตั้งเงื่อนไขสำหรับกรณีและคำสั่งต่าง ๆ ที่จะดำเนินการเมื่อตรงตามเงื่อนไขแต่ละข้อ หากโหมด = 0, มอเตอร์ที่ขา 5 ริ่งที่ความเร็ว 0 (เช่นไม่หมุน) หากโหมด = 1 มอเตอร์ที่ขา 5 ริ่งที่ความเร็ว 100 (เช่นหมุนช้าๆ) หากโหมด = 2 มอเตอร์ที่ขา 5 ริ่งที่ความเร็ว 255 (หมุนที่ความเร็วเต็ม) หากโหมดไม่ใช่ 0,1 หรือ 2 ให้ตั้งโหมดเป็น 0 หากกดสวิตช์ที่ขา 2 เพิ่มค่าของโหมดตัวแปร 1 หลังจากปล่อยสวิตช์ หากโหมด = 3 ให้ตั้งค่าของโหมดกลับเป็น 0 |



ไวยากรณ์ของซอร์สโค้ด

1. เช่นเดียวกับคำสั่ง if, switch ... case ควบคุมการไหลของโปรแกรมโดยอนุญาตให้โปรแกรมเมอร์ระบุบรรทัดของโค้ดที่จะถูกเรียกใช้เมื่อตรงตามเงื่อนไขที่ตั้งไว้ สิ่งนี้มีประโยชน์เมื่อคุณจำเป็นต้องใช้สวิตช์รูปแบบเดียวกันในการรันโค้ดที่ต่างกันเมื่อตรงตามเงื่อนไขต่าง ๆ

```
switch (var) {  
  case 1:  
    //ทำอะไรบางอย่างเมื่อ var เท่ากับ 1  
    break;  
  case 2:  
    //ทำอะไรบางอย่างเมื่อ var เท่ากับ 2  
    break;  
  default:  
    // หาก var ไม่ใช่ 1 หรือ 2 (var จะเท่ากับค่าอื่น ๆ)  
    // ให้เรียกใช้โค้ดต่อไปนี้  
    // default คือเงื่อนไขที่ไม่จำเป็นเสมอไป  
    break;  
}
```

1. หากต้องการหยุดการหมุนของมอเตอร์คุณสามารถเลือกซอร์สโค้ดหนึ่งในสองบรรทัดด้านล่าง:

```
digitalWrite (pin, LOW);
```

หรือ

```
analogWrite (pin, 0);
```

```
//ในกรณีของเรา pin ถูกตั้งค่าเป็น 2  
//เราสามารถใส่
```

```
digitalWrite(2,LOW);
```

หรือ

```
analogWrite(2,0);
```



ดีแล้วที่รู้

3. คุณสังเกตเห็นหรือไม่ว่าเราไม่ได้ใส่เครื่องหมายเซมิโคลอน ";" หลังจาก "while" (1) ในกิจกรรมก่อนหน้านี้ของเรา แต่เราเพิ่มเซมิโคลอนหลังจากนั้น (digitalRead (2) == LOW) ในกิจกรรมนี้หรือไม่

หากไม่มี ";"

```
while (เงื่อนไข) {  
  //หากเงื่อนไขเป็นจริงโปรแกรมจะรันโค้ดทุกบรรทัดภายใน {}
```

หากมี ";"

```
while (เงื่อนไข) ;  
//หากเงื่อนไขเป็นจริงมันจะรันเฉพาะบรรทัดนี้จนกว่าเงื่อนไขจะเป็นเท็จ
```

ตัวอย่าง:

```
while(digitalRead(2)==LOW);  
// หากกดสวิตช์ 2 (เงื่อนไขเป็นจริง) โปรแกรมจะยังคงอยู่ที่  
บรรทัดของซอร์สโค้ดนี้จนกระทั่งปล่อยสวิตช์ 2 (เงื่อนไขจะเป็น  
เท็จ) จากนั้นและจะย้ายไปยังเพื่อดำเนินการซอร์สโค้ดบรรทัดถัด  
ไป
```



ความท้าทาย

การฝึกหัด: ตั้งโปรแกรมให้มอเตอร์หมุนด้วยความเร็วคงที่เมื่อคุณกดสวิตช์ค้างไว้ มอเตอร์ควรหยุดหมุนเมื่อคุณปล่อยสวิตช์และเมื่อคุณกดสวิตช์เป็นครั้งที่สอง ความเร็วของมอเตอร์ควรเปลี่ยน

[คลิกที่นี่หรือสแกน
รหัส QR เพื่อดูวิดีโอ
สาธิต](#)



ขอแสดงความยินดี: คุณได้เรียนบทที่ 6 เสร็จสิ้นแล้วและได้เรียนรู้วิธีการทำรายการต่อไปนี้

1. วิธีสร้างวงจรอิเล็กทรอนิกส์เพื่อควบคุมมอเตอร์ DC
2. วิธีการควบคุมความเร็วของมอเตอร์ DC
3. วิธีใช้คำสั่ง "switch .. case"



บทเรียนที่ 7

เซ็นเซอร์อัลตราโซนิก
(ULTRASONIC SENSOR)

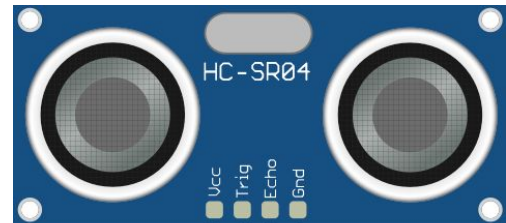


Project 16: Setting Up Ultrasonic Sensor

1. เตรียมอุปกรณ์เหล่านี้ให้พร้อม

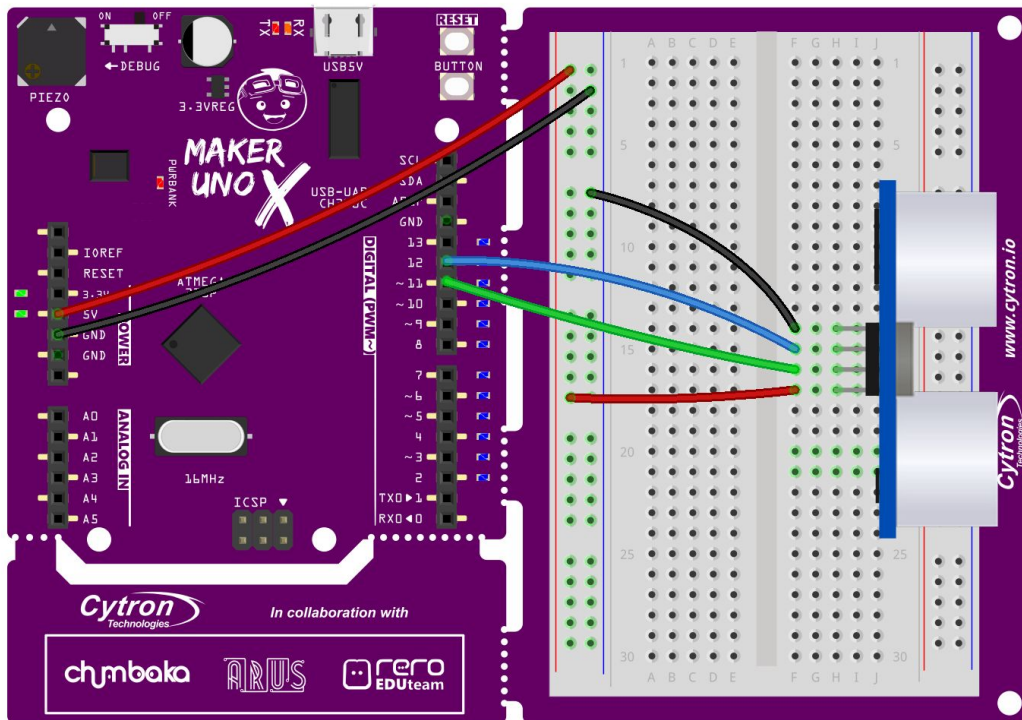


เซ็นเซอร์อัลตราโซนิก
(Ultrasonic Sensor)



VCC Trig Echo GND

1. ต่อดังต่อไปนี้ตามรูปที่แสดงด้านล่าง



fritzing

| Ultrasonic Sensor | VCC | TRIG | ECHO | GND |
|-------------------|-----|--------|--------|-----|
| Maker UNO X | 5V | PIN 11 | PIN 12 | GND |

เชื่อมต่อเซนเซอร์เข้าที่บริเวณขอบของบอร์ดและ
ตรวจสอบให้แน่ใจว่า สายไฟจัมเปอร์ไม่ไปขวางเซนเซอร์



- เขียนและอัปโหลดโค้ดดังต่อไปนี้ลงในบอร์ดของคุณ.

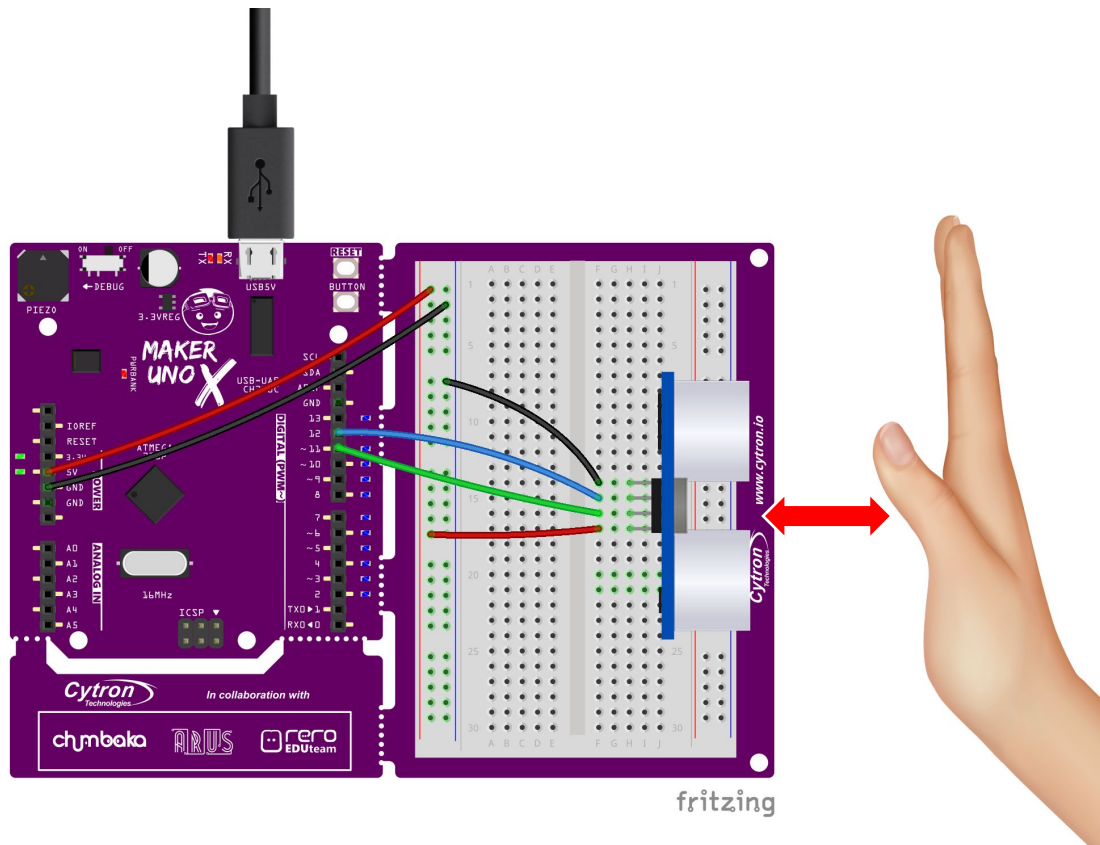
```
Project_16

long duration;
int distance;

void setup() {
  pinMode(11, OUTPUT);
  pinMode(12, INPUT);
  Serial.begin (9600);
}

void loop() {
  digitalWrite(11, LOW);
  delayMicroseconds(2);
  digitalWrite(11, HIGH);
  delayMicroseconds(10);
  digitalWrite(11, LOW);
  duration = pulseIn (12, HIGH);
  distance = duration*0.034/2;
  delay(50);
  Serial.print ("Distance = ");
  Serial.print (distance);
  Serial.println ("cm ");
}
```

4. ทันทีที่การอัปโหลดเสร็จสิ้น คลิกที่ปุ่ม Serial Monitor เพื่อดูผลลัพธ์ นำมือของคุณไปบังข้างหน้าเซนเซอร์ สังเกตค่าที่แสดงบนหน้าต่างของ Serial monitor ขณะเดียวกันเลื่อนมือของคุณหาเซนเซอร์อัลตราโซนิก หลังจากนั้นเลื่อนมือออก



4. ตรวจสอบผลลัพธ์ของคุณ.

คลิก หรือ สแกน
คิวอาร์โค้ด เพื่อรับ
ชมวิดีโอสาธิต.



คุณสังเกตเห็นอะไรบ้าง ค่าที่อ่านได้จากอัลตราโซนิกเพิ่มขึ้นหรือลดลงตามที่คุณเลื่อนกล่องเข้าหาเซนเซอร์หรือไม่ แล้วค่าที่อ่านได้หลังจากนำกล่องออก (ไม่มีสิ่งกีดขวาง) เป็นอย่างไร





How it works

Project_16

```
long duration;  
int distance;  
  
void setup() {  
  pinMode(11, OUTPUT);  
  pinMode(12, INPUT);  
  Serial.begin (9600);  
}  
  
void loop() {  
  digitalWrite(11, LOW);  
  delayMicroseconds(2);  
  digitalWrite(11, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(11, LOW);  
  
  duration = pulseIn (12, HIGH);  
  
  distance = duration*0.034/2;  
  delay(50);  
  
  Serial.print ("Distance = ");  
  Serial.print (distance);  
  Serial.println ("cm");  
}
```

กำหนดตัวแปรชื่อ “duration” เป็นตัวแปรแบบ long
หนดตัวแปรชื่อ “distance” เป็นแบบ Integer

ตั้งค่าขาพิน11(trigger pin)เป็นเอาต์พุต
ตั้งค่าขาพิน12 (echo pin) เป็นอินพุต
คั้งค่า serial monitor (9600 baud)

ตั้งค่าของขาพิน11(trigger pin) เป็นlow
หน่วงเวลา 2 มิลลิวินาที
ตั้งค่าขาพิน 11เป็น high(ส่งสัญญาณออก).
หน่วงเวลา 10 มิลลิวินาที
ตั้งค่าขาพิน 11 เป็น low อีกครั้ง

ตรวจสอบ pulse duration (ช่วงเวลาคลื่น) ที่ขาพิน12 (echo pin) แล้วส่งค่าไปเก็บไว้ที่ตัวแปร “duration”

แปลงค่าของตัวแปร “duration” ไปเป็นระยะทางในหน่วยเซนติเมตรแล้วแสดงค่าค้างไว้ 50 ms

แสดง “Distance = “ ใน serial monitor
ตามด้วย “distance value” และหน่วย “cm”

6. ขั้นตอนถัดไป ลองแก้ไขโปรแกรมให้เป็นเหมือนโค้ดดังต่อไปนี้แล้วอัปโหลดเข้าไปในบอร์ดอีกครั้งหนึ่ง



The screenshot shows the Arduino IDE interface with a project named "Project_16a". The code is written in C++ and is designed to measure distance using an ultrasonic sensor. It includes variable declarations for duration and distance, a setup function to initialize pins and serial communication, a loop function to read the sensor and print the distance, and an ultrasonic function to handle the sensor's timing.

```
long duration;
int distance;

void setup() {
  pinMode(11, OUTPUT);
  pinMode(12, INPUT);
  Serial.begin (9600);
}

void loop() {
  ultrasonic ();
  Serial.print ("Distance = ");
  Serial.print (distance);
  Serial.println ("cm ");
}

void ultrasonic() {
  digitalWrite(11, LOW);
  delayMicroseconds(2);
  digitalWrite(11, HIGH);
  delayMicroseconds(10);
  digitalWrite(11, LOW);
  duration = pulseIn (12, HIGH);
  distance = duration*0.034/2;
  delay(50);
}
```

7. ตรวจสอบผลลัพธ์ที่ได้ของคุณ.

[คลิก หรือ สแกน](#)
[คิวอาร์โค้ด เพื่อรับ](#)
[ชมวิดีโอสาธิต](#)



คุณควรจะได้ผลลัพธ์เหมือนกับโปรแกรม
ก่อนหน้านี้



How it works

```
Project_16a

void loop() {
  ultrasonic ();
  Serial.print ("Distance = ");
  Serial.print (distance);
  Serial.println ("cm ");
}

void ultrasonic() {
  digitalWrite(11, LOW);
  delayMicroseconds(2);
  digitalWrite(11, HIGH);
  delayMicroseconds(10);
  digitalWrite(11, LOW);
  duration = pulseIn (12, HIGH);
  distance = duration*0.034/2;
  delay(50);
}
```

เรียกใช้ฟังก์ชัน “ultrasonic” (เป็นการ
เรียกใช้โค้ดทุกบรรทัดที่อยู่ในฟังก์ชัน
“ultrasonic”

โค้ดทุกบรรทัดเหล่านี้จะถูกรวมเป็นกลุ่ม
อยู่ในฟังก์ชันที่ชื่อว่า “ultrasonic”

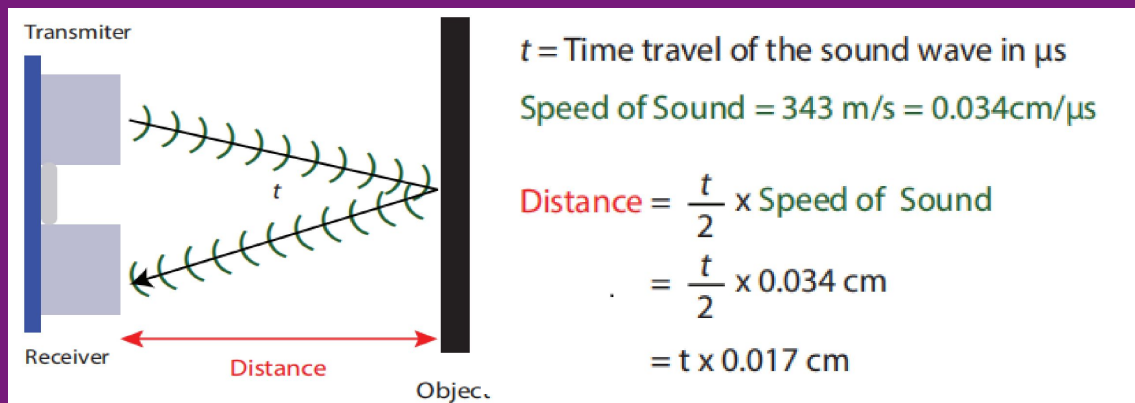


Good To Know

Ultrasonic Sensors

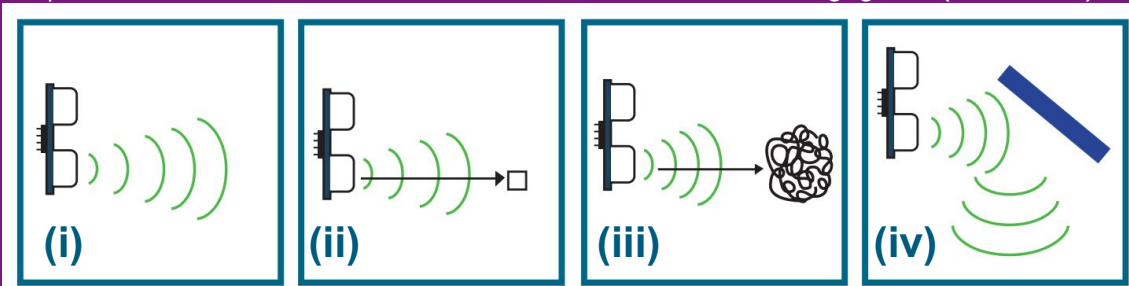
เซนเซอร์อัลตราโซนิกถูกใช้เหมือนกับเซนเซอร์อินฟราเรด (IR sensor) ใช้ในการตรวจจับสิ่งกีดขวางและวัดระยะทาง กลไกการทำงานคล้ายกับเซนเซอร์อินฟราเรด ตัวส่งสัญญาณ (transmitter) ในเซนเซอร์จะส่งคลื่นความถี่เหนือเสียงออกไป ซึ่งเมื่อคลื่นไปกระทบกับวัตถุที่อยู่ในแนวของคลื่น คลื่นจะถูกสะท้อนกลับมาและถูกตรวจจับได้โดยตัวรับสัญญาณ

ด้วยการคำนวณระยะเวลาจากที่คลื่นถูกส่งไปสะท้อนกลับวัตถุจนกลับมายังตัวรับสัญญาณจะทำให้เราสามารถหาระยะทางของวัตถุกับเซนเซอร์ได้



เซนเซอร์อัลตราโซนิกถูกใช้อย่างกว้างขวางในรถยนต์ (car reverse sensor) เพื่อช่วยผู้ขับรถประมาณระยะทางและตรวจจับสิ่งกีดขวาง ข้อดีของเซนเซอร์อัลตราโซนิกที่เหนือกว่าเซนเซอร์อินฟราเรดคือ ไม่อ่อนไหวต่อแสงรอบๆ บริเวณ

อย่างไรก็ตาม เซนเซอร์อัลตราโซนิกอาจจะทำงานได้ไม่เป็นไปตามที่คาดหวัง ถ้าวัตถุมีลักษณะเป็น (i) โค้งเกินไป, (ii) เล็กเกินไป, (iii) นุ่มเกินไป หรือ (iv) วัตถุสะท้อนคลื่นเสียงไปในทิศทางอื่นออกไปจากตัวรับสัญญาณ (receiver)





ARDUINO FUNCTION

1. คุณสามารถใช้ฟังก์ชันนี้ในการอ่านค่าคาบ (time cycle) ของพัลส์ (pulse)
pulseIn (pin, value)
pin: ตัวเลขของขาพินที่คุณต้องการอ่านค่าพัลส์ (pulse)
value: ประเภทของพัลส์ที่ต้องการอ่านค่า แบ่งเป็น HIGH และ LOW

ถ้าค่า value เป็น HIGH ฟังก์ชัน pulseIn() จะรอรับค่าตอนที่พินเปลี่ยนจาก LOW เป็น HIGH แล้วจะเริ่มนับเวลา จากนั้นรอจนกว่าค่าจะเปลี่ยนกลับเป็น LOW และหยุดจับเวลา แล้วฟังก์ชันนี้จะส่งค่าความยาวของพัลส์กลับมาในหน่วยมิลลิวินาที (milliseconds)

2. ประเภทของข้อมูล (Data Type)
คุณอาจจะมีคำถามว่า ทำไมส่วนมากค่าของเวลาเรามักจะใช้ “int” (integer : จำนวนเต็ม) ในการกำหนดตัวแปร แต่ในโปรเจกต์นี้เราใช้ตัวแปรประเภท “long” แทน เหตุผลเพราะว่า เราต้องการบอก Arduino ให้ทราบถึงชนิดของตัวเลขที่เราต้องการเก็บไว้ในตัวแปร (ว่ามันเป็นเลขขนาดเล็ก, เลขขนาดใหญ่ เลขทศนิยมและอื่นๆ ในโปรเจกต์นี้ ตัวเลขที่เราต้องการเก็บไว้ในตัวแปร “duration” เป็นตัวเลขขนาดใหญ่ (large number)

| Variable | Number Range |
|----------|---------------------------------|
| char | -128 to 127 |
| int | -32,768 to 32,767 |
| long | -2,147,483,648 to 2,147,483,647 |
| float | 3.4028235E+38 to 3.4028235E+38 |



Project 17: Build a Car Rear Bumper Sensor

1. M1.แก้ไขโค้ดของคุณในโปรเจกต์ที่ 16 ตามโค้ดดังต่อไปนี้:

```
Project_17
long duration;
int distance;

void setup() {
  pinMode(8, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, INPUT);
}

void loop() {
  ultrasonic ();
  if (distance < 2) {
    tone(8, 349);
  }
  else {
    tone (8, 349);
    delay(50);
    noTone(8);
    delay(distance*10);
  }
}

void ultrasonic() {
  digitalWrite(11, LOW);
  delayMicroseconds(2);
  digitalWrite(11, HIGH);
  delayMicroseconds(10);
  digitalWrite(11, LOW);
  duration = pulseIn (12, HIGH);
  distance = duration*0.034/2;
  delay(50);
}
```

2. อัปโหลดโค้ดเข้าไปใน Maker UNO X และตรวจสอบผลลัพธ์ที่ได้

[คลิก หรือ สแกน](#)
[คิวอาร์โค้ด เพื่อรับ](#)
[ชมวิดีโอสาธิต](#)



เพียโซบัสเซอร์ (piezo buzzer) ส่งเสียง beep ซ้ำๆ เมื่อเซนเซอร์ตรวจจับสิ่งกีดขวางได้หรือไม่?

คุณสังเกตเห็นหรือไม่ว่าเสียง beep ดังถี่ขึ้นเรื่อย ๆ เมื่อเซนเซอร์ยิ่งเคลื่อนที่เข้าใกล้วัตถุ?





How it works

```
Project_17

long duration;
int distance;

void setup() {
  pinMode(8, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, INPUT);
}

void loop() {
  ultrasonic ();

  if (distance < 2) {
    tone(8, 349);
  }
  else {
    tone (8, 349);
    delay(50);
    noTone(8);
    delay(distance*10);
  }
}

void ultrasonic() {
  digitalWrite(11, LOW);
  delayMicroseconds(2);
  digitalWrite(11, HIGH);
  delayMicroseconds(10);
  digitalWrite(11, LOW);
  duration = pulseIn (12, HIGH);
  distance = duration*0.034/2;
  delay(50);
}
```

ตั้งขาพิน 8 (piezo buzzer) เป็นเอาต์พุต

เรียกใช้ฟังก์ชัน “ultrasonic”

ถ้าระยะทางน้อยกว่า 2 เซนติเมตร บัซเซอร์ (buzzer) จะส่งเสียง beep เป็นโน้ตเสียงฟากลางหรือ F4 (ความถี่ 349 เฮิร์ตซ) อย่างต่อ

นอกจากนี้บัซเซอร์จะเล่นโน้ต F4 เป็นเวลาเท่ากับ ‘distance x 10’ มิลลิวินาที เช่น เมื่อ distance = 5 เซนติเมตร บัซเซอร์จะส่งเสียง beep ทุก ๆ 50 มิลลิวินาที ถ้า distance = 10 เซนติเมตรบัซเซอร์จะดังทุก ๆ 100 มิลลิวินาที



Challenge

รู้หรือไม่?

“เธอเรมิน (theremin) เป็นเครื่องดนตรีไฟฟ้าที่ควบคุมโดยปราศจากการสัมผัสทางกายภาพจาก thereminist (นักดนตรีเธอเรมิน) เครื่องดนตรีนี้ถูกตั้งชื่อตาม เลออน เธอเรมิน (Léon Theremin) ผู้ประดิษฐ์มันขึ้นมาและจดลิขสิทธิ์ในปี 1928.”

ที่มา: <https://en.wikipedia.org/wiki/Theremin>



Task: สร้าง Theremin โดยใช้ Maker UNO X และเซนเซอร์อัลตราโซนิก แบ่งช่วงการตรวจจับของเซนเซอร์อัลตราโซนิก ให้เป็นช่วงเล็ก ๆ เพื่อเล่นเสียงในโทนที่แตกต่างกัน

คลิก หรือ สแกน
คิวอาร์โค้ด เพื่อรับ
ชมวิดีโอสาริต



คลิกหรือสแกน
คิวอาร์โค้ด เพื่อรับ
ชมโชว์เล่น
Theremin



ขอแสดงความยินดีด้วย! คุณได้พิชิตบทเรียนที่ 7 สมบูรณ์แล้ว และได้เรียนรู้เรื่องดังต่อไปนี้:

1. เซ็นเซอร์อัลตราโซนิกทำงานอย่างไร.
2. การสร้างฟังก์ชันแบบ self-declare



CONGRATULATIONS!

คุณได้สำเร็จทุกบทเรียนและมีความเชี่ยวชาญในเบื้องต้นแล้ว เราหวังว่าคุณจะรู้สึกสนุกสนานกับสิ่งที่ผ่านมาตลอดเส้นทาง

อย่างไรก็ตาม นี่ไม่ใช่จุดจบการเดินทางของคุณกับ Arduino ด้วยความรู้และทักษะที่คุณได้รับ คุณพร้อมแล้วที่จะเดินบนเส้นทางของคุณ เพื่อสร้างโปรเจกต์ของคุณเอง

สำหรับไอเดียในการทำโปรเจกต์ที่มากมาย คุณสามารถเข้ามาเยี่ยมชมเว็บไซต์ฝึกสอน (tutorial page) ของเราได้ หรือติดตามเราบน Facebook นอกจากนี้ถ้าคุณต้องการอุปกรณ์เพิ่มเติม คุณสามารถไปค้นหาและซื้อมันมาได้จาก Cytron webstore *ฟรีค่าจัดส่งสำหรับนักเรียนและคุณครูอาจารย์

อย่าลืมแบ่งปันการสร้างสรรค์ผลงานของคุณกับเรา คุณสามารถแนะนำหรือทิ้งข้อความไว้ให้เราได้ทุกเวลา เราชอบมากที่จะได้ยินสิ่งเหล่านี้จากคุณ!

ขอแสดงความยินดี~

[คลิกที่นี่หรือสแกน
คิวอาร์โค้ดเพื่อชม
CytronWebstore.](#)



[คลิกที่นี่หรือสแกน
คิวอาร์โค้ดเพื่อเยี่ยมชม
Tutorial Page](#)



[คลิกที่นี่ หรือสแกน
คิวอาร์โค้ดเพื่อชม
Cytron FB Page](#)



ใช้พื้นที่นี้สำหรับการ ..

บันทึก & ไอเดีย



Simplifying Digital Making

ใช้พื้นที่นี้สำหรับ ..

บันทึก & ไอเดีย



Simplifying Digital Making

MAKER UNO X

brought to you by Cytron Technologies